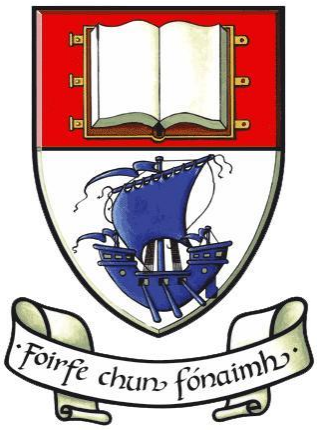


Reconfigurable Adaptive Wireless Sensor Node

A THESIS

BY

RENTAO WANG

IN PARTIAL FUFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Ken Deevy, Principal Supervisor

Joe O'Mahony, Supervisor

Submitted to Waterford Institute of Technology, 2019

Abstract

This thesis describes the design and development of a new wireless sensor node technology called the Reconfigurable Adaptive Wireless Sensor (RAWS) Node. The RAWS node technology has the ability to adaptively support different analogue sensors by reconfiguring its hardware resources in an autonomic, flexible and scalable way.

The research employs programmable mixed-signal hardware as the enabling technology and utilises the hardware reconfigurability as the foundation for implementing the research concepts. A sensor identification scheme has been developed which enables the RAWS node to identify a large number of sensors and acquire the key parameters and attributes of the connected sensor. The research has also developed three adaptive reconfiguration techniques which combine the sensor parameter and attribute information with the dynamic reconfigurability to autonomically adapt the mixed-signal hardware in real-time to different analogue sensors. These features are realised while keeping the power consumption of the system at a low level.

Table of Contents

Table of Contents i

List of Figures iv

List of Tables viii

List of Publications x

Acknowledgements xi

Chapter 1 Introduction 1

1.1 Research Overview 1

1.2 Research Context 4

1.3 Problem Statement 10

1.4 Research Questions 15

1.5 Research Concepts 16

1.6 Objectives 21

1.7 Contribution 22

1.8 Thesis Organisation 23

Chapter 2 Literature Review 24

2.1 Programmable Hardware Technologies 24

2.2 Smart Sensor Standards 47

2.3 Wireless Communication Protocols 61

Chapter 3 System Architecture and Design 73

3.1 Design Approach 73

3.2 RAWS Node System Architecture 76

3.3 Wireless Network Design and Development 81

Chapter 4 Sensor Identification Scheme 96
4.1 Feasibility Analysis of the Sensor Characteristics Classification Approach 96
4.2 IEEE 1451.4 TEDS based Sensor Identification 105
4.3 Sensor Identification Scheme Summary 120
Chapter 5 Adaptive Reconfiguration Techniques 121
5.1 The Principle of Adaptive Reconfiguration 121
5.2 Adaptive Reconfiguration Techniques 136
5.3 Adaptive Reconfiguration Techniques Summary 157
Chapter 6 System Functionality Testing and Analysis 158
6.1 Testing Approach 158
6.2 Wireless Communication Functionality Test 160
6.3 Sensor Identification Scheme Test 167
6.4 Adaptive Reconfiguration Test and Analysis 177
6.5 Power Consumption Test 207
Chapter 7 Conclusion 216
Appendices 223
Appendix A – WSN Node Platforms Analogue Capabilities 223
Appendix B – Switched Capacitor Circuit Basic Theory 230
Appendix C – IEEE 1451 Standards 232
Appendix D – Wireless Communication Protocols Review 236
Appendix E – Principles of ZigBee Networking and Communications 253
Appendix F – IEEE 1451.4 Relevant Information 261
Appendix G – 1-Wire Signalling 265
Appendix H – PSoC Blocks Complete Circuit Diagrams 268

Appendix I – Other Test Results 270

Appendix J – RAWS Research Publication 277

References 284

List of Figures

Figure 1 Illustration of WSN 4
Figure 2 Overview of WSN applications 6
Figure 3 Architecture of Wireless Sensor Node 7
Figure 4 The traditional technology for supporting multiple analogue sensors & actuators 12
Figure 5 Illustration of the key research concepts 16
Figure 6 Illustration of the programmable hardware technology 17
Figure 7 Programmable mixed-signal hardware technology for supporting multiple analogue sensors 18
Figure 8 Illustration of idea of the first sensor identification approach 19
Figure 9 Illustration of idea of the second sensor identification approach 20
Figure 10 Illustration of the Programmable Analogue Hardware Architecture 25
Figure 11 Continuous time CAB 26
Figure 12 Switches capacitor Integrator 27
Figure 13 Different configuration variations of a switched capacitor circuit 28
Figure 14 General architectural block diagram of the FPAA device 30
Figure 15 General architectural block diagram of the PSoC device 32
Figure 16 Reference model of the IEEE 1451 Standard Family 48
Figure 17 Block Diagram of a general IEEE 1451.4-capable sensing system 50
Figure 18 1451.4 TEDS Structure 51
Figure 19 ZigBee and IEEE 802.15.4 Protocol Layers 70
Figure 20 Progression of the RAWS Node System Design 74
Figure 21 Basic Framework of the System Architecture 76
Figure 22 System architecture of the RAWS node platform 78
Figure 23 Wireless RAWS Node Network Diagram 81
Figure 24 RAWS Node Wireless Hardware Design 83
Figure 25 XBee Module API Frame Structure 87
Figure 26 RAWS Node Terminal Application 88
Figure 27 RAWS node firmware ZigBee transmission operation flow 94
Figure 28 RAWS node terminal application ZigBee transmission operation flow 95
Figure 29 Thermistor R-T curves 98

Figure 30 Example of the cross point of two thermistor R-T curves.....99
Figure 31 Exploit self-heating effect to distinguish thermistors.....101
Figure 32 Matlab simulation results of the self-heating effects in special cases.....102
Figure 33 V-T curves of the four most common thermocouple types.....103
Figure 34 1-Wire Interface Connections.....107
Figure 35 TEDS Transmission Procedure.....109
Figure 36 Basic TEDS Format.....111
Figure 37 TEDS Parsing Routine Flowchart.....113
Figure 38 Bridge sensor type Standard TEDS parsing.....118
Figure 39 Analogue System Top Level Architecture.....122
Figure 40 PSoC Continuous Time Configurable Analogue Block Diagram.....124
Figure 41 Configurable Resistor String Diagram.....126
Figure 42 Configuring a CT block into a PGA module.....128
Figure 43 PSoC Switched Capacitor Configurable Analogue Block Diagram.....130
Figure 44 Configurable Capacitor Array Diagram.....132
Figure 45 Configuring an SC Block into a first-order modulator of the delta-sigma ADC.....135
Figure 46 Intelligent Algorithm High-Level Flow Diagram.....137
Figure 47 Intelligent algorithm operations for building signal chain circuits.....141
Figure 48 Intelligent algorithm operations for sensor reading calculations.....142
Figure 49 Signal conditioning and processing chain diagram for bridge sensor.....143
Figure 50 the 9-bit DAC configuration with two SC blocks.....144
Figure 51 Three-opamp INS-AMP configuration using two CT block and one SC block.....146
Figure 52 Second-order modulator of delta-sigma ADC using two SC blocks.....148
Figure 53 Reconfiguration process of the intelligent algorithm technique for C9B full bridge force sensor.....150
Figure 54 Signal conditioning and processing chain diagram for thermocouple.....151
Figure 55 Reconfiguration process of the intelligent algorithm technique for 101-K thermocouple.....152
Figure 56 Signal conditioning and processing chain for RTD.....152
Figure 57 Reconfiguration process of intelligent algorithm technique for 101-10B RTD.....153
Figure 58 Overall reconfiguration process of intelligent algorithm technique.....154
Figure 59 Overall reconfiguration process of local and remote configuration settings technique.....155
Figure 60 General diagram of system test setup.....159

Figure 61 ‘Set 64-bit PAN ID’ AT Command Frames.....161
Figure 62 ‘Set Scan Channels’ AT Command Frames.....161
Figure 63 ‘Set 64-bit PAN ID’ AT Command Response Frames.....162
Figure 64 ‘Set Scan Channels’ AT Command Response Frames.....162
Figure 65 ‘Network-Success’ Status Messages.....163
Figure 66 ZigBee Transmit Request Packet.....164
Figure 67 ZigBee Transmit Status Packet.....165
Figure 68 ZigBee Receive Packet.....165
Figure 69 ZigBee Transmitted Data Displayed on RAWS Terminal Software.....166
Figure 70 C9B Force Sensor TEDS data.....167
Figure 71 Parsing process of the C9B sensor Basic TEDS section.....168
Figure 72 C9B sensor Basic TEDS section parsing results from the TEDS parsing routine.....169
Figure 73 Parsing process of the C9B sensor Standard TEDS section.....170
Figure 74 C9B Sensor Standard TEDS section parsing results by the TEDS parsing routine.....171
Figure 75 101-K Thermocouple TEDS data and parsing process.....172
Figure 76 101-K Thermocouple TEDS parsing results by the TEDS parsing routine.....173
Figure 77 101-K Thermocouple TEDS parsing results on RAWS node terminal application.....173
Figure 78 101-10B RTD TEDS data and parsing process.....174
Figure 79 101-10B RTD TEDS parsing results by the TEDS parsing routine.....175
Figure 80 101-10B RTD TEDS parsing results on RAWS node terminal application.....175
Figure 81 C9B signal process chain placement in PSoC.....178
Figure 82 9-bit DAC Configuration.....178
Figure 83 DAC control registers in PSoC.....182
Figure 84 Three opamp instrumentation amplifier configuration.....182
Figure 85 INS-AMP control registers in PSoC for C9B sensor.....186
Figure 86 Second-order modulator configuration for Delta Sigma ADC.....187
Figure 87 Delta Sigma ADC control registers in PSoC.....190
Figure 88 Compete Signal Chain Test Setup for C9B Force Sensor.....191
Figure 89 101-K Thermocouple signal process chain placement in PSoC.....194
Figure 90 INS-AMP control registers change in PSoC for 101-K thermocouple.....197
Figure 91 Delta Sigma ADC control registers change in PSoC for 101-K thermocouple.....197
Figure 92 Delta Sigma ADC control registers change in PSoC for 101-K thermocouple.....197

Figure 93 101-K Thermocouple signal process chain placement in PSoC 199
Figure 94 PGA Configuration 199
Figure 95 PGA control registers in PSoC for 101-10B RTD 201
Figure 96 INC ADC control registers in PSoC for 101-10B thermocouple 203
Figure 97 Wireless Initialisation Stage Current 208
Figure 98 Sensor Identification Stage Current 210
Figure 99 Sensor Reading Stage Current – C9B Force Sensor 211
Figure 100 Sensor Reading Stage Current – 101-K Thermocouple 212
Figure 101 Wireless Transmission Stage Current 213

List of Tables

Table 1 Important PSoC device features 33
Table 2 WSN Wireless Protocols 64
Table 3 Comparison of Different Wireless Protocols 64
Table 4 Wireless protocols for existing WSN platforms 65
Table 5 PSoC CY8C29466 Features 79
Table 6 XBee ZF RF Transceiver Module Key Specification 80
Table 7 XBee Module Pins used for Prototyping 83
Table 8 XBee API Frame Types used in the RAWS node research 87
Table 9 AT Command API Frame Example 91
Table 10 AT Command Response API Frame Example 92
Table 11 IEEE 1451.4 TEDS Sensor Type Template 114
Table 12 Standard TEDS Template Example – Bridge Sensors Template 114
Table 13 Physical Measurand Selector Enumeration Example 116
Table 14 Continuous Time Block Control Registers 125
Table 15 CT Block Key Register Bits Settings for PGA module 129
Table 16 Switched Capacitor Block Control Registers 130
Table 17 SC Block Key Register Bits Settings for First-Order Modulator of Delta-Sigma ADC 134
Table 18 C9B Force Sensor Key Attributes 177
Table 19 Control Register Settings for 9-bit DAC Part 1 179
Table 20 Control Register Settings for 9-bit DAC Part 2 180
Table 21 Control Register Settings for 9-bit DAC Part 3 181
Table 22 MSB Control Registers of 9-bit DAC 181
Table 23 LSB Control Registers of 9-bit DAC 181
Table 24 CT Blocks Control Register Settings for INS-AMP Part1 183
Table 25 SC Block Control Register Settings for INS-AMP Part1 184
Table 26 CT Blocks Control Register Settings for INS-AMP Part2 185
Table 27 SC Block Control Register Settings for INS-AMP Part2 185
Table 28 Control Register Settings for INS-AMP Part 3 185
Table 29 CT Blocks Control Registers of INS-AMP 186

Table 30 SC Block Control Registers of INS-AMP 186
Table 31 Control Register Settings for Delta Sigma ADC 2nd order Modulator Part 1 188
Table 32 Control Register Settings for Delta Sigma ADC 2nd order Modulator Part 2 188
Table 33 Control Register Settings for Delta Sigma ADC 2nd order Modulator Part 3 189
Table 34 First Stage Control Registers of Delta Sigma ADC 2nd order Modulator.....189
Table 35 Second Stage Control Registers of Delta Sigma ADC 2nd order Modulator 189
Table 36 Decimator Control Registers of Delta Sigma ADC 189
Table 37 PWM period setting of 12-bit Delta Sigma ADC.....190
Table 38 Complete Signal Chain Test Results of C9B Force Sensor.....192
Table 39 Key Attributes of the 101-K Thermocouple.....193
Table 40 INS-AMP CT Blocks Control Register Settings Change for 101-K Thermocouple 195
Table 41 INS-AMP SC Block Control Register Settings Change for 101-K Thermocouple 195
Table 42 PWM period setting of 14-bit Delta Sigma ADC.....195
Table 43 Function Modules settings for 101-K Thermocouple 196
Table 44 Data Conversion Test Results of 101-K Thermocouple 198
Table 45 CT Block Control Register Settings for PGA Part1.....200
Table 46 CT Blocks Control Register Settings for PGA Part2 200
Table 47 PGA Control Registers.....200
Table 48 First Stage Control Registers of Incremental ADC 2nd order Modulator during Conversion 202
Table 49 Second Stage Control Registers of Incremental ADC 2nd order Modulator during Conversion .202
Table 50 First Stage Control Registers of Incremental ADC 2nd order Modulator before Conversion.....202
Table 51 Second Stage Control Registers of Incremental ADC 2nd order Modulator before Conversion .203
Table 52 Decimator Control Registers of INC ADC.....203
Table 53 Data Conversion Test Results of 101-B-10 RTD Sensor 205
Table 54 RAWS Node System Sleep Current (at around 22°C and 3.3V) 207

List of Publications

1. R. Wang and K. Deevy, "Reconfigurable adaptive wireless sensor node," in *IET Irish Signals and Systems Conference (ISSC 2012)*, 2012, pp. 1-6.
2. R. Wang and K. Deevy, "Reconfigurable adaptive wireless sensor node technology using IEEE 1451.4 standard," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, pp. 3988-3993.

Acknowledgements

Firstly, I would like to express my gratitude to my supervisors. I would like to thank my principal supervisor Ken Deevy for his support, guidance and patience over this long research project and thesis writing period. I would like to thank my supervisor Dr Joe O'Mahony for all his help and support during this period of time as well.

I would like to thank my parents, especially my mom, thank them for all the little and big things they have done for me and for being so supportive.

I'd also like to thank my friends who helped me through some difficult times during this time, thank Jing Yunqi for all the delicious dinners, cakes and tarts she made for me.

Finally, thank myself for finishing this, it will be a good start for many better things to come.

Chapter 1 Introduction

1.1 Research Overview

Wireless sensor network (WSN) technology has been playing an increasingly important role in a wide variety of areas, including home automation, smart buildings, environmental monitoring, energy monitoring, health monitoring, industrial process control, and the Internet of Things (IoT) [1] [2]. As a technology that spans multiple disciplines and one of the key technologies of the Internet of Things (IoT), WSN is a diverse and challenging research area and has been gaining significant interest [3] [4]. This research project is called Reconfigurable Adaptive Wireless Sensor (RAWS) Node and it falls within the WSN area and focuses on the wireless sensor node aspect of the technology. The RAWS node research develops a new wireless sensor node technology which can adapt itself to accommodate different analogue sensors in an autonomic, flexible and scalable way and have wide applicability for deployment in low power high utility wireless sensor networks.

A wireless sensor network is a network typically built of small size, low cost and low power devices referred to as *nodes.* A WSN node is essentially a device equipped with a controller for computing and a wireless transceiver for communications, but more importantly, it is a platform for supporting sensors and actuators. WSN nodes have the ability to sense and/or control the environment and to cooperate with other nodes to accomplish one or more particular tasks by communicating information and data wirelessly, facilitating interaction between people and the environment. With these features and capabilities, WSNs make it easy to accomplish many previously difficult tasks and open up new possibilities for many applications that used to be considered as impractical or impossible [3].

While wireless communication capability and other features are necessary, the sensing capability is the key feature of wireless sensor networks. Wireless sensor networks are deployed in various scenarios, where many of them have the requirement of monitoring different physical parameters. Many applications also have the need for the WSN sensor node to add in new sensors or modify its original sensing functions in order to adjust to changes of

environment or application requirements. Therefore, it is a very common need for a wireless sensor network to be able to support multiple sensing functions [5], i.e. WSNs with multi-sensing capability.

Wireless sensor nodes are the elements that form a wireless sensor network, and the sensors on each node are the actual components that are capable of sensing. Therefore, the sensor nodes' ability to accommodate different sensors is the key to realise a WSN with multi-sensing capability. Furthermore, WSN nodes need to be able to accommodate multiple different sensors in a flexible and scalable way. Only in this way does the multi-sensing capability of a WSN become feasible and widely applicable.

Analogue sensors are still the dominant sensor type in use [6] [7]. However, analogue sensors vary greatly in their characteristics, so each analogue sensor normally requires its own particular analogue signal processing chain. An analogue signal processing chain usually contains signal conditioning circuits and function modules to handle the sensor outputs and convert analogue information to digital format. The traditional approach for supporting multiple analogue sensors is simply adding different signal processing chains for each individual sensor. Most existing wireless sensor node technologies still take this approach to cope with multiple sensors because it is straightforward. However, this traditional approach will cause the following problems which will be explained in more detail later:

- Multiple sensors usually need different signal conditioning circuits and processing functions. To support a large number of sensors, the complexity and scale of the sensor node will increase greatly.
- Accumulating signal processing chains also brings other potential problems such as higher power consumption, higher cost and bigger physical size. This goes against the inherent philosophy of low power, low cost and small size WSN sensor nodes.
- Lack of flexibility and scalability. When different or extra sensors are needed, adding more signal processing chains usually requires substantial changes to the WSN nodes. The entire system may have to be redesigned just because of one sensing function modification.

This research aims to overcome these disadvantages and solve the multi-sensing challenge for WSN sensor nodes. The research developed the RAWS node technology with the following features:

- Reconfigurability and adaptability as the key features for supporting multi-sensing capability
- Capable of supporting analogue sensors in a flexible and scalable way
- Accommodating different sensors autonomically with optimised system performance
- Realising the above features without compromising power consumption, size and cost
- Overall, a more generic WSN node platform with high flexibility and scalability, and wide applicability for low power high utility WSNs

The following key research concepts have been devised for the RAWS node technology and the proof-of-concept is demonstrated through the design, development, and testing of the RAWS platform prototypes presented in this thesis. These concepts will be illustrated in more detail later.

1) Programmable mixed-signal hardware as the enabling technology to build the reconfigurable and adaptive sensor node platform. The RAWS node adaptively reconfigures its signal conditioning and processing functions in real-time to accommodate different analogue sensors.
2) Sensor identification scheme to provide the sensor's identity, parameters and other attributes. This information is utilised by RAWS nodes to recognise the connected analogue sensors and to decide how to reconfigure the hardware for the connected sensors.
3) Adaptive reconfiguration techniques that utilise the sensor information provided by the sensor identification scheme, and combine it with the reconfigurability of the programmable hardware technology to adaptively and autonomically configure the RAWS node hardware system to accommodate different analogue sensors, realising a WSN sensor node technology with autonomic multi-sensing capability.
4) Dynamic power state management to achieve low power consumption.

1.2 Research Context

Because wireless sensor networks merge a wide range of information technologies that span hardware, software, networking, and programming methodologies, this section gives a basic overview of the WSN technology in order to provide a broad context for this research.

1.2.1 Wireless Sensor Network

The definition of a wireless sensor network is that it is essentially a network of spatially distributed individual devices referred to as *nodes* which have the ability to sense and/or control the environment, to perform at least some basic computation, to communicate information and data through wireless mediums, to cooperate with other nodes to accomplish one or more particular tasks, and to facilitate people interacting with the environment. Nodes are also called *motes* or sometimes just simply *sensors*. Because WSN applications may need a large number of nodes to be non-invasively deployed in the environment for a longest possible time period without any maintenance or energy supplement, wireless sensor nodes are normally low cost, small size and low energy consumption. It is worth mentioning that some WSN nodes also have actuation capability to control the environment, and these nodes can be referred to as *actuators*.

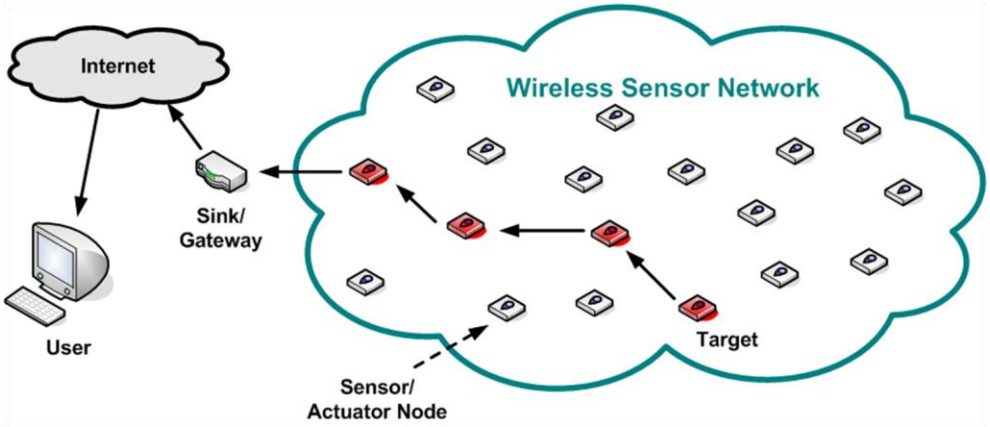


Figure 1 Illustration of WSN

WSNs may also contain devices denoted as *sinks* which collect information and data from nodes and *gateways* which have the ability to connect to other networks such as the internet. Gateways sometimes can also function as sinks, and vice versa.

The unique features of wireless sensor networks bring many advantages:

- Enable significant reduction in wiring and maintenance cost and labour
- Have negligible impact on the surrounding environment
- Can be fast, easily and flexibly deployed, installed and extended in various conditions including unknown and hostile environments
- Could last for years in some applications where it may not be feasible for humans to be present all the time
- Can obtain localised 'hard-to-get' measurements and large amounts of detailed real time information which may not be possible via other known methods

The advantages of WSN technology make it possible to perceive what takes place in the physical world in ways not previously possible [3]. These advantages make WSNs powerful platforms for a variety of applications with greatly varying requirements and characteristics [8] [9] [2]. WSNs have been employed in a wide range of real-world scenarios including scientific research [10] [11] [12] , health care [13], environmental monitoring [14] [15] [16] [17], public utilities [18] [19], industry [20] [21] [22] [23], security and surveillance [24] [25] [26], sustainable technologies [27], and military applications [28]. An overview of the types of WSN technology applications is shown in Figure 2. It is also worth mentioning that one of the most important WSN applications is the Internet of Things but there is also a big overlap between the IoT and WSN applications as WSN is one of the key technologies of the IoT.

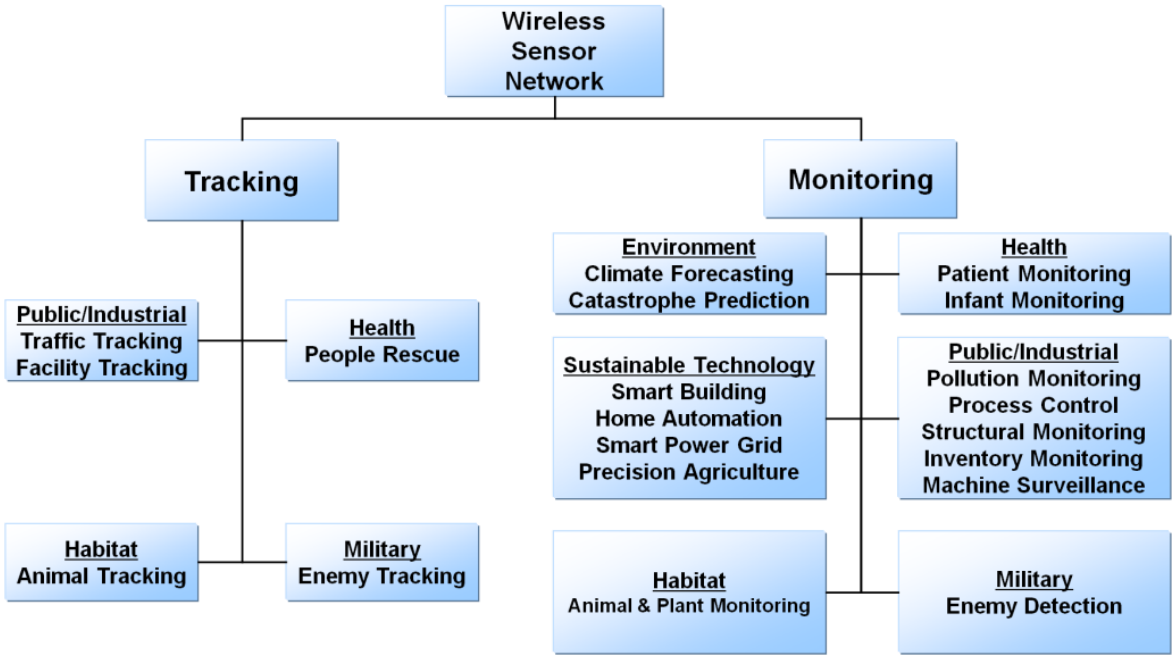


Figure 2 Overview of WSN applications

1.2.2 Sensor Node

The wireless sensor nodes are the essential elements of a WSN. The functionalities of a WSN are built on the primary tasks of the nodes, and the five main tasks include sensing and/or actuation, computation, data storage, and wireless communication. Therefore a node comprises five main parts to support its functions, as shown in Figure 3. This subsection gives an overview of these five main parts.

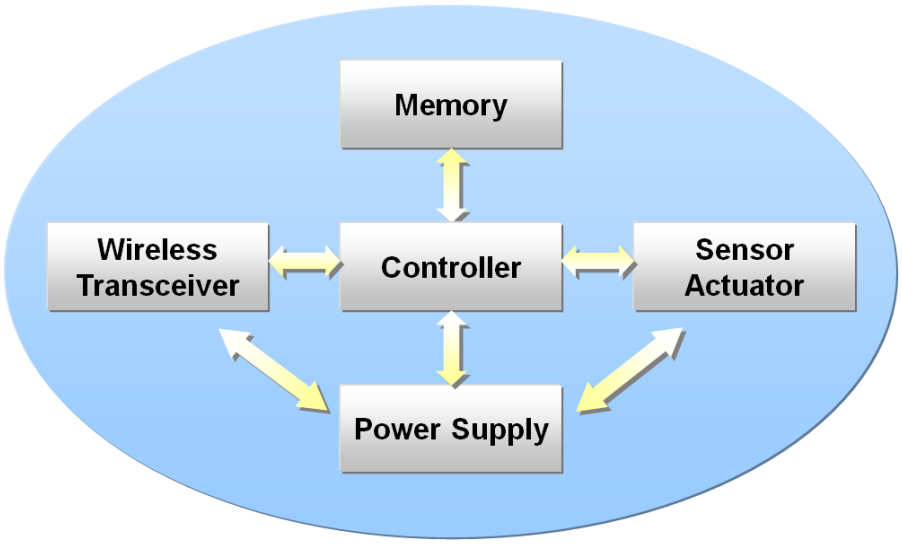


Figure 3 Architecture of Wireless Sensor Node

Power Supply

Unlike the power supply in many other electronic systems which are not the most important part as long as it meets the voltage and current specifications of the system, for a WSN sensor node the power supply plays a key role from the beginning of the node design. Generally power sources in WSN node systems are some sort of battery with a small physical size which can only provide a very limited amount of energy, typically a few hundreds of milliamp-hours. Therefore, low power is one of the most important features for most WSN designs as the limited energy budget demands low power consumption and high energy efficiency for all components in the WSN node system shown in Figure 3.

Controller

The controller is the core of a WSN node, managing all the tasks such as collecting data from sensors, processing and storing the sensor data, controlling the wireless transmissions, and executing different programmes to fulfil application goals. As the core of the node, the choice of the controller is important for developing the node platform. In general, microcontrollers are the common choice as the sensor node platform controller in many WSN applications. This is because microcontrollers are designed for use in embedded systems, so their features meet the general WSN application requirements. They have relatively low power consumption, typically in the tens of milliamp range or lower. They can provide high flexibility because they have in-

circuit re-programmability and they can easily switch between different programmes. Moreover, modern microcontrollers already incorporate many common analogue and digital peripheral functions to offer System-on-Chip (SoC) capabilities. High-integration microcontrollers, especially the ones with mixed-signal capabilities, can help to reduce the circuit complexity of the system and provide better flexibility. A detailed analysis of other WSN controller options, including general purpose processors, digital signal processors (DSPs), field-programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), can be found in Appendix A.

Wireless Transceiver

Wireless transceivers connect the nodes into a network and they interface the WSN sensor nodes with each other. The general characteristics of WSNs and WSN applications, such as low power, low cost, small size, non-line-of-sight transmission environments, determine that the RF based wireless communication is usually the most suitable option because RF communications and transceivers can meet these requirements. There are many mature low energy RF wireless protocol options available for use in the WSN area, such as ZigBee, Bluetooth, Bluetooth Low Energy, and Wi-Fi. The wireless aspect of WSNs will be discussed in more detail in Chapter 2.

Sensor

The sensor part in the WSN sensor node architecture diagram (Figure 3) actually contains two parts: the actual sensors themselves and then the hardware and software components associated with the sensor signal processing chain.

Sensors are the interface between the physical environment and the WSN system. Sensors can be generally classified into two categories based on the output signal type: analogue sensors and digital sensors. But as the direct interface with the real world, sensors are still analogue at the core. For the following reasons, analogue sensors are still the most common kind of sensor in use and in some applications they are irreplaceable:

- Low cost
- Reliable

- Robust and can survive harsh environments where its digital counterpart often cannot

The main disadvantage for analogue sensors is that they have a wide of variety of characteristics and non-standardised output signals. This can cause many challenges for the sensor signal conditioning and processing.

The hardware components of the sensor signal conditioning and processing chain include different functions such as for example gain stages, filters and data converters. The software algorithms that participate in signal conditioning, processing and conversion are also part of the chain. The processing chain hardware circuits can be formed by discrete components or IC chips. However, many functions that are commonly used for handling signals have been integrated into SoC type microcontrollers, which can help to reduce the circuit complexity of the sensor node. Meanwhile, programmable hardware technologies offer new options for building signal processing chains.

In many cases, the sensors themselves are considered as independent to the node platform. However, the sensor signal processing chain is always an indispensable part of a WSN node platform. It determines a node platform's sensing capabilities as it determines the two perspectives from which the sensing capabilities can be measured: the types of supported sensors and the quality of acquired data. As discussed before, the first feature, i.e. the types and number of sensors that a node platform can support, is one of the key factors that decide the applicability of a WSN.

Software System

Although it is not shown in the WSN node architecture diagram (Figure 3), the software is also a very important part of the system. A well-designed hardware platform provides the foundation that allows a WSN system to support its desired functions and features. The software system is built on the hardware platform and it instructs the hardware to realise these vital functions and features.

The software systems for different WSNs have their own particular characteristics due to the different design requirements and objectives, so it is difficult to describe them in general. But

there is one important characteristic featured by most of the WSN software systems and it is the dynamic energy and power management. The purpose of the dynamic energy and power management is to switch each individual component into different power states in order to reduce the system energy consumption.

1.3 Problem Statement

1.3.1 Multi-Sensing Challenges

The multi-sensing challenge was briefly explained at the beginning of this chapter and it is discussed in more detail here.

WSNs have diverse applications as discussed previously and often require the sensor nodes to support different types of sensors for different measurement functions. Also in many applications additional sensors have to be added to the existing system because extra measurement functions are required. An example is setting up an environmental monitoring WSN system. When initially setting up a basic environmental monitoring system, the sensor nodes may only need to support functions like temperature sensing, relative humidity sensing, and ambient light sensing. When it needs to be expanded into a more comprehensive monitoring system, other sensing functions such as atmospheric pressure sensing, ultraviolet radiation sensing, and CO_2 sensing may also be required.

The following situations may occur as well. Due to the changes of application environments or requirements, some modifications or updates may have to be made to the WSN being used. Sensor nodes may need to support alternative models of the original sensors, or need to support sensors that were not included in the original design. For instance, a WSN equipped with vibration, temperature, and humidity sensors is deployed in a manufacturing plant for monitoring the conditions of the production equipment, and after a few years a percentage of the vibration sensors may become faulty, but the original sensor model has been discontinued. The alternative replacement options have some small differences compared to the original one. So in this case the WSN will need to support an alternative sensor with slightly different specifications or performance.

With the IoT applications rapidly expanding, the WSN as one of the key technologies is increasingly involved in many more diverse applications. In this context, a sensor node platform that is able to interchangeably work with different sensors for handling diverse scenarios can reduce development time and cost and bring many other benefits. Therefore, as WSNs are becoming increasingly pervasive, the demands on wireless sensor nodes are also increasing, and it becomes more common that wireless sensor nodes are required to accommodate different sensors as well as to support new additional sensors to cope with various situations in a flexible and scalable way in order to be widely applicable.

From the perspective of sensing capabilities, a flexible WSN node platform should be able to support different sensors without greatly increasing the complexity, physical size and power consumption of the system; a scalable platform should be able to support new or extra sensors without fundamental hardware alteration. Unfortunately, the traditional WSN sensor node technologies are not able to meet these requirements in many cases.

As discussed before, analogue sensors are still the dominant type of sensors in use. They are cheap and reliable. More importantly, they are robust. Sensors often have to face harsh environments as they are the direct interface to the real world. In many cases analogue sensors can be the only choice because digital sensors can be easily damaged by punishing environments where analogue sensors can survive. For example, to measure the high temperature inside a blast furnace, most digital sensors cannot do the job while the simple and cheap analogue thermocouples can. For the above reasons, analogue sensors still prevail in this digital world and provide information needed by the digital world. For the same reasons, analogue sensors are also important in wireless sensor network applications, especially in areas like industrial control, building structure monitoring, pollution and disaster monitoring.

To utilise information provided by analogue sensors in digital systems, output signals from analogue sensors have to be processed and converted into the digital domain. This requires the support of analogue signal processing chains. However, analogue sensors vary greatly in their characteristics and usually provide non-standardised output signals. To accommodate different analogue sensors, different sets of signal processing chains are needed. This feature makes it

difficult to support a number of different analogue sensors especially in a system that has only limited local resources.

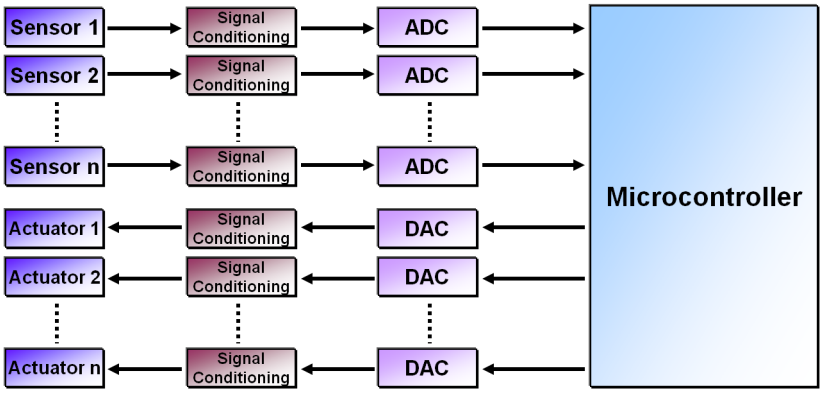


Figure 4 The traditional technology for supporting multiple analogue sensors & actuators

The traditional approach for coping with different analogue sensors is to stack signal processing chains as illustrated in Figure 4. This approach needs a large scale of circuits for supporting a number of different sensors. As a result the system complexity is greatly increased and this means bigger physical size, higher power consumption and higher cost. More importantly, these systems lack flexibility and scalability. Because a traditional signal processing chain usually is for one sensor, adding different or additional sensors into the system will require support from different signal processing chains. This not only means stacking more signal conditioning and processing circuitry and causing more issues on size, cost and power but also can lead to requiring fundamental changes to the node or even redesigning the whole system. The traditional approach is not an ideal solution for the wireless sensor node system to generally support a number of different analogue sensors.

In this context, how to accommodate different analogue sensors within the WSN sensor node system in a flexible and scalable way while not compromising power consumption, size and cost, becomes a significant challenge, i.e. the multi-sensing challenge.

1.3.2 Research Gap

As discussed in the previous section, to solve the multi-sensing challenge and meet the wide applicability requirement, a wireless sensor node platform should have the ability to support sensors in a flexible and scalable manner. However, this research reviewed the literature and

found that existing wireless sensor node technologies lack this ability. This section discusses the reasons why the existing WSN node technologies lack the ability to solve the multi-sensing challenge. The relevant research in this field will be described later in Chapter 2.

Accommodating analogue sensors relies on the support of signal processing chains. The analogue/mixed-signal capabilities of a WSN node platform determine its analogue signal processing chains and therefore decide its sensing capabilities. To solve the multi-sensing challenge the requirement for analogue capabilities is even higher. So the analogue capabilities can be considered as the foundation for WSN node platforms to support sensors. However, the literature review results show that most of the wireless sensor node platforms have very limited analogue capabilities, i.e. lacking a proper hardware infrastructure (see Table A1 in Appendix A).

The literature review indentifies that the analogue capabilities of a WSN platform are largely or even solely dependent on its controller. This is because node platforms are built around the controller and it is not practical to reserve a lot of extra external signal processing circuits and ICs in a WSN node system with limited resources. With regard to the choice of the controller, according to the review results, except for a few platforms that employ FPGAs or DSPs as the controller, many of the WSN node platforms choose a microcontroller as the platform controller. This result matches the theoretical analysis of the WSN node controller choice described in subsection 1.2.2, i.e. microcontrollers are the common choice for WSN nodes.

In terms of controllers' analogue capabilities, the DSPs and FPGAs are normally pure digital controllers hence having little or no analogue capabilities. On the other hand, most of the microcontrollers used in WSN platforms are the system-on-chip type, and they integrate some analogue peripherals. However, these microcontrollers mainly just include one type of ADC for data conversion, but lack other analogue functions such as the functions for signal conditioning. The literature review shows that the analogue capabilities of the WSN node platforms are mostly only a 10-bit SAR (Successive Approximation) ADC or a 12-bit SAR ADC plus analogue comparators (Table A1 in Appendix A). Most of these microcontrollers feature low or ultra low power consumption, and this is one of the reasons why they are selected for WSN platforms

(The detailed reasons for these microcontroller selections are analysed in Appendix A). But with limited analogue capabilities, these WSN sensor node platforms cannot establish complicated signal processing chains to provide sufficient support for different sensors. For example, the Atmel ATmega128 is one of the most widely used microcontrollers in WSN platforms as shown in the review results. It is a low power controller with an 8mA typical active current and 20uA sleep current. But in terms of integrated analogue functions, it mainly only has a 10-bit SAR ADC with a sample rate of 15ksps which is a reasonably high rate. Yet in temperature measurement applications where a thermocouple is required as the sensor, this ADC is not suitable because thermocouple measurements normally do not need a high sampling rate but do require a high resolution ADC as their voltage outputs are very small.

Due to the limited analogue capabilities, these WSN node platforms have to resort to external circuits and ICs to support sensors. The review shows that the existing WSN node technologies do in fact take this approach. Node platforms like the eWatch [29] and the SunSPOT [30] employ external amplifier and ADC ICs to work with certain analogue sensors. But if different sensors are required, these platforms will have to add more external analogue ICs. This is the same way that the traditional approach copes with sensors and as discussed before this approach has many significant disadvantages.

More importantly, because of the inadequate infrastructure, the existing WSN node technologies lack flexibility and scalability and it is impractical to further build any high-level techniques to generally support different analogue sensors.

In this context, there is a need for a new WSN node technology and the new WSN node should have the proper infrastructure and ability to interchangeably work with different analogue sensors to support various WSN applications. This leads to the research questions for this research project.

1.4 Research Questions

The key research question is how to develop a new WSN sensor node technology that is able to support different types of analogue sensors in a flexible and scalable way to solve the multi-sensing problem. More importantly, it should be able to cope with different analogue sensors in an autonomic manner, i.e. without any manual intervention, to further enhance the applicability of the technology. At the same time it should minimise any compromises to the power consumption, size and cost.

The detailed research questions involved in the research are:

1. What kind of embedded systems technology and architecture is required in order to develop a hardware platform to provide the foundation to generally handle a number of different analogue sensors, i.e. supporting multi-sensing capability?
2. In order to enable the autonomic operation capability, the technology needs to have the ability to acquire important information about the unknown sensors for the system, and this leads to the following questions:

   How can the system obtain the characteristics from analogue sensors themselves to identify different sensors? What are the important characteristics that can be utilised for identification? Can these characteristics provide sufficient information? If yes, what methods are required to obtain these characteristics and information? How can these methods be implemented on the WSN sensor node platform?
3. What high level techniques including software algorithms are required to collaborate with the hardware platform to autonomically accommodate different analogue sensors in a flexible and scalable way?
4. How can the above objectives be achieved with optimised overall energy efficiency?

1.5 Research Concepts

The RAWS node research project was designed to answer the research questions just described. The research concepts were described at the beginning of the thesis but further details are discussed here. The RAWS node research took the reconfigurability and adaptability as key attributes to solve the multi-sensing challenge, and developed a technology that can support different analogue sensors within each sensor node in a flexible, autonomic and scalable way, as shown in Figure 5. The key feature of the RAWS node is that it is able to adaptively and autonomically reconfigure itself to accommodate a wide range of analogue sensors, i.e. autonomic multi-sensing capability.

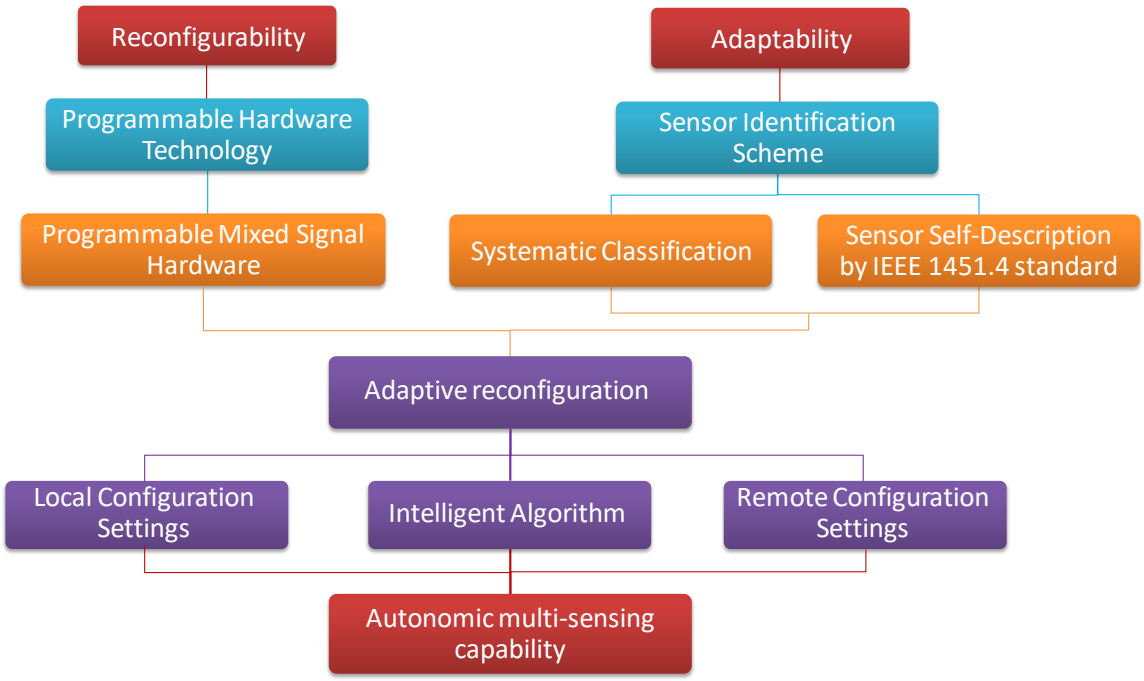


Figure 5 Illustration of the key research concepts

Programmable Hardware

The RAWS node research identifies that programmable hardware technology, especially the more recent generation of mixed-signal programmable hardware, as one of the enabling technologies. Programmable hardware, as the name suggests, has the ability to make changes to the structures or configurations of the hardware itself so that different functionalities can be

achieved as shown in Figure 6. Moreover, this capability becomes even more potent when the devices can be reconfigured in real-time.

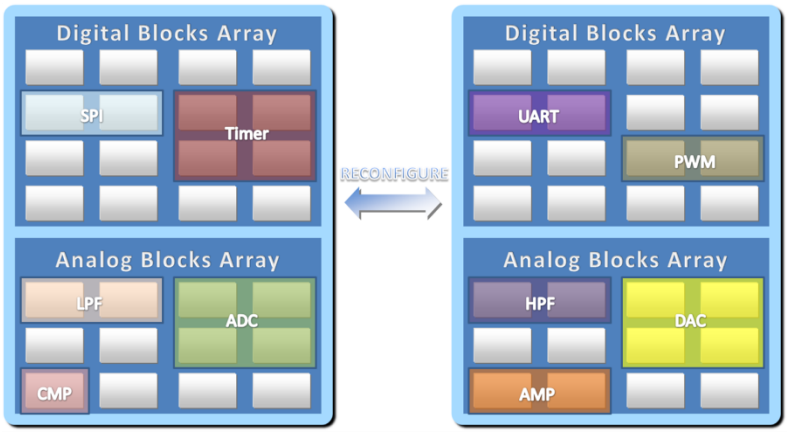


Figure 6 Illustration of the programmable hardware technology

The research exploits this reconfigurability to use the same hardware resources to support different sensors as illustrated in Figure 7. This approach can help overcome many limitations of the traditional technology which simply stacks different signal processing chains for different analogue sensors.

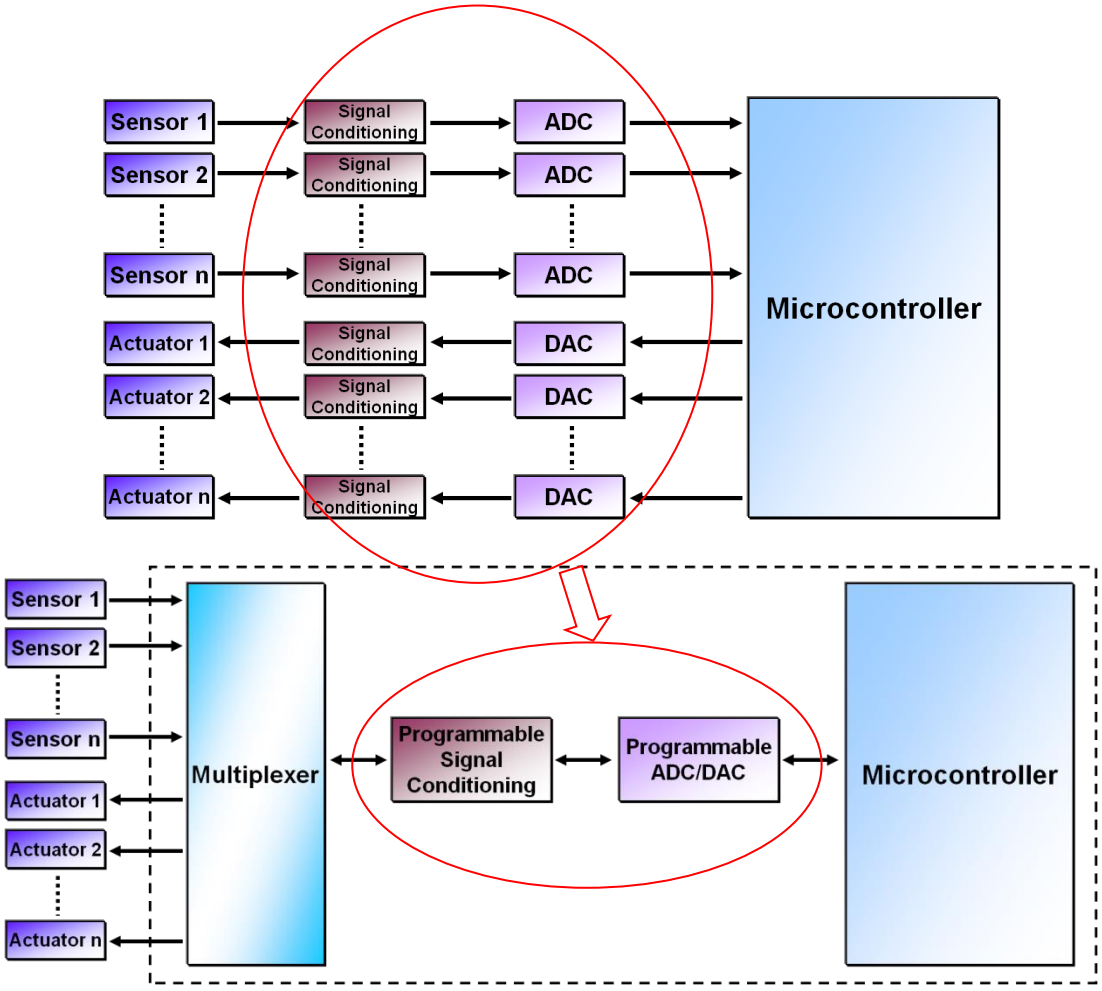


Figure 7 Programmable mixed-signal hardware technology for supporting multiple analogue sensors

The RAWS technology utilises the hardware reconfigurability as the foundation stone. By reconfiguring the hardware into suitable signal processing functions in real-time, the RAWS node is able to adapt to different sensors and it can also support new sensors that are not included in the original design. This greatly improves the flexibility and scalability of the system.

Sensor Identification Scheme

For the RAWS nodes to adaptively and autonomically reconfigure their programmable hardware to accommodate analogue sensors, the RAWS nodes need to have adequate information about the connected sensor so that they can decide what kind of signal conditioning and processing chain is suitable. This research developed a sensor identification scheme that can be used by the RAWS nodes to identify the connected sensors and acquire key parameters about the sensors.

The research investigated two approaches for realising such a sensor identification scheme. The first approach is a systematic classification process of sensor identification, which consists of a series of systematic tests for the RAWS nodes to acquire key attributes of the connected sensor and to classify and then to eventually identify the sensor. To be more specific, when an unknown sensor is connected, the RAWS node executes a number of systematic tests to obtain the important characteristics from the sensor, such as the output voltage, resistance, frequency response, and utilise these characteristics to identify the sensor. If the sensor is identified, the RAWS node reconfigures its hardware to support the sensor. This research investigated the feasibility of developing such a systematic classification process of sensor identification, but found some inherent shortcomings in its ability to be widely applicable.

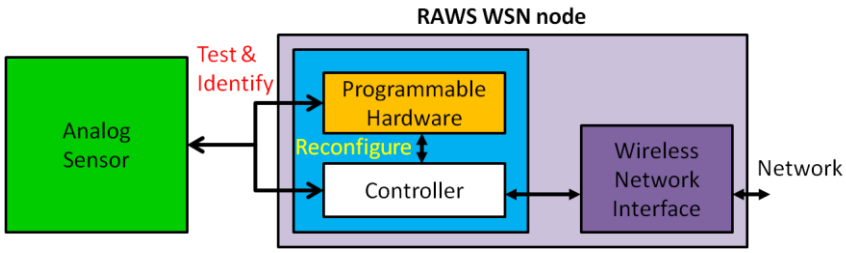


Figure 8 Illustration of idea of the first sensor identification approach

The second approach is based on the sensor self-description and self-identification capability provided by the IEEE 1451.4 standard. IEEE 1451.4 standard (the rest of this thesis will refer to the 1451.4 standard as dot 4 standard for short) is a member of the IEEE 1451 family of open, industry consensus standards which aims to bring smart features into transducers. The most important feature of the dot 4 standard for this research is the transducer electronic data sheet (TEDS) concept for traditional analogue sensors. The TEDS memory can provide identification information and important characteristics of the sensor to the host system. This second sensor identification scheme approach includes the TEDS interface hardware circuit and software routines developed for the RAWS sensor node to retrieve data from TEDS-enabled sensors. It also includes the software algorithm for parsing the TEDS and extracting sensor identification and description information.

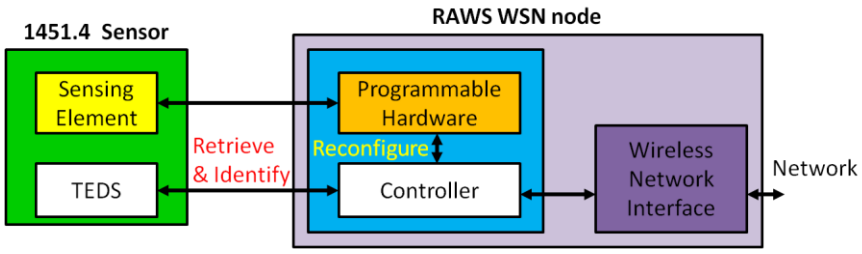


Figure 9 Illustration of idea of the second sensor identification approach

Adaptive Reconfiguration Techniques

This research further developed the three adaptive reconfiguration techniques in the form of software algorithms that utilise the acquired sensor information to manage the RAWS node hardware reconfigurations in an autonomic way to support analogue sensors. The adaptive reconfiguration is the key process that combines the hardware reconfigurability with the sensor identification scheme to realise the main project aim, i.e. autonomic multi-sensing capability.

- The intelligent algorithm technique

  The software algorithm is able to intelligently select suitable signal conditioning and processing function modules and also decide suitable parameters for the function modules based on the important sensor attributes such as sensor type, electrical output range, etc. The intelligent algorithm method can generally support a large number of sensors, and the configuration created by the algorithm can be used by the other two adaptive techniques below. The intelligent algorithm is the central adaptive reconfiguration technique of the RAWS system.

- The local configuration settings technique

  The RAWS node based on the sensor identification information including manufacturer ID and model number is able to search and load the corresponding hardware configuration settings from the local memory.

- The remote configuration settings technique

  Due to the limited memory space, a large number of configuration settings cannot be stored locally on the sensor nodes. Therefore, this research also developed this technique which stores the hardware configuration settings in a remote database

located on the RAWS node network gateway. The configuration settings are transferred wirelessly to the RAWS node when required.

These three techniques can function individually and they could also cooperate with each other and achieve a highly adaptive reconfiguration scheme.

In addition to the key research concepts discussed above, dynamic power management was incorporated into the system so that RAWS nodes can achieve the reconfigurability and adaptability in a way that does not compromise on the power consumption as low power consumption is one of the essential and important requirements of WSN nodes.

1.6 Objectives

The detailed objectives of the research are as follows:

- Develop a reconfigurable signal conditioning and peripheral functions wireless sensor node hardware platform by utilising mixed signal programmable hardware technology.
- Develop a sensor identification scheme that can provide important sensor information to the sensor node platform.
- Develop techniques for the platform to utilise the sensor information to adaptively reconfigure the hardware to support the corresponding sensors, and more importantly, endue the reconfiguration algorithm with the intelligence to build best possible signal conditioning and processing chain with the available hardware resources and set up suitable parameter for the connected sensor to achieve optimised system performance.
- Accommodate different analogue sensors in an autonomic, flexible and scalable way, i.e. the platform can adapt to the connected sensor without any manual operation and aid, realising an autonomic multi-sensing capability for the WSN node.
- Develop a low power, low cost, and small size sensor node technology.
- Develop the WSN node technology with wide applicability for deployment in low power high utility wireless sensor networks.
- Implement a fully operational WSN testbed prototype.

- Devise and carry out system functionality, performance, and power consumption tests and analyses.

1.7 Contribution

The results of this research programme realise a new wireless sensor node technology with autonomic multi-sensing capability, i.e. an analogue sensor *plug & play* solution in the context of wireless sensor node technology. The RAWS node platform is able to autonomically reconfigure its hardware to adapt to different analogue sensors, supporting analogue sensors in a flexible and scalable way.

The results of this research programme can offer novel features and significant advantages in the area of WSNs, across a range of application platforms. The primary novelty and contribution of the RAWS node technology is the intelligent adaptive hardware reconfiguration algorithm. It enable the RAWS system to be able to synthesise important sensor attributes and build signal conditioning and processing chain and set up signal chain parameters most suitable for the sensor aiming for optimised measurement performance without any manual intervention, i.e. adaptive reconfiguration in autonomic operations, whilst for other research work in the similar field the system hardware configuration for sensors needed to be manually predesigned by users. This is also the major difference between the RAWS technology and other research work that makes RAWS technology flexible and scalable.

The multi-sensing capability supported by adaptive reconfiguration of the sensor hardware with autonomic sensor interfacing capability are central pillars of the approach as well as small size, low power and low cost. These characteristics will yield a number of benefits:

1. Improved applicability, scalability, and flexibility in the wireless sensor node system

2. Optimal balance between highly capable functionality and energy efficiency

3. Lower cost and smaller size wireless sensor nodes for a range of diverse application areas

1.8 Thesis Organisation

The rest of this thesis is organised as follows:

Chapter 2 reviews and analyses the relevant technologies and standards of the RAWS node technology and their literature, including programmable hardware technologies, smart sensor standards, and wireless communication technologies.

Chapter 3, 4 and 5 illustrate the detailed RAWS system design and implementation for the proof-of-concept including the overall system architecture, the wireless sensor node platform testbed, the sensor identification scheme, and the adaptive reconfiguration techniques.

Chapter 6 discusses the system functionality and power consumption tests. The theoretical analysis and tests results of the main parts of the system design are described and examined in this chapter.

Chapter 7 summarises and concludes the RAWS node research project and restates the key research question and concepts, the design and testing of the system, and the contribution of this research work. The chapter also discusses possible future work.

Chapter 2 Literature Review

Three main types of technologies and standards serve as the basis of this research, including programmable hardware technologies, smart sensor standards, and wireless communication technologies. This chapter reviews and analyses the relevant state-of-the-art technologies and standards and concludes with indicating the choice of technology, standard and protocol for the proof-of-concept as well as the rationale for this choice. The existing and contemporary research and publications in the relevant area are also reviewed and examined in this chapter in order to provide a more detailed context for the RAWS node research and a more detailed discussion of the research gap.

2.1 Programmable Hardware Technologies

This research utilises the reconfigurability of programmable hardware as the foundation upon which to build high-level techniques for adaptively and autonomically accommodating different analogue sensors. This section reviews and analyses the analogue programmable hardware technologies, laying the groundwork for the design and development of the hardware platform for the RAWS node system. In the first subsection, different analogue programmable technologies are analysed with their characteristics to illustrate how different techniques can be utilised. Different programmable analogue and mixed-signal hardware is then discussed and compared in order to identify a suitable hardware platform for prototyping. Finally the relevant research work and applications of programmable hardware in the WSN area are examined and discussed.

2.1.1 Programmable Analogue Hardware Technologies Overview

2.1.1.1 General Hardware Architecture

The programmable analogue hardware is comprised of two main components, an array of configurable analogue blocks (CABs) and a programmable interconnection network as shown in Figure 10.

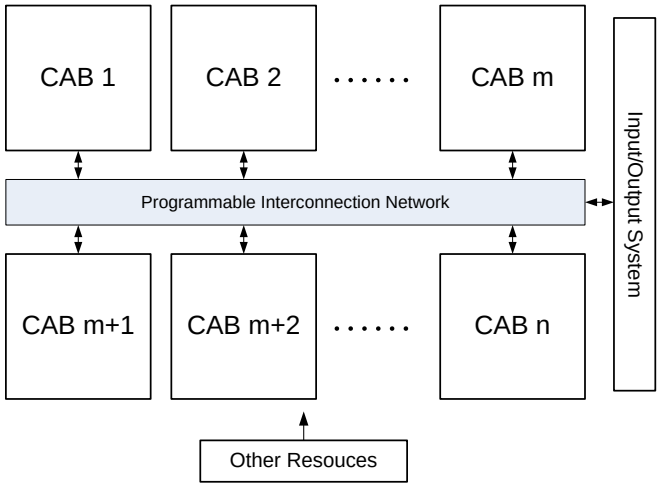


Figure 10 Illustration of the Programmable Analogue Hardware Architecture

The programmable interconnection network can route the signal path among the configurable analogue blocks and between the I/O (input/output) system and CAB arrays. A configurable analogue block contains analogue circuits of which the hardware configuration can be reshaped to achieve different analogue functions such as different types of amplifiers, converters, filters, etc. Some programmable hardware allows multiple configurable analogue blocks to chain together in order to implement the required function when one CAB is not sufficient. The configurable analogue blocks can be broadly categorised into two types: the continuous time CAB and the discrete time CAB, each displaying their own advantages and disadvantages which will be discussed next. With different characteristics, each type of CAB shows its own strength at the different stages of the analogue signal processing chain. This is discussed in the following subsections and also will be reflected and used in the RAWS node system design.

2.1.1.2 Continuous Time Technology

The continuous time (CT) CAB usually consists of computational elements, programmable passive components and a configurable switch array. The CT CAB is similar to the combination of the traditional continuous time circuit and an array of configurable switches which can control the signal flow path within the CT circuit so that different functions can be achieved. The computational element is usually an operational amplifier or an array of transistors. Programmable passive components are often resistor matrixes of which the resistance value is variable and configurable. The configurable switches array interconnects the computational

elements and the passive components, and structures the CAB circuit into different configurations. A simple continuous time CAB is shown in Figure 11 below as an example.

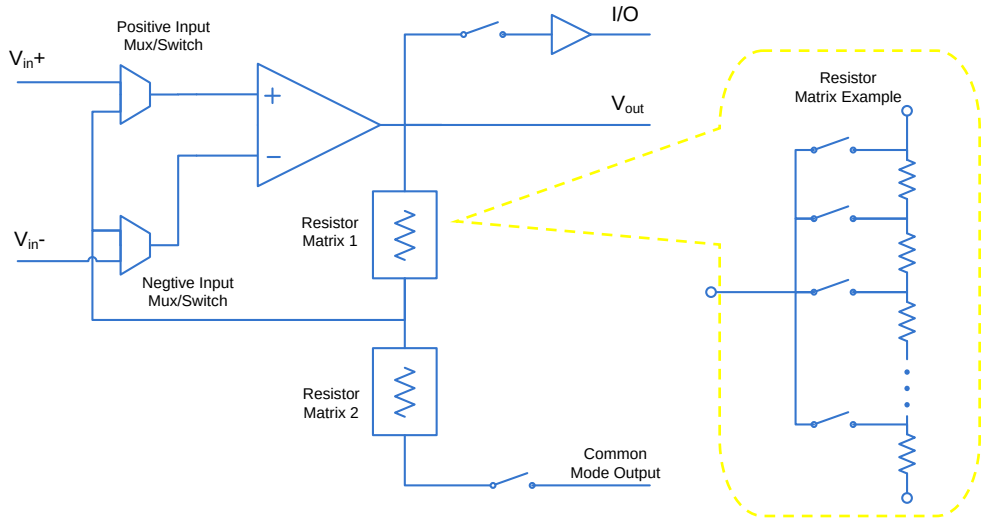


Figure 11 Continuous time CAB

Compared to discrete time CAB, normally the continuous time CAB can support larger signal bandwidth; the CT CAB design is relatively easy as the switch array is usually not very complicated; it works in continuous time so no anti-aliasing filters are required. Due to these advantages, the CT blocks are often applied at the front stage of a signal processing chain for functions like amplifiers and comparators.

The flexibility of a single CT CAB is however normally restricted due to the relatively simple switch array. The switches introduce parasitic inductances and capacitances into the CT CAB circuits which will reduce the maximum signal bandwidth, and the switches can introduce noise and other sources of error. Also the precision or the tolerance of the resistor matrixes is difficult to control in the integrated circuit fabrication.

2.1.1.3 Discrete Time Technology

The discrete time configurable analogue hardware in essence uses sampled-data technologies to economically and accurately reproduce the functions of their continuous time hardware counterparts. The switched capacitor (SC) technology is dominant in discrete time programmable devices.

The idea behind switched capacitor technology is to use capacitors and alternately opened and closed switches to replace resistors and control signal flow paths. The detailed theory of the SC circuit is explained in Appendix B. Figure 12 illustrates the basic SC theory with an example of the switched capacitor circuit replacing the resistor in an integrator which is an important component in many analogue designs such as filters.

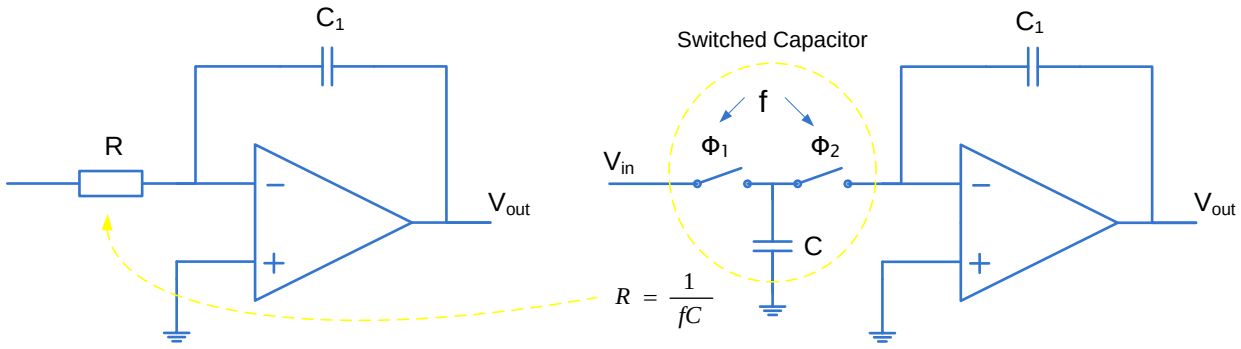


Figure 12 Switches capacitor Integrator

SC technology has many advantages. First of all, in integrated circuit fabrication capacitors can be precisely fabricated whilst it is relatively difficult to manufacture resistors with high precision, and also capacitor ratios can be more accurately implemented than resistor ratios [31] [32] [33]. In addition, the effective resistance of a switched capacitor can also be controlled by the switching clock which allows the effective resistor to be varied over a relatively wide range [33]. These advantages of SC technology can benefit many analogue circuits. In the integrator example above, when the integrator is constructed with a resistor, the performance of the integrator is subjected to the tolerance of the components in the circuit. Replacing the resistor with a switched capacitor, the integrator's performance will depend on the ratio of the two capacitance values and will no longer be affected by the absolute value of components thereby not relying on the component tolerances. As the capacitor ratios can be accurately controlled and the equivalent resistance of the switched capacitor can be configured over a wide range, the SC integrator is more precise and flexible than the resistor version.

Secondly, the switched capacitor circuit has better flexibility than the traditional continuous time circuit. A well structured SC circuit can easily be programmed into different configurations

performing different functions. As illustrated in Figure 13, the SC circuit can be flexibly configured and reconfigured for functions including inverting amplifier, non-inverting amplifier, integrator, and comparator by assigning appropriate switching sequences to the five switches.

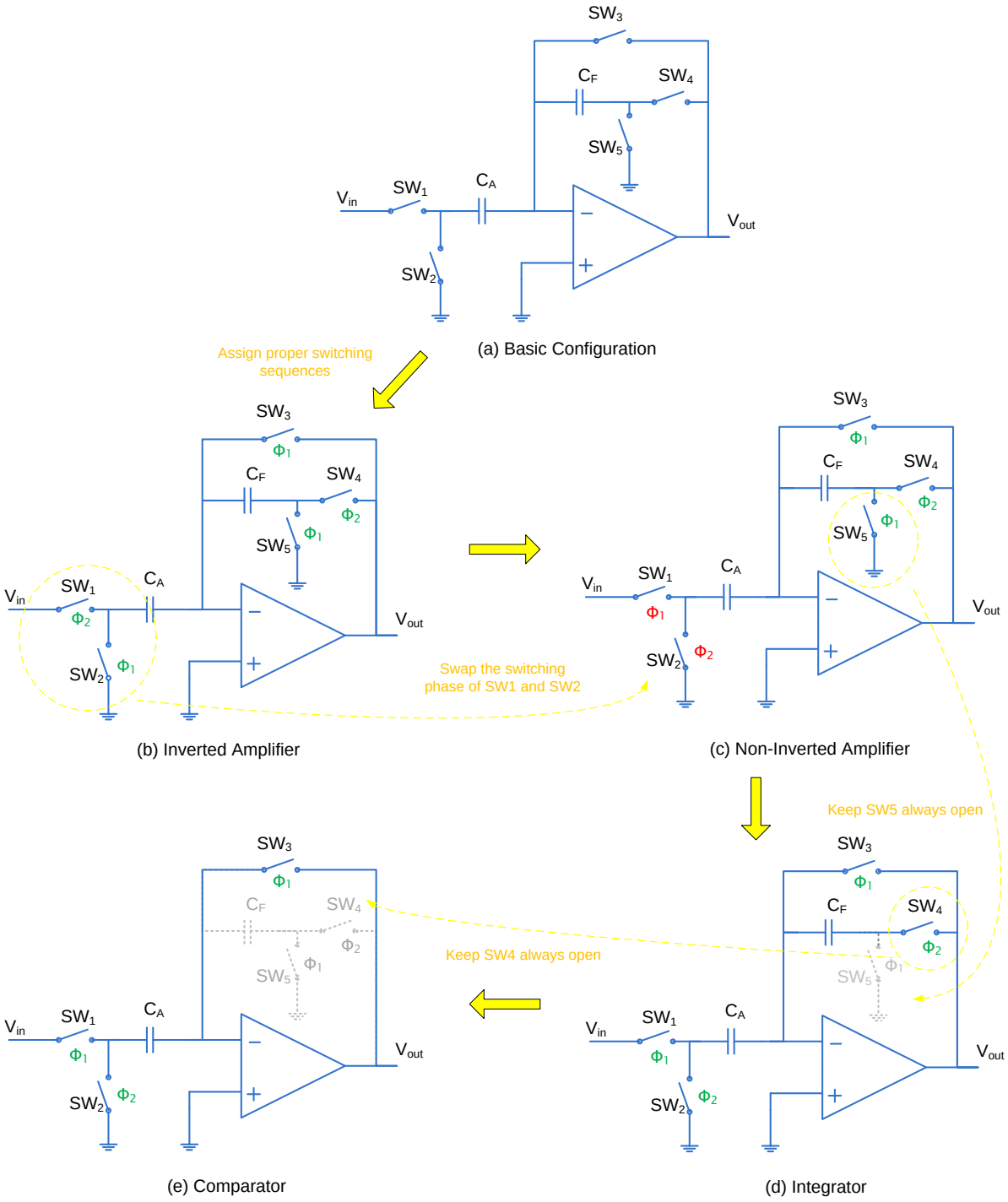


Figure 13 Different configuration variations of a switched capacitor circuit

SC circuits have their own disadvantages though. As a sampled data technology, the bandwidth of the switched capacitor CAB is limited due to the finite sampling and switching rate, also anti-aliasing filters are needed which can complicate the system design. Switches and capacitors can introduce nonlinearity and more noise than resistors do, so the SC CABs may require a more complex design and layout process.

2.1.2 Programmable Hardware Platform Analysis and Selection

An option for the programmable hardware platform for the RAWS node technology is to design a custom IC or ASIC which integrates a highly configurable analogue subsystem with an array of CT and SC blocks. These CABs should allow different signal conditioning and processing chains to be flexibly created to provide support for a large number of different types of sensors. This custom IC should also contain a low power CPU subsystem with adequate computing capability and memory space (such as the ARM Cortex-M cores), common digital peripheral functions or a small array of digital configurable blocks, and maybe even an RF wireless communication subsystem as well. In conclusion, what would be required is a highly reconfigurable and low power custom IC platform. But considering the design and development time, cost and complexity of this solution, a more practical approach is to select a suitable programmable hardware from the commercially available devices for the proof-of-concept.

The two major commercial programmable analogue/mixed-signal hardware platforms are the Field-Programmable Analogue Arrays (FPAA) device and the Programmable System-on-Chip (PSoC) device. There are also some other programmable analogue ICs and microcontrollers, but they are not comparable to the FPAA and the PSoC devices in terms of the flexibility and the scale of analogue functionalities integrated into one chip. The FPAA and the PSoC are the two programmable hardware platforms that are considered here.

2.1.2.1 Field-Programmable Analogue Array

The field-programmable analogue array is a typical programmable analogue hardware device. An FPAA is essentially a set of configurable analogue blocks connected through a programmable interconnection network. The FPAA is purely for analogue functions as

suggested by its name. Although continuous time FPAAs and discrete time FPAAs are both available, generally the commercial FPAA devices available are based on switched capacitor designs. They typically contain 4 SC blocks with 2MHz signal bandwidth, programmable interconnects and other components such as clock sources and voltage references. The general architecture of the FPAA device is shown in Figure 14.

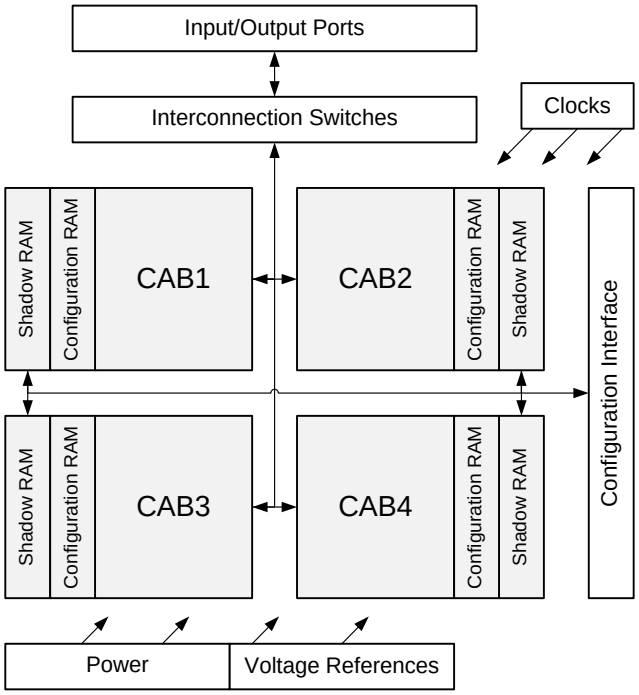


Figure 14 General architectural block diagram of the FPAA device

The programmable interconnects include switches and multiplexers to provide connections for the CABs and I/O subsystem of the device. The hardware configuration settings are transferred over the configuration interface, normally from a host controller, and stored in the configuration memory for arranging configurable analogue blocks and programmable interconnects. The older FPAA devices only support static reconfiguration which means the reconfiguration progress cannot be carried out when the FPAA device is operating and a reset is required before any new configurations become effective. The newer devices are dynamically reconfigurable, i.e. the hardware configuration can be updated during runtime.

With regard to analogue capabilities, off-the-shelf FPAA devices can be used to create complete analogue signal conditioning and processing chains. But because they normally only feature

relatively low resolution ADCs, typically 8-bit, they are more often used for signal conditioning functions like amplification, filtering, summing, rectification.

The power consumption of FPAA devices depends largely on the usage of the configurable analogue blocks. For nominal to high CAB usage, the typical current consumption of FPAAs is around 80 to 220mA in full power mode, and 25 to 73mA in low power mode. Many FPAAs do not have power saving modes, but the latest devices have started to support this feature and the typical deep sleep current is about 4uA.

2.1.2.2 Programmable System-on-Chip

The PSoC is a programmable embedded system on a chip as the name suggests. The PSoC contains more than just an array of configurable blocks. A PSoC also includes a CPU subsystem, configurable digital blocks, programmable interconnects, and other system resources [34]. So PSoCs are effectively programmable mixed-signal microcontrollers. The block diagram of a typical PSoC is shown in Figure 15.

The analogue subsystem of the PSoC encompasses a configurable analogue blocks array, analogue references, analogue input multiplexers, and analogue drivers. The CAB array contains both continuous time CABs and switched capacitor CABs as well as other resources including internal analogue multiplexers. The CAB array is arranged in the form of columns as shown in Figure 15. The analogue subsystem contains up to 4 analogue columns with up to 12 configurable analogue blocks depending on the device. Some devices also have 4 extra limited function analogue blocks. The typical signal bandwidth of the block is around 2.8MHz. These configurable analogue blocks can be used individually or in combination to implement different functions.

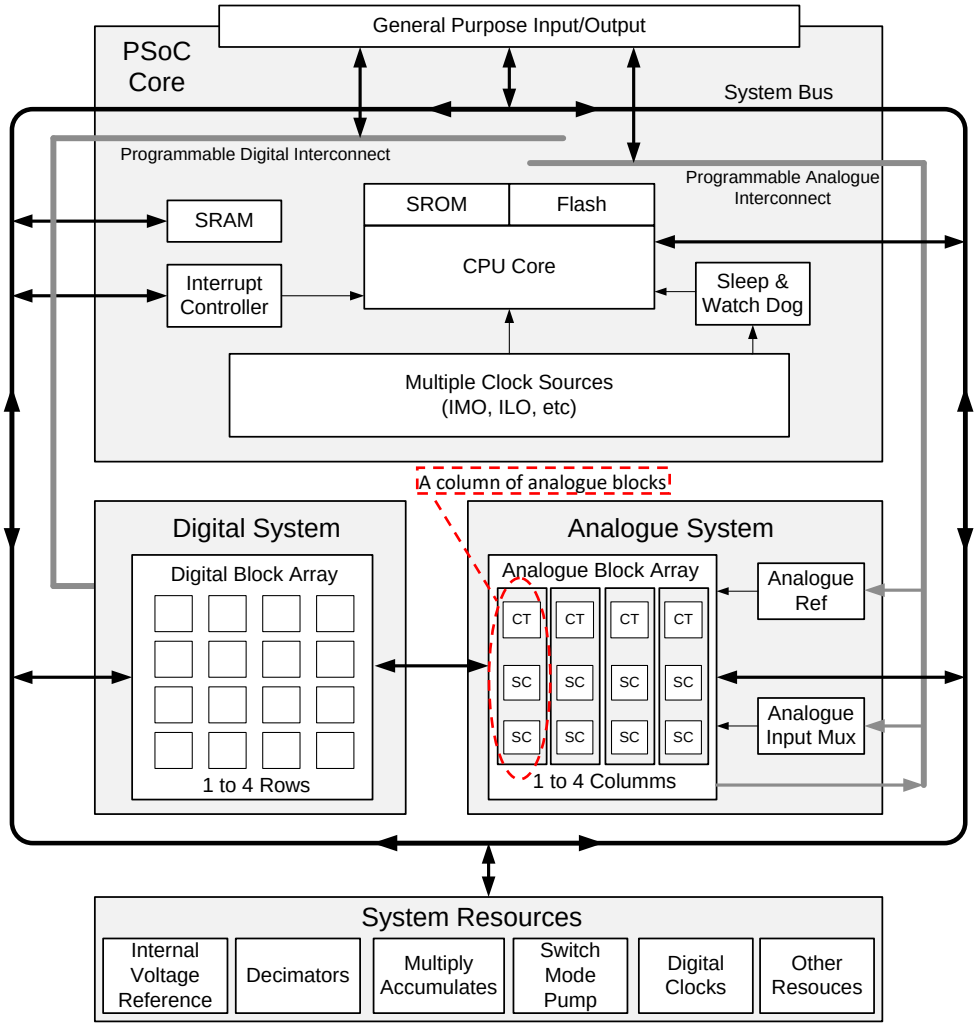


Figure 15 General architectural block diagram of the PSoC device

The digital system of the PSoC is composed of 4 to 16 digital configurable blocks depending on the device. Digital blocks are organised in rows of four. Each block can be used alone as an 8-bit logic block. 16, 24, and 32-bit wide logic function modules can be formed by combining multiple blocks in the same row. Supported digital functions include PWMs, counters, timers, UART, SPI, and I^2C series communication modules.

The analogue and digital capabilities of the PSoC devices are summarised in Table 1. As shown in the table, it is possible to build different complicated and complete signal conditioning and processing chains with these functions.

Analogue Functions	6- to 14-bit incremental, delta sigma, and SAR ADCs 2-, 4-, 6-, and 8-pole band pass, low pass, and notch filters programmable gain amplifier with gain up to 48x instrumentation amplifiers with gain up to 93x Comparators 6- to 9-bit DACs Other configurations can also be implemented using the generic SC blocks 1.024 V ±0.1% internal voltage reference across –40°C to +85°C (14 ppm/°C)
Digital Functions	8- to 32-bit timers, counters 8-, 16-bit PWMs UART 8-bit with selectable parity SPI master and slave I^2C slave and multi-master 8- to 32-bit Pseudo-random sequence (PRS) generator 8- to 32-bit CRC generator

Table 1 Important PSoC device features

The CPU subsystem includes a CPU core, SRAM data memory, and flash program memory, and other essential system resources for a microcontroller. The configurable general purpose inputs/outputs (GPIOs), programmable internal interconnects and global buses provide connection to the CPU subsystem, and the digital and analogue resources of the device.

The PSoC device programming is through two GPIO pins which also work as the ISSP (In-system serial programme) pins, and there is no dedicated configuration interface on PSoC. The PSoCs are dynamically reconfigurable and the reconfiguration process is conducted internally by the CPU.

The typical current consumption of the PSoC device is less than 15mA while the sleep current is less than 5uA.

2.1.2.3 Comparison and Summary

Although both the FPAA and the PSoC are able to provide the common functions that are necessary for constructing analogue signal conditioning and processing chains and both can be reconfigured at runtime to support different sensors, the PSoC has advantages that make it a more suitable choice for this research compared to the FPAA:

- The FPAA contains only switched capacitor blocks. The PSoC employs both continuous time and switched capacitor blocks. This heterogeneous design is more flexible because different analogue functions can be implemented with different types of blocks utilising the advantages of both techniques. For example, the CT blocks can be used for front-end functions such as amplifiers because they can better handle signals with large bandwidth, while the SC blocks have more flexible configurations and they can be used for oversampled ADCs and filters. Also signal conversion is the weakness of the FPAA, while the PSoC is superior as it provides different types of ADCs for different scenarios.
- An FPAA is an array of CABs and needs an external controller. A PSoC contains a built-in CPU subsystem with memory. This can reduce the complexity of the sensor node as well as the overall cost and power consumption. The PSoC also contains digital blocks that can be programmed to perform different digital functions. This can improve the flexibility, scalability and applicability of the system.
- The 'all-the-functions-in-one-chip' feature means PSoCs can be utilised to develop a system with high integration and also help to reduce the size, cost, and the complexity of the system and save development time.
- An FPAA needs the host controller to reprogramme it. Even though the new FPAA model can be dynamically updated, configuration settings still have to be sent from the host controller to the FPAA and then the FPAA can start the reconfiguration process based on the new settings. The PSoC can dynamically reconfigure itself via controlling its internal registers and this internal reconfiguration approach is more flexible and powerful. The importance of this feature will be shown later in the thesis.
- The most important advantage for the PSoC is the relatively low power consumption. The current consumption is less than 15mA for the whole PSoC under the nominal chip

resources usage condition, whilst the FPAA device itself will consume more than 150mA at full power mode with 50% CAB usage. Even in the low power mode with 25% CAB usage, the typical current is still more than 25mA and the power consumption of the external host controller is not included yet. The high power consumption of FPAAs is generally not a good fit for the low power requirement for wireless sensor nodes.

In summary, the PSoC programmable mixed-signal microcontroller was selected as the hardware platform for the RAWS node prototyping and proof-of-concept.

It is worth mentioning that when the RAWS node research started there was only one PSoC device family available which is now referred to as PSoC 1. As the research progressed, three new PSoC families were released and they are the PSoC 3, 4, and 5 families. The PSoC technology reviewed and analysed in this section is mainly based on the PSoC 1 family, and the PSoC 1 is also the main hardware platform for the RAWS node prototyping, although a PSoC 5 based RAWS node prototype is also implemented in the research as well.

The PSoC 3, 4, and 5 families are similar to each other except for featuring different CPU cores. Compared to PSoC 1, the PSoC 3, 4, 5 families have the following main advantages: they are equipped with faster and more powerful CPU cores; they offer more sleep mode options and have lower current consumption in sleep modes; some of the modules support higher resolution ADCs, up to 20-bit versus the 14-bit ADC of the PSoC1; some of the analogue function modules have better specifications.

However, the PSoC 3, 4, 5 families employ a different architecture for the analogue subsystem which brings two main changes. Firstly, the number of CABs is reduced to 4 at maximum. Secondly, many of the analogue functions are provided in the form of dedicated modules. An example is the ADCs available for PSoC 3, 4, 5 families. There are two types of dedicated ADC available, the SAR ADC and the Delta Sigma ADC. Some device models support both but some only support one of them. The main advantage of this design approach is that the dedicated function modules like the ADCs just mentioned can have better specifications and performances, because they are based on the circuits specifically designed to realise their

functionalities rather than based on generic configurable blocks. But the trade-off of this architecture change is the flexibility which is an important feature of the RAWS node design.

As a result of the architecture changes, the flexibility of the analogue subsystem reduces, because the analogue subsystem is no longer fully configurable and the number of generic CABs is greatly reduced. In some circumstances, it becomes difficult to reconfigure the hardware for different sensors.

For this research project, the PSoC 1 is better in terms of the flexibility and the extent of the reconfigurability of the analogue subsystem, so the PSoC 1 was the main programmable platform for prototyping and concept proofing even though the PSoC 5 has also been used. In the rest of this thesis, the PSoC 1 will be generally referred as PSoC. When the PSoC 3, 4, 5 devices are discussed, their family number will be explicitly indicated so they can be distinguished from the PSoC 1.

2.1.3 Programmable Hardware Literature

Because programmable hardware technology itself is a huge research area, this review will focus on its literature in the WSN area. This review is organised into two parts. The first subsection will provide a general context of the application of different types of programmable hardware in WSN research. Then, because one of the key concepts of this research is utilising the reconfigurability of programmable hardware to support different sensors, the review on this topic is split from the general overview and will be discussed in detail in the second subsection.

2.1.3.1 General Context of Programmable Hardware Technologies in WSNs

Programmable hardware has been employed in many sensor research work and applications. This research reviewed the literature and found that a big percentage of them are in the digital domain, whilst the number of analogue ones is relatively small.

Within the digital domain, programmable hardware is employed mainly because of the advantages brought by their flexibility and/or their processing power. Field-programmable gate arrays (FPGAs) are the dominant programmable hardware in the digital domain. The flexibility

of FPGAs allows them to be widely used for quick prototyping, concept proofing, and low cost product development [35] [36] [37]. Some models of FPGAs also have computing power comparable to or even exceeding DSPs, making them a valuable choice as the processor in WSN node platform designs for digital signal processing tasks which involve a lot of heavy computation operations [38] [39] [40] [41] [42].

Typical application scenarios where digital programmable hardware is employed to perform heavy computation locally on WSN nodes include: image data processing [43] [44] [45], routing and networking functions [46] [47] [48], encryption algorithm operations [49] [50] [51], signal processing or modulation functions for the digital front-end of wireless transceivers [52] [53], and many other different types of arithmetic operations and algorithms such as digital filtering functions [54] [55], artificial neural networks and machine learning algorithms [56] [57]. Some applications also make use of the reconfigurability to enable WSN nodes to take different hardware configurations to process different types of computation operations so that the node can efficiently handle different heavy computing tasks in an adaptive way, improving the overall energy efficiency of the system [58] [59] [60] [61].

Similar to digital programmable hardware, analogue programmable hardware is also employed for quick prototyping, concept proofing, and low cost product development. But the scope of application fields is relatively narrow and they were mainly utilised as the analogue front-end in signal chains [62] [63] [64]. Analogue programmable hardware is also equipped in sensor systems for handling heavy tasks locally, like the analogue signal processing (ASP) tasks, to reduce the wireless communication overhead and improve the overall performance and energy efficiency of the whole sensor node system [65] [66] [67]. New analogue programmable hardware such as FPAAs with new architectures has also been developed for sensor applications focusing on different features such as low power operations [68] [69] or better performance and flexibility [70] [71] [72].

2.1.3.2 Application of Programmable Hardware for Supporting Sensor in WSNs

The previous subsection presented a general picture of how programmable hardware has been utilised in WSN applications. This subsection reviews the research projects and applications that employ programmable hardware for accommodating sensors.

The literature review indicates that there is a relatively small amount of research on programmable hardware for supporting sensors in WSN nodes. These relevant research publications and applications, especially the ones in the analogue domain, are analysed in detail here to illustrate the features, advantages and disadvantages of different designs and to indicate the differences with the RAWS node technology.

2.1.3.2.1 Digital Domain Research

In the digital domain, a representative example of the utilisation of the hardware reconfigurability for accommodating different sensors on WSN node platforms is presented [73]. This research [73] reported a reconfigurable sensor node platform which consists of a microcontroller as the core controller of the platform and an FPGA as the coprocessor for the sensors. The FPGA works as a digital sensors controller and can be programmed to process a set of digital protocols including I^2C, PWM, and 1-Wire. Digital sensors with the abovementioned protocols can be supported by this FPGA based sensor controller. The FPGA processes the digital output information and sends the data to the microcontroller. This platform has the reconfigurability for accommodating different sensors. Some other research publications also presented similar wireless sensor platforms with structure and functionalities described above [74] [75] [76] [77] [78].

These designs naturally are for working with digital sensors because FPGAs are digital. In these node systems, the FPGAs are interposed between sensors and the main controller of the node, i.e. the microcontroller, working as a coprocessor for controlling digital sensors. The microcontroller is also in charge of reprogramming the FPGA because a host controller is needed for carrying out the FPGA device programming operations. It is relatively easy for these platforms to support different digital sensors, because there is only a small set of commonly used digital protocols for sensors.

However, these WSN node technologies can still only work with the pre-known sensors as they do not have sensor identification ability and cannot adaptively reprogramme the node hardware. Different hardware configurations for sensors, i.e. different digital protocol implementations on the FPGA, have to be pre-designed. The configurations are loaded into the FPGA by manual operation or based on a predefined sequence.

Also, FPGAs are not the best choice for this kind of design even though they are fully capable of handling different digital peripheral communication protocols and the reasons are analysed as follows.

As discussed above, the main advantages of FPGAs in WSN applications include the high processing power and reconfigurable computing capability, but power consumption is not one of them. In general, FPGAs are high power devices compared to microcontrollers. As an example, the platform presented in [73] consumes around 30mA even in standby mode, and more than 50mA for the FPGA re-programming phase.

FPGAs have been employed in many WSN applications but usually in the applications that are required to handle computation-intensive tasks. The FPGA can be loaded with different specific hardware configurations for different tasks to increase the system performance. By utilising the reconfigurable computing of FPGA-based DSPs, computation-intensive tasks can be executed locally and efficiently through hardware processing rather than executed by the microcontroller via slow software methods or carried out remotely which entails wireless transmissions of a large amount of data. Therefore, for computation-intensive applications, even though the FPGA processor itself is relatively high power, from the whole system point of view they could improve the overall energy efficiency and in turn reduce the system power consumption, i.e. the advantage of the FPGA-based DSPs could counter-balance their high power disadvantage.

However, if the FPGA devices were just used as the digital sensor controllers, the wireless sensor node cannot benefit greatly from the advantages of the FPGAs described above while the system power consumption remains at a high level. In other words, in these cases the FPGAs' main advantages are not fully utilised and cannot compensate for their high power disadvantage in the limited-energy WSN systems. For accommodating digital sensors in WSN

node systems, the FPGA devices can be replaced by other low power options including microcontrollers with configurable digital function modules or mixed-signal programmable hardware. The RAWS node technology can also be applied to adaptively and autonomically accommodate digital sensors, even though this research project focuses on analogue sensors.

2.1.3.2.2 Analogue Domain Research

Compared to digital sensors, the greatly varied characteristics and non-standardised outputs make it relatively complicated to accommodate different analogue sensors in WSN nodes, even with flexible programmable hardware. In terms of the hardware involved, the FPAA, as one of the major analogue programmable hardware technologies, has been naturally deployed in this field, including both continuous time FPAAs and switched capacitors FPAAs. The PSoC as the main mixed-signal programmable hardware available has been utilised as well. Besides the FPAA and PSoC, some other off-the-shelf analogue ICs and ASICs with configurability have also been involved.

This review shows that although analogue programmable hardware has been employed for sensors, it is not always due to their reconfigurability and it is not always to adaptively cope with different sensors. The relevant research is analysed in detail here and it is categorised based on how the programmable hardware is utilised for sensors.

i. Programmable hardware used but not for reconfigurability

The programmable hardware has many other important features which can bring benefits to a design besides the reconfigurability. In many scenarios, these features are the reasons that the programmable hardware is utilised in sensor system designs and the reconfigurability is not needed. In these designs, the system is usually developed to work with a few specific sensors.

FPAAs are often employed because they can be easily reprogrammed so that the design can be built and modified more conveniently. In one typical example [79] a switched capacitor FPAA was used for quick prototyping an ASIC designed for weather monitoring applications. This prototype handles three analogue sensors which measure temperature, humidity and CO_2 level respectively. The FPAA is in charge of signal conditioning and processing for these three sensors. In this design the traditional approach for supporting different sensors was adopted. Three

separate and fixed analogue signal processing chains were built for each sensor, and three identical SAR ADCs are included in each chain.

The PSoCs are often employed because of the advantages brought by their 'all-in-one' feature. The PSoC as a mixed-signal SoC contains CPU subsystem, memory, and programmable analogue and digital arrays in one IC. This 'all-in-one' feature helps to reduce the number of component parts onboard improving system integration. This feature can also help to reduce the size, cost, power and the complexity of the system and save development time. Just as important, it can help to improve the system reliability as well. Many WSN research projects and applications include the PSoC because of these advantages. An example is where a PSoC together with a 2.4GHz wireless transceiver are used to equip a custom-designed thin-film gas pressure sensor with digital and wireless interfaces, turning the basic gas pressure sensor into a small-size smart wireless sensor [80]. In this smart gas pressure sensor system, the PSoC microcontroller plays the key role. It manages the overall system operations, and it takes charge of the signal conditioning, filtering, and data conversion for the gas pressure sensing element, and it also controls the wireless transceiver. The advantages that are brought by the all-in-one high integration feature plus other characteristics such as low power consumption and ease-of-use are the main reasons that the mixed-signal SoC were employed in not only the applications just mentioned but also many other WSN research projects and applications [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] .

The kind of system discussed above is designed to just work with one or a very few particular sensors, and the reconfigurability of the programmable hardware is not exploited.

ii. Programmable hardware used due to reconfigurability but not for supporting different sensors

In some cases, the reconfigurability of the programmable hardware is used in the sensor systems but just to control the parameters of the signal processing chain. This kind of system usually works with one particular sensor at a time, and the reconfigurability is utilised to tweak the system for this particular sensor so only a small part of the reconfigurability is needed and used.

One example is where an SC FPAA is employed to implement a signal conditioning circuit for working with radiation sensors [91]. In this system [91] ,the FPAA was constructed as a second order low pass filter and an amplifier which were the two main elements in the radiation sensor signal conditioning circuit. The main structure of this circuit was fixed, while the parameters of the circuit were adjusted in order to tune the cut-off frequency of the filter and the gain of the amplifier. The FPAA was utilised in order to enable the parameters of the analogue front-end to be tuned more easily for radiation sensors to achieve high precision.

In a WSN system developed for sound localisation [92], PSoCs were employed to build the sensor nodes. The PSoC was the core of the node. The CAB array of the PSoC was set up for signal conditioning, filtering, and conversion. The PSoC also handled the digital processing operations including Hanning windowing, FFT, etc. The reconfigurability of the PSoC was applied for adjusting the two important parameters of the signal processing circuits, the cut-off frequency of the low pass filter and the gain of the input amplifier, to enhance the localisation accuracy.

Besides the two examples above, many other WSN designs also utilise programmable hardware in the same way [93] [94] [95] [96] [97] [98] [99] [100] [101] [102], i.e. exploiting the reconfigurability of the programmable hardware for tuning the parameters of the analogue front-end to achieve better system performance and to save development time.

Reconfigurability is also utilised to control the circuit parameters for reasons such as improving the system robustness. One example is where an FPAA is employed as the analogue front-end of a sensor system which was designed to be robust with respect to sensor failures [103]. This system was set up in a way that it could connect three identical sensors at a time to create a sensor redundancy so that as long as the three sensors did not fail at the same time the system could still keep operating. Three identical RTDs were used with the system and the three same signal conditioning paths were created in the FPAA. The structure of the signal chains was fixed, but the parameters could be controlled. When a sensor failed, the FPAA was reprogrammed to set the gain of the input amplifier of the sensor to 0 which effectively disconnected the sensor.

Overall, despite the different objectives, in these kinds of designs, reconfigurability is used to improve the sensor system to better support one specific sensor. The overall structure of the signal processing circuit built on the programmable hardware is fixed and does not need to be changed. The reconfigurability is utilised to make small changes, i.e. altering the parameters of the circuit.

iii. Reconfigurability utilised to support different sensors

The analogue/mixed-signal programmable hardware has been used for supporting different analogue sensors. One example was a wireless health monitoring device based on a PSoC microcontroller [104]. The PSoC was the core of the device. Three analogue sensors shared a signal processing chain constructed in PSoC through the internal multiplexer. The signal processing chain was predesigned and its structure was fixed, but the gain of the input amplifier of the chain, a PGA, was adjusted in run-time as the PSoC cycled through the three sensors. In another example [105], a sensor system had two gas pressure sensing elements connected to a PSoC which was the main controller of the system. Two configurations were built for these two sensing elements, and the PSoC could switch between the configurations to work with the two elements. Some other designs also utilised the programmable hardware in a similar way to the above examples. However, in this kind of design, the sensors were pre-known to the system and the hardware configurations were already pre-designed for them. The system automatically iterated through sensors and loaded pre-set parameters, and the programmable hardware was used to assist the system in accommodating a small number of specific analogue sensors.

The reconfigurability feature has also been a key consideration for generally supporting different sensors. One example was a wireless sensor platform with a continuous time FPAA as the analogue sensor front-end [106]. This platform had a microcontroller as the main processor of the platform. The FPAA front-end was in charge of conditioning the analogue signals before they were fed into the ADC of the microcontroller. This FPAA front-end was set to work with a tilt sensor but it could be reprogrammed for other sensors. A similar programmable analogue sensor front-end but using two FPAAs was also reported [107]. In some other sensor node

designs [108] [109], FPAAs were also included in node systems in the same way as in the example above [106] except that these systems had FPGAs as the main controller. These are the typical examples of how FPAAs are utilised in WSN node platforms for supporting different analogue sensors, i.e. FPAAs are interposed between the sensors and the microcontroller of the node platform working as a sensor front-end or controller, like the role of the FPGAs in the systems discussed in the digital domain subsection due to their similar natures and they both need a host controller for device reprogramming.

PSoC devices were also used in a very similar way in WSN node designs like the FPAA examples discussed above.

An example is where the idea of applying the over-the-air (OTA) programming techniques was explored to remotely reprogramme the PSoC based sensor controller board developed to work with WSN nodes [110]. In this system the PSoC extension board was connected to the off-the-shelf WSN node called IRIS mote featuring an Atmel ATMega1281 microcontroller which has very limited analogue capabilities as described in Chapter 1, Section 1.3.2. This PSoC extension board could greatly expand the analogue capability of the IRIS sensor node. The firmware of the PSoC could be wirelessly transferred to the IRIS mote, and then the IRIS carried out the PSoC reprogramming process to enable the PSoC (thus the node system) to support different sensors.

Another WSN node reported [111] [112] also had a similar hardware structure as the example [110] described above. In this node system [111] [112], a PSoC device was also used for handling sensors and the PSoC was connected to an off-the-shelf WSN node called TelosB which has similar limited analogue capabilities as the IRIS mote mentioned in the example [110] above. The TelosB acted as the main controller of the node system. The hardware configuration settings of the PSoC could be pre-designed by users for sensors to be connected, and transferred wirelessly to the main controller and then to the PSoC for the reconfiguration process. In this design the hardware configuration settings could also be manually pre-loaded into the system for a specific sensor, and the PSoC could load the configuration settings based on the certain pre-defined sequence or logic. This second method is similar to the design in [105] described before.

In another example [113], a wireless sensor node platform was developed for high performance industry control applications. The platform was equipped with a PSoC for handling analogue sensors and actuators and a 32-bit ARM-based Atmel microcontroller as the main controller. The ARM controller communicated with the PSoC device via UART interface. Like the FPAA sensor front-end example in [106] above, this node platform also included programmable hardware because of the consideration that it could be reprogrammed to support different sensors, and there were also other designs included programmable hardware for the same reason [114] [115] [116].

Two similar sensor controller devices were described [117] [118] for supporting different analogue sensors. They were very alike in design concepts except that one was built with discrete configurable ICs and the other was based on the PSoC. Both were designed to be interposed between the analogue sensor and the processor of the sensor system, working as external signal conditioning circuits to make up for the shortage of analogue abilities of the node processor, similar to the role of the programmable hardware in the WSN platforms described above.

Similar to the two afore-mentioned sensor controller devices, there are also programmable ASICs which were developed to support different sensors [119] [120] [121]. These ASICs are normally designed for one specific application area, such as the biomedical area [120] or the entomology area [121], and their analogue capability and programmability are usually limited, for example one ASIC only features one amplifier and one ADC such as a single-input programmable gain amplifier and a 10-bit SAR ADC [119].

Overall, programmable hardware is employed in these systems or devices described above for the purpose of supporting different sensors and improving the system flexibility. However, a significant shortcoming of these kinds of systems is the fact that these devices must still be pre-programmed depending on the analogue sensor that will be connected to it. Also the programming process is executed by manual operation. These sensor nodes lack the ability to dynamically reconfigure its hardware to adapt to different sensors. This can greatly reduce the flexibility and limit the versatility of the system.

2.1.3.2.3 Programmable Hardware Literature Summary

Programmable hardware has been utilised in different WSN applications as it can benefit sensor systems in many different ways as presented in this subsection. The flexibility offered by reconfigurability is one of the main reasons that programmable hardware is widely employed.

However, in terms of utilising programmable hardware to generally support different sensors, especially analogue sensors, the existing WSN node technologies have shortcomings and the applicability of these designs is limited. These shortcomings of different designs have been commented in the subsection above, and they are summarised here and the reasons are also analysed to provide a clearer context of the RAWS technology based on the review so far.

The major limitation of the existing designs discussed above is that the configuration settings and circuit parameters have to be predesigned and programmed into the hardware for each sensor connected to the system. The hardware reprogramme or reconfiguration process is launched manually when switching sensors for the system, or the system automatically cycles through the pre-loaded hardware configurations for the pre-known sensors. So in theory these systems can support a large number of different sensors, but in reality it is not practical to achieve.

One of the main reasons for this limitation is that these WSN node technologies have no sensor identification techniques. Lacking the sensor identification capability implies that these node systems have no means to acquire the information of the connected sensor. Without sensor information, it is not possible for the node system to adaptively utilise its hardware resources to support sensors. As a result, hardware configurations have to be predesigned by developers for the known sensors that will be connected. Another reason is that these WSN node technologies do not have any adaptive reconfiguration techniques to manage how the programmable hardware reconfigures itself according to different sensors. So the system can only either wait for the manual instructions to start the reprogramme process to load in the hardware configuration required for the currently connected sensor or loops through the pre-loaded hardware configurations based on certain predefined sequences for a small number of specific sensors.

In summary, the existing WSN node technologies lack the ability to perform dynamic reconfiguration in an adaptive and autonomic way, therefore, it is not practical for them to support a large number of sensors in a flexible and scalable manner. On the other hand these capabilities are the key features and the novelties of the RAWS node research, and more details are discussed in the next subsection after another important part of the technology, the sensor identification scheme, is reviewed.

2.2 Smart Sensor Standards

As discussed in the research concepts section, the sensor identification scheme plays an important role for the RAWS node to utilise the reconfigurability of the programmable hardware. It provides the important sensor identification and characteristics information, and this information is critical for the adaptive reconfiguration techniques to instruct the hardware reconfiguration process.

Two approaches for realising such a sensor identification scheme are investigated in this research. The first approach is a systematic classification process for sensor identification. An analysis of the characteristics of different sensors was required to verify the feasibility of such a systematic classification process, and this analysis is presented in Chapter 4. The second approach involves smart sensor standards that provide analogue sensor self-identification and self-description capabilities. This section reviews the relevant standards, i.e. the IEEE 1451 standard family.

2.2.1 IEEE 1451 – Smart Transducer Standard Family Overview

IEEE 1451 is a family of standards for smart sensors and actuators. Sensors and actuators are generally referred as transducers. The aim of the IEEE 1451 family is to provide a framework for the design of smart transducers and the transducer control and data acquisition systems with network capability.

The 1451 family categorises the elements of a complete transducer system into two types of devices by introducing the concepts of the network capable application processors (NCAPs) and the transducer interface modules (TIMs). The standard members in the 1451 family are devised around these two device type concepts.

A TIM is a digital module that contains one or many sensors and/or actuators, the signal conditioning and analog-to-digital and/or digital-to-analog conversion circuits for these sensors and/or actuators, and an interface to the NCAP devices [122]. A TIM is a similar concept to a sensor node. Several different types of TIMs are defined in the 1451 family.

An NCAP device is like a gateway for the TIMs to the external user network. An NCAP has two types of interfaces. One is for connecting to the user network, such as a WLAN, an Ethernet, or the Internet. The other one is for interfacing TIMs. NCAPs function as bridges between different forms of TIMs and different types of user networks.

Besides TIM and NCAP concepts, the transducer electronic datasheet (TEDS) is the key concept of the IEEE 1451 family. The TEDS can contain the identification, characteristics, calibration, and manufacturer related information of the transducers within the TIMs. It describes the type, operations and attributes of TIMs for NCAPs, assisting the NCAPs and TIMs interfacing with each other.

The 1451 family defines the specifications for the two types of devices and the interface between them. The reference model of the IEEE 1451 family shown in Figure 16 illustrates the roles of different standards members.

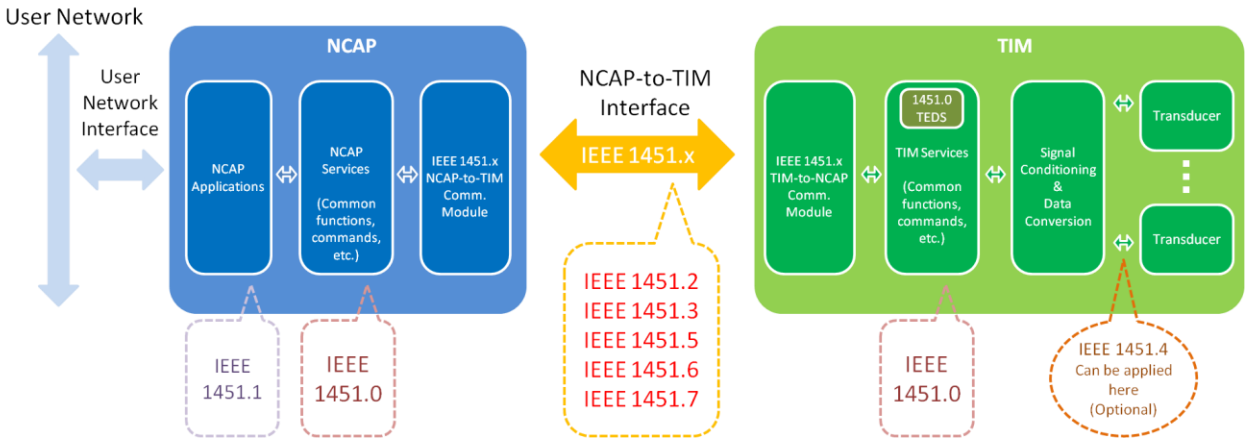


Figure 16 Reference model of the IEEE 1451 Standard Family

Different forms of TIMs with different TIM-to-NCAP interface specifications are defined by different members of the 1451 family. The 1451.2 describes the specifications of the first TIM

type introduced in the 1451 family, which is referred to as smart TIM (STIM). The 1451.2 standard also defines a digital point-to-point interface for the communications between STIMs and NCAPs. The 1451.3 defines a TIM type with a multi-drop bus interface. The 1451.5 introduces the concept of wireless TIMs (WTIMs) and defines the wireless communication methods and data format between WTIMs and NCAPs. The 1451.6 and 1451.7 define the TIM-to-NCAP interface based on the CANopen network and the RFID communications respectively.

The specifications of the common NCAP and TIM services, such as the common functions and commands for these devices, are defined in the IEEE 1451.0. The IEEE 1451.0 also introduces a general TEDS format for the 1451.5 and onward standards like the 1451.6 and 1451.7, and proposes to solve the format compatibility issue between the 1451.2 and 1451.3 standards. The 1451.1 defines the information model of the NCAP applications which serves as a software programming paradigm.

A more detailed discussion of 1451 family is in Appendix C. Overall, the 1451 members discussed above in essence define the interface and protocol specifications which aim to standardise the communications between TIMs (sensor nodes) and NCAPs (gateways). The TEDS defined in these standards are also for describing the TIM modules for the NCAP devices, i.e. for assisting the NCAP to know the connected TIM, to establish the NCAP-to-TIM interface, and to communicate and work with the TIM module.

The 1451.4 is an exception in the 1451 family. Unlike the other 1451 members, it is the only one that attends to the interface between the actual sensors and TIMs.

2.2.2 IEEE 1451.4 Standard

The IEEE 1451.4 standard focuses on traditional analogue sensors. The dot 4 standard is considered as optional in the 1451 family reference model. Its aim is to provide compatibility between traditional analogue sensors and IEEE 1451 systems. But it has unique features which can be utilised to facilitate the WSN node platforms in supporting analogue sensors, and the key feature is the dot 4 TEDS concept. As will be explained below, in essence this concept adds the digital TEDS for sensor self-identification and self-description in parallel with the analogue

output, providing an approach to recognise analogue sensors and obtain the important sensor characteristics as shown in the left part of Figure 17 below.

The IEEE 1451.4 standard firstly introduces the mixed-mode interface concept which allows the transducer incorporate both the analogue interface and the digital interface. The analogue interface provides signals representing a physical phenomenon and the digital interface is for reading the transducer electronic data sheet (TEDS). This concept is a foundation for the dot 4 TEDS concept to be widely accepted because it allows the TEDS to be added to sensors in a very simple way. The combination of the reliability, robustness and cost-effectiveness of analogue sensors and the intelligence of digital components can bring great benefits to sensor applications.

IEEE 1451.4 is unique among the smart sensor standards because it maintains the analogue output of the sensor. This feature makes the dot 4 standard flexible as it allows for seamless use with the traditional analogue sensors without changing the original design, and the advantages of the analogue sensor can be kept. Also a sensor system can be implemented more easily and flexibly without lots of complicated requirements and restraints as in the digital sensor standards. Figure 17 below illustrates the diagram of a general IEEE 1451.4-capable sensing system.

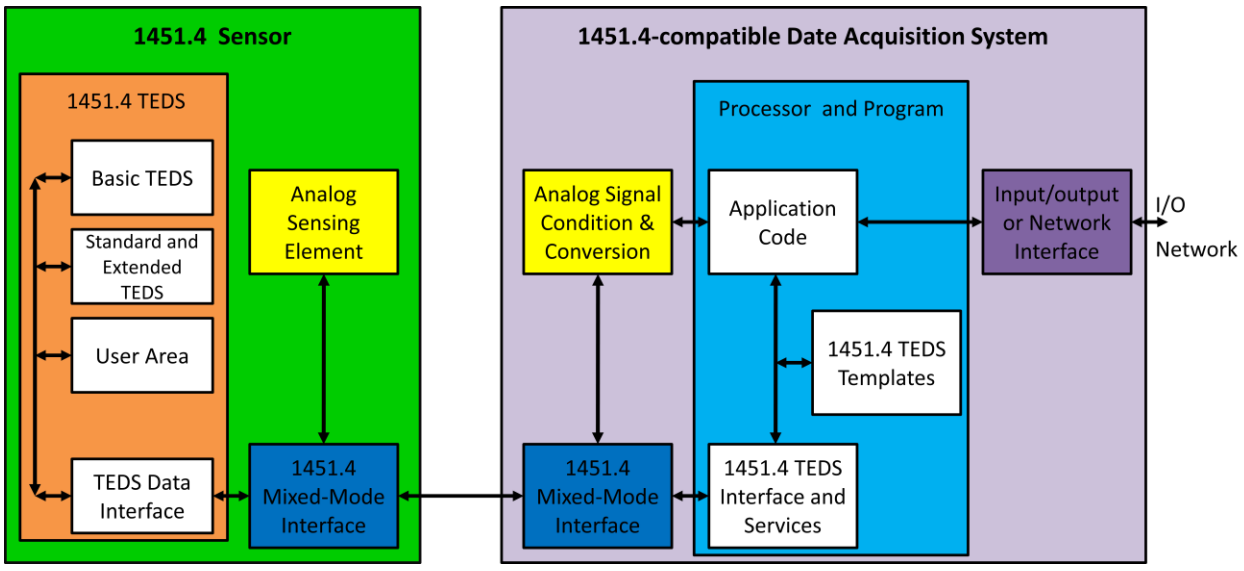


Figure 17 Block Diagram of a general IEEE 1451.4-capable sensing system

The dot 4 TEDS is the key concept that has been utilised by researchers and manufacturers. The dot 4 TEDS contains the identity information for the analogue transducer such as the manufacturer ID, model number, and also describes important attributes of the transducer such as the transducer type, physical measurand, measurement range, electrical output range. The TEDS information is categorised into sections and has a standardised structure and format as shown in Figure 18 (a). The identification information is contained in the first section called Basic TEDS which can uniquely identify the transducer. The transducer characteristics information is contained in the Standard TEDS section which follows the Basic TEDS section. The dot 4 TEDS provides a simple mechanism for adding self-identification and self-description features into analogue sensors. These two key features enable data acquisition systems to recognise sensors more easily and interpret the analogue information correctly.

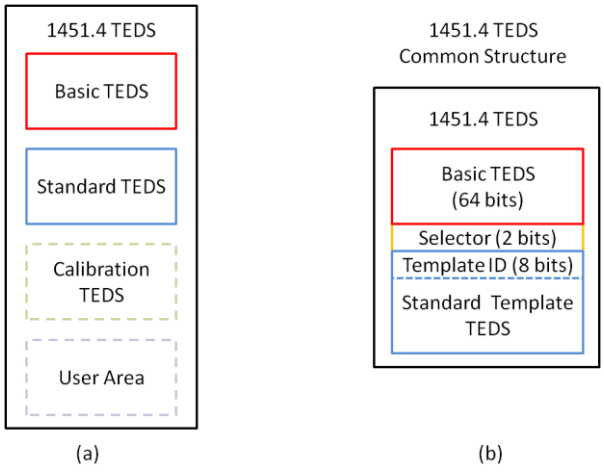


Figure 18 1451.4 TEDS Structure

The dot 4 TEDS can also include the optional Calibration TEDS and user area sections as shown in Figure 18 (a). The Calibration TEDS section is for storing calibration information such as a calibration table or polynomial. The calibration information can be used to compensate errors such as nonlinear errors, gain and offset errors and help to achieve high accuracy. The user area can be used to store custom information, e.g. the sensor location or the calibration date. These two information sections have not been widely utilised. Therefore, the common 1451.4 TEDS structure contains the Basic and Standard TEDS as shown in Figure 18 (b).

The 1451.4 TEDS is different from the general 1451 TEDS format. As discussed in the previous subsection, the general TEDS defined for the other 1451 standards are for providing the information of the TIM to the NCAP device, assisting the interfacing between TIMs and NCAPs. The general 1451 TEDS includes a large amount of miscellaneous information about the TIM modules which can help the NCAPs understand the TIMs but will complicate the TEDS information processing procedure and also the whole sensor system. The 1451.4 TEDS however focuses on describing the sensors themselves in a concise way. The dot 4 TEDS is defined to cover a wide range of sensor types and keep memory usage to a minimum at the same time through the use of the 1451.4 TEDS templates as shown in Figure 18 (b). The dot 4 TEDS templates specify the data structure and format of the sensor characteristics information in bitwise form to maximise the memory usage. A set of templates for different sensor types is defined in the dot 4 standard. The standardised templates can be stored in the DAQ systems for interpreting the dot 4 TEDS so no formatting, declaration, and commentary information that describes the TEDS itself has to be stored within the sensor TEDS, further reducing the memory usage and avoiding unnecessary data transmission to save power. The maximum size of the dot 4 TEDS mandatory fields is less than one third of the minimum size of the general 1451 TEDS.

Also the dot 4 standard adopts a simple addressable and multi-drop serial digital protocol called '1-Wire' which only needs one signal wire and one ground wire for data communications. All these features enable the 1451.4 TEDS to be implemented with small size, low cost and low power consumption memory devices, reducing the complexity of adding the dot 4 TEDS feature into sensor systems. Commercial low cost EEPROM solutions are already available for adding TEDS to existing analogue sensors so that the traditional analogue sensors can be easily equipped with self-description and self-identification capability [123] [124] .

Additionally, the standard allows the TEDS to be either physically or virtually associated with the transducer [125]. This means that the TEDS can either be stored in an embedded memory located on any place near or within the sensor or be placed in a remote digital medium. This flexible TEDS concept can further extend the benefits of the TEDS for the legacy sensors as well

as for the applications where it is not practical to physically place the memory device together with the actual sensing element.

Compared to other smart sensor standards, e.g. IEEE 1451.2, which have considerably complicated constraints that ultimately led to virtually no adoption in practical applications [126] [127], many features and concepts of the IEEE 1451.4 are flexible to employ. The dot 4 standard provides a way to add smart features into existing analogue sensors or to develop new smart sensors with minimal effort [125]. Compared to the other 1451 standards with restricted rules, the risks of adoption associated with the more flexible IEEE 1451.4 standard for sensor manufacturers is low [128] [129].

IEEE 1451.4 standard has been supported by more than 40 manufacturers worldwide including many major sensor manufacturers and data inquisition vendors such as Honeywell, HBM, B&K, Endevco, Kistler, National Instruments, VTI Instruments, m+p International, Maxim/Dallas Semiconductor, and so on [130]. Common sensor products built on the 1451.4 standard include thermocouples, RTDs, force sensors, pressure sensors, accelerometers, vibration sensors, and microphones [131] [132] [133] [134] [128]. The 1451.4 TEDS has been supported in different types of DAQ systems and devices such as signal conditioners, recorders, and analysers from manufacturers like National Instruments, Endevco, B&K, VTI Instruments, HBM and many others [133] [134] [135] [136] [137] [138] [139]. Other 1451.4 compliant hardware such as PC based 1451.4 TEDS readers/writers and software such as TEDS editors from many different manufacture such as The Modal Shop, HBM and B&K has also been developed to provide a convenient approach to view and programme the 1451.4 sensors' TEDS information [140] [141] [142] [143]. Due to the low cost and low complexity of adding TEDS to traditional analogue sensors as described above, there are also manufacturers and third party companies provide this service to turn existing analogue sensors into dot 4 compatible [144] [145] [134]. The 1451.4 TEDS has also been supported from the software perspective in the industry, for example National Instruments has added the 1451.4 supports into their programming environment LabVIEW and many other manufacturers also support TEDS in their DAQ software [146] [147].

This research employs the self-identification and self-description features of the dot 4 standard to build a sensor identification scheme because these features ideally fit into the context of this research. The RAWS nodes parse the TEDS to obtain the sensor characteristics information so that the platform can reconfigure its programmable hardware system accordingly. Also the 1451.4 is an IEEE standard that already has support from many sensor manufacturers. With the rapid development of the IoT applications, IEEE 1451.4 could have even wider supports.

However, the RAWS node technology does not solely rely on the dot 4 standard because the concept behind the dot 4 TEDS is the more important element for this research, i.e. using a simple but flexible mechanism to supply identification information and the important characteristics of analogue sensors. The RAWS node technology could just as easily employ other kinds of sensor self-description formats by updating the RAWS node firmware (e.g. IEEE 1451.4's successor standard).

2.2.3 IEEE 1451.4 Literature

The 1451.4 standard as a relatively new standard is not very widely employed yet in academic research and in the wireless sensor network area. This review summarises the 1451.4-related research and applications and provides an overview of how the dot 4 standard has been utilised.

2.2.3.1 Literature

Some research projects studied the 1451.4 standard itself and its possible usages. For example a web-based virtual learning software tool was developed with the National Instruments LabVIEW for helping students to understand the 1451.4 TEDS and the different types of templates defined for different types of sensors [148]. Also, two new 1451.4 TEDS templates were proposed and implemented for electronic tongue devices and the gas sensors of the electronic nose system respectively [149] [150]. Another example describes a new approach to store the 1451.4 TEDS information has been reported which encodes the binary TEDS data into printable 2D bar codes [151]. The National Instruments LabVIEW is used for the conversion between the TEDS data and the 2D bar codes.

1451.4 compatible devices were developed in sensory applications and research projects. A 1451.4 TEDS 1-Wire interface prototype for reading the TEDS information was built based on a

single board computer (SBC) [152]. This interface prototype was developed by the National Institute of Standards and Technology (NIST) researchers who drafted the 1451 standards as an example prototype after the dot 4 standard was released. An example of reading the dot 4 TEDS from a 1451.4-enabled accelerometer with this prototype was presented. This prototype employed discrete components for digital functions including a pulse generator, a latch, etc., to constitute the dot 4 TEDS interface hardware part and the SBC controlled these digital components. It also contained on-board analogue ICs including an ADC to support the accelerometer used in the system. Other 1451.4-compatible modules [153] [154] [155] [156] were also reported with similar functionalities to the SBC based TEDS 1-Wire interface prototype described above. These modules were based on some kind of microcontroller. Some of them [153] [154] could read TEDS from the 1451.4-enabled sensors while some others [155] [156] could have on-board TEDS memory chip for adding the TEDS feature to traditional analogue sensors. They could be connected to the processor of the system, e.g. communicating with PCs via the USB interface for displaying the TEDS information [153] [154], or can be connected into an external network through the on-board network interface such as Ethernet port [155] [156]. Like the SBC based prototype above [152], these modules also contained onboard analogue signal conditioning circuits built with analogue ICs and other passive components for supporting the specific sensor used in the system. There also are some other interface devices or sensor systems claimed to be 1451.4-compatible but without implementation details [127] [157] [158] [159] [160] [161] [162] [163] [164] .

Custom made 1451.4 compatible sensors were also developed in research projects and 1451.4 TEDS was incorporated in these designs for describing sensor attributes such as a vibration sensor based on digital MEMS accelerometer [165] , radiation sensors including [166]

The 1451.4 TEDS were also incorporated into custom made sensors in research projects, for example, vibration sensor based on digital MEMS accelerometer [165] , radiation sensors including a Geiger-Muller detector and a ionisation chamber [166]

In some sensor applications and research projects, the IEEE 1451.4 TEDS has been simply used for the identification feature in order to distinguish the sensors or sensor nodes in a network.

A low power transmit-only wireless sensor node was developed for temperature sensing [167]. This WSN node had a microcontroller with built-in RF transmitter as the controller of the node and an on-board digital temperature sensor. The 1451.4 TEDS concept was employed to allow the network gateway to distinguish the sensor nodes. Each node was assigned a dot 4 TEDS style ID. This ID has the same format as the 8-byte Basic TEDS section of the 1451.4 TEDS. This 8-byte TEDS ID was sent to the network gateway in addition to the temperature readings, and it was used by the gateway to differentiate the source of the temperature data. This design chose the dot 4 TEDS over the general IEEE 1451 TEDS introduced by 1451.0 because the dot 4 TEDS is much more concise. This feature can reduce the data packet size, helping to keep the power consumption to a minimum.

A microcontroller based sensor system for monitoring the vital signs and motor activity of wheelchair users was reported where the 1451.4 TEDS was also utilised to distinguish the sensors used in the system described by [168] [169] [170] [171]. Several analogue biomedical sensors and an accelerometer were used in the system. Each sensor was attached to an extension board with an on-board 1-Wire EEPROM which stored a custom-compiled 8-byte Basic TEDS for the attached sensor, turning the traditional analogue sensors into 1451.4 enabled sensors. Similar to the temperature sensing nodes discussed above [167], the minimal Basic TEDS was also used as an ID for this system to differentiate the measurement reading sources. The system had separate signal conditioning chains for each sensor in the system which is how the traditional approach copes with multiple analogue sensors.

The self-description feature of the TEDS has also been utilised to provide the sensor attribute information for the sensor systems. In these cases, the TEDS is more than an ID, and not just the first 8-byte Basic TEDS section was used.

A wireless sensor network for monitoring indoor air quality was developed and the 1451.4 TEDS was utilised to assist the measurement data processing [172]. Each sensor node of this network contained three sensors for detecting the intensity of the deoxidising gases and measuring the temperature and relative humidity of the air respectively. The dot 4 TEDS for these sensors were created via LabVIEW and stored in the virtual TEDS form on a PC which was also the

gateway of the network. Unlike the minimal dot 4 TEDS used in the aforementioned research and applications, these TEDSs have the most common form of the dot 4 TEDS, i.e. they include the Basic TEDS section for identification information as well as the Standard TEDS section for sensor parameters and attributes. Based on the sensor characteristics provided by the TEDSs, the LabVIEW programme on the PC processed the raw data retrieved from the sensor nodes and presented meaningful readings for the users.

In a wired surveillance sensor network [173], the 1451.4 TEDS was utilised for discovering the types of the sensors to provide relevant data in the example of the robotics navigation application. The surveillance network mainly consisted of cameras, but there were also other types of sensors like infrared sensors. Each of these sensors was equipped with a modified version of the 1451.4 TEDS which contained additional information including the sensor location. Based on the TEDS information, the system could discover the cameras in the network and their geographical locations, and could then select the suitable cameras to provide the obstacles information in the experiment environment for guiding the robot to reach the destination. The same modified 1451.4 TEDS version was also utilised for the same purpose in a similar surveillance network [174] which was tuned for the position tracking of the robot instead of navigation.

In some applications, the TEDS concept has also been borrowed to describe other devices rather than sensors. For example, in a small form factor satellite developed for testing and comparing different memory devices in space environment [175], the TEDS concept was extended to develop a TEDS-like structure for describing the important features and attributes of the components of the satellite's communication subsystem. Based on the information provided by this TEDS-like structure, the microcontroller of the communication subsystem can set the parameters of different components such as the working mode of the transceiver as required.

2.2.3.1 IEEE 1451.4 Literature Summary

The sensor self-identification and self-description capabilities are the two key features that the IEEE 1451.4 TEDS provide, and the previous subsection presents a review of how these two key

TEDS capabilities are used. As described above, some sensory applications and research projects utilise the self-identification feature and employ the TEDS as ID to facilitate the sensor systems to distinguish the known sensors in the system [168] [169] [170] or the sensor nodes in the network [167]; some based on the TEDS description obtain more information about the sensor to determine the type of the sensors in the network [173] [174] or assist the processing of the sensor raw data [172]. However, none of them exploits the 1451.4 TEDS concept to develop a sensor identification scheme that enables the wireless sensor node platform to identify the connected sensors which are unknown to the platform, and none of them combines the TEDS concept with programmable hardware.

The literature review finds that there are a few WSN research projects and applications [117] [176] [177] that investigate the combination of programmable hardware technology and the other members of the 1451 standard family (such as 1451.3, 1451.5) to develop 1451 compatible TIMs. But as discussed in the previous section, the main purpose of these 1451 family members is to develop standardised interface protocols between the TIMs and NCAPs. So the TIMs developed in these research projects can automatically interface with compatible 1451 NCAPs. But in terms of supporting sensors, they still do not have the capability to identify sensors or obtain important sensor information. So even though they employ programmable hardware, they cannot adaptively accommodate sensors in an autonomic way. The hardware configurations have to be manually pre-designed and pre-programmed for different sensors or the system automatically cycles through the pre-loaded configurations.

This review found one sensor system that did use the IEEE 1451.4 standard in combination with the programmable hardware technology to support sensors [108]. However, this system also had similar issues to the ones just mentioned and the main shortcoming was that the hardware configurations had to be pre-designed and pre-loaded for the known sensors used in the system. The minimal version of the dot 4 TEDS version, i.e. the 8-byte Basic TEDS, was used as ID for the sensors in a similar way as in [167-170]. The system just switched between configurations based on the sensor's dot 4 TEDS style ID. Also the system was not designed for or suitable for WSN applications due to the very high power consumption. This review also

found a 2013 research publication [178] that described a wired sensor platform that utilises the programmable hardware to support 1451.4 compatible sensors. This paper cited a paper which was published in 2012 from the RAWS node research [179].

As discussed before in the programmable hardware literature subsection 2.1.3, even though there are many WSN node platforms equipped with programmable hardware are designed to be flexible to generally support different sensors, but in reality it is not practical for them to achieve. One of the main reasons is that they lack the means to acquire the information of the connected sensor. In this section, the literature review identifies that the 1451.4 TEDS can provide sensor self-identification and self-description capabilities hence the possibility for WSN node system to acquire important sensor information. However, the literature review also shows that even though some of the WSN designs have been used TEDS to distinguish a small set of known sensors in the network, the TEDS concept is not fully exploited. It is not utilised to develop a complete sensor identification scheme for the WSN node platform to generally identify unknown analogue sensors and acquire the critical attributes, whilst such a scheme can be further utilised as the foundation for developing high-level techniques for adaptive and autonomic supporting different sensors like the RAWS node technology does.

Therefore, in terms of supporting different sensors, shortcomings of the traditional WSN platforms still exist. Even though the many WSN node systems have programmable hardware as infrastructure, it is still not possible for them to adaptively utilise their hardware resources without the important sensor information and attributes. Furthermore, they do not have the techniques to autonomically execute reconfiguration processes to adapt to different sensors. Different hardware configurations still have to be predesigned for the known sensors that will be connected, and hardware reprogramme or reconfiguration processes have to be manually managed or predefined. These limitations can greatly reduce the flexibility and applicability of the WSN system.

In summary, the review indicates that no research work has been reported which combines both the dynamic reconfigurability of programmable hardware technologies and the self-

identification and self-description capability of IEEE 1451.4 to develop a wireless sensor node technology with autonomic hardware reconfiguration techniques for adaptive support of analogue sensors, whilst this is the main goal of the RAWS node research. The key novelty of the RAWS node platform is that it is able to acquire and parse information of the connected sensor and extract the important attributes, and then based on the sensor information dynamically reconfigure the hardware resources to intelligently build the suitable signal processing chain which aims to achieve optimised measurement performance for the connected sensor, realising an autonomic multi-sensing capability for the WSN node to support sensors in a flexible and scalable way.

2.3 Wireless Communication Protocols

Wireless communication capability is one of the key features of wireless sensor networks. Wireless communication technology plays a key role in establishing a reliable WSN with good energy efficiency, long life time, and flexible network structure. Even though the main focus of the RAWS node research is not the wireless side of WSNs, the wireless networking and communication is still an important part of the research. A suitable wireless technology allows the research to quickly set up a WSN environment for prototype testing, and more importantly it is one of the critical factors to control the power consumption of the RAWS nodes at the hardware level. This section analyses and compares different wireless communication technologies and protocols involved in the WSN area, concluding with the selection of a wireless technology suitable for the RAWS node prototyping.

2.3.1 WSN Communication Characteristics & Protocols Selection Considerations

The wireless communications in the WSN area have unique features because of the characteristics and the requirements of wireless sensor networks and WSN applications. One of the most important characteristics of WSNs is the limited energy budget. Therefore, one of the key requirements for the WSN wireless communication is low power as it is one of the most power-consuming operations for WSN nodes. In fact, the typical energy for transmitting a single bit can be more than 2000 times that for a microcontroller computing one instruction [180] [181, 182].

The WSN wireless communication also has the following features.

1. Transmission Range

   WSN nodes usually have small transmit power and thus a relatively small transmission range. On the other hand, sensor nodes usually are not deployed in a way that will leave long distances between each node. The requirement for sensor node transmission range is typically on the order of several meters to hundreds of meters, which falls into the transmission range definitions of the wireless personal area network (WPAN) and the wireless local area network (WLAN).

2. Data Rate & Duty Cycle

   Communications in many WSN applications are low data rate, and also are low duty cycles, i.e. transmissions happen infrequently. One reason for these two features is that the usually limited energy budget of the WSN nodes cannot sustain high throughput for long duration continuous transmissions. However, it is also because in many WSN applications the sensor nodes only need to wake up periodically in minutes or even hours or days, or only when an event has been detected. Therefore, the transmission can be intermittent or irregular with long time intervals, and there may not be a large amount of data to be transmitted.

3. Reliability

   Although the data rate requirement is not very high, the WSN communications have to be reliable since the wireless transmission conditions may not be ideal and re-transmitting will waste power.

4. Network Topology

   Flexible network structures and topologies are needed to cope with the non-ideal complicated transmission environment.

5. Network Capacity

   The wireless network capacity cannot be too small as a large number of sense nodes might be required.

The features discussed above are also the main considerations for choosing the wireless protocols in this research project.

As discussed in Chapter 1, RF based wireless communication can meet the requirements of most WSN applications because it can offer low energy wireless transmission while providing adequate data rate, transmission range and acceptable error rate in non-ideal conditions such as indoor environments. Therefore, this research focuses on the RF based short range (WPAN and WLAN range) wireless technologies and protocols.

There were no standardised wireless protocols that had been widely used at the early stage of the WSN history (around 1999 to 2003). Most of the WSN sensor node platforms employed

some sort of simple low power single chip RF transceiver such as the RFM TR1000 and TI (Chipcon) CC1000. These transceivers provide platforms based on which software implementations of user-defined Medium access control (MAC) sublayer protocols and higher layer protocols can be developed, i.e. researchers and developers need to build a communication protocol stack upon the physical (PHY) layer of the transceivers.

As the wireless technologies began to progress quickly, many WPAN and WLAN standards and especially some IEEE standards became significant in the WSN area. These IEEE standards include: IEEE 802.15.1, 802.15.4, and 802.11 standards. These standards defined the PHY and MAC layers specifications for a range of wireless communication protocols which are widely utilised in wireless applications including Bluetooth family over IEEE 802.15.1, ZigBee and a few others over IEEE 802.15.4, Wi-Fi over IEEE 802.11. There are also many proprietary protocols aiming for the WSN area, such as ANT and Z-Wave. The next subsection provides a comparative analysis of these wireless standards and protocols. A more detailed review is presented in Appendix D.

The analysis and review focus on the features important to this research including the ones discussed above, like the power performance, transmission range, transfer rates, network topologies and scalabilities. Some other features are also considered, like network setup complexities, the availability and ease-of-use of the transceivers. The low level technical details such as modulation or multiplexing techniques will not be emphasised.

2.3.2 Wireless Protocols Analysis

2.3.2.1 Protocol Specification Overview

The wireless standards and protocols mentioned in the previous subsection can cover most short range wireless applications, i.e. WPAN and WLAN applications with the communication range on the order of several meters to hundreds of meters, including the wireless sensor network applications. These protocols are listed in Table 2.

IEEE 802.15.1 based	Bluetooth, Bluetooth Low Energy
IEEE 802.15.4 based	ZigBee, 6LoWPAN, WirelessHART, ISA100.11a
IEEE 802.11 based	Wi-Fi
Other	ANT, Z-Wave, etc

Table 2 WSN Wireless Protocols

Each of these protocols has unique features and all these protocols have been employed in wireless sensor node platform developments to a greater or lesser extent. Their specifications and application in WSNs are summarised in Table 3 and Table 4.

	Bluetooth	Bluetooth Low Energy	ANT	ZigBee/802.15.4	Z-Wave	Wi-Fi
Freq Band	2.4GHz	2.4GHz	2.4GHz	868/915MHz,2.4GHz	868/915MHz	2.4GHz, 5GHz
Max Data Rate	1 Mbps(BR) 2or 3Mbps(EDR)	1Mbps	1Mbps	250kbps	<100kbps	54Mbps(802.11a/g) 11Mbps(802.11b) 150 ~ 600(802.11n)
Max Throughput	721.2kbps(BR) 2.1Mbps(EDR)	~300kbps (in theory) < 100kbps (most devices)	Up to 20kbps	<130kbps	~40kbps	~24Mbps(a/g) 5.9~7.1Mbps(b) ~100 to 300Mbps(n)
Basic Topology	Piconet	Piconet	Star	Star	Mesh	BSS, IBSS
Max Nodes	8	>2 billion	>2 billion	>60,000	232	Device depended
Ext Topology	Scatternet	N/A	Tree, Mesh	Tree, Mesh	N/A	ESS
Typ Tx Power	4dBm (Class 2)	~0dBm	~0dBm	0~3dBm	-5~2dBm	10~18dBm
Approx Range	<10m (indoor) <30m (outdoor)	<10m (indoor) <30~50m(outdoor)	<10m (indoor) <30m (outdoor)	10~30 (indoor) 70~100 (outdoor)	<30 (indoor) <100 (outdoor)	30~50 Indoor,b/g 90~120outdoor, b/g
Tx Current	~30 to 50mA	~12 to 20mA	~15 to 28mA	~24 to 31mA	~24 to 40mA	~180 to 300mA
Rx Current	~ 30 to 50mA	~14 to 18mA	~17 to 24mA	~19 to 24mA	~23mA	~30 to 90mA
Sleep Current	50~100uA much higher	<1uA	<1uA	<1uA	~1 to 2.5uA	<6uA

Table 3 Comparison of Different Wireless Protocols

Wireless protocols	WSN platforms
IEEE 802.15.4 and 802.15.4 based or compatible protocols Including ZigBee, 6LoWPAN, WirelessHART	BSN node[183], Cookie[73], Egs mote[184], Ember node [185], EPIC Mote[186], FireFly[187], Imote2[188], Indriya_DP_03A Series[189], Indriya_DP_03B Series[189], Indriya_DP_01A Series [190], Indriya_DP_04A11 [191], Iris/XM2110CB [192], Jennic JN5148[193], Lotus[194], MeshScape[195], n-Core[196], OpenMote[197], panStamp AVR[198], Pluto[199], PowWow[200], Preon32[201], SAND[202], SenseNode[203], SensiNode[204], Shimmer[205], SmartMesh PM2511/LM2650/Linear LTC58xx/LTP59xx [206], Smartmote[207], SunSPOT[30], Telos [208], TelosB [209]/Tmote Sky CM3000/CM4000/CM5000[210]/Kmote[211], Tyndall 25mm Mote[38], Waspmote[212], WiSense[213], WisMote[214], WMSN.P-2812[215], XM1000[216], XYZ[217], Zolertia Z1[218]
Bluetooth (IEEE 802.15.1)	BTnode[219], Egs mote[184], eWatch[29], n-Core[196], SAND[202], Shimmer[205], Tyndall 25mm Mote[38], Waspmote[212]
General RF communications using non-standard format	BEAN [220], BTnode[219], CIT Sensor Node[221], Dot [208], DSYS25[222], Eco[223], Everlast[224], EYES [225], EyesIFX v1 [226] / EyesIFX v2 [226], Fleck [227], G-Node G301[228], i-Bean[229], M1010[229], MANTIS Nymph[230], Mica[180]/Mica2/MicaZ/Mica2Dot[231], MiniMote [232], MITes[233], Muller[234], panStamp NRG[235], Particle series [236], ScatterWeb[237], SNoW5[238], TCM120 [239], Tinynode[240], Tyndall 25mm Mote[38],Tyndall 10mm Mote[241], uNode[242], weC/Rene [208],
Wi-Fi	Indriya_DP_03A Series[189], Waspmote[212]

Table 4 Wireless protocols for existing WSN platforms

2.3.2.2 Bluetooth

Bluetooth (classic Bluetooth) is one of the most widely implemented wireless technologies and has been also employed by many WSN platforms as shown in Table 4. Because of the wide adoption of Bluetooth technology, one benefit for the WSN platforms to employ Bluetooth as the wireless interface is the wide compatibility with a large number of Bluetooth devices such as laptops and mobile phones which can be easily turned into Bluetooth WSN gateways.

Originally designed to replace cables for data exchange, classic Bluetooth has several features to provide a robust and reliable wireless connection even in industrial conditions. Because it is

designed for data exchange, classic Bluetooth devices can provide relatively high data rate, but this leads to high power consumption as well. The typical transmit current of the common Bluetooth Class 2 devices already goes up to around 30mA to 50mA while the transmission range is only around 30 to 50 meters outdoors and less than 10 meters indoors. For the Class 1 devices with 100 meters or higher transmission range, the transmit current is normally more than 100mA or even more than 200mA for some devices. Also, the classic Bluetooth devices do not always support sleep mode, and even for some new device models the sleep current is around 50uA to 100uA which is not considered low.

The basic network configuration of the classic Bluetooth, the piconet which is like a star topology, can only support a maximum of 8 devices which is very limited for most WSN applications. Even though a number of classic Bluetooth piconets can constitute a bigger network called scatternet, which is similar to a cluster tree topology, data routing between the piconets of a scatternet are not natively supported by classic Bluetooth devices. The nature of the scatternet determines the high complexity of establishing a large size classic Bluetooth network.

In summary, classic Bluetooth can be a choice for wireless networks with moderate data rate requirement and abundant energy budget but where there is not a high demand on network capacity and topology.

2.3.2.3 Bluetooth Low Energy

Bluetooth Low Energy (BLE) is also becoming one of the most widely implemented wireless technologies after the burgeoning adoption in 2013. This is an advantage for WSN platforms with BLE in terms of interoperability.

Bluetooth Low Energy has the lowest power consumption among the protocols discussed here. Compared to classic Bluetooth that was designed for replacing cables in data exchange, BLE aims for low power applications and adopts different techniques to make sure BLE devices can operate at ultra low power consumption, e.g. BLE adopts different radio techniques and a different set of protocol stack to reduce the complexity of creating a connection between devices and reduce the connection time.

BLE has a much bigger network capacity than the classic Bluetooth and most other protocols discussed here. In theory, more than 2 billion nodes can be contained in one BLE piconet so a large size network can be formed. But BLE only supports the piconet (star) topology and does not support multi-hop transmission. The point-to-point range is usually less than 10 meters indoor and 30 meters outdoor. These features reduce the effective network coverage and the benefit of the huge network capacity. Although user defined high level protocols over BLE and software programmes can be developed to equip BLE with more complex networking and routing capabilities, the complexity of establishing such a BLE network is also increased.

The data rate of BLE is much lower than the classic Bluetooth. The maximum practical data rate of BLE is about 300kbps in theory but for most BLE devices this number is well under 100kpbs, normally only 30kpbs to 60kbps, and average practical date rate is less than 20kpbs.

All these features indicate that BLE can be a choice for WSNs that need to be ultra low power but have relatively low requirements on other features such as data rate, flexible network structure, and large coverage.

2.3.2.4 ANT

The proprietary ANT wireless protocol is similar to BLE in many ways. ANT also aims for ultra low power, low data rate, low duty cycle applications and ANT has a similar practical data rate, transmission range, and network capacity as BLE. Due to the different techniques adopted at the PHY layer, ANT may not be able to cope with the RF interference as well as BLE and the power consumption is also higher. But ANT can provide more flexible network topology options as it supports tree, mesh topologies in addition to the basic star network.

However, a big disadvantage for ANT and many other proprietary solutions, such as Z-Wave and other similar protocols which are not listed in this review like MeshScape, SOWNet, SensiNet, is the limited support from academia and industry. From the perspective of developing a WSN technology with wide applicability, a standardised wireless technology is preferred in this research because the standard solutions can ensure interoperability and future proofing.

2.3.2.5 Wi-Fi

Wi-Fi (802.11) is the most widely implemented wireless technology and has become the synonym for WLAN. A huge benefit for sensor node platforms equipped with Wi-Fi based communications is the direct interconnectivity with the enormous number of Wi-Fi networks. Wi-Fi based sensor nodes can utilise the existing Wi-Fi network infrastructure and there is no need to build a separate framework for the wireless sensor network from the ground up. Also the Internet connectivity can be easily set up for Wi-Fi based WSNs and this can facilitate the data and information collection from the WSNs. Another big advantage of Wi-Fi is the very high data rate. Generally the raw data rate of embedded Wi-Fi modules can be up to 100Mbps.

The main issue for Wi-Fi in many WSN applications is its power consumpsion. Wi-Fi is designed to be high data rate rather than low power. The transmit power of embedded Wi-Fi modules or SoCs is usually 10mW to 63mW (10~18dBm) and the transmit current is around 180mA to 300mA.

Wi-Fi has many advantages but because of the very high power consumption, it tends to be restricted to WSNs with a plentiful energy budget and very high date rate requirements.

2.3.2.6 IEEE 802.15.4

The IEEE 802.15.4 has been the most widely used PHY and MAC layers standard in wireless sensor network research and also widely supported by manufacturers. Many low-rate short range wireless protocols are built upon the IEEE 802.15.4 PHY and MAC layers including ZigBee, 6LoWPAN, WirelessHART, ISA100.11a, as well as some proprietary ones like MiWi, DigiMesh.

The techniques employed by IEEE 802.15.4 ensure the simplicity of PHY and MAC layer protocols and allow the protocol to be implemented with low cost, low complexity and low power consumption. The standard allows wireless connections to be established between devices with little or no infrastructure which enables power efficient wireless communications for devices with limited energy resources. The transmit current of IEEE 802.15.4 devices is usually around 24mA to 31mA and some models can achieve less than 15mA. The receive current is usually around 19mA to 24mA and the deep sleep current can be lower than 1uA.

The IEEE 802.15.4 PHY and MAC layers protocol provides a scalable low data rate from less than 20kbps to around 100kbps to suit different energy budgets. The transmission range is around 10 meters to 100 meters. The network capacity is more than 60,000 nodes. Also a flexible peer-to-peer topology is defined in the standard to provide the foundation to support multi-hop transmissions and to implement a variety of complex topologies such as tree or mesh topology.

2.3.3 Analysis Conclusion and Protocol Selection

As discussed before, many WSN applications have the following requirements for wireless communications:

- Low power consumption as a key requirement
- Normally a low data rate, typically on the order of tens to hundreds of kilobits per second
- Low duty cycle, usually the wireless transceiver will stay in the sleep mode as long as possible for energy conservation
- The point-to-point transmission range is usually on the order of several meters to hundreds of meters
- Flexible network topologies for non-ideal transmission environments
- Big network capacity
- Low complexity

Based on the protocols analysis in the last several subsections, the features of IEEE 802.15.4 fit well with the above WSN wireless communication requirements. Compared to classic Bluetooth (802.15.1) and Wi-Fi (802.11), IEEE 802.15.4 devices have much lower RF power consumption while offering similar transmission range, and the sleep current is lower as well. The scalable data rates of the IEEE 802.15.4 device are much lower than Bluetooth and Wi-Fi but still adequate to meet the requirement for typical WSN applications. Compared to BLE, IEEE 802.15.4 solutions can offer better practical data rate and longer transmission range, plus the multi-hop transmission mode can effectively expand the data transmission range and the network coverage. It also supports more flexible network topologies than BLE. As indicated in Table 4, IEEE 802.15.4 has been broadly accepted as the MAC and PHY layer standard for many

WSN platforms. Therefore, because of the above-mentioned advantages, the RAWS node research focuses on the IEEE 802.15.4 based wireless protocols.

Within IEEE 802.15.4 based protocols, ZigBee is the most widely adopted one. ZigBee inherits the low power, low complexity features from IEEE 802.15.4 and defines the upper layer specifications including the networking, routing, and security functions as well as the tree and mesh topologies on the top of 802.15.4. ZigBee completes the full set of protocol layers on top of IEEE 802.15.4 (Figure 19) and frees researchers and developers from repeatedly designing their own non-standardised upper layer functions and protocols.

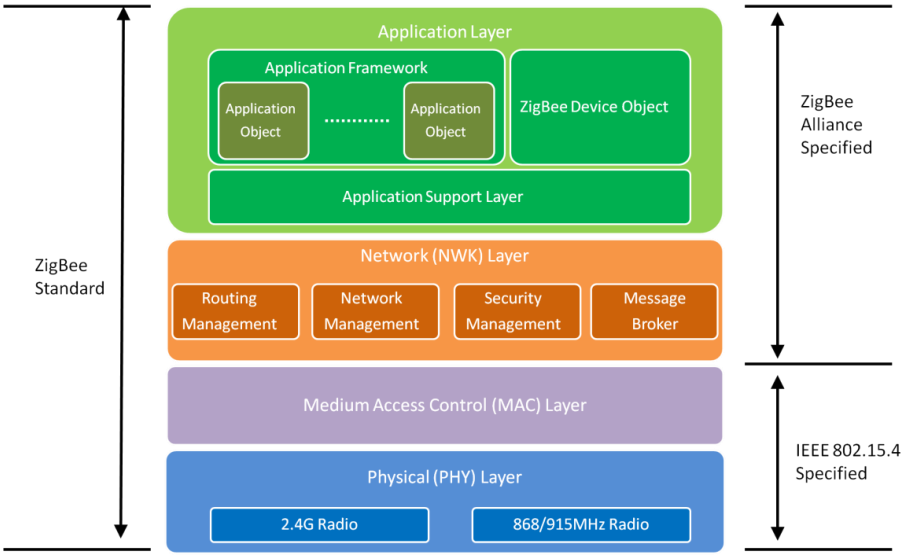


Figure 19 ZigBee and IEEE 802.15.4 Protocol Layers

ZigBee was standardised in the same year as the IEEE 802.15.4 and it is the earliest one built on IEEE 802.15.4. It has wide support from academia and industry. As indicated in Table 4, ZigBee has been employed in many wireless sensor node platforms. Also, a large range of embedded ZigBee transceiver modules is available from many manufacturers including Texas Instruments, Freescale, NXP, Atmel, etc., with free protocol stacks which can allow the host controller to interface and manage the transceiver. These features can reduce the development time, cost and complexity.

Compared to ZigBee, other IEEE 802.15.4 based protocols have also their unique features. The 6LoWPAN (Internet Protocol version 6 (IPv6) over Low power Wireless Personal Area Networks)

brings the interconnectivity between the IEEE 802.15.4 based WSN platforms and the existing large numbers of the Internet Protocol based networks such as Ethernet, Wi-Fi networks, and of course, the Internet. The 6LoWPAN combines the wide connectivity of the IP network and the low power feature of the IEEE 802.15.4. WirelessHART and ISA100.11a aim for automation and control applications and can provide more reliable and secure communications in harsh industrial environments, and they can be ultra low power as well, e.g. the WirelessHART devices from Linear Technology can transmit at 8dBm with less than 10mA current. However, 6LoWPAN, WirelessHART, and ISA100.11a are all relatively new standards. The support for these protocols is limited. The devices available for academic purposes as well as the supporting materials such as documents and protocol stacks are still limited. Especially for the standards intended for the industrial area like WirelessHART and ISA100.11a, suppliers tend to not offer their protocol stacks publically. For example, in some R&D projects [243] [244] the researchers have to use IEEE 802.15.4 embedded modules as hardware platforms and implement their own WirelessHART software protocol stack three years after the protocol has been released.

As discussed at the beginning of this section, features like ease-of-use, development complexity, and availability are also important factors for choosing the wireless protocol and transceiver for the RAWS node prototyping, and also taking into account the time and cost for the wireless network development and implementation. Therefore, after synthesising the features of the wireless protocols and the requirements of the WSN applications, ZigBee wireless communication protocol is selected for the following reasons:

- Relatively low power with adequate data rate and range for WSN applications
- Flexible network topology options and sufficient network capacity
- Wide availability of devices, protocol stacks and other supports which will facilitate the wireless network development and the wireless testing environment setup

ZigBee was selected for the proof-of-concept and RAWS node prototyping, however, as the research progressed and the technology advanced, many other very promising wireless

communication protocols can also be utilised for further developing the wireless part of the RAWS node technology.

In fact, with the rapid development of the IoT technologies, many new wireless communication technologies have appeared or become popular after the RAWS node development work was finished, such as LTE-M, LoRa, SigFox, Narrowban-IoT (NB-IoT), Dash7, Weightless, just to name a few. These protocols were developed for low power wide area networks (LPWAN), aiming mainly for IoT applications and naturally for WSNs as well since WSN is one of the key technologies of IoT. These LPWAN technologies have their own advantages and disadvantages. In general, they have much bigger transmission ranges, which can be from a few kilometres to tens of kilometres, than the WPAN or WLAN wireless technologies discussed above in this section which were traditionally often used in WSNs. Even though these LPWAN technologies usually have higher power consumption and are more expensive, they open up new possibilities for WSN applications.

Many existing wireless technologies and standards have also been further developed or expanded to add in new features. For example, the networking capability of Bluetooth technology has been enhanced by supporting the mesh networking, which can allow more flexible Bluetooth network to be built to cope with scenarios that the simple star topology may not be ideal for. For another instance, Wi-Fi family had a new amendment called Wi-Fi HaLow (IEEE 802.11ah) protocol. It was created as a low power Wi-Fi protocol for low data rate and long range sensor and controller applications, and it aims to extend Wi-Fi further into IoT and WSN area.

As mentioned before, the main focus of this research project is not the wireless side of WSNs. However, the wireless networking and communication have always been considered as an important part of the research, and surely the aforementioned wireless protocols can be useful for further developing the wireless networking and communication capabilities of the RAWS technology for different application requirements and objectives.

Chapter 3 System Architecture and Design

The design and implementation of the RAWS node technology for the proof-of-concept are detailed in Chapter 3, 4 and 5, including the overall system architecture, the wireless sensor node hardware platform testbed, the sensor identification scheme, and the adaptive hardware reconfiguration techniques. But first of all, the design approach is discussed in the next section to provide the framework and rationale for the sequence and progression of the design of the RAWS technology.

3.1 Design Approach

The RAWS research sets out to demonstrate the feasibility of the research concepts through the design and development of the RAWS node system. The research concepts consist of the following three key elements as discussed in Chapter 1:

- WSN node hardware platform based on programmable mixed-signal hardware
- Sensor identification scheme for acquiring key information of analogue sensors
- Adaptive reconfiguration techniques that combines the above two elements to realise the autonomic multi-sensing capability

Therefore, the sequence and progression of the RAWS system design followed the flow of the research concepts logic hierarchy (Figure 5 in Chapter 1) to implement and evaluated each of key concepts, as illustrated in Figure 20.

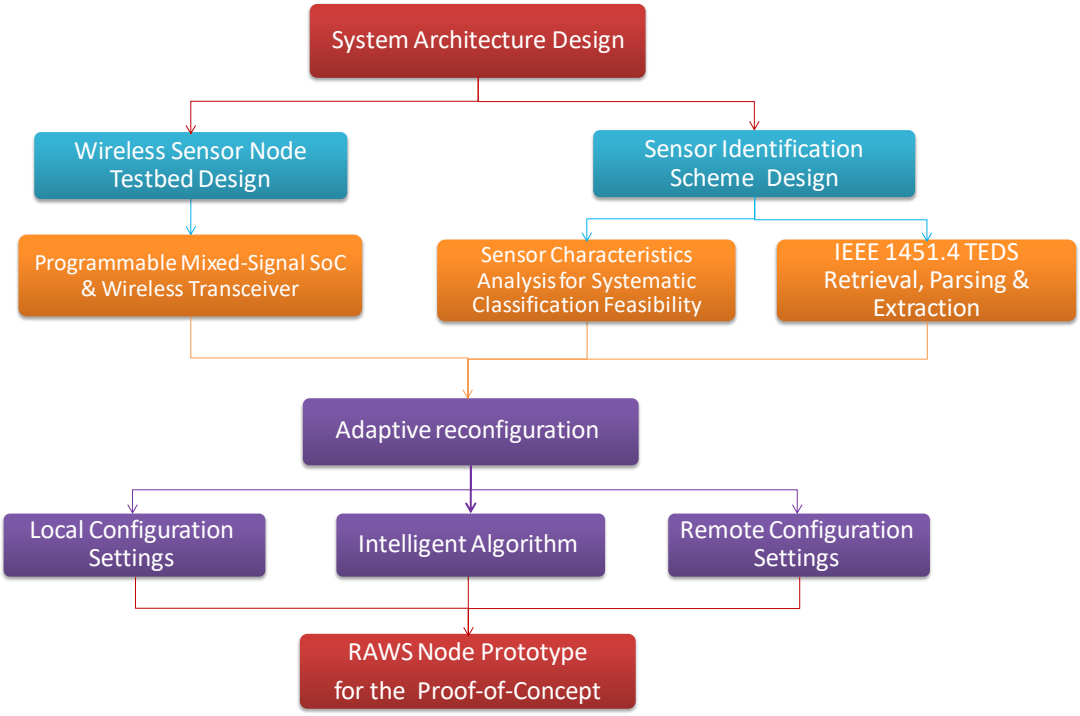


Figure 20 Progression of the RAWS Node System Design

First of all, the overall system architecture was developed. The framework and functionalities of the RAWS node platform and the RAWS network gateway were assigned in order to provide a guideline for the detailed system design.

The basic RAWS node platform was then designed and implemented after the main components of the platform were selected. The main components include a programmable mixed signal microcontroller and a wireless transceiver module. This basic platform was implemented only with the necessary functions of a WSN node as a testbed, including mainly the wireless networking and communication functions. However, it supported the dynamic reconfigurability and other key features of the system. The sensor identification scheme and the adaptive reconfiguration techniques were further developed based on this RAWS hardware platform.

The sensor identification scheme is another foundation of the system. Two approaches were investigated for realising such a scheme for acquiring important analogue sensor information so that the RAWS node can adaptively utilise its hardware resources. In the first approach, the

characteristics of different types of analogue sensor were analysed to verify the feasibility of developing a systematic classification process for identifying different analogue sensors. In the second approach, hardware interface circuits and software APIs and algorithms were developed to retrieve, parse and extract sensor identification and characteristics information from the 1451.4 TEDS.

Once the sensor information could be obtained, the next key step of the system design was how to reconfigure the hardware. Three adaptive reconfiguration techniques were developed to utilise critical sensor information in different ways to reconfigure the hardware in order to adapt to different analogue sensors in an autonomic manner, where the key technique developed in the research is the intelligent algorithm that enables the RAWS node to autonomically build suitable signal processing chains to support different sensors.

Of the four parts of the system design described above, the overall system architecture design and the wireless network platform development are illustrated in the rest of this chapter, while the other two major parts of the design, sensor identification scheme and adaptive reconfiguration techniques are detailed in two standalone chapters, Chapter 4 and Chapter 5 respectively.

3.2 RAWS Node System Architecture

3.2.1 System Architecture Design

The basic framework of the system architecture of the RAWS node platform based on the research concepts is shown in Figure 21 below. This basic framework is further developed after the technical approach was decided as described in Chapter 2 and is further discussed here.

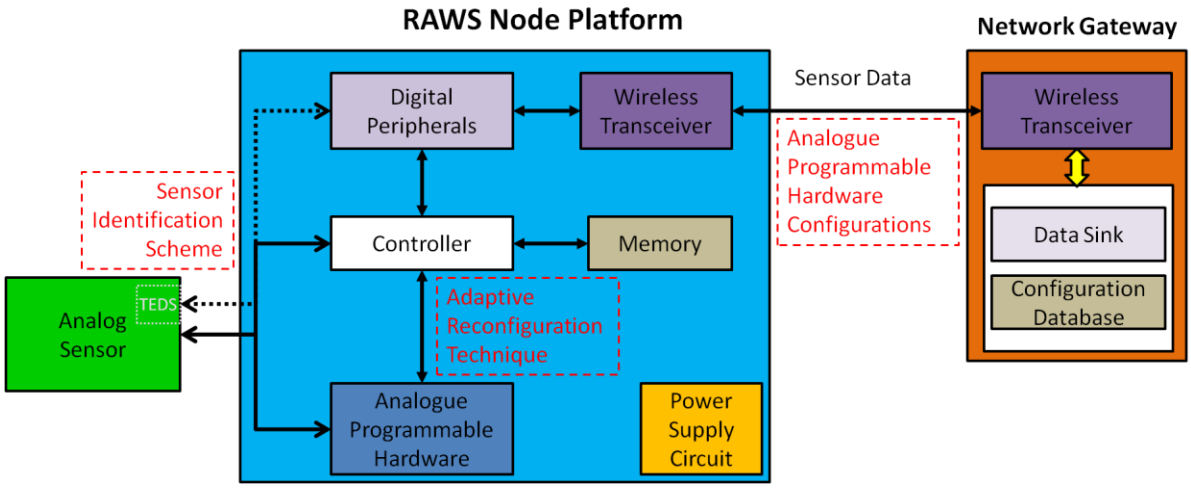


Figure 21 Basic Framework of the System Architecture

As discussed in Chapter 2, the PSoC programmable hardware and the ZigBee wireless technology have been selected for the proof-of-concept.

PSoC is more than just a programmable analogue array but is a system on-a-chip microcontroller including CPU and memory subsystem, reconfigurable analogue and digital subsystems, and programmable interconnections. This all-in-one feature of PSoC minimises the amount of onboard IC chips and simplifies the hardware structure and design. To be more specific, the PSoC microcontroller incorporates the functionality of four main components in the basic framework of the system architecture illustrated in Figure 21, which are the controller, memory, digital peripherals and analogue programmable hardware. Therefore, from the hardware perspective, the RAWS node prototype is comprised of two major components, a PSoC microcontroller as the core controller taking multiple key roles and a ZigBee transceiver as the wireless interface.

The core of the RAWS node hardware platform is the PSoC microcontroller, but the RAWS node firmware is the soul that makes the hardware platform functional. The RAWS node firmware is in charge of managing the hardware platform to realise the following functions:

- Perform the sensor identification process in order to acquire the identification and attributes information of the connected sensor
- Parse the sensor information and extract the important parameters for the hardware reconfiguration
- Deploy the adaptive reconfiguration techniques to reconfigure the programmable subsystem to accommodate the connected sensors
- Manage the transceiver to accomplish wireless communications tasks including: the transfer of sensor readings to the network gateway; the process of retrieving the hardware configuration settings from the network gateway when the remote configuration settings technique is applied
- Dynamically manage the power states of the wireless transceiver and the subsystems of the microcontroller for energy saving

A network gateway has also been built during the research and the two major hardware components of the gateway include a ZigBee transceiver and a Windows-based laptop, where the laptop can be replaced by other similar devices such as a PC or a tablet. A gateway software application called RAWS node terminal has been designed and developed to realise the following functions.

- Creating and managing the wireless RAWS node network
- Collecting sensor measurement data and working as a data sink
- Supporting the remote adaptive reconfiguration technique of the RAWS node platform, i.e. storing the hardware configuration settings and transferring them to RAWS node when requested

The RAWS terminal application mentioned above can be seen as the controller of the wireless transceiver of the gateway.

The more detailed and developed system architecture of the RAWS node platform has been designed as illustrated in Figure 22.

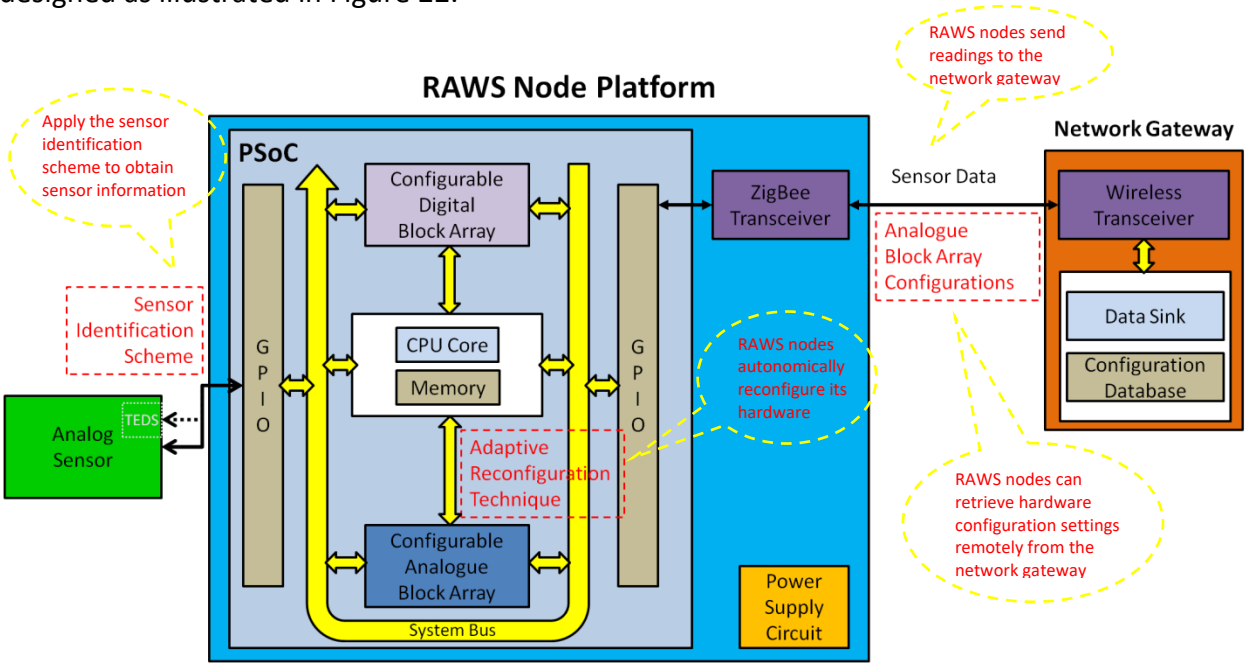


Figure 22 System architecture of the RAWS node platform

3.2.2 Hardware Components Selection and Specification

The PSoC microcontroller selected for the RAWS node prototyping is the Cypress CY8C29466, a PSoC 1 device. As discussed in Section 2.1.2.3, although a RAWS node prototype has also been implemented based on the PSoC 5 CY8C5568 microcontroller, the proof-of-concept is mainly based on the PSoC 1 and the main reason is that the configurable analogue arrays of the PSoC 1 device have better flexibility. Overall, the CY8C29 family has the highest specifications of all the PSoC 1 devices. All the models within the CY8C29 family have the same general specification except for the number of GPIOs [245].

The key specifications of the PSoC CY8C29466 and the main peripheral functions that can be implemented with the programmable analogue and digital subsystems are summarised in Table 5 below. The analogue system is composed of 4 CT CABs and 8 SC CABs and they are all generic analogue blocks and can be fully reconfigured into a wide range of peripheral functions using a single block individually or multiple blocks together. This architecture of the analogue

subsystem can offer high flexibility for the RAWS node platform to cope with different analogue sensors.

PSoC CY8C29466 Features	
CPU Core	8-bit M8C Core can work at 6MHz or 24MHz
Memory	2KB SRAM, 32KB Flash ROM
Operating Voltage	3.0V to 5.5V (down to 1.0 V using on-chip switch mode pump (SMP))
Supply Current	8mA Typical Supply Current <5uA Typical Sleep Current
Analogue System	4 columns of configurable analogue blocks Each column includes 1 CT block and 2 SC blocks, 12 CABs in total Main CT block functions: PGA, Instrumentation Amplifier, Comparator Main SC block functions: Incremental ADC (up to 14-bit), Delta Sigma ADC (up to 14-bit), SAR ADC(6-bit), DAC (up to 9-bit), Filter (Low Pass, Band Pass)
Digital System	4 rows, 16 programmable digital blocks Main functions: UART, SPI Master and Slave, 8 and 16-bit PWMs, 8 to 32-bit Timers and Counters, Cyclical Redundancy Check (CRC) modules, Pseudo Random Sequence (PRS) modules
Other	24 configurable GPIOs, I^2C Masters and Slaves, Decimator, Internal Voltage References, Watchdog and sleep timers

Table 5 PSoC CY8C29466 Features

In terms of the ZigBee transceiver, the selection process gave ease-of-use a high priority simply because the main focus of the research is the sensor and analogue electronics rather than advanced wireless networking and communication functions. The objective therefore is to select a transceiver that allows for quick design and implementation of the wireless part of the RAWS node. This however compromised the power consumption a little which will be explained below but it does not affect the overall proof-of-concept. The power consumption of the wireless transceiver and the platform can be further optimised in the future.

The ZigBee transceiver selected for the RAWS node implementation is the Digi International XBee ZB RF module. Compared to other off-the-shelf ZigBee transceiver chips or modules available at the time this research project started, the XBee ZigBee module has many

advantages in terms of ease-of-use. The XBee ZigBee module interfaces with the host controller via UART protocol which is relatively straightforward to implement. The module is equipped with an application processor to run the built-in ZigBee protocol stack and operate the RF transceiver circuitry and other components, so the module can perform the low level operations with minimum host controller intervention or management, such as RF channel scans and selections, network forming and joining processes, etc. The XBee ZigBee module also offers adequate specifications with regards to other features such as transmission range, data rate, etc.

The trade-off for the ease-of-use feature is the higher power consumption. The transmit current of the XBee module is around 35mA to 40mA at the max transmit power of 1dBm or 3dBm with boot mode on, while for other ZigBee transceivers this number is usually around 24mA to 31mA. The receive current of the XBee module is above 38mA which is also higher than the typical ratings of many other ZigBee transceivers. Its power consumption can be reduced by configuring the transmit power to a lower level such as -2dBm or -4dBm, but this also reduces the transmission range. The key specifications of the XBee ZB RF module are summarised in Table 6.

	XBee ZB RF module Specification
Indoor Range	Up to 40m
Outdoor Range	Up to 120m
Raw Data Rate	250kbps
Practical Data Rate	Up to 35kbps
Transmit Power	Max: 1.25mW (1dBm) or 2mW (3dBm) with boost mode Min: -10 dBm or -8 dBm with boost mode
Transmit Current	35mA (at max transmit power level, normal mode) 40mA (at max transmit power level, boost mode)
Receiver Sensitivity	-95 dBm or -96 dBm with boost mode
Receive Current	38mA (normal mode) 40mA (boost mode)

Table 6 XBee ZF RF Transceiver Module Key Specification

3.3 Wireless Network Design and Development

This section describes the design and implementation of the wireless RAWS node network. The network is based on ZigBee technology. The design principle is to equip the RAWS node with fully functional but uncomplicated wireless networking and communication capability, and establishing a simple ZigBee network as the testbed for the node platforms. This principle is reflected in the ZigBee transceiver selection as explained in the previous section and is applied to the ZigBee network development described in this section as well. More advanced wireless networking and communication functions and different wireless protocol options can be investigated and developed in the future.

The RAWS node network design and development illustrated in the next few subsections will involve the principles of ZigBee networking and communications, network parameters, and device operations, but the more detailed technical descriptions have been assigned to Appendix E.

3.3.1 Network Architecture

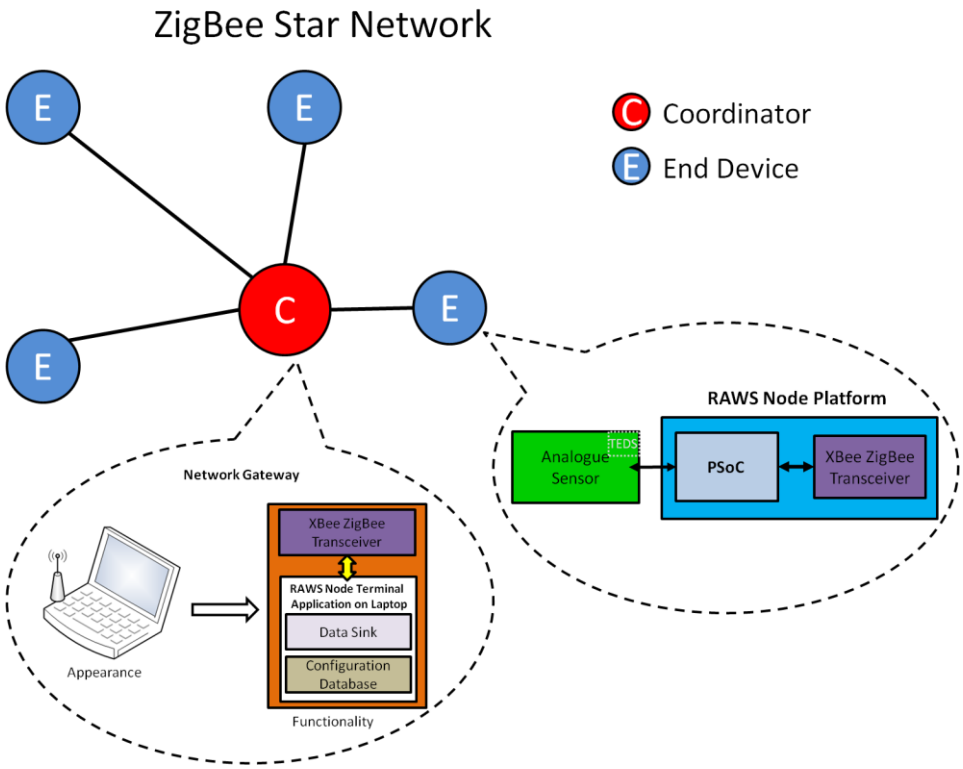


Figure 23 Wireless RAWS Node Network Diagram

The wireless network designed and developed in this research consists of multiple RAWS nodes and a network gateway forming a ZigBee network with a star topology as shown in Figure 23. The gateway is the centre of the network and is powered by mains so naturally it also performs the ZigBee coordinator role, while RAWS nodes act as ZigBee end devices in this star network.

The network gateway from the hardware perspective is a Windows-based laptop (or a PC) equipped with an XBee ZigBee coordinator transceiver module. The gateway software application is the RAWS node terminal and was developed as an important component of the RAWS system. It instructs the coordinator module to initialise and create the ZigBee network. It also manages the network and the end devices. The gateway takes the role of the data sink as well to collect the readings from sensor nodes. It also supports the remote configuration settings feature of RAWS nodes by keeping a hardware configuration database.

RAWS nodes are designed to be low power and have sleep mode, so according to the characteristics of ZigBee device types the RAWS nodes can only act as ZigBee end devices. The RAWS node firmware directs the XBee ZigBee end device transceiver module to join the network through the coordinator. After joining the network, the RAWS node can start transmitting and receiving data.

3.3.2 Hardware Design

The XBee ZigBee transceiver modules are the wireless hardware platform. The XBee ZigBee module can install different types of firmware to act as different ZigBee device types, so the same model of XBee ZigBee module hardware is employed by all RAWS node platforms and the network gateway but different types of firmware are installed. The network gateway performs as the ZigBee coordinator so the XBee module of the gateway is loaded with the coordinator type firmware to work as a coordinator module, and RAWS nodes act as ZigBee end devices so the XBee modules on RAWS nodes are loaded with end device type firmware and work as end device modules.

The wireless hardware design is illustrated in Figure 24.

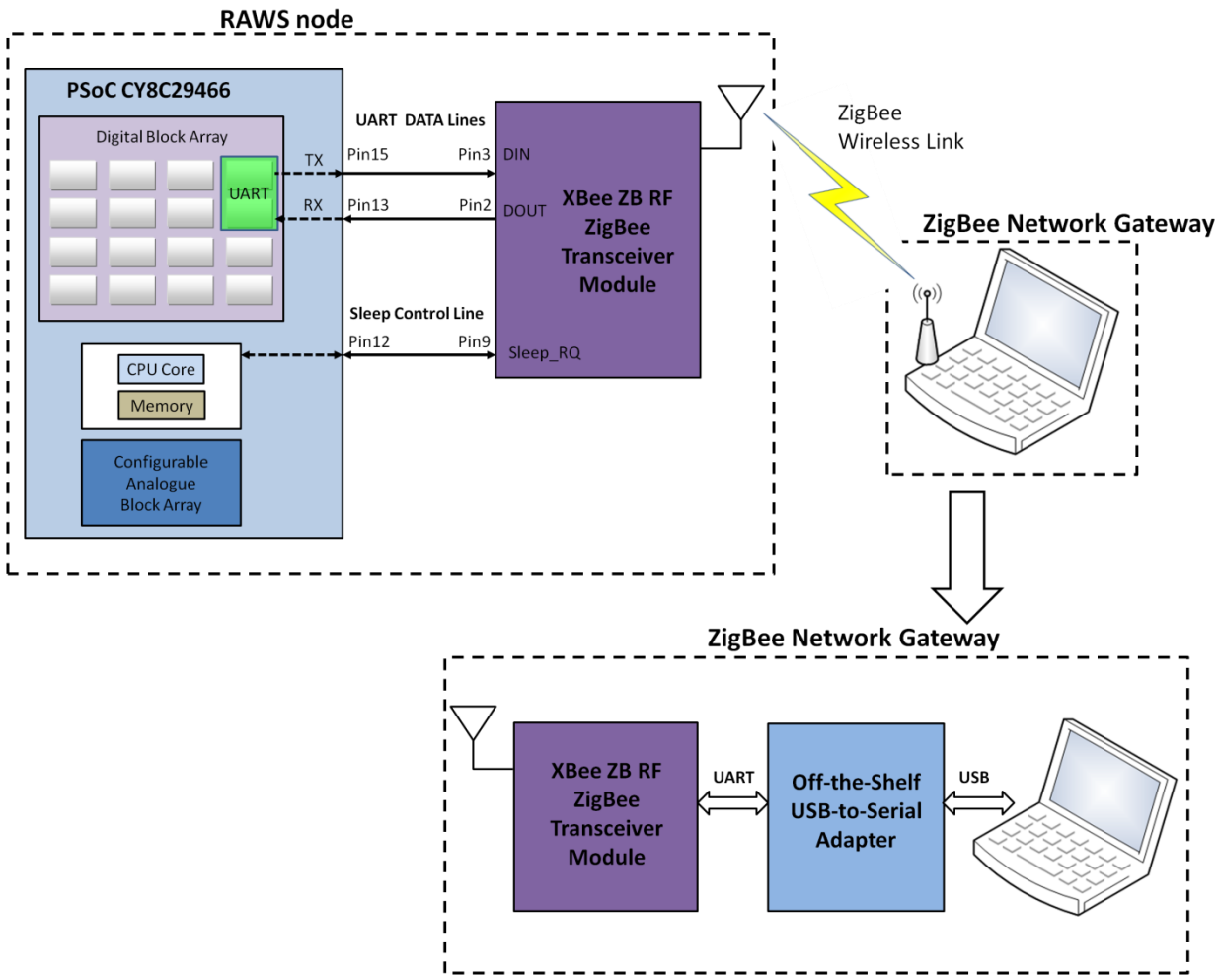


Figure 24 RAWS Node Wireless Hardware Design

The XBee module is a 20-pin device and the pins that are needed in the hardware design are described in Table 7.

Pin Name	Pin Number	Description
Vcc	1	Power Supply
GND	10	Ground
DOUT	2	UART Data Output
DIN	3	UART Data Input
Sleep_RQ	9	Pin Sleep Control

Table 7 XBee Module Pins used for Prototyping

The sleep feature of the XBee module is used only by the RAWS nodes since the coordinator must always remain on. To conserve energy on RAWS nodes, the ZigBee transceiver is woken up only when wireless transmissions are needed. Once wireless communications finish, the transceiver is put back into sleep mode. The XBee ZigBee module supports a sleep mode called 'pin-sleep' where the module enters and exits sleep mode according to the state of the Sleep_RQ pin. The pin-sleep mode is utilised in the design to allow the controller of the RAWS node, the PSoC microcontroller, to flexibly control the sleep behaviour of the XBee ZigBee module. One of the GPIO pins, Pin12 (Port1, Pin2), of PSoC CY8C29466 microcontroller is connected to Sleep_RQ pin for this purpose.

The XBee ZigBee module is equipped with a UART serial interface. To control and communicate with the XBee ZigBee module, two digital blocks of the PSoC microcontroller are configured as a UART function module. The RX and TX (input and output) of the UART function module are routed to the GPIO pins P13 and P15 through the internal digital multiplexer to connect to the UART I/O pin of the XBee module.

In terms of the hardware setup of the network gateway, because most laptops (or PCs) no longer have serial COM ports, the XBee module cannot connect to the laptop or PC directly. So in this research the gateway also employs an off-the-shelf USB-to-Serial adapter to interface the XBee module with the laptop. When plugged into the USB port of the laptop, the adapter emulates a serial COM port in the operating system. It works as a bridge that enables UART transmission between the laptop and the XBee ZigBee transceiver and allows the RAWS node terminal application to communicate with the ZigBee module.

3.3.3 Software Design

As mentioned before, the software implementation of the wireless functions is included in the RAWS node firmware and the RAWS node terminal application. The wireless part of the software design and development is detailed in the next few subsections, including how these two pieces of software communicate with the ZigBee transceiver modules, how they manage

the network establishing and joining processes, and how they perform the wireless transmission operations.

3.3.3.1 Interface Operation Protocol between RAWS node software and XBee module

The two pieces of software communicate with XBee ZigBee modules through the UART transmission to perform wireless networking and data transmissions. The XBee ZigBee module supports two types of UART interface operation protocols which determine how the host software application communicates with the module. The two types of operation protocols are the transparent operation and the API operation.

The transparent operation allows the XBee modules act as if they are transparent between the transmission devices. The transparent operation has two modes, data mode and AT command mode. In data mode, all data received from the UART input of the module is queued up for ZigBee RF transmission, also data received wirelessly is sent out through the UART output. The XBee module works like a UART cable linking two transmission devices. For the host controllers and software of the transmission devices, the XBee modules are transparent in the data mode.

The AT mode allows the host controller to control the transceiver module. A special 3-character command sequence plus at least a one second idle time interval before and after the 'special command sequence' compose an enter-AT-mode command for switching the module from data mode to AT command mode. In AT mode the host software application can send AT commands to read or change the module status or settings. The key network parameters for creating and joining a ZigBee network such as the 64-bit PAN ID, ScanChannels, stack profile, PermitDuration, and the parameters for wireless transmissions such as destination MAC and network addresses can all be set and changed in the command mode using AT commands. An exit-AT-mode command is used to switch the module back to the data mode for wireless transmissions.

As an alternative to the transparent operation, the API operation requires that the host software application communicates with the XBee module through structured API frames. Different types of API frames with different functions are used for interactions between the host application and the module. The API frames sent from the host application to the XBee module contain two main types of contents:

- Data to be sent through ZigBee RF transmission
- AT commands to check or change the settings of the module

The frames received from a module usually contain the following types of content:

- Data received from RF transmission
- Response of the AT command
- Response of the RF transmitting request
- The notification messages of the module status

The API operation is chosen for the RAWS software designs for the following reasons.

- In API operation, the host application can perform many operations in a faster and more convenient way through API frames, such as reading and changing the module status and parameters, and setting the destination addresses for the transmitting data. However, in transparent operation the host application has to go through the entering and exiting AT command mode process which takes a long time and wastes power.
- In API operation, the RF data recipient can obtain the information of the source device from the received data frames including the MAC and network addresses which are useful for the gateway terminal software for distinguishing different network nodes, while the transparent operation cannot provide such information.

Overall, the RAWS node firmware and the RAWS node terminal application can manage the wireless networking and communications more efficiently through the API operation mode of the XBee module, avoiding unnecessary power consumption especially for the battery-powered RAWS nodes.

The API frames have the common structure shown in Figure 25.

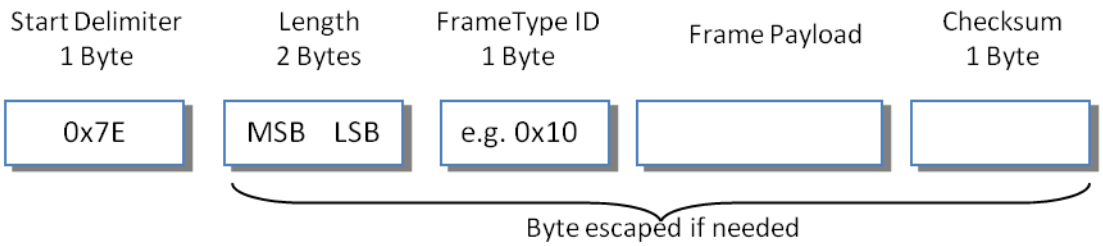


Figure 25 XBee Module API Frame Structure

The RAWS node firmware and the terminal application assemble API frames to communicate with the XBee modules and parse the frames received from the module following this structure.

The RAWS node firmware and the terminal application utilise the FrameType ID field to declare the intention of the outgoing frames to the transceiver module and to identify the purpose of the incoming frames and then interpret the frame payload accordingly. The API frame types important for the RAWS node firmware and the terminal application are summarised in Table 8.

	API Frame types	Type ID	Description
Host → XBee module	AT Command	0x08	Send AT command to read or set a module parameter
	ZigBee Transmit Request	0x10	Send data to another device via ZigBee transmission
XBee Module → Host	AT Command Response	0x88	Response message of the AT command frame
	Module Status	0x8A	Indicating the module status at specific conditions such as hardware reset
	ZigBee Transmit Status	0x8B	Response message of the Transmit Request frame indicating the transmitting status
	ZigBee Receive Package	0x90	Data package received from other device via ZigBee transmission

Table 8 XBee API Frame Types used in the RAWS node research

3.3.3.2 RAWS Node Terminal Application Operations for Creating Network

The RAWS node terminal application instructs the network coordinator to create the ZigBee network. The RAWS node terminal application was developed with Microsoft Visual C++ and

the .Net Framework. The main window of the terminal application, as shown in Figure 26, contains multiple tab pages where the first tab page is the coordinator settings tab and others are RAWS node tabs. The RAWS node tabs are dynamically added when new RAWS nodes join the network and these tab pages are for displaying the information of the joined RAWS node including the sensor identification information and parameters as well as the sensor measurement readings. The RAWS node tabs will be shown later in the thesis. The coordinator settings tab is for setting the key parameters of the ZigBee network.

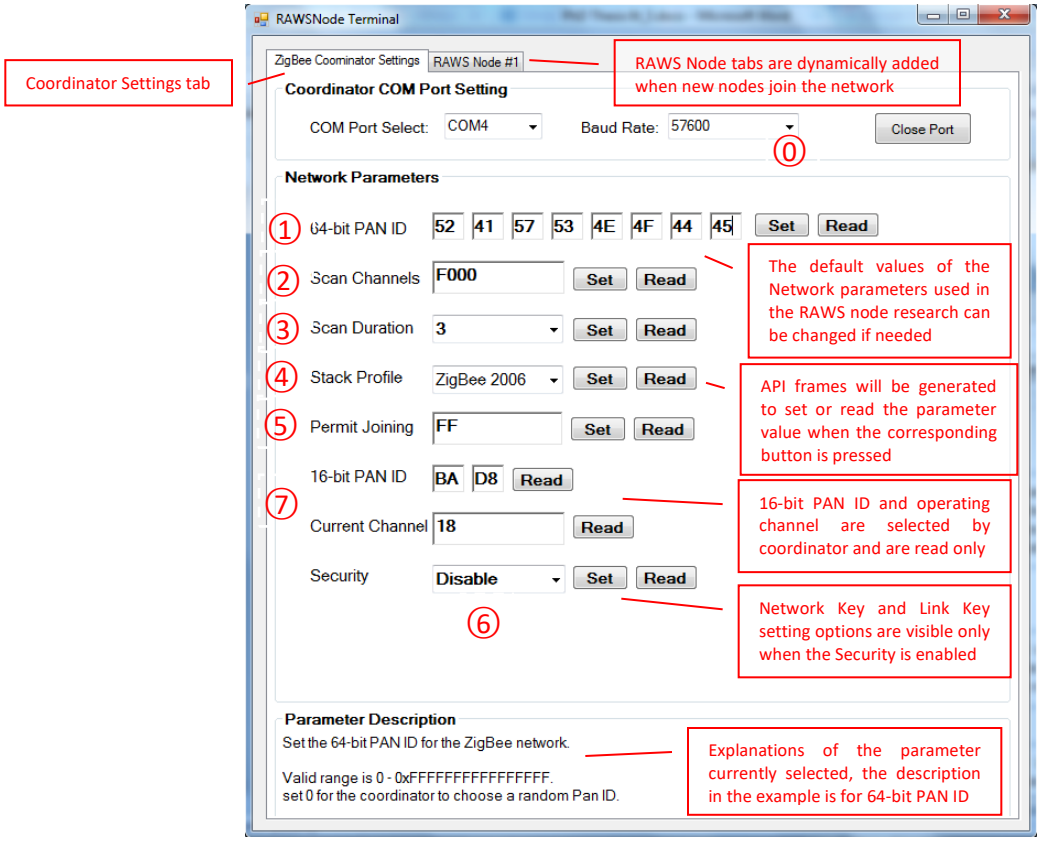


Figure 26 RAWS Node Terminal Application

The RAWS node terminal application communicates with the XBee module through the virtual COM port. The COM port for interfacing the ZigBee coordinator module and its baud rate can be selected in the coordinator settings tab. The default baud rate used in the research is 57600 (shown at ⓪ on Figure 26). The other UART settings are hard coded in the software as they do not need to change, and these settings include 8 data bits, no parity, 1 stop bit, no flow control.

Once the COM port is opened, the terminal application sends a series of AT command API frames (FrameType ID = 0x08) to the coordinator module to set up the values of the key network parameters for starting the ZigBee network. These network parameters are explained in Appendix E. The values selected as default parameter settings for the RAWS node network are explained as follows. The values can be changed in the RAWS terminal application.

- 64-bit PAN ID

  The value of the 64-bit PAN ID is "0x 52 41 57 53 4E 4F 44 45" (shown at ① on Figure 26) which is the ASCII code for "RAWSNODE". The coordinator starts the network with this PAN ID. In the test, no ZigBee network nearby has the same ID.

- ScanChannels

  The ScanChannels parameter is set to 0xF000 (② on Figure 26) so that the coordinator will only scan four ZigBee channels, the channel No. 23, 24, 25, and 26, and selects one of these channels to create the network. This is to avoid the operating channels of the other two ZigBee networks in the same testing area and avoid potential interference. Also, limiting scanning channels to four can also reduce the total scanning time of the end devices in the network and help RAWS nodes to save power.

- ScanDuration

  The ScanDuration parameter is set to 3 (③ on Figure 26), which means that it takes about 0.5 seconds to scan all the four channels defined by the ScanChannels parameter.

- Stack Profile

  The ZigBee 2006 stack profile (④ on Figure 26) is used by the RAWS node network because the new features introduced by the ZigBee Pro stack profile such as advanced routing and security functions are not required in this wireless network testbed. The network can switch to ZigBee Pro stack profile when the new features are needed in the future.

- Permit Joining

  The PermitDuration parameter of the coordinator is set to 0xFF (⑤ on Figure 26) to always allow RAWS nodes to join in the test.

- Security

  The security feature is disabled in the testbed (⑥ on Figure 26) but it is implemented and can be enabled when needed and the network key and the link key can be assigned in the terminal application as well (see Appendix E).

After the RAWS node terminal application is finished setting up the network parameters, the coordinator module starts to perform the channel scans and chooses an operating channel to create the network based on these parameters. A 16-bit PAN ID will be also generated by the coordinator during the network creating process. The terminal application does not need to micromanage these low level operations but it still needs to keep monitoring the status of the coordinator and handling the exceptional situations when errors occur for the whole network lifetime.

Once the network starts, the RAWS node terminal application reads the 16-bit PAN ID and the operating channel selected by the coordinator and displays them on the coordinator settings tab. In the example shown at ⑦ on Figure 26 above, the 16-bit PAN ID is 0xBAD8 and the operating channel is 0x18.

3.3.3.3 RAWS Node Firmware Operations for Joining Network

The RAWS node firmware also sets a series of parameters of the onboard XBee end device module to guide the RAWS nodes to join the ZigBee network. The parameters to be set for the XBee end device modules and their default values are the same as the ones for the coordinator discussed above, except that the end devices do not have the Permit Joining parameter. Once the parameters are set up, XBee end device modules perform the PAN ID scan to find the RAWS node ZigBee network in order to join it. Like the terminal application, the firmware does not need to micromanage the low level operations. However, it also needs to monitor the transceiver status and handle the errors that occur during the network joining process.

Same as for the terminal application, the firmware also sends a series of AT command API frames to the XBee module to perform the parameter setup process. Table 9 shows the AT command frame for setting the 64-bit PAN ID as an example of the API frames that are

assembled by the RAWS node terminal application and the RAWS node firmware for controlling the transceiver module. The basic structure of the API frame is illustrated in Figure 25 in subsection 3.3.3.1. The API frame example below also follows the structure.

Frame Field		Offset	Value	Description
Start Delimiter		0	0x7E	
Length	MSB	1	0x00	Number of bytes between the length field and checksum, 12 in this example
	LSB	2	0x0C	
Frame Type ID		3	0x08	AT Command frame type
Frame Number		4	0x01	Number of the frame, 0x01 in this example
Frame Payload	**AT Command**	5	0x49 (I)	AT command field. AT command "ID" is used to set or read 64-bit PAN ID.
		6	0x44 (D)	
	AT Command Parameter Value	7	0x52 (R)	ASCII code of "RAWSNODE" is used as the PAN ID. Send the AT command only (i.e. without the parameter value field) to read/query the current module setting.
		8	0x41 (A)	
		9	0x57 (W)	
		10	0x53 (S)	
		11	0x4E (N)	
		12	0x4F (O)	
		13	0x44 (D)	
		14	0x45 (E)	
Checksum		15	0x06	0xFF minus the 8-bit sum of the bytes from offset 3 to the one before checksum

Table 9 AT Command API Frame Example

The RAWS node terminal application and the RAWS node firmware parse the response frames received from the XBee module to make sure the API frame sent to the module have been successfully processed. Both the RAWS terminal application and firmware have error handling functions based on the feedback in the response frames. The error handling includes resending API frames, prompting error messages, and resetting the module or the network. Table 10 below shows the normal "OK" response to the above 64-bit PAN ID setting frame as an example of the API frame responses.

Frame Field		Offset	Value	Description
Start Delimiter		0	0x7E	
Length	MSB	1	0x00	Number of bytes between the length field and checksum, 5 in this example
	LSB	2	0x05	
Frame Type		3	0x88	AT Command Response frame type
Frame Number		4	0x01	Response to the 0x01 frame
Frame Payload	**AT Command**	5	0x49 (I)	Response to the AT command "ID"
		6	0x44 (D)	
	AT Command Status	7	0x00	0 = OK, 1 = ERROR, 2 = Invalid Command, 3 = Invalid Parameter, 4 = Tx Failure The command status is used for debugging.
	AT Command Data			If the AT command frame is to read the setting value, the value will be returned in this field. This field is not used is the AT command is to set the value, as in this example.
Checksum		8	0xE9	0xFF minus the 8-bit sum of the bytes from offset 3 to the one before checksum

Table 10 AT Command Response API Frame Example

3.3.3.4 Wireless Transmission Operations

The transmissions within the wireless RAWS node network mainly include the following three types:

- The unidirectional sensor measurement data packet from RAWS nodes to the gateway
- The remote hardware configuration request for retrieving configuration settings from the gateway
- The hardware configuration response contains configuration settings sent from the gateway back to the RAWS nodes.

The wireless communications originate from the RAWS node end and the general communication flow is described below:

i. The RAWS node firmware packs the sensor readings or the remote hardware configuration request plus the content type ID into a ZigBee transmit request API frame

ii. The firmware also sets the destination of the transmit request frame which usually is the coordinator. The reserved coordinator MAC address is 0x000000000000000000 and the network address is 0x0000.

iii. The transmit request frame is sent to the transceiver module for ZigBee transmission

iv. After RF transmission attempts, the transceiver module responds with a ZigBee transmit status API frame to indicate whether the transmission has succeeded. If it fails the firmware calls the error handling routine.

v. On the gateway end, if the RF transmissions succeed, the RAWS node terminal application will receive the content sent from the RAWS node contained in a ZigBee receive packet frame. The ZigBee receive packet frame is packed and sent by the coordinator module and collected by the terminal application via the virtual COM port.

vi. The terminal application identifies the frame content by checking the content type ID.

vii. If the received frame contains sensor readings from RAWS node, the terminal application displays sensor data in the corresponding RAWS node tab pages and stores them into the data sink as well.

viii. If a remote hardware configuration request is received, the terminal application searches the corresponding configuration settings and replies to the source device with the source MAC and network addresses contained in the ZigBee receive packet frame. This section only covers the transmission sequences, and the remote hardware settings technique will be illustrated later in Chapter 5.

The detailed flowcharts of wireless transmission operations executed by the RAWS node firmware and the terminal application are presented in Figure 27 and Figure 28 respectively.

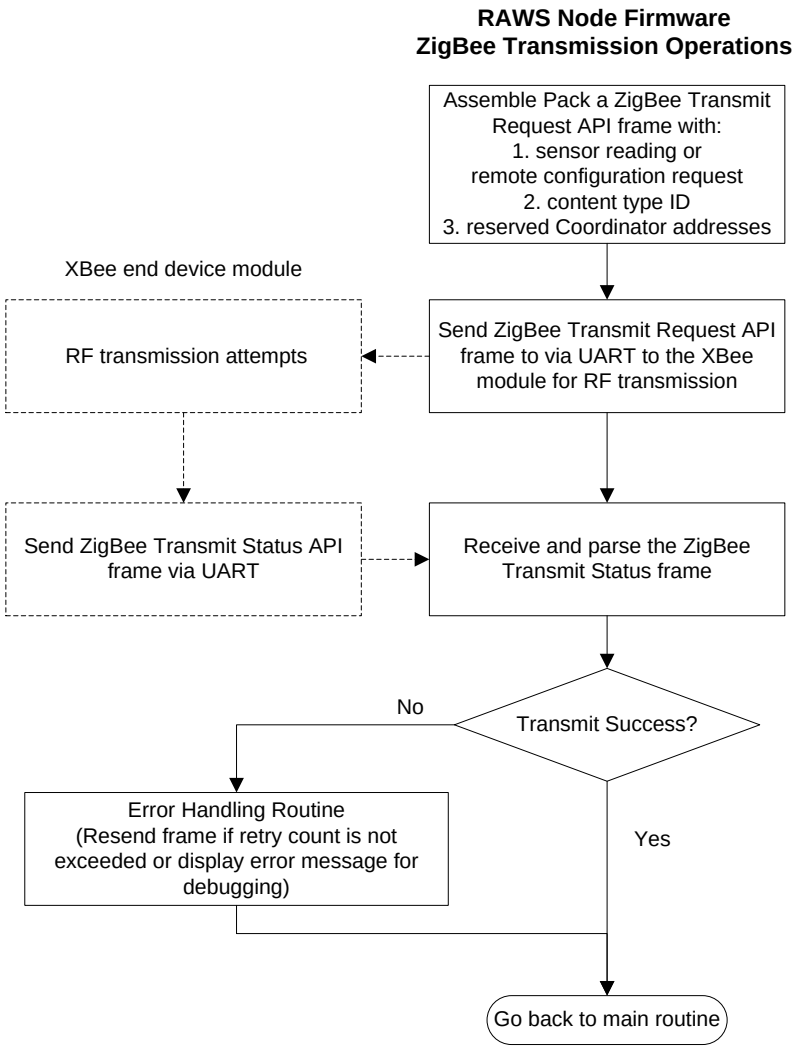


Figure 27 RAWS node firmware ZigBee transmission operation flow

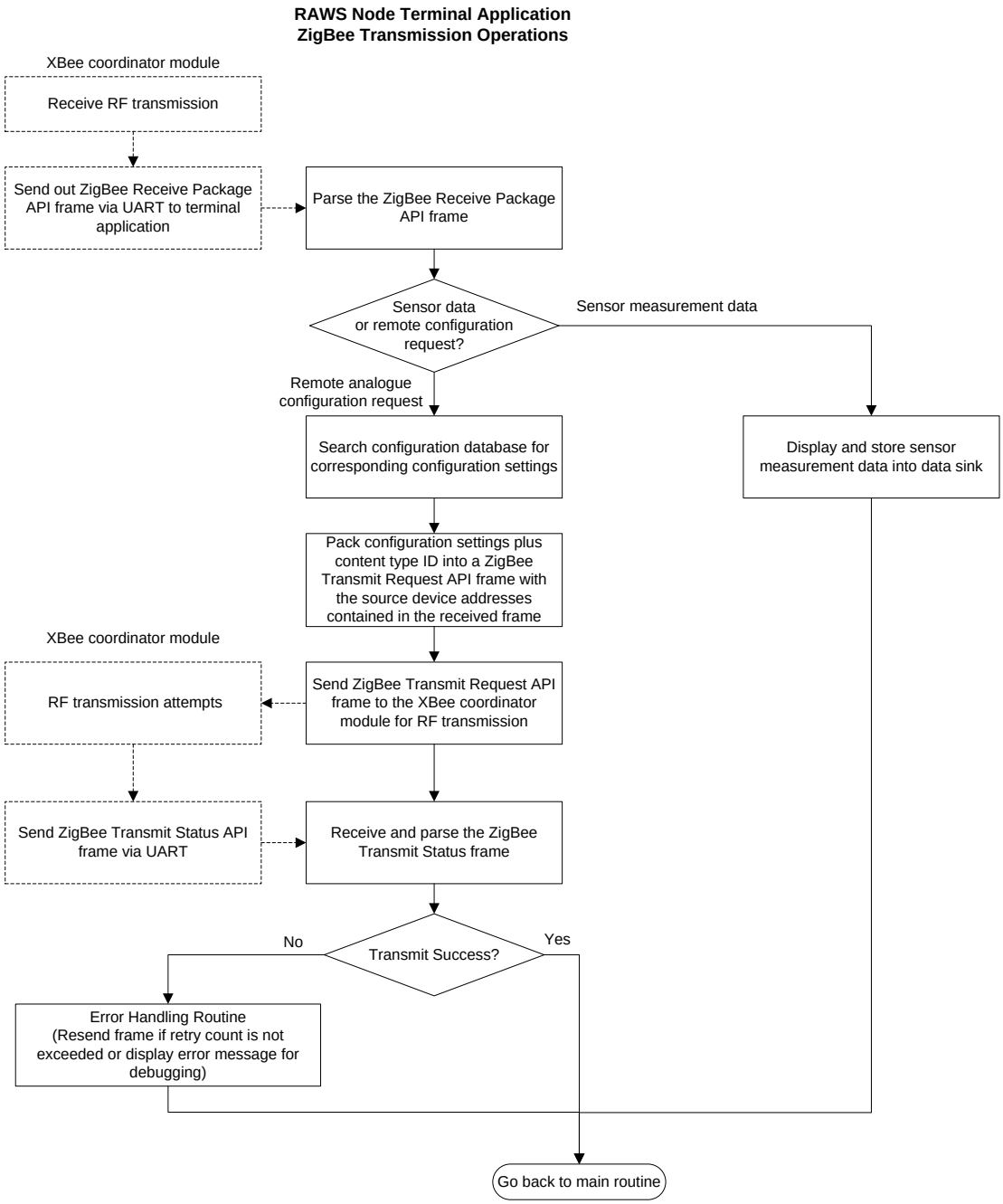


Figure 28 RAWS node terminal application ZigBee transmission operation flow

Chapter 4 Sensor Identification Scheme

This chapter describes the development of the sensor identification scheme. The task of the sensor identification scheme is to acquire the important identification and characteristics information of any connected sensors. The sensor information is prerequisite for the RAWS node to adaptively construct signal conditioning and processing chains in an autonomic way to support the connected sensors. Therefore, the sensor identification scheme plays a vital role for the RAWS node technology to be able to flexibly utilise the reconfigurability of programmable hardware to achieve the autonomic multi-sensing capability.

Two approaches were investigated in the research for realising such a scheme. The first approach involved devising a systematic characteristics classification process of sensor identification. The feasibility of this first approach requires analyses of the characteristics of different sensors and is described and analysed in section 4.1. The second approach is based on the sensor self-identification and self-description capability provided by the IEEE 1451.4 TEDS. The development and implementation of the second approach will be illustrated later in section 4.2.

4.1 Feasibility Analysis of the Sensor Characteristics Classification Approach

The concept of the systematic characteristics classification approach is based on developing a series of systematic tests which can be conducted by the RAWS node to acquire important characteristics information from the connected analogue sensor so that the RAWS nodes can classify and identify the sensor based on these characteristics. The premise of this approach is to determine what characteristics can be utilised for sensor identification. So the characteristics of different sensors first need to be analysed for investigating the feasibility of this approach.

As temperature sensing is a very common type of measurement used in a wide range of applications, the analysis begins with temperature sensors. Three major categories of analogue temperature sensors are: thermistors, thermocouples, and resistance temperature detectors (RTDs). The analysis of the thermistor and thermocouple characteristics is discussed in detail in the next two subsections.

4.1.1 Thermistor

A thermistor is a temperature sensitive resistor that exhibits a large resistance change in response to temperature change. Depending on the sign of the temperature coefficient, thermistors can be classified into two types, positive temperature coefficient (PTC) thermistors and negative temperature coefficient (NTC) thermistors. Most thermistors used for temperature measurement are negative temperature coefficient. The thermistors discussed in this section are the NTC type. Thermistors have the highest sensitivity within the three major categories of temperature sensors. The negative temperature coefficient can be as large as several percent per degree Celsius, allowing NTC thermistors to detect minute changes in temperature which could not be observed with a thermocouple or RTD [246].

Resistance-Temperature (R-T) Characteristic

There is no industrial standard for thermistors. Since thermistors express the temperature in the form of resistance values, to identify different thermistors the resistance-temperature characteristic is the first feature to be analysed. Thermistors are non-linear devices and the generic resistance-temperature curve of an NTC thermistor can be approximated using the equation below.

$$R_T = R_{T0}\, exp\left[\beta\left(\frac{1}{T}-\frac{1}{T_0}\right)\right] \text{ or } R_T = R_{T0}\, exp\left[\frac{\beta(T_0-T)}{T_0 T}\right] \qquad \text{Equation 1}$$

Where:

- R_T is the thermistor resistance at the absolute temperature T in Kelvin.
- β is the beta value, is the slope of the thermistor R-T characteristic over the specified temperature range. Normally the β value of a thermistor is provided in the datasheet by the manufacturers. The common range of the β value is from 3000 K to 4500 K.
- R_{T0} is the thermistor resistance at the reference temperature T_0. The de facto standard reference temperature used by thermistor manufacturers is 25°C (298K).

As the β value can be unique for different types of thermistors, if the β value can be measured or determined, it would be a straightforward way to identify different types of thermistors. However, the R-T exponential approximation is based on curve fitting to experimental data of

the R-T characteristics of a thermistor. Because the R-T characteristic of a thermistor is determined by the materials used during the fabrication process, the β value can be interpreted as an empirical value that reflects the characteristics of the materials. It is determined by solving the equation using two known data points on the R-T curve. It is not a parameter that can be simply measured with electrical methods in an embedded system environment. Therefore, it would be impractical for a WSN sensor node to directly detect the β value in order to straightforwardly identify unknown thermistors.

Although the β value cannot be measured, a WSN node platform with its own onboard temperature sensor can obtain one resistance-temperature value pair (R-T pair) of a thermistor, which can be the first step to distinguish different thermistors.

To further analyse the R-T characteristic, a number of thermistors were selected as the objects of study. As the commonly used and commercially available thermistors have the 25°C reference resistances ranging from 1kΩ to 10k Ω and the common range of the β value is from 3000 K to 4500 K, the four thermistors selected as the objects of the analysis also have the 25°C reference resistance and the β value within these ranges. The R-T curves of the thermistors were calculated based on Equation 1 and formatted and plotted using Excel. These four R-T curves are shown in Figure 29. The 25°C reference resistances of these four thermistors are 3 kΩ, 5 kΩ, 5 kΩ and 6.8 kΩ and the β values are 3470K, 3470K, 4080K and 3910K respectively.

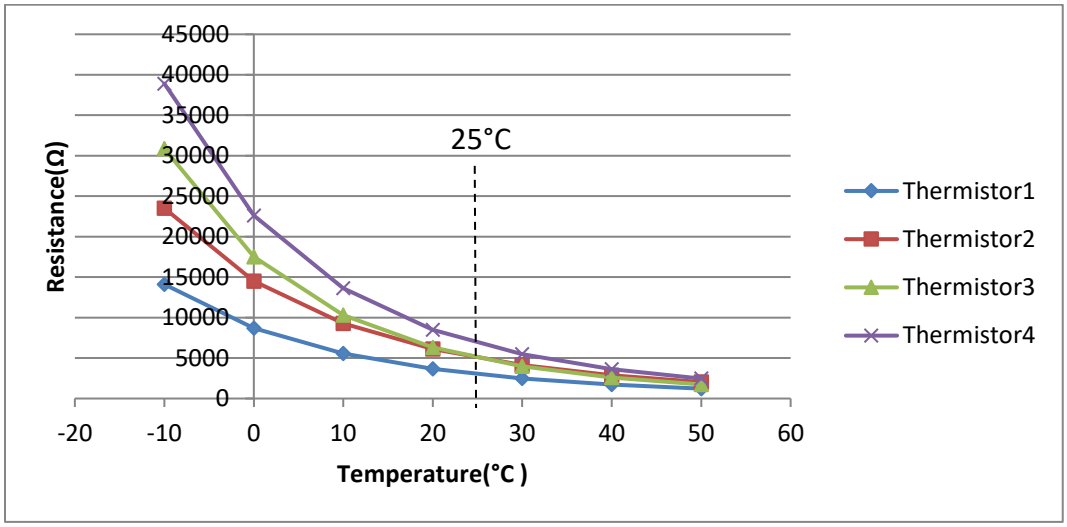


Figure 29 Thermistor R-T curves

Based on Figure 29, one pair of resistance and temperature measurement results can assist distinguishing three thermistors within the common working temperature range (-10 to 50°C). This means if the RAWS node has the pre-given knowledge of these four thermistors, at least three of them can be identified.

However, it becomes clear that one pair of R-T measurement results is not enough to distinguish thermistors, because thermistors that have different β values could have a cross-over point on their resistance-temperature curves. One example is thermistor 2 and 3 in Figure 29. Another example of this is provided in Figure 30 below where one thermistor is 5 kΩ at 25°C and $\beta = 3480K$, the other one is 4.7 kΩ at 25°C and $\beta = 4080K$. The calculation result shows that they have about the same resistance of 7160.5 Ω at around 16.1°C. So even though one pair of R-T values of a thermistor can be measured by the WSN node platform with onboard temperature sensor and it could be utilised as the first step to distinguish between different thermistors, the information provided by one R-T pair is not conclusive. A possible solution to this dilemma would be to limit the list of allowable thermistors to ones which do not have a cross-over point within the temperature range of interest. However, while this method has some merit, it suffers from significant limitations and is far from an ideal solution.

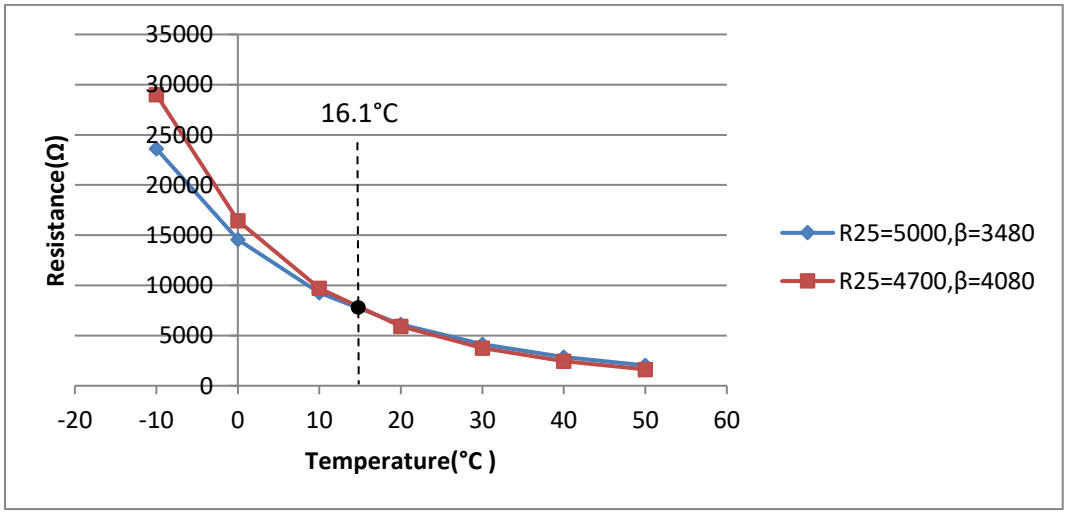


Figure 30 Example of the cross point of two thermistor R-T curves

Self-Heating Effects

Another characteristic that could be utilised to distinguish thermistors is the self-heating effect of the thermistor. The current flowing through a thermistor will generate heat that raises the body temperature of the thermistor. This is referred to as the self-heating effect. When it reaches a state of equilibrium, the self-heating effect can be expressed by the following equation:

$$\delta(T - T_A) = I^2R = \frac{V^2}{R} \qquad \text{Equation 2}$$

Where:

- T_A is the ambient temperature, and T is the body temperature of the thermistor.
- *V* or *I* is the voltage or current applied to the thermistor. *R* is the resistance of the thermistor at body temperature *T*.
- δ is the dissipation constant, usually expressed in units of milliwatts per degree Celsius (mW/°C). The value of δ indicates the amount of power necessary to raise the body temperature of the thermistor by 1°C [247].

When a thermistor is used for measuring temperature, self-heating should be avoided since it will introduce errors in the results. But this effect could be exploited to provide additional information to distinguish different thermistors in addition to the R-T value pair discussed above.

If certain voltage/current is applied to a thermistor, because of the self-heating effect the resistance of the thermistor will change. So even though the two thermistors may have the same resistance at a certain temperature, their resistance value can be changed by applying the self-heating effect and the two thermistors can be distinguished. For example, the two thermistors in Figure 30 can be distinguished by applying the self-heating as illustrated in Figure 31, i.e. the self-heating can be applied to provide the second R-T value pair to distinguish thermistors.

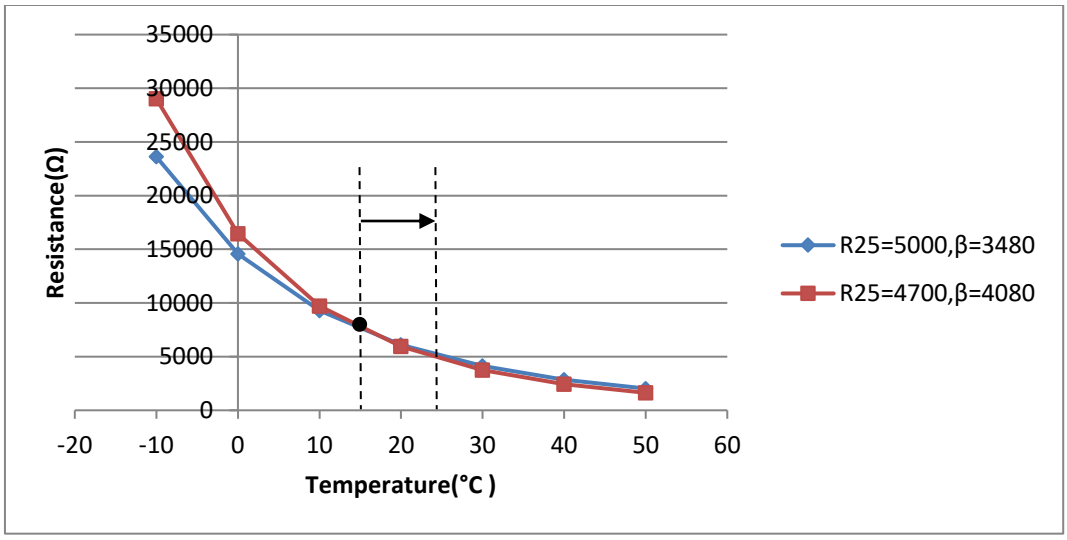


Figure 31 Exploit self-heating effect to distinguish thermistors

However, Matlab simulations carried out during this research indicated that in some cases applying the self-heating effect to different thermistors can result in the same amount of change in resistance. One of the Matlab simulations showed that, for two thermistors, both were 10kΩ at 25°C but one had $\beta = 3630K$ and $\delta = 4mW/$°C and the other had $\beta = 4080K$ and $\delta = 4.5mW/$°C , the resistance change due to the self-heating effect by applying different voltages was the same. The simulation results are shown in Figure 32. In a case like this, the self-heating effect cannot provide the second R-T value pair useful for distinguishing thermistors.

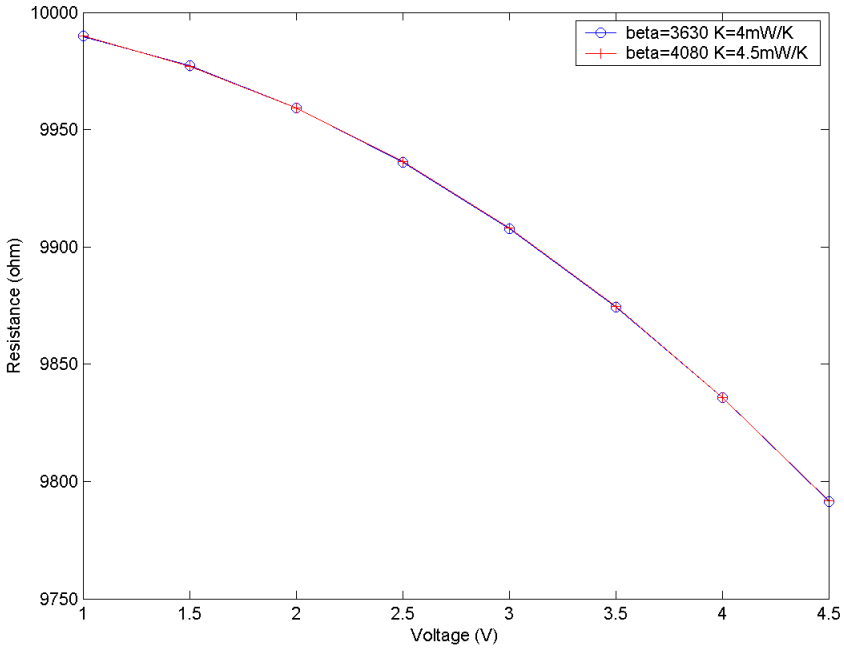


Figure 32 Matlab simulation results of the self-heating effects in special cases

Also, the dissipation factor δ is dependent on a number of factors including the thermal conductivity and relative motion of the medium in which the thermistor is located, the heat transfer from the thermistor to its surroundings by conduction through the leads, by free convection in the medium and by radiation [248]. Thus the value of δ can vary a lot when the thermistor is placed in different environments. Due to the uncertainty of the dissipation factor, although the self-heating effect can be exploited for additional information that can help to distinguish and identify different thermistors, it is very difficult to procure this information especially for a WSN sensor node platform with limited resources and means to perform tests.

In summary, it is possible for the RAWS node to distinguish a particular set of thermistors with different R-T characteristics given that the RAWS node has been preinstalled with information about these thermistors. However, it is not generally practical to identify thermistors under the usual conditions of a WSN sensor node system.

4.1.2 Thermocouple

The thermocouple is a type of temperature sensor that consists of two dissimilar metals that are joined together at one end and can produce a voltage corresponding to the temperature difference between the open end and the junction end. Thermocouples are generally inexpensive and robust, and they can measure a wide range of temperatures from -270°C up to 2320°C. Compared to thermistors, the main limitation of thermocouples is that high resolution measurements are relatively difficult to achieve due to their microvolts-level outputs [246].

Analysis indicates that the voltage-temperature characteristic is the only characteristic that can be utilised to identify a thermocouple under the usual conditions of a WSN sensor node system. There are many types of thermocouples and all are produced under industrial standards. Their voltage-temperature (V-T) characteristics are usually standardised against a reference temperature of 0°C. The most common thermocouple types are J, K, T and E [249]. This analysis studied the V-T characteristics of these four types and their standardised V-T characteristic curves across the range of 0 to 30°C have been plotted in Excel and are shown in Figure 33. As indicated in the figure, each type of thermocouple generates a unique voltage at any given temperature except 0°C, even though the type T and K thermocouples have very close output voltages in this range where the difference is less than 7 µV.

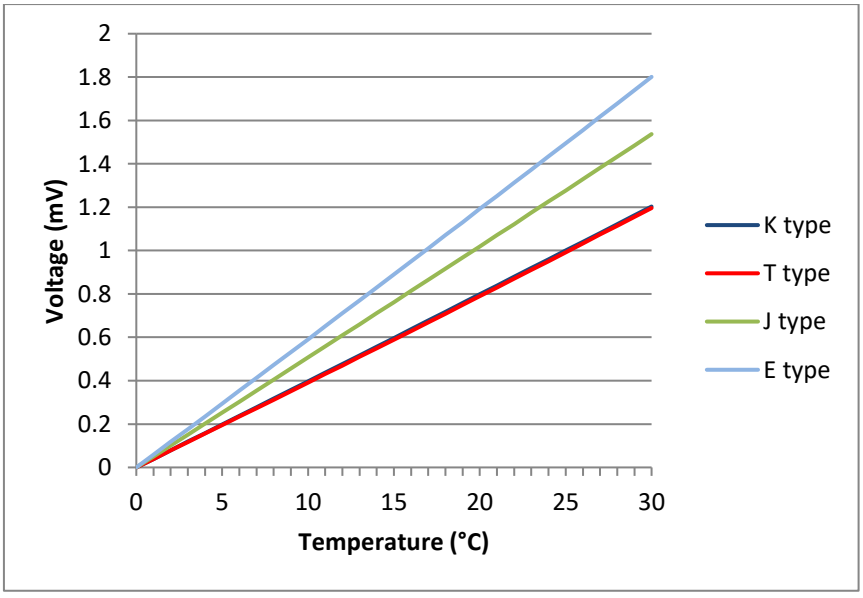


Figure 33 V-T curves of the four most common thermocouple types

Because different types of thermocouples generate unique voltage outputs at a given temperature, theoretically in the condition where the temperature difference between the junction end and open end of the thermocouple can be controlled, the type of the thermocouple can be identified by simply measuring the output voltage. So if the testing system can create a known temperature for the junction end of a thermocouple different to the temperature of the open end, measuring the output voltage of the thermocouple is a straightforward way to identify the thermocouple type. However, it would not be practical for a WSN sensor node to create a known temperature difference between the two ends of the thermocouple.

4.1.3 Other types of sensors

Besides the two major types of temperature sensors discussed above, the research also analysed several other types of sensors such as force sensors and pressure sensors. The analysis results also show that it is not practical to generally identify different sensors under the usual conditions of a WSN node system. For example, many different force sensors share the same Wheatstone bridge structure. For a Wheatstone bridge type sensor, the two common characteristics that a WSN node can measure are the impedance of the bridge and the output voltage at a particular value of the measurand unknown to the node. However, it is impossible to distinguish different force sensors with only the two aforementioned characteristics, nor to detect the other important attributes of a bridge sensor such as the sensitivity of the sensor and the suitable excitation level. Also, many other types of sensors such as gas and liquid pressure sensors and torque sensors share the same Wheatstone bridge structure as well. So it would also be impossible to distinguish different sensor types or know the actual units of the measurand.

4.1.4 Conclusion of the feasibility analysis

The idea of the systematic classification process of sensor identification is to devise a series of systematic tests to acquire key characteristics from the connected analogue sensor, and the RAWS node can utilise these characteristics to classify and then to eventually identify the sensor. The feasibility study analysed the characteristics of several different types of sensors,

including temperature sensors, force sensors, pressure sensors. The analysis results indicate it is not practical to develop such a systematic classification scheme to generally identify a large number of analogue sensors with the limited testing methods and resources that a WSN sensor node would have.

However, based on the sensor characteristics analysis, it may be possible to distinguish or identify a particular set of analogue sensors with the proposed systematic classification approach under the right conditions. For example, if in a system only several specific thermistors will be used and these thermistors do not have a cross-over point on their resistance-temperature curves such as the thermistor No.1, No.2 and No.4 shown in Figure 29 (Section 4.1.1), it is possible for the system to distinguish and identify these thermistors with the knowledge of these thermistors pre-installed into the system. Another example is when only a small set of Wheatstone bridge sensors will be used in a system and the bridge impedances of these sensors are all different. In this case the system is also able to identify these sensors with the pre-installed knowledge of these sensors. Therefore, the systematic sensor characteristics classification process approach may still be useful for developing a sensor distinguishing scheme in applications where a specific limited number of sensors are to be supported. But due to the limitations of this sensor classification approach, this research project investigated a second approach to develop a more general identification scheme.

4.2 IEEE 1451.4 TEDS based Sensor Identification

The research then investigated a second approach to realise the sensor identification scheme. This second approach is based on the sensor self-identification and self-description capability provided by IEEE 1451.4 TEDS. The sensor identification scheme retrieves and parses the TEDS data and extracts the identification information and important attributes of the analogue sensor, and it provides this information to the adaptive reconfiguration techniques.

The software routine designed for parsing the TEDS and extracting sensor information is the core of the sensor identification scheme. This routine is referred to as TEDS parsing routine for short. The sensor identification scheme also includes the 1-Wire interface circuit and the 1-Wire protocol transmission routines for communicating with the TEDS memory chip, because the

1451.4 TEDS is stored in a small capacity memory chip with 1-Wire serial interface attached to the sensor. The 1-Wire interface circuit is implemented within the PSoC microcontroller. The 1-Wire communication routines and the TEDS parsing routine are part of the overall RAWS node firmware.

Four sensors with built-in 1451.4 TEDS were selected for assisting the design of the RAWS node technology. This subsection also uses the TEDS of these sensors to assist the illustration of the 1451.4 TEDS specifications and the sensor identification scheme design. These four sensors include HBM C9B (50N) force sensor, HBM P8AP pressure sensor, Weed Instrument 101-K thermocouple and Weed Instrument 101-10B RTD.

This section firstly discusses the retrieval of 1451.4 TEDS raw data as it is the basis of the sensor identification scheme. The TEDS parsing algorithm is detailed later in this section.

4.2.1 TEDS Data Transmission

As discussed previously in Chapter 2, the 1451.4 TEDS is stored in bitwise form and the data transmission is according to the 1-Wire serial communication protocol developed by Dallas Semiconductor (now acquired by Maxim Integrated). The 1-Wire hardware interface and API functions are implemented as the necessary part of the sensor identification scheme to retrieve TEDS data and further extract sensor identification and characteristic information. The 1-Wire is a simple low cost protocol and only one signal line in addition to ground reference is needed for communications between 1-Wire devices. The signal line also serves as power supply line. The signal line is also referred to as 1-Wire bus.

1-Wire Interface Circuit

The 1-Wire communication follows the master-slave model and one master and one or more slave devices comprise a 1-Wire network. The TEDS memory chips embedded into analogue sensors are the Maxim/Dallas 1-Wire EEPROMs and they work as 1-Wire slave devices. The RAWS node takes the master device role.

The 1-Wire interface circuit is straightforward. One of the GPIO pins of the PSoC microcontroller is set as the 1-Wire pin for connecting to the 1-Wire data pin of the TEDS

memory. The 1-Wire pin is set to GPIO Pin17 (Port1, Pin7) by default in the RAWS node firmware but can be reconfigured to other pins at runtime if needed. The 1-Wire signal line of the TEDS memory requires a pull-up resistor of approximately 5kΩ. Because the GPIO pins of the PSoC can be configured as the resistive pull up drive mode with resistance around 5kΩ, no external pull-up resistor is needed. The RAWS node firmware sets up the drive mode for the 1-Wire pin to pull up the signal line. The 1-Wire interface circuit of RAWS node is shown in Figure 34.

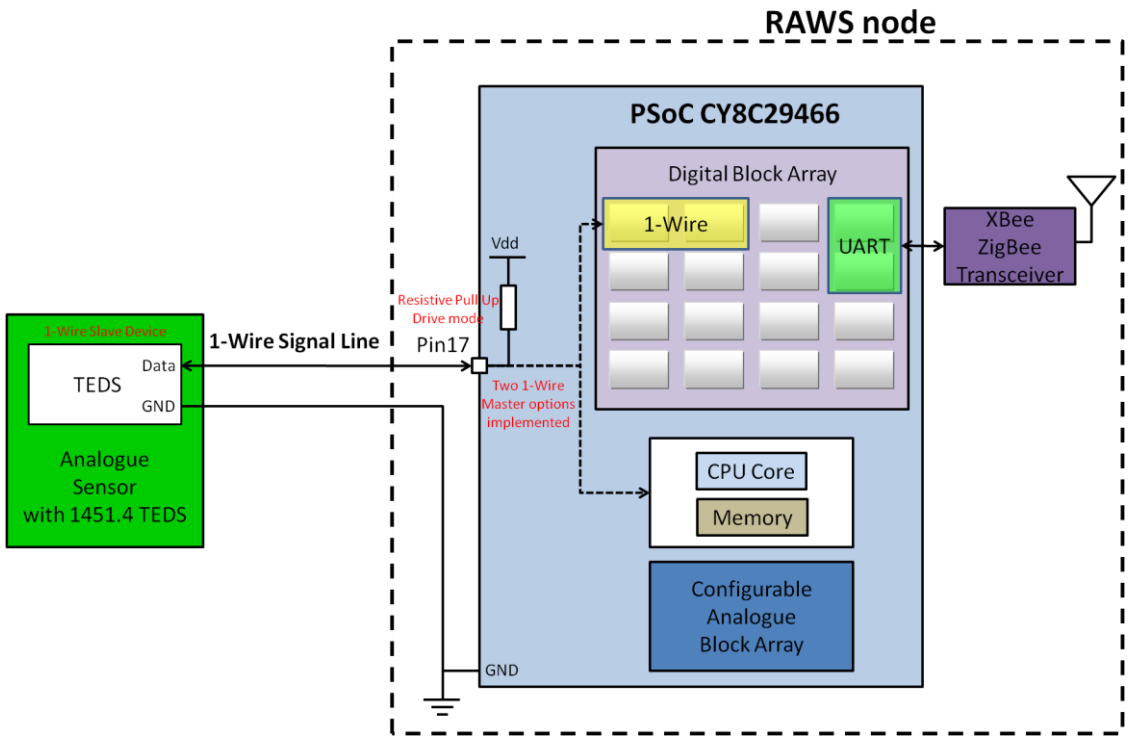


Figure 34 1-Wire Interface Connections

1-Wire Protocol Implementation

The 1-Wire communication protocol has been implemented using two methods in this research project. The first method takes a pure software approach to control the 1-Wire pin and drive the 1-Wire signal line for carrying out the data transmission. The second method utilises two PSoC digital blocks forming a 1-Wire master module to achieve the same functionality.

The principle behind both methods is the same and it is to generate the four types of 1-Wire signalling that compose all 1-Wire data transmissions. The four types of signalling are: Reset Sequence with Reset Pulse and Presence Pulse, Write 0, Write 1 and Read Data. The reset sequence is also called initialisation sequence as it is required to start any 1-Wire communications. The functionalities of the other three 1-Wire signals are indicated by their names.

All four 1-Wire signals are initiated by the master device as bit level transmission requests to slave devices. The slave devices can only respond to requests from the master and reply to the master if needed. This means the master device initiates all communications and the slave devices cannot operate without a master. The four 1-Wire signals are generated by driving the signal line (the 1-Wire pin) according to the signal timing specifications. To be more specific, they are generated by alternately pulling down and then releasing the normally pulled up signal line for precise time intervals. The technical details of the four types of signalling are explained in Appendix G.

As mentioned, this research implements the 1-Wire protocol using two methods. Both methods in essence produce the four types of signalling discussed above as the basic 1-Wire functions but through two different approaches. Each has its own advantages and disadvantages.

The pure software method generates these signals in a straightforward way. It directly reads and writes the GPIO port data register of the 1-Wire pin on PSoC (Port1 Pin7 by default). The register allows for access to the logical equivalent of the voltage on the pin. So the register enables the direct control of the 1-Wire pin. Software precision delay functions are designed for this method to reproduce the exact signal timing. Via the register and delay functions, the 1-Wire software APIs can sample and control the state of the 1-Wire signal line for reading and writing 1-Wire signals. This software method does not occupy PSoC hardware resources like the digital blocks but it requires more CPU time than the second method described below due to the software delay functions.

The second method takes a hardware approach to produce the 1-Wire signalling as mentioned above. The 1-Wire master function module formed by two digital blocks provides the clock

frequency for reproducing the signal timing and it drives the 1-Wire signal line as well. The advantage of this hardware method is that it does not use the software delay functions which send NOP instructions to the CPU and waste clock cycles.

In both methods, low level 1-Wire API functions for producing the four basic types of 1-Wire signals were developed for bitwise operations. Based on the low level functions, high level 1-Wire API functions were developed to write or read one byte or a certain number of bits at a time to facilitate the retrieval of the TEDS data. These high level functions are part of the sensor identification scheme as well.

TEDS Data Transmission Procedure

The RAWS node, as the master device, controls the 1-Wire transmissions. The sensor identification scheme utilises the high level 1-Wire API functions described in the previous subsection to communicate with the 1-Wire TEDS memory slave device. The RAWS node platform performs the following operations to interface with the TEDS memory and retrieve TEDS data:

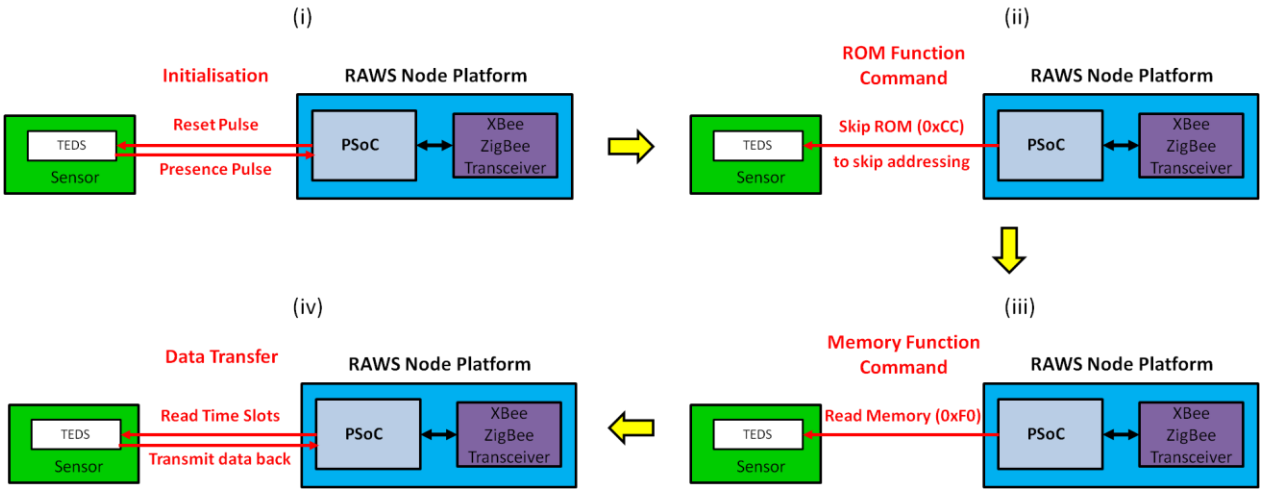


Figure 35 TEDS Transmission Procedure

i. Firstly, the PSoC sends out the initialisation sequence to set the 1-Wire TEDS memory chip of the connected sensor into the ready-to-operate state.

ii. According to the 1-Wire protocol, the master device then needs to address the target TEDS memory for data transmission by sending out a type of 1-Wire function commands

called ROM function commands. If device addressing is not required, the addressing operation can be skipped. In the RAWS node prototyping, different 1-Wire pins are used for interfacing the TEDS memory of different sensors, so on each 1-Wire signal line there is only one slave device so device addressing is not needed. PSoC skips addressing by issuing a 'Skip ROM' (0xCC) function command.

iii. The PSoC issues a 'Read Memory' command (0xF0) to set the TEDS memory ready for transmitting data. The 'Read Memory' is one of the memory function commands which are another type of 1-Wire function commands. The master sends the 1-Wire function commands by writing the 1-byte value that represents the command on the signal line. The PSoC calls the developed high level 'Write Byte' API to send out commands.

iv. The PSoC calls high level 1-Wire read functions that consist of a number of basic 1-Wire read signals to retrieve TEDS data from the memory chip.

4.2.2 Parsing TEDS Data and Extracting Sensor information

The TEDS parsing routine processes the retrieved TEDS data and extracts the sensor information. As discussed in Chapter 2, the 1451.4 TEDS information is stored in bit stream form without separation between the data fields and it is categorised into sections. Therefore, the TEDS parsing routine performs a series of bit-level operations to parse the raw data and acquire information from each TEDS sections. The first two TEDS sections, the Basic TEDS and Standard TEDS, provide identification information and characteristic attributes respectively, important for hardware reconfiguration. The parsing algorithm was also developed to focus on these two sections.

Sensor Identification Information Extraction

The TEDS parsing routine firstly processes the Basic TEDS section which is the first section in the dot 4 TEDS structure. It is the compulsory section of the dot 4 TEDS, and it contains the information that uniquely identifies the transducer [125]. In the RAWS node system, the content of the Basic TEDS is used by the local and remote configuration settings techniques to search for the hardware configuration for the identified sensor.

The Basic TEDS contains 64 bits of sensor identification information including five fields which are the 14 bits manufacturer ID, 15 bits model number, 5 bits version letter, 6 bits version number, and 24 bits serial number. The parsing algorithm extracts the value in each field of the Basic TEDS according to the standardised format shown in Figure 36 below which is the same for every sensor. The total 64 bits uniquely identify a sensor, but the routine still parses these bits and extracts the five fields within the section for informative purpose. The structure and length of the Basic TEDS are fixed so the parsing operation is straightforward.

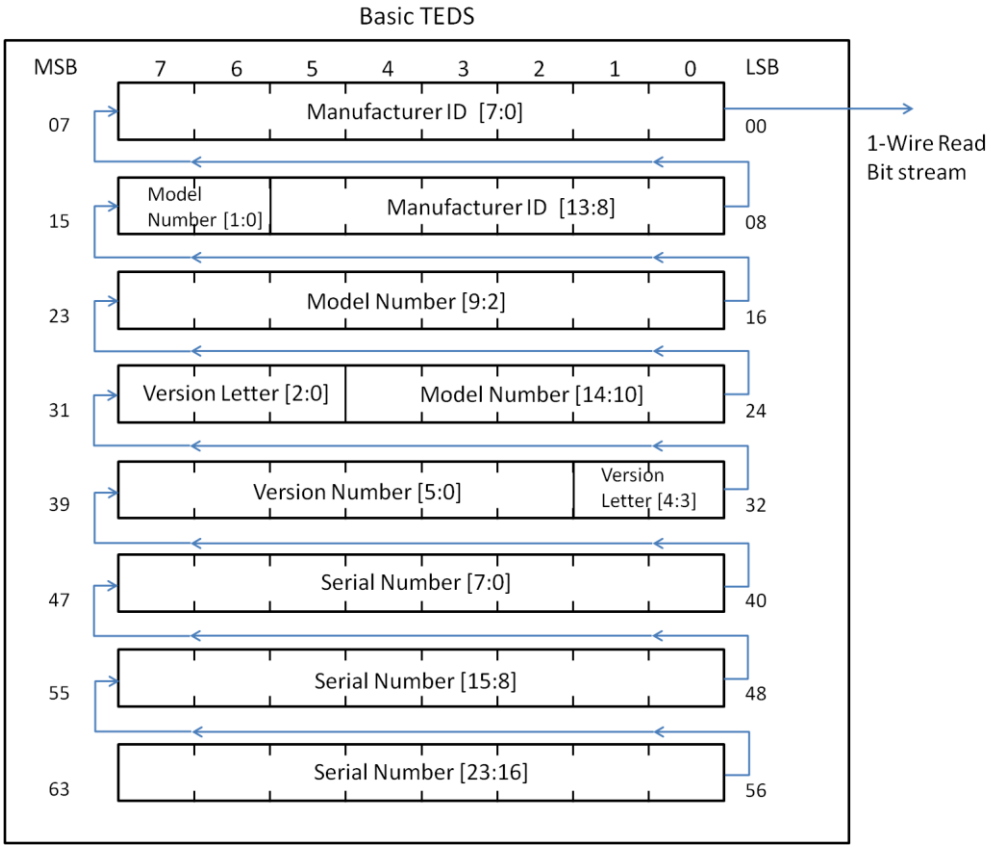


Figure 36 Basic TEDS Format

Based on the dot 4 standard, at the end of one TEDS section, a 2-bit 'selector of descriptor' is used to indicate the 'type' of the following section if there is a following section (see Figure 18 (b) in Chapter 2). The TEDS parsing routine also handles the value of this 'selector of descriptor' at the end of the Basic TEDS in order to operate accordingly. The common 1451.4 TEDS, including the TEDS of the four sensors used in this research project for design and testing, has the Standard TEDS section as the second section. Therefore, the parsing routine continues to

process the Standard TEDS section to extract the key sensor attributes vital for the RAWS node system to intelligently utilise its hardware resources.

Sensor Attributes Information Extraction

The TEDS parsing routine extracts the sensor attributes information from the Standard TEDS section. However, unlike the parsing of the Basic TEDS, the TEDS parsing routine handles the Standard TEDS section for different types of sensors in different ways due to the use of 'sensor type template' for this section. Further to the discussion in Chapter 2 Section 2.2.2, the concept of templates is employed to define the data structure of the Standard TEDS for different sensor types. A 'sensor type template' specifies the key attributes and the attribute data fields format of the Standard TED section for one type of sensor, i.e. a template specifies what key attributes are included in the Standard TEDS section for a type of sensor and how these attributes are stored in the memory. Different templates are defined for different sensor types. Therefore, the TEDS parsing routine also supports different sensor type templates.

The template specifies the Standard TEDS format, so it is also the way to interpret the Standard TEDS content. The template itself is not contained in the TEDS memory. Instead, the template has an ID number and this number is included in the TEDS to indicate which template is to be used for the current Standard TEDS. The developed TEDS parsing routine has many subroutines designed based on the templates to interpret different Standard TEDS and extract different types of sensor information, and it selects the corresponding subroutine based on the template ID. The TEDS parsing algorithm's high-level operation sequences described so far are summarised in the flowchart below.

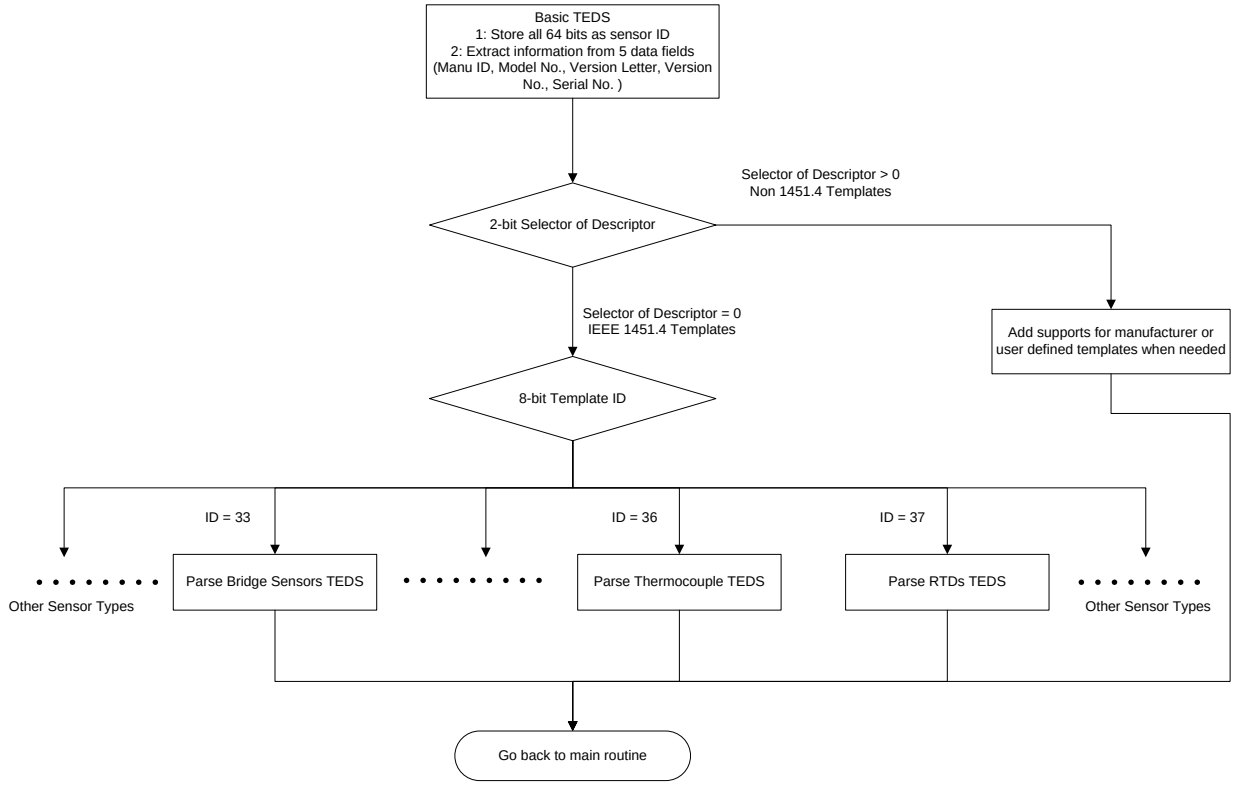


Figure 37 TEDS Parsing Routine Flowchart

The low level details of how the TEDS parsing routine parses the Standard TEDS section are described as follows.

The parsing routine first extracts the 8-bit template ID located at the beginning of the Standard TEDS bit stream in order to determine the template type. The IEEE 1451.4 standard defines a set of templates for the most common sensor types. The sensor types of the four sensors used in this research are also covered. Within the four sensors, the force sensor and the pressure sensor are both bridge type sensors and the other two are the thermocouple type and the RTD type respectively. The template IDs for these three sensor types are shown in Table 11 and indicated in the flowchart Figure 37 as well. The full list of the IEEE 1451.4 templates and their template ID can be found in Appendix F.

Name of Template	Template ID
Bridge sensors	33
Thermocouple	36
Resistance temperature detectors (RTDs)	37

Table 11 IEEE 1451.4 TEDS Sensor Type Template

The routine supports most of the 1451.4 templates. Here the bridge sensors template shown in Table 12 below is used to help to further explain how the parsing algorithm acquires sensor information because this template has all of the common sensor attributes areas (marked as ⓪ on Table 12), including the ID area, measurement attributes area, electrical signal output attributes area, and excitation/power supply attributes area.

⓪ Attributes Area

③

Function	Select	Property	Description ①	Bits	Data Type	Units
ID	② —	TEMPLATE	Template ID	8	Integer (Value=33 in the example)	—
Measurement	Select Case – Physical Measurand		Selector - Enumeration	6	Select Case	—
	Cases 0-45	%MinPhysVal	Minimum physical value	32	Single Float	Depend on Measurand
		%MaxPhysVal ④	Maximum physical value	32	Single Float	
Electrical Signal Output	Select Case – Full-Scale Electrical Value		Selector-Decide next field	2	Select Case	—
	Case 0	%MinElecVal	Minimum electrical output	11	ConRes (±1, step 1E-3)	V/V
	④	%MaxElecVal	Maximum electrical output	11	ConRes (±1, step 1E-3)	V/V
	Case 1	%MinElecVal	Minimum electrical output	19	ConRes (±6.55E-3, step 25E-9)	V/V
		%MaxElecVal	Maximum electrical output	19	ConRes (±6.55E-3, step 25E-9)	V/V
	Case 2	%MinElecVal	Minimum electrical output	32	Single Float	V/V
	Field Options	%MaxElecVal	Maximum electrical output	32	Single Float	V/V
	—	%MapMeth	Mapping Method		Assign = 0, "Linear"	—
	—	%BridgeType	Bridge Type	2	Enumeration: Quarter | Half | Full	—
	—	%SensorImped	Bridge element impedance	18	ConRes (1 to 26.2k, step 0.1)	Ω
	—	%RespTime	Response Time	6	ConRelRes (1E-6 to 7.9, ±15%)	s
Excitation Supply	—	%ExciteAmplNom	Excitation level, nominal	9	ConRes (0.1 to 51.1, step 0.1)	V
	—	%ExciteAmplMin	Excitation level, min.	9	ConRes (0.1 to 51.1, step 0.1)	V
	—	%ExciteAmplMax	Excitation level, max	9	ConRes (0.1 to 51.1, step 0.1)	V
Calibration Information	—	%CalDate	Calibration Date	16	DATE	—
	—	%CalInitials	Calibration initials	15	CHR5	—
	—	%CalPeriod	Calibration period	12	UNINT	days
Misc.	—	%MeasID	Measurement location ID	11	UNINT	—
			Total bits:	209 to 251 bits		

Table 12 Standard TEDS Template Example – Bridge Sensors Template

Sensor Type Information - ID Area

As discussed above the TEDS parsing routine firstly extracts the 8-bit template ID value (① on Table 12) from the ID area, which is always the first data field of the Standard TEDS section, and then employs the corresponding template to parse the rest of the section. More importantly, the parsing routine also obtains the sensor type information from this field and will provide it to the adaptive reconfiguration techniques, which then utilises this information as a factor for the selection of suitable function modules for building the signal processing chain. For example, if the parsing routine finds out that the connected sensor is a bridge type (Template ID = 33), then during the reconfiguration process the RAWS system will consider the instrumentation amplifier suitable for the gain stage of the signal processing circuit based on this information. In another example, low resolution ADCs will not be selected for thermocouples due to the precision requirements in this situation.

Sensor Physical Input Information - Measurement Attributes Area

The TEDS parsing routine acquires and synthesises sensor input information, such as the physical measurand, the minimum and maximum physical values the sensor can measure (i.e. the sensor's measurement range), from the measurement attributes area, and provides them to the data conversion routine of the RAWS node firmware for calculating the sensor physical readings. Starting from this area, the parsing routine has to parse the bit stream based on the sensor type template.

The parsing routine needs to cope with the selection structure within the templates which is introduced by the 'select-case' selector defined in the dot 4 standard. In the Standard TEDS, a 'select-case' selector is often used to either enumerate sensor attributes or indicate that the format of the subsequent data field shall be chosen from a number of options listed in the template. The expression of the physical measurand is the former case (② on Table 12). As shown in the example template, the measurand of the sensor is denoted by the value of the 6-bit physical measurand 'select-case' selector. Based on the value of this selector, the parsing routine finds out the actual meaningful measurand according to the table of the enumeration of 'select-case' values for the physical measurand defined by the dot4 standard. For the two

bridge sensors used in this research, their 'select-case' selector values and the corresponding physical measurand is shown in Table 13. The full physical measurand enumeration table can be found in Appendix F.

Sensor	Physical Measurand Selector Value	Physical Units
C9B force sensor	4	Newton
P8AP pressure sensor	12	Pascal

Table 13 Physical Measurand Selector Enumeration Example

For some data fields like the physical measurand that are expressed by the 'select-case' enumeration, the TEDS parsing routine can determine their meanings straightforwardly. However, for most of the data fields, the TEDS parsing routine also needs the information from the 'Data Type' column and the 'Bits' column (③ on Table 12) of the template to convert the content in the data field into meaningful values. For example, the parsing routine extracts the minimum and maximum physical value as 32-bit floating-point numbers from the TEDS raw data stream based on the indications of the two aforementioned columns.

Sensor Output Information - Electrical Signal Output Attributes Area

The parsing algorithm acquires important information about the sensor electrical output signal from this area, such as the minimum and maximum electrical output values, output signal mapping method, sensor impedance, etc. Many of these attributes are critical for the adaptive reconfiguration technique to select function modules and module parameters. For example, the maximum electrical output is one of the important attributes for determining the gain of the input amplifier of the signal chain. Some of these attributes are also utilised by the data conversion routine for converting the raw data from the connected sensor into actual readings, e.g. the signal mapping method is important for data conversions.

In this attributes area, the 'select-case' selector is often utilised to indicate the format options of the following data field. In the bridge sensors template example in Table 12, three data format options are available for the minimum and maximum electrical output fields, and the field length and the data type of these options are different (④ on Table 12). The 2-bit full-scale electrical value 'select-case' selector specifies which option is actually being used for the

current TEDS. As a result of this selection structure, the date fields and the length of the Standard TED can vary even if the same template is applied. The length of the TEDS bit stream is also varied. Therefore, the TEDS parsing routine also handles the same TEDS template differently according to the 'select-case' selector.

Sensor Excitation Information - Excitation Attributes Area

The parsing routine obtains the key information on the power requirements of the connected sensor from this area of the Standard TEDS section. Attributes including minimum, nominal and maximum excitation levels are utilised by the system for providing the correct excitation.

After extracting the key sensor attributes from the four areas discussed above, the TEDS parsing routine continues to process the rest of the Standard TEDS section where useful data can be acquired for information purposes such as the calibration date of the sensor. This kind of information is not critical for the reconfiguration of the RAWS system so it is not detailed here.

The flowchart of the subroutine for the bridge sensors type TEDS parsing discussed above is shown in Figure 38 as an example of the Standard TEDS parsing process.

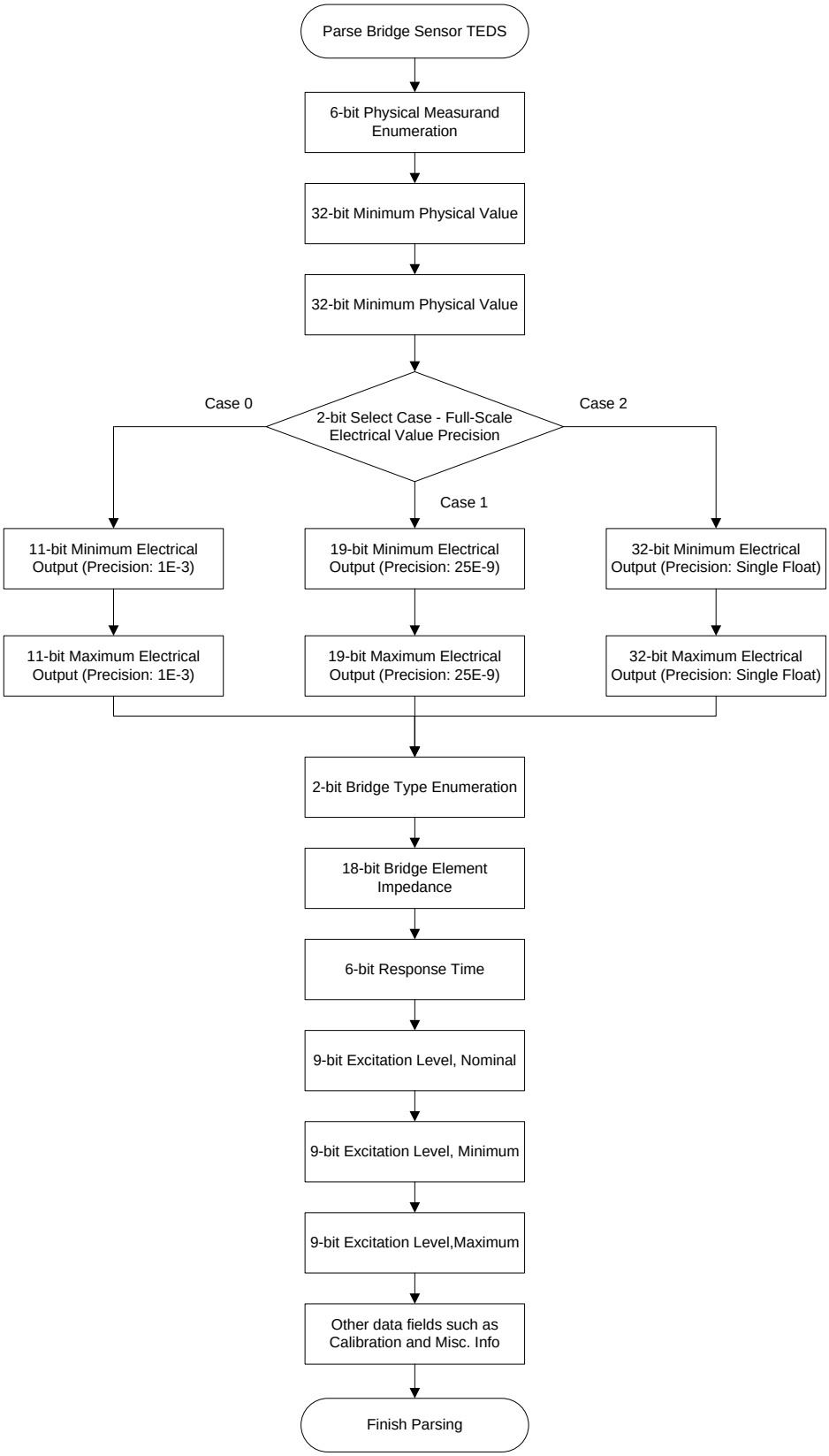


Figure 38 Bridge sensor type Standard TEDS parsing

TEDS Parsing Summary

The TEDS parsing routine is in charge of parsing the retrieved TEDS data and extracting sensor identification and characteristics information. It performs a series of bit-level operations to parse and extract information from the TEDS raw data bit stream.

Because the TEDS uses the 'selector of descriptor', templates, and 'select-case' selectors to construct different data structures for different types of sensors, the TEDS parsing routine itself also features a complex selection structure for coping with various TEDS data structures as shown in Figure 37 and Figure 38. Despite the complex algorithm structure, the routine follows a sequence of general steps to parse the TEDS and is summarised below:

i. The routine firstly extracts the sensor identification information from the Basic TEDS. The total 64 bits of the Basic TEDS uniquely identify a sensor, so the whole section of Basic TEDS is stored in the RAWS node as the ID of the connected sensor. This sensor ID is utilised in the local and remote configuration settings techniques for searching for the matching hardware configuration settings. The routine still parses the Basic TEDS section and extracts the five fields within the section for information purposes.

ii. The routine then branches based on the 2-bit 'selector of descriptor' at the end of the Basic TEDS (Figure 37). The 'selector of descriptor' indicates whether the following section, which is the Standard TEDS section in this case, is described by the 1451.4 defined templates or other types of templates defined by manufacturers or users. The TEDS parsing routine supports most of the 1451.4 defined templates. The manufacturers or users specified templates can also be added in when needed.

iii. The routine further branches based on the template ID to employ corresponding sensor type templates to parse the Standard TEDS content. The routine then goes into the corresponding subroutine that processes the rest of the Standard TEDS based on the template. During the Standard TEDS parsing process, the routine will furthermore branch if 'select-case' selectors are met (Figure 38). In this setup, the routine extracts and stores sensor characteristics information from each data field, and these sensor

attributes will be utilised by the intelligent algorithm adaptive technique for instructing hardware reconfiguration.

4.3 Sensor Identification Scheme Summary

The sensor identification scheme plays an important role in the RAWS system as it acquires the sensor identification and characteristics information and this information is critical for the adaptive reconfiguration techniques of the RAWS node platform to dynamically reconfigure the hardware to support the connected sensor.

The research explored the proposed systematic characteristics classification of sensor identification and indicated that it has limitations in terms of supporting a large number of sensors. However, it still provides a possibility of distinguishing sensors for the WSN node system under the proper conditions. The potential use of this approach can be further investigated.

The RAWS technology implemented a more general sensor identification scheme to acquire sensor information by exploiting the TEDS concept and sensor self-description capability. The RAWS system can parse and extract the key features and attributes of the unknown sensors, and this important sensor information is utilised to enable the RAWS node to adaptively and autonomically support different sensors. Overall the research developed a sensor identification scheme for the WSN sensor node to generally identify sensors and obtain key information and utilised this scheme as one of the foundations to achieve the autonomic multi-sensing capability. As discussed before this is one of the novelties of the research compared to other similar research work.

Chapter 5 Adaptive Reconfiguration Techniques

This chapter describes the development and implementation of the adaptive reconfiguration techniques. The adaptive reconfiguration techniques utilise the sensor identification and characteristics information to adaptively reconfigure the RAWS node hardware to accommodate different sensors. These techniques combine the reconfigurability of the programmable hardware and the sensor identification ability to achieve the autonomic multi-sensing capability.

Three techniques have been developed to conduct adaptive hardware reconfigurations, the intelligent algorithm technique, the local configuration settings technique, and the remote configuration settings technique. These three techniques have been implemented as software algorithms as part of the RAWS node firmware. The three techniques adopt different methods to utilise sensor information to perform reconfigurations. The intelligent algorithm is key technique of the RAWS technology and it is able to autonomically assemble analogue function modules and select parameters for the modules to build a suitable signal conditioning and processing chain to support the connected sensor based on the sensor characteristics. Both the local and remote configuration settings techniques search and load the corresponding hardware configuration settings based on the sensor ID.

The high level concepts of these three techniques could be applied to different suitable programmable hardware platforms. However, the implementation of these techniques still involves a lot of low level platform-dependent hardware micromanagement. Therefore, in order to better explain the design and testing of the three adaptive techniques especially the intelligent algorithm approach, the underlying principle of the reconfiguration processes including the analogue hardware resources and the low level hardware reconfiguration operations are discussed first in the next section 5.1. The design of the adaptive reconfiguration techniques is discussed in more detail later in section 5.2.

5.1 The Principle of Adaptive Reconfiguration

As mentioned above, the adaptive reconfiguration techniques implementation includes low-level hardware-dependent micromanagement. In the RAWS node platform, this means that the

low-level operations need to directly deal with the hardware resources of the PSoC microcontroller, especially the analogue resources. Therefore, how the adaptive techniques utilise the analogue resources of the platform, especially the configurable analogue blocks, are discussed in detail in this subsection.

5.1.1 Analogue Resources

The adaptive reconfiguration techniques mainly operate on the analogue subsystem, and the main component that they operate on is the configurable analogue blocks array. Further to the discussion in Chapter 2, the blocks array is comprised of continuous time CABs and switched capacitor CABs, and blocks are organised into columns of three, where each column consists of one CT block and two SC blocks. The PSoC microcontroller used for proof-of-concept contains four columns of analogue blocks, and the architecture diagram of its analogue subsystem is shown in Figure 39 below (The overall PSoC architectural block diagram can be referred to Figure 15 in Chapter 2).

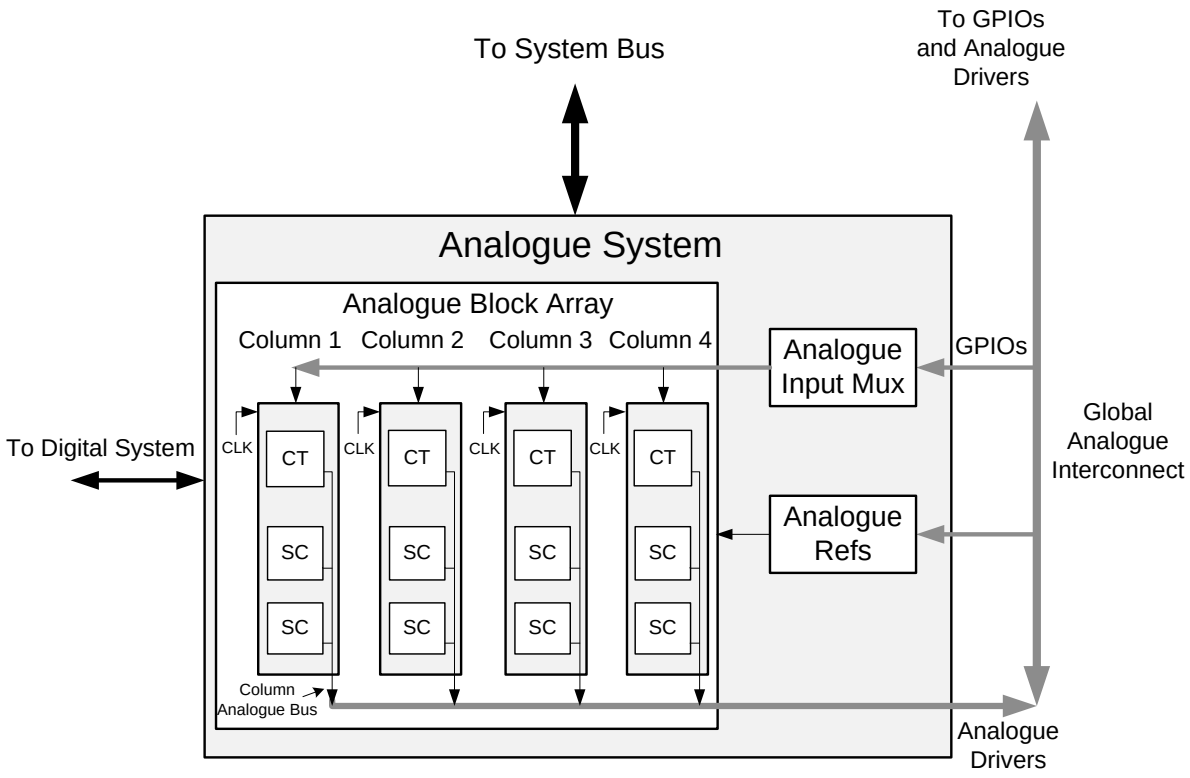


Figure 39 Analogue System Top Level Architecture

The adaptive reconfiguration techniques configure all the blocks by managing the registers that control the circuit structure as well as the input and output connections. These registers can be accessed at runtime by the software programme being executed on the PSoC CPU core, so the hardware reconfiguration can be performed at runtime. This is also the basis of how the RAWS node dynamically changes the configurations of its hardware.

The RAWS system is able to freely configure the blocks into various analogue functions since all the blocks are generic configurable blocks and they can be used individually or chained together in certain patterns to form more complicated modules. However, the system utilises the continuous time CABs and the switched capacitor CABs differently because the two types of CABs have their own advantages at the different stages of the signal processing chain. This will be analysed in more detail in the next two subsections.

The adaptive techniques also employ other analogue resources shown in Figure 39 to support the CABs and to build signal processing chains for sensors. One of the frequently reconfigured hardware resources is the analogue reference generator. It produces a set of reference voltage levels for the analogue blocks including the analogue ground (AGND), the high reference (RefHi) and the low reference (RefLo). These three reference voltages are generally used by analogue function modules for signal comparisons and data conversions, and they are also configurable because the voltage source of the reference generator is selectable. In terms of other frequently used analogue resources such as the global analogue interconnect, analogue input multiplexers, and analogue drivers, their functionalities are explicit and can be referred to Chapter 2 Section 2.1.2. Same as for the analogue blocks, the RAWS system can reconfigure all the analogue resources mentioned above during runtime through the corresponding control registers.

5.1.2 Continuous Time Block Configuration

In the RAWS node system, the adaptive reconfiguration algorithm mainly utilises the continuous time block at the front end of a signal chain for building signal conditioning functions such as different types of amplifiers due to its relatively high maximum signal bandwidth.

The schematic of the kernel circuit of the CT block is shown in Figure 40. The continuous time block is built around an operational amplifier (opamp). A configurable precision resistor string is located in the feedback path of the opamp. Multiple analogue multiplexers and switches determine the circuit topology and the signal flow inside the block as well as the input and output signal connections.

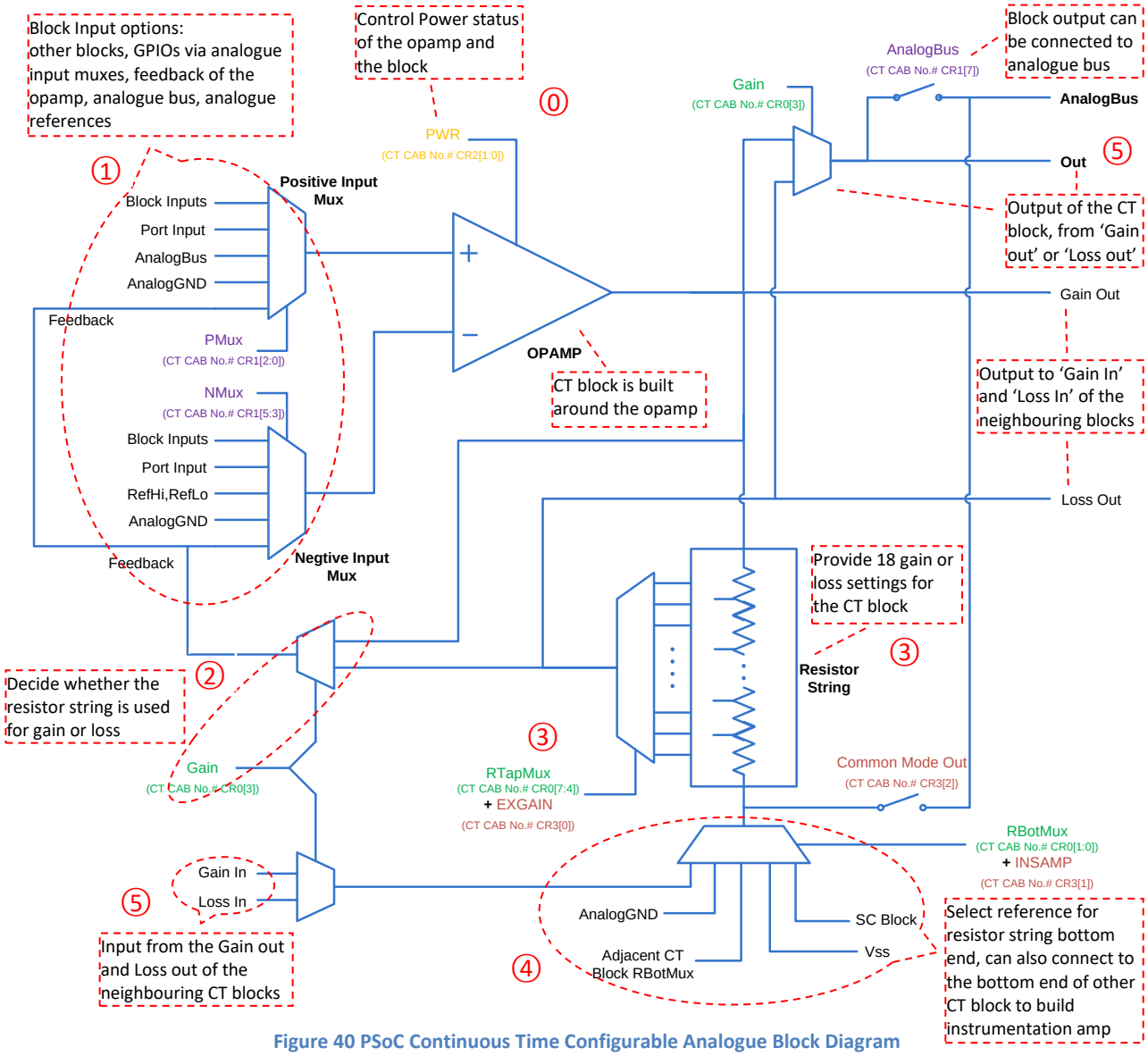


Figure 40 PSoC Continuous Time Configurable Analogue Block Diagram

The adaptive reconfiguration algorithm manages the configuration settings of the CT block by controlling the registers called continuous time block control registers. Four 8-bit CT block control registers (these four registers will be referred as CT CAB CR0 – CR3, or CT_CR0 – CR3 for

short) contain the full configuration of one CT block. The key bits of the CT block control registers used by the reconfiguration algorithm are highlighted in Figure 40 and summarised in Table 14 as well.

CT CAB CR#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	RTapMux[7:4]				Gain		RBotMux[1:0]	
CR1	AnalogBus		NMux[5:3]			PMux[2:0]		
CR2							PWR[1:0]	
CR3						CMOut	INSAMP	EXGAIN

Table 14 Continuous Time Block Control Registers

These key configuration settings of the CT block are detailed as follows:

- Power Mode

  The opamp is the core of the CT block. The PWR bits of CR2 control the power states of the opamp and the CT block. The RAWS node firmware manages the working mode of the block via the PWR bits: full-power-mode during reading sensors and off-mode for power saving, as shown at ⓪ on Figure 40.

- Inputs

  The positive (non-inverting) and negative (inverting) inputs of the opamp are multiplexed through the two input multiplexers inside the CT block as shown at ① on Figure 40. The adaptive techniques select different input sources accordingly based on the type of the function module being built and where the module is positioned within the signal processing chain. For example, when the RAWS system configures a PGA which is located at the front-end stage of the signal chain for amplification, the selections of the positive and negative inputs need to be the GPIO of the input signal and the opamp feedback respectively as shown in Figure 42 at the end of this subsection. The adaptive techniques control the multiplexers of the positive input and the negative input via the PMux bits and NMux bits of CR1 respectively.

- Configurable Resistor String

  The adaptive reconfiguration techniques adjust the gain setting of the CT block by configuring the resistor string. The resistor string consists of 48 units of resistor

elements of equal value in series and 18 resistor taps (switches), as shown in Figure 41. The selection of the tap can divide the resistor string into two resistor segments with 18 different resistance ratios. The Gain bit of CR0 decides whether the resistor string is connected around the opamp for gain (i.e. amplification) or for loss (i.e. attenuation) as shown at ② on Figure 40. For the four sensors involved for the proof-of-concept, the gain mode is used. The combination of RTapMux bits of CR0 and EXGAIN bit of CR3 (③ on Figure 40) specifies the selection of the resistor tap and in turn chooses the gain setting of the CT block from 18 gains in the range of 1 to 48 or from 18 losses in the range of 0.02 to 1.

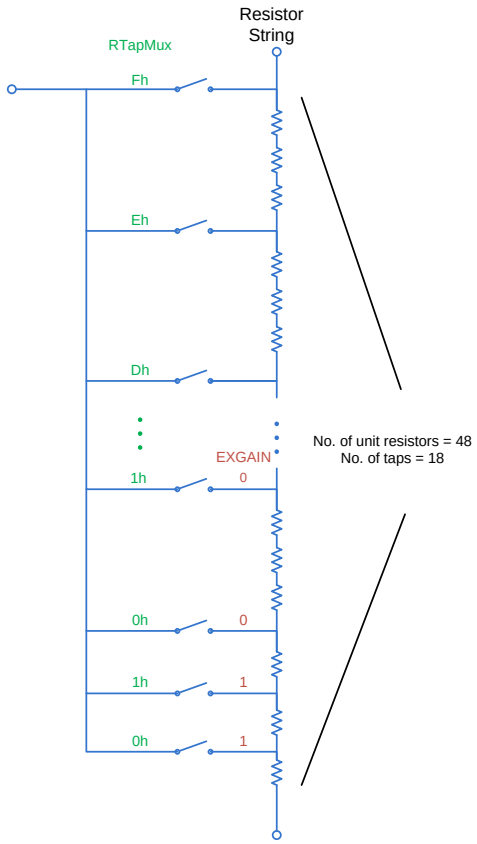


Figure 41 Configurable Resistor String Diagram

- Reference Point

  The bottom end of the resistor string is the reference point of the opamp circuit of the CT block as shown at ④ on Figure 40. The adaptive techniques also need to properly select the reference point connection to correctly build a function module. For example,

when the RAWS system configures a PGA module for amplifying external input signals, V_{SS} is often the choice for the reference point. For another example, when the adaptive algorithm builds a three opamp instrumentation amplifier, the bottom ends of the two CT blocks are connected together to build the differential input stage of the amplifier. The reconfiguration algorithm selects the bottom end connection via the resistor string bottom multiplexer which is controlled by the combination of the RBotMux bits of CR0 and the INSAMP bit of CR3.

- Outputs

  The output connection of a CT block also has different options, as shown at ⑤ on Figure 40. One option often used by the RAWS system is connecting the output of a CT block to the other analogue blocks as input, e.g. the system connects the output of the PGA to the input of an ADC block. Another useful option for the system is to connect the output to the analogue bus of the column, and through the analogue bus the output signal can be routed to the GPIOs with analogue drivers, e.g. the system routes the output of a DAC module to a GPIO pin for providing an excitation voltage to a connected sensor. The AnalogBus bit of CR1 controls the CMOS switch for connecting the block output to the analogue bus.

  A few other components are not considered as the kernel circuit structure of the CT block in the RAWS node design so they are not discussed here but can be found in the CT block complete circuit diagram in Appendix H.

Overall, the adaptive reconfiguration techniques manage the CT block control registers discussed above to configure the block circuit and programme CT blocks into different analogue functions, mainly signal conditioning functions. Figure 42 shows an example of the RAWS node platform turning a generic CT block into a PGA using the key register bits settings in Table 15. In this example, the PGA amplifies the external input signal from GPIOs by 4 times.

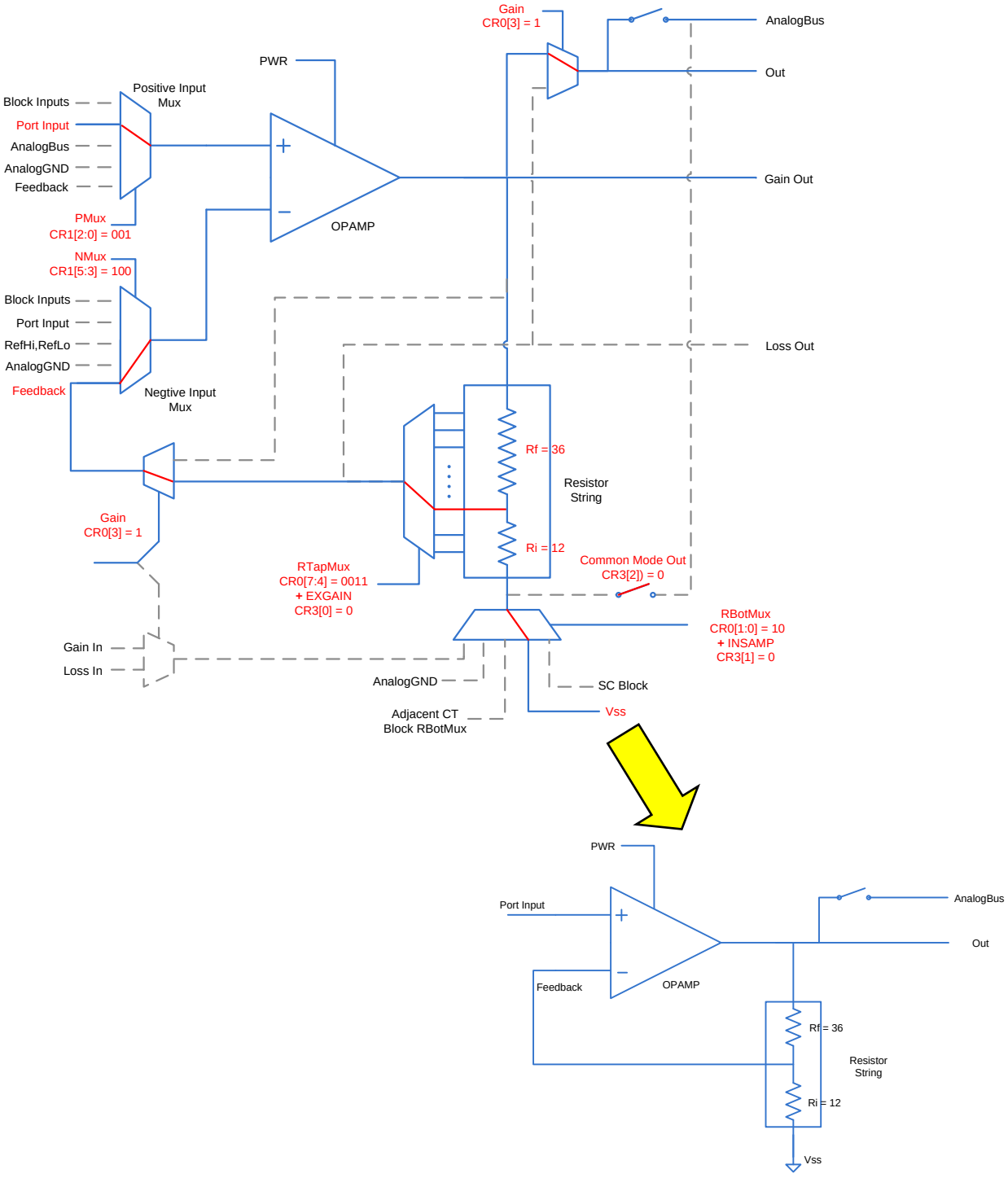


Figure 42 Configuring a CT block into a PGA module

Register Bits	Value (Binary)	Description
PMux (CR1[2:0])	001	Non-inverting input is connected to GPIOs via input mux
NMux (CR1[5:3])	100	Inverting input is connected to the feedback path of opamp
Gain (CR0[3])	1	Block is configured as for gain
RTapMux (CR0[7:4])	0011	Gain is set as 4 (The resistor tap is selected to Rf : Ri = 36 : 12, and gain = 1+ Rf/Ri)
EXGAIN (CR3[0])	0	
RBoxMux (CR0[1:0])	10	Vss is connected to the resistor string bottom end as reference
INSAMP (CR3[1])	0	

Table 15 CT Block Key Register Bits Settings for PGA module

5.1.3 Switched Capacitor Configurable Analogue Block

In RAWS node system, the adaptive reconfiguration algorithm utilised the switch capacitor block for more complicated function modules, such as delta-sigma, successive approximation, and incremental analogue-to-digital conversion, digital-to-analogue conversion, and SC filters, because the SC blocks can support more flexible and complicated circuit structures.

There are two types of switched capacitor blocks in PSoC and one analogue column has one of each. The main components and the kernel circuit structure of the two types of SC blocks are the same as shown in Figure 43. The switched capacitor block is also built around an opamp. Two types of capacitor arrays with selectable unit values are located at the input paths and the feedback path of the opamp. A group of switches, synchronous to the block internal clocks Φ_1 and Φ_2 and with modifiable behaviour, together with configurable capacitors comprise the switched capacitors and determine the signal topology inside the block.

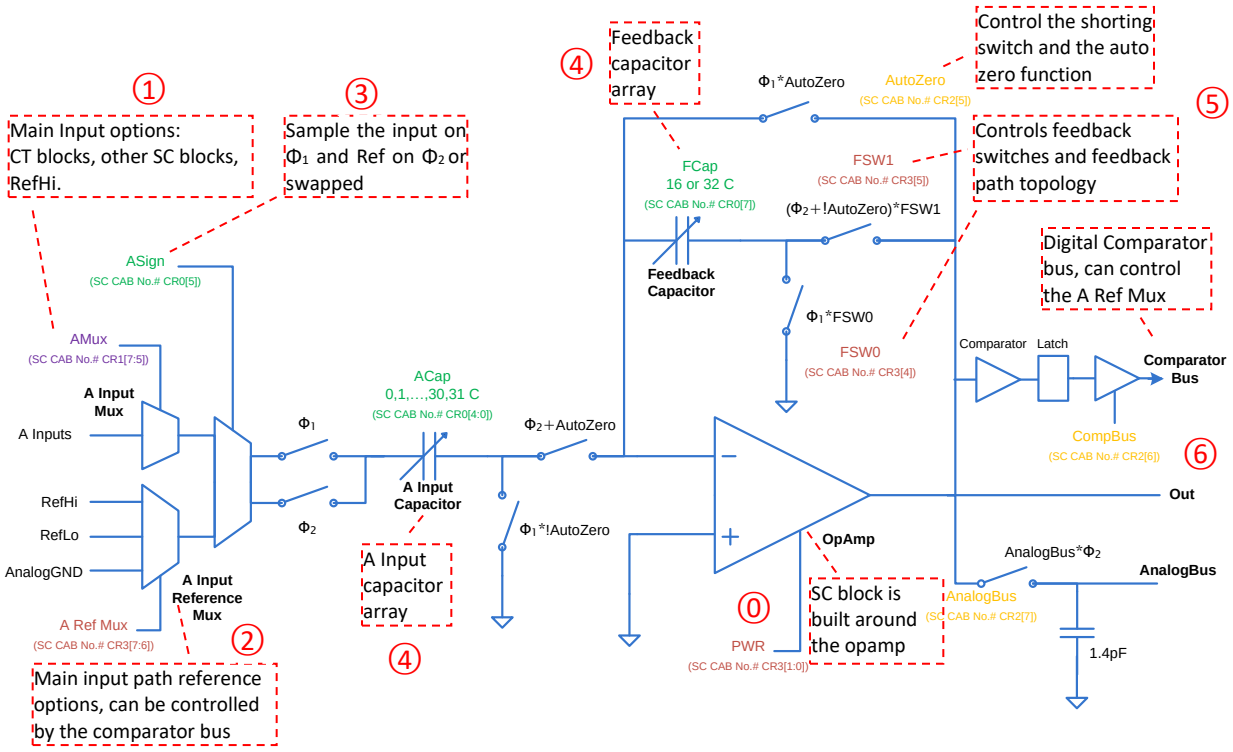


Figure 43 PSoC Switched Capacitor Configurable Analogue Block Diagram

Similar to the CT block, the adaptive reconfiguration techniques also manages the configuration settings of an SC block via four 8-bit switched capacitor block control registers (SC CAB CR0 – CR3 or SC_CR0 – CR3 for short). The key register bits for the reconfiguration algorithm are highlighted in Figure 43 and summarised in Table 16 below as well.

SC CAB CR#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	FCap	ClockPhase	ASign	ACap[4:0]				
CR1	AMux[7:5]			BCap[4:0]				
CR2	AnalogBus	CompBus	AutoZero					
CR3	ARefMux[7:6]		FSW1	FSW0			PWR[1:0]	

Table 16 Switched Capacitor Block Control Registers

These key configuration settings of the SC block are detailed as follows:

- Power Mode

The RAWS node firmware manages the power states of the opamp and the SC block through the PWR bits of CR3: full-power mode for sensor readings and off mode for saving power (⓪ on Figure 43).

- Inputs

  The A input is the main signal input of the block. The RAWS system can select different sources for the A input signal via the A input multiplexer by controlling the AMux bits of CR1 as shown at ① on Figure 43. Because the system mainly uses SC blocks for conversion functions which usually do not need to directly handle input sensor signals, the input options of SC blocks are usually the outputs of CT blocks or other SC blocks. The system can also specify the reference source for the main A input. The reference is multiplexed into the signal path via the A reference multiplexer and controlled by ARefMux bits of CR3, as shown at ② on Figure 43. For the similar reason mentioned above, the reference options for the SC blocks are the three reference voltages (AGND, RefHi and RefLo), while the references that are commonly used in CT blocks, such as V_{SS}, are not used by SC block functions in the RAWS system. The system also controls whether the input to be sampled at Φ_1 for positive output or at Φ_2 for negative output with reference to the analogue ground. This is managed by the sign of the main input, i.e. the ASign bit of CR0 (③ on Figure 43). It is an important parameter to set up when chaining analogue blocks together to form a function module, for instance when the system builds DACs and instrumentation amplifiers which will be shown later in the thesis.

  In addition to the main A input, both types of SC blocks have a secondary switched capacitor input path, the B input. The B input is needed when building some function modules like the DAC and instrumentation amplifiers. In the two different types of SC blocks, the B input has slightly different circuit structures, but in both types the B input path is connected to the inverting input of the opamp and overall its circuitry is similar to the A input path except without the reference inputs. The B input path is not detailed here but can be found in the SC block complete circuit block diagram in Appendix H.

- Configurable Capacitor Arrays

The configurable capacitors and switches (④&⑤ on Figure 43) are the key components to be managed during the reconfiguration process as their combinations and their settings determine the functionality of the block.

The capacitor array with selectable unit values is comprised of multiple capacitor elements in parallel which can constitute different capacitor sizes according to the setting of the switches connected to each of the capacitor elements inside the array, as illustrated in Figure 44.

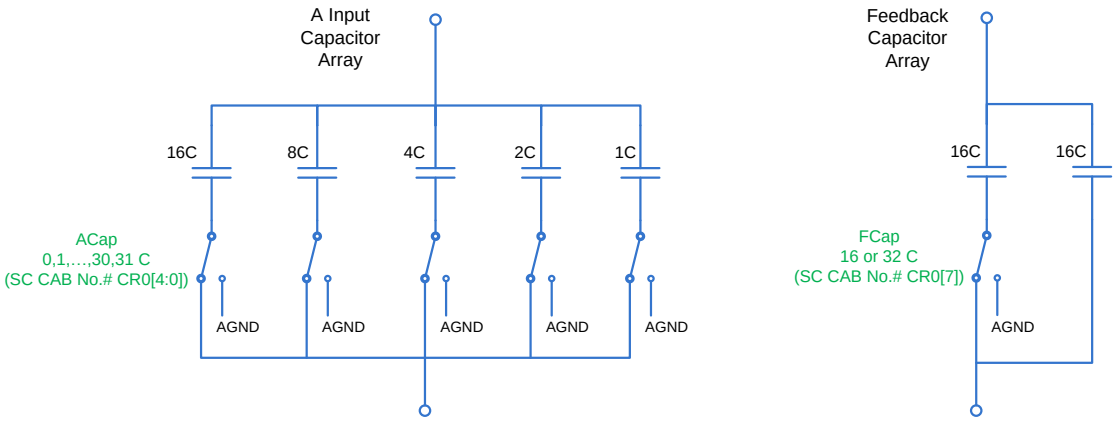


Figure 44 Configurable Capacitor Array Diagram

The switch in series with the capacitor element can swap the element in or out of the array so different elements can be selected for composing different sizes of capacitance. Therefore, a capacitor array is also considered as a variable capacitor in the circuit. The capacitor array in the A input path consists of five elements that support 32 possible capacitor sizes in the range of 0 to 31 capacitor units. The A input capacitor array (A capacitor) is controlled by the ACap bits of CR0. The feedback path capacitor array (F capacitor) supports two capacitor sizes and the FCap bit of CR0 selects the F capacitor value between the 16 or 32 capacitor units. The ratio of these two capacitors often decides the key parameter of the block. For example, when configuring an SC block into the output amplification stage (i.e. the conversion stage) of a three opamp instrumentation amplifier, the RAWS node controls the ratio of the ACap and FCap to

control the gain of SC output stage, i.e. the conversion gain of the instrumentation amplifier, in the range of 0.03125 to 1.9375 (ACap/FCap -> 1C/32C to 31C/16C).

- Configurable Switches

  The behaviour of the group of switches in the SC circuit can be modified by configuring the control register bits, which is how the reconfiguration techniques shape the SC circuit into different topologies for different signal processing functions. The switch behaviour control register bits include the ASign bit of CR0, AutoZero bit of CR2, FSW0 and FSW1 bits of CR3 as shown at ⑤ on Figure 43. The ASign bit, which is associated with the main A input as discussed before, determines whether the input is sampled on Φ_1 or Φ_2 which give positive or negative gain respectively. The AutoZero bit controls the auto zero functionality of the SC block which is useful for reducing the opamp offset and gain errors. When AutoZero is on, the opamp output is shorted to the inverting input to measure the offset during Φ_1 and is removed from the actual input signal during Φ_2. The FSW0 and FSW1 bits control the two feedback switches connected to the F capacitor for constructing different feedback signal paths. Also there is a ClockPhase bit in CR 0 that can be used to swap the internal clocks Φ_1 and Φ_2.

- Outputs

  The output of the SC block (⑥ on Figure 43) is the opamp output and can be routed to other analogue blocks as input. This is a useful option often used by the reconfiguration algorithm, e.g. when the RAWS system connects the SC output stage of a three opamp instrumentation amplifier to the next module in the chain like an ADC, or when the system builds a function module that needs two SC blocks like the DAC. Another important output option for SC block is the output to the comparator bus of the analogue column. The comparator bus is a digital bus shared by all the blocks in the column but it is more important for SC blocks because the comparator output can control the reference multiplexer of the main input of the block, the A reference multiplexer. This feature is required for constructing function modules such as the

modulators of delta-sigma and incremental ADCs. When the comparator output drives the A reference multiplexer, the reference input is set to RefHi when the comparator output is high, and it is set to RefLo when the comparator output is low. The SC block output can also be connected to the analogue bus of the column.

The adaptive reconfiguration techniques also configure the SC block into different analogue function modules by managing the SC block control registers especially the key register bits discussed above. Figure 45 shows how the RAWS node platform configuring an SC block into the first-order modulator of the delta-sigma ADC using the key register bits settings in Table 17. The modulator is essentially a 1-bit over-sampling integrator/comparator circuit, so the main circuit of the SC block is configured as an integrator with the comparator output enabled.

Register Bits	Value (Binary)	Description
AMux (CR1[7:5])	001	Main input is connected to CT block 0 (PGA in last example)
ARefMux (CR3[7:6])	11	Reference selection is driven by the comparator
ASign (CR0[5])	0	Input sampled on Φ_1 and reference sampled on Φ_2, together with the two switches after the A cap composing the switch sequencing of the input path of an integrator
ACap (CR0[4:0])	10000	A capacitor = 16 capacitor units
FCap (CR0[7])	1	F capacitor = 32 capacitor units
AutoZero (CR2[5])	0	AutoZero function is off
FSW1 (CR3[5])	1	FSW1 switch is always close and FWS0 switch is always open, forming the feedback path of an integrator
FSW0 (CR3[4])	0	
CompBus(CR2[6])	1	Enable comparator output for driving the A reference mux

Table 17 SC Block Key Register Bits Settings for First-Order Modulator of Delta-Sigma ADC

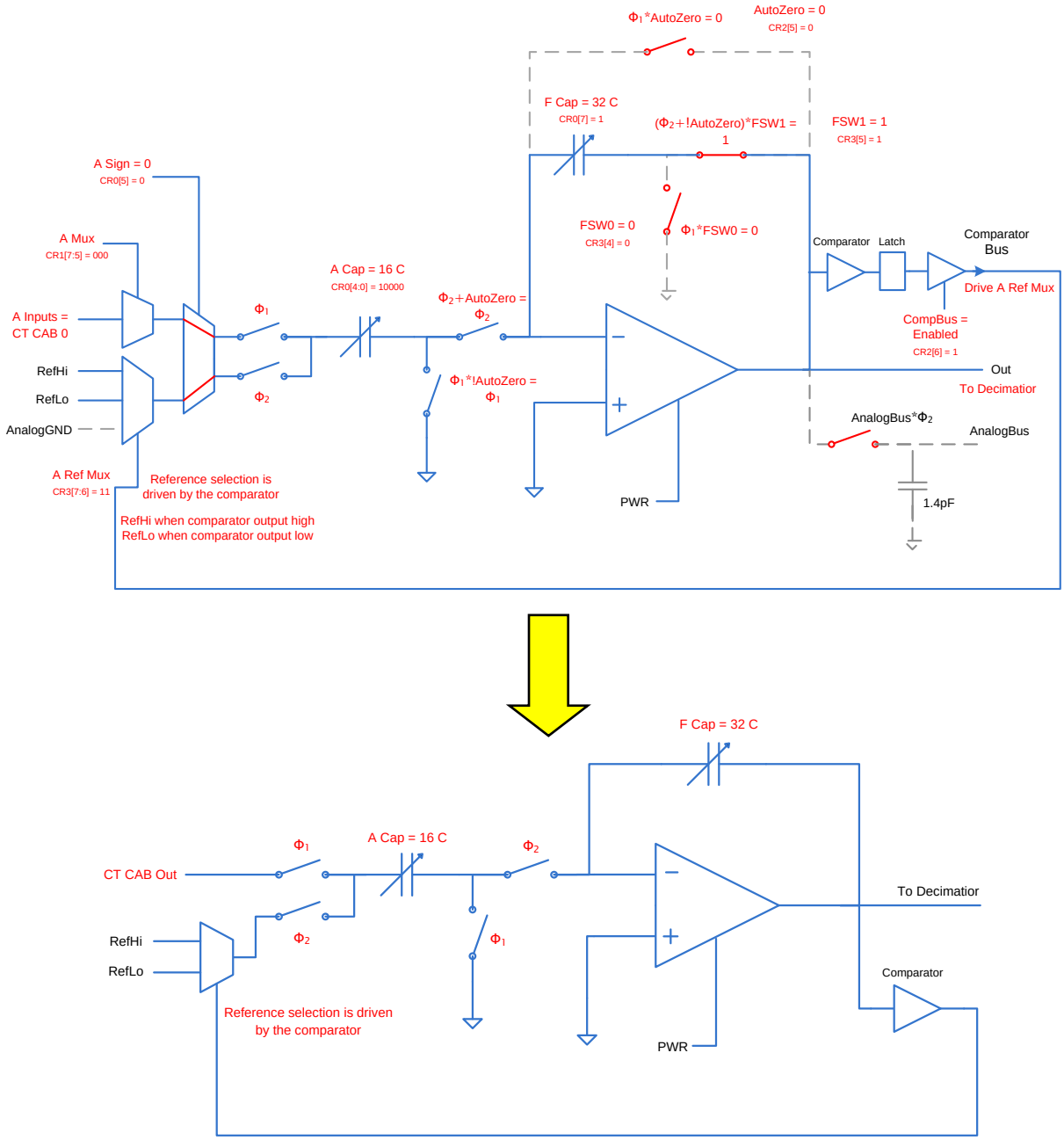


Figure 45 Configuring an SC Block into a first-order modulator of the delta-sigma ADC

5.2 Adaptive Reconfiguration Techniques

The previous section explained the principle and low level operations of the RAWS node hardware reconfigurations. This section illustrates the design details of the three adaptive reconfiguration techniques. Of the three adaptive techniques designed and developed in this research project, the intelligent algorithm is the central pillar of the RAWS node reconfiguration techniques and brings a number of novel features and advantages.

5.2.1 Intelligent Algorithm Technique

The intelligent algorithm adaptive reconfiguration technique enables the RAWS platform to autonomically reconfigure its hardware into suitable analogue function modules and select parameters for the modules. It is the key adaptive reconfiguration technique. It integrates the dynamical reconfigurability of the hardware platform and the sensor self-description capability to achieve an autonomic multi-sensing capability in the WSN area. The intelligent algorithm technique is able to establish a signal conditioning and processing chain based on the sensor characteristics acquired by the sensor identification scheme without any manual intervention. It can intelligently construct the analogue function modules and set up module parameters to build the signal conditioning and processing chain most suitable for the sensor in order to achieve optimised performance. It can generally support a large number of sensors. Also, the hardware configurations created by the intelligent algorithm for the identified sensors can be stored in the RAWS node configuration database and be utilised by the other two reconfiguration techniques. This subsection firstly explains the intelligent algorithm from a high-level. More technical details are then illustrated through examples with the four TEDS sensors used in RAWS node development.

The flow diagram in Figure 46 illustrates how the intelligent algorithm works to build a complete signal processing chain for the connected sensor.

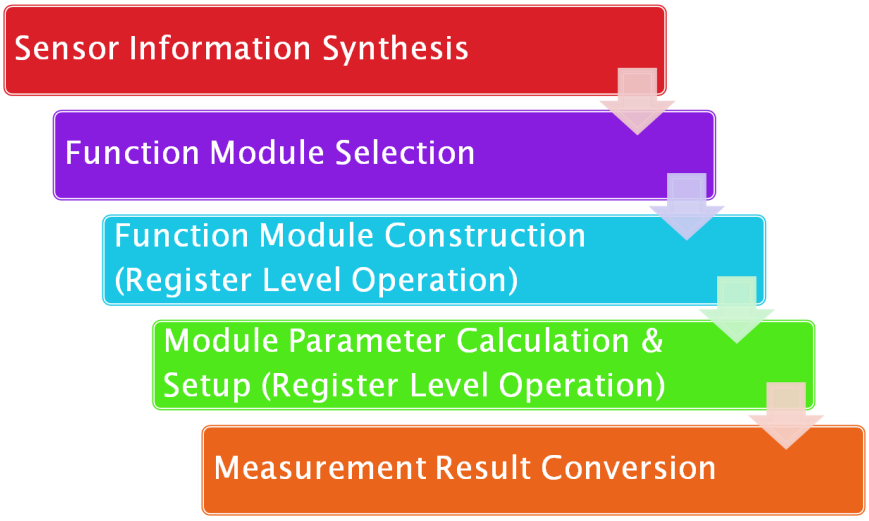


Figure 46 Intelligent Algorithm High-Level Flow Diagram

As shown in Figure 46, the algorithm starts the adaptive reconfiguration process by synthesising the sensor information as well as the available hardware resources information. It then based on this information decides the general structure of the signal processing chain and what function modules are required to build the signal chain for the connected sensor. Once the function modules are decided, the algorithm reconfigures the hardware platform to build the modules and establish the signal processing chain circuit. The algorithm then sets up the parameters of the function modules based on the sensor attributes as well. After the signal chain hardware is completed, the algorithm is also able to read the connected sensor and convert the raw data into meaningful readings. The whole process is performed autonomically.

To be more specific, the algorithm firstly parses the sensor information acquired by the sensor identification scheme and further extracts the important features of the connected sensor from the general attributes provided by the TEDS. For example, the algorithm can find out the sensor output signal features based on the basic attributes including sensor type, sensor configuration (such as resistive full bridge configuration), excitation requirements and so on. The intelligent algorithm also has the knowledge of hardware resources. This information including the characteristics of the different available function modules is integrated into the algorithm as a part of the node firmware during the design phase.

The intelligent algorithm then performs a decision-making process to autonomically determine the structure of the signal processing chain and select the suitable function modules for building the signal chain circuit. During this decision-making process, the algorithm essentially carries out a series of selections based on the extracted sensor features and the function modules features provided to the algorithm.

There are two main steps for the algorithm during the decision-making process. In the first step the algorithm decides the general structure and the stages of the signal processing chain. In this step, the sensor features such as the type of the connected sensor are the important factor for the algorithm. For example, for a bridge type sensor the algorithm decides that an excitation source, an amplification module, and a conversion module are the necessary main stages for building the signal chain, whilst for a thermocouple the algorithm does not include the excitation source module in the signal chain.

In the second step, the algorithm decides the suitable function modules for each stage of the signal chain. Both the sensor features and the function module features play important roles in this step for the algorithm to autonomically select function modules. For example, the algorithm can decide that an instrumentation amplifier with three-opamp topology is suitable to handle differential sensor output signal with common-mode voltage. In the algorithm design, different routines are implemented to perform the selections for modules at different stages of the signal chain. Overall, the principle in this decision-making step is to select the most suitable module for optimised system performance. In order to achieve this, the algorithm is designed to considered different factors as a whole to make the decision, such as the sensor type and output signal features and the characteristics of different types of function modules, and not just simply picking the modules with the highest specification on one certain aspect (e.g. the highest resolution). This is reflected in the detailed examples illustrated later in this subsection.

Once the components are decided, the algorithm performs the reconfigurations on the programmable platform to actually build the signal chain circuit. This reconfiguration process includes constructing the selected function modules in the hardware platform and setting up the parameters for the modules. This process involved a lot of register level operations and the

underlying principle of low level reconfiguration operations has been illustrated in the previous section. In the algorithm implementation, many low level APIs, similar to the driver level APIs on PC, are developed for building different function modules and setting up the module parameters.

From a high level perspective, the algorithm carries out a sequence of procedures to build the signal chain, including locating the programmable analogue blocks for the select function module, reconfiguring the block circuit into the proper structure and topology for realising the desired functionality, and connecting function modules together to realise the signal chain circuit.

The algorithm then calculates and sets up the module parameters based on the sensor features and hardware resources information. For the modules at different stages of the signal chain, the algorithm takes different sensor attributes and module specifications into consideration to determine the suitable parameters. For example, when selecting the amplifier gain, the algorithm carries out the calculation based on the factors including sensor output range, the gain setting range of the amplifier itself, and the input range of the next module in the chain; when deciding the excitation source output level, the algorithm does the calculation based on the sensor excitation attributes and the excitation module output range.

The more detailed flowchart of the intelligent algorithm's high level operations described so far for building the signal processing chain is presented in Figure 47.

The intelligent algorithm is also able to perform the sensor measurement result calculations in an autonomic way, completing the full signal chain. The algorithm starts the calculation process by employing the corresponding formula or equation for different kinds of sensors including linear sensors and non-linear sensors. The algorithm determines the equation based on the sensor attributes such as the sensor type and the sensor signal mapping method. For instances, for non-linear sensors such as a thermistor the algorithm uses the Steinhart-Hart equation for calculating results; it selects the Callendar-Van Dusen equation for an RTD; it uses a generic linear equation for a linear output sensor such as a linear bridge sensor.

The algorithm determines the coefficients of the equation according to the sensor attributes and the function module parameters set up during the signal chain reconfiguration process. For example, the algorithm can determine Steinhart-Hart coefficients and Callendar-Van Dusen coefficients based on the sensor TEDS data; it can based on the thermocouple type retrieve the corresponding thermocouple polynomial coefficients which are stored as a part of the RAWS node firmware; for the calculations of linear sensor readings, it often needs the sensor physical measurement range and output signal range as coefficients; Also, function module parameters, such as the amplifier gain and converter input range and resolution, are often used in data conversion calculations for many different types of sensor. Once the equation coefficients are determined, the algorithm calculates and converts sensor outputs into meaningful readings. It is also worth mentioning that when digital signal processing is needed the RAWS node's microcontroller could handle the processing operations in software methods. However, as discussed in Chapter 2 this approach can be slow and inefficient especially when handling computation-intensive tasks, so in this case a DSP or an FPGA as the co-processor of the node platform for heavy computation operations would be a better approach.

The flowchart of the intelligent algorithm's sensor reading calculation sequences is presented in Figure 48 in the next page.

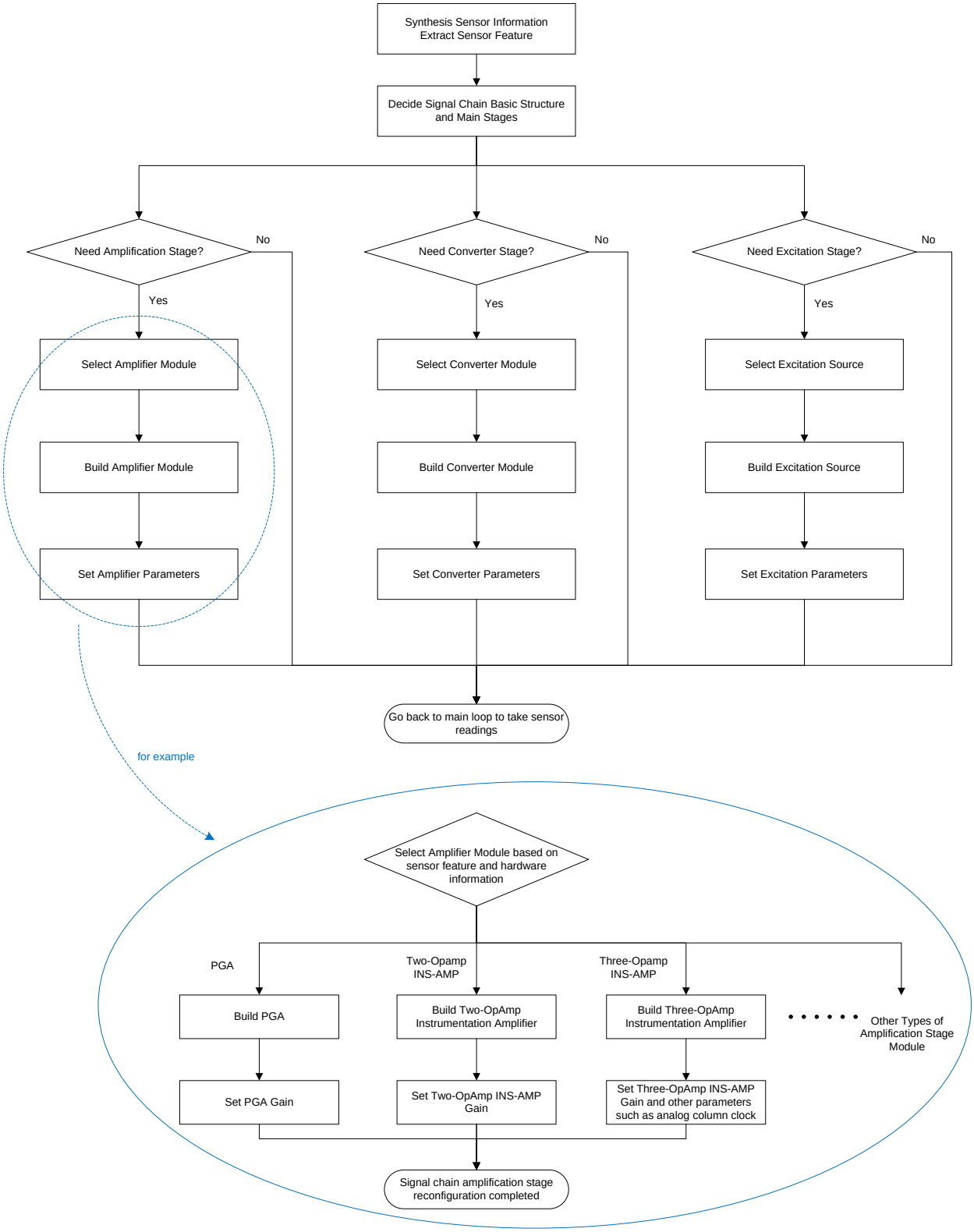


Figure 47 Intelligent algorithm operations for building signal chain circuits

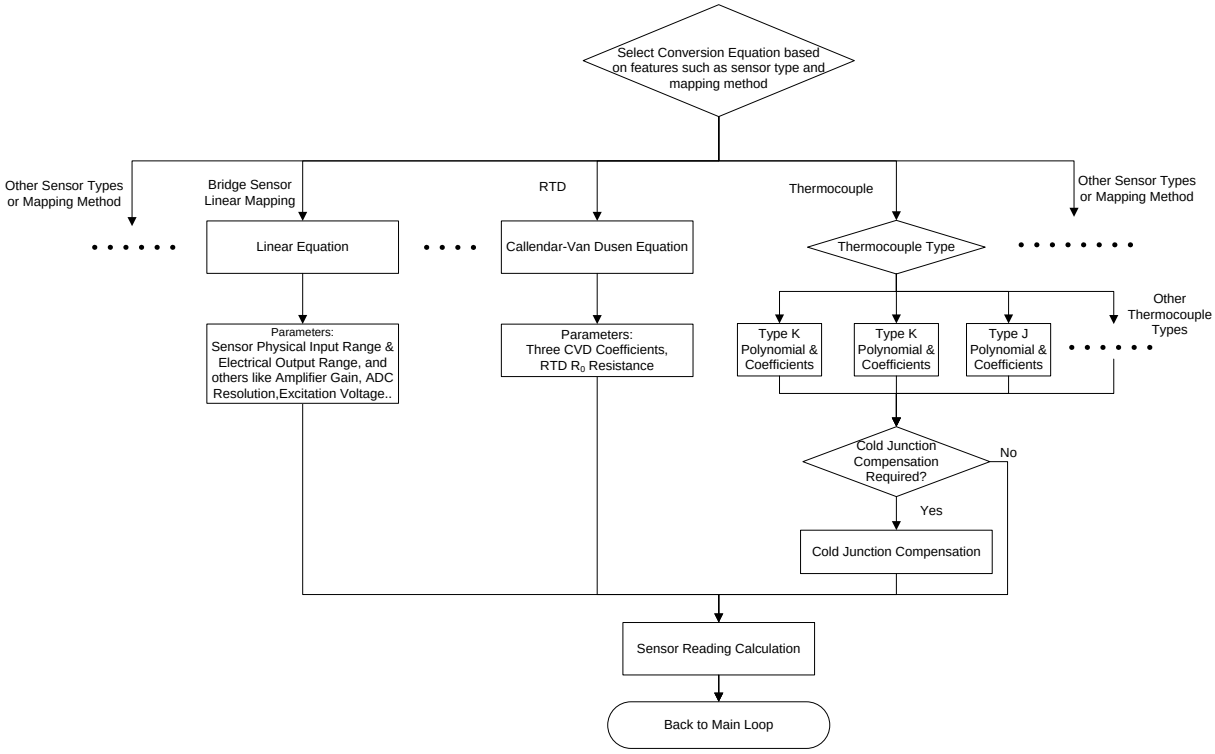


Figure 48 Intelligent algorithm operations for sensor reading calculations

The rest of the subsection illustrates how the intelligent algorithm reconfigures hardware to support analogue sensors in more detail with TEDS sensor examples. The four sensors used for the proof-of-concept to assist the design and development of the RAWS node technology, including C9B force sensor (load cell), P8AP pressure sensor, 101-K thermocouple and 101-10B RTD, are also used in this section for illustrating the adaptive reconfiguration techniques.

In the case where the C9B force sensor is connected to the RAWS node platform, the intelligent algorithm obtains the following key characteristics information provided by the sensor identification scheme:

1) Full bridge type sensor
2) Nominal excitation level 2.5V
3) 1mV/V sensitivity
4) Measurand: force, Measurement range: 0 to 50N

5) Mapping method from the physical value to the electrical output: linear mapping

Based on the sensor type and the physical measurand, the algorithm decides that the following analogue function modules are needed for this sensor and it builds a signal conditioning and processing chain as shown in Figure 49.

- A DAC module for supplying excitation voltage
- An instrumentation amplifier for conditioning the sensor differential output
- An ADC to convert the sensor output

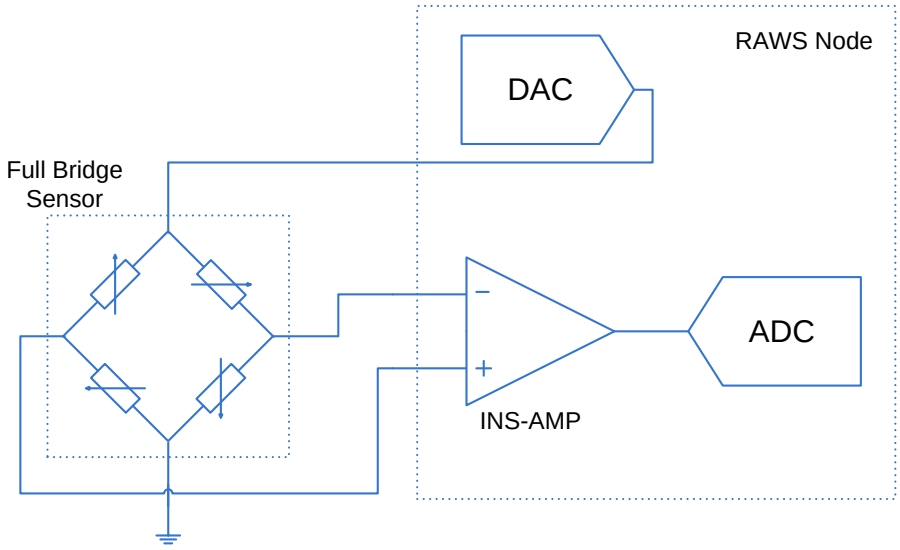


Figure 49 Signal conditioning and processing chain diagram for bridge sensor

The configuration process and parameters setup for supporting the sensor are detailed as follows:

- DAC module

  The DAC module is needed to supply excitation voltage for the connected full bridge sensor. The PSoC supports up to a 9-bit DAC. For excitation voltage supply, the voltage output of the DAC is not varied so the DAC update rate is not an important factor and the intelligent algorithm is designed to select the highest resolution DAC available, which is the 9-bit DAC in this case, to achieve the highest precision output voltage. The algorithm constructs the 9-bit DAC by configuring two SC blocks into the circuitry as

shown in Figure 50. Essentially, the two SC blocks are configured as two SC amplifiers with gains less than 1 (i.e. the feedback capacitor is fixed to 32C so the gain is always less than 1 because the input capacitor can be set to 31C max). They are chained together to scale the reference voltage down to the desired voltage level according to the ratio of the input capacitors to the feedback capacitors settings. One of the SC amplifiers (which are in fact attenuators) works as the LSB stage of the attenuation chain, and its output is coupled into the MSB stage via the secondary B input of the MSB block. This LSB output is further scaled down in the MSB stage and is combined into the MSB output to provide the final voltage output of the DAC. The output of the DAC is fed to the sensor so the analogue bus connection of the MSB block has to be enabled for routing the DAC output to the GPIO pin to provide the excitation voltage.

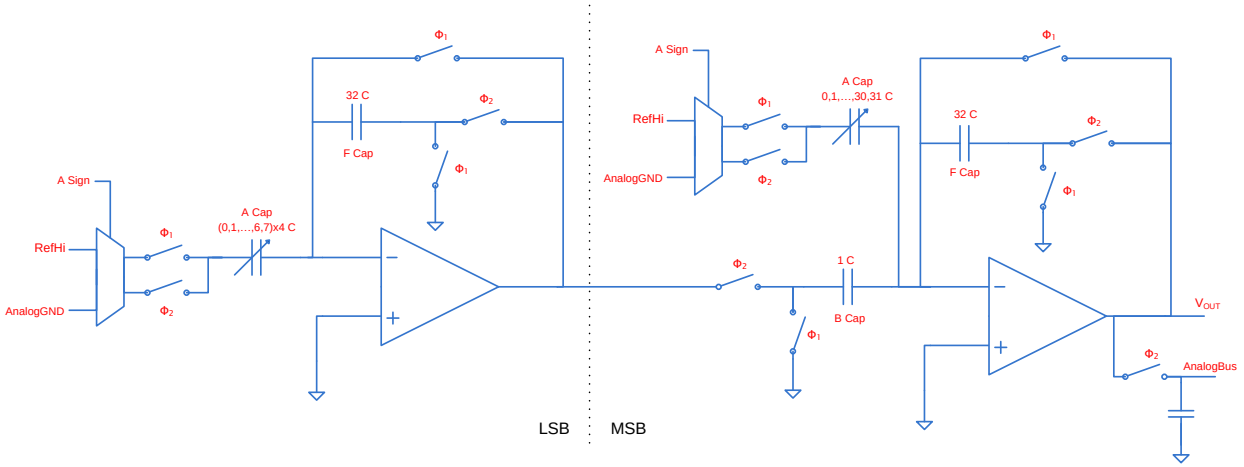


Figure 50 the 9-bit DAC configuration with two SC blocks

The intelligent algorithm then sets up the parameters after the configuration of the function modules is completed. Firstly, based on the nominal excitation level of the sensor, the algorithm selects a suitable voltage reference for the DAC in order to reduce the output range of the DAC to the smallest possible range which is still able to cover the nominal excitation voltage level. The purpose of reducing the DAC output range is to further increase the output precision. The algorithm changes the voltage reference by adjusting the source of the reference generator.

The algorithm then calculates and sets up the digital value that represents the nominal excitation level based on the DAC resolution and the output range.

$$DAC_{Value} = (2^n - 1) \times \frac{V_{Nom}}{V_{OutFullRange}} \quad \text{Equation 3}$$

Where:

n is the DAC resolution. n is equal to 9 in this case.

V_{Nom} is the nominal excitation level of the sensor.

$V_{OutFullRange}$ is the DAC output range and it is decided by the setting of the reference generator source.

- Amplifier

  The output of the full bridge sensor is a differential voltage signal with a common-mode voltage. Due to this feature, the algorithm constructs a three-opamp instrumentation amplifier (INS-AMP) to cope with the sensor output signals because the instrumentation amplifier has a differential input and also the three-opamp INS-AMP features wide common mode input range and high common mode rejection. The INS-AMP is configured with two CT blocks forming an input stage (i.e. differential stage) amplifier and an SC block as the output stage amplifier (i.e. conversion stage), as shown in Figure 51. The input stage amplifier has both a differential input and a differential output. The two CT blocks have the same resistor string configuration setting thus identical gain, and the resistor string bottom ends of the two blocks are connected together. The output stage SC amplifier converts the differential output of the input stage to a singled ended voltage. The conversion stage also supports various gain settings. So the combination of the input stage and the output stage generates a large number of useful gain settings for creating the signal chain for sensors.

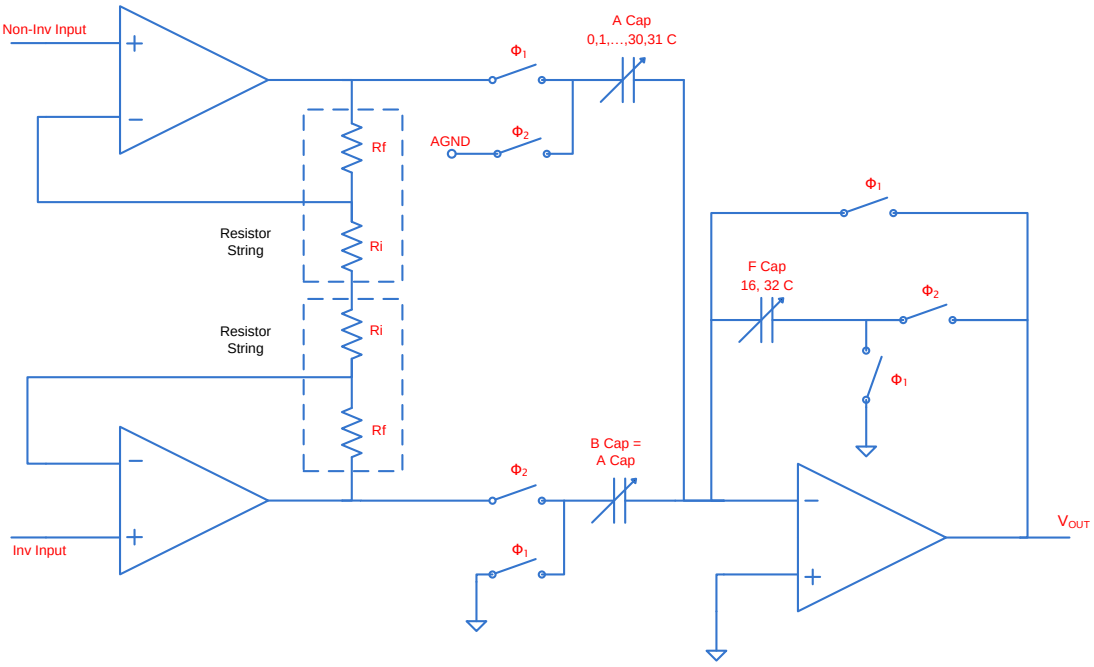


Figure 51 Three-opamp INS-AMP configuration using two CT block and one SC block

The algorithm then calculates and sets the gain setting of the instrumentation amplifier based on various factors including the sensitivity attribute of the sensor, the nominal excitation level, and the ADC input range which is decided by the reference voltage setting. The three opamp topology INS-AMP of PSoC supports programmable gains up to 93 (48 CT stage gain X 1.9375 SC stage gain) based on the discussion in the previous section 5.1. The algorithm selects the highest possible gain setting to amplify the sensor output within the ADC input range. The following equations express the abovementioned guideline for selecting the amplifier gain.

$$\begin{cases} Gain \times (C_{Nom} \times V_{Nom}) < V_{InFullRange} \\ Gain \leq MaxGain \end{cases}$$ Equation 4

Where:

C_{Nom} is the nominal sensitivity of the sensor in mV/V.

V_{Nom} is the nominal excitation level of the sensor. $(C_{Nom} \times V_{Nom})$ is the maximum sensor output in mV.

$V_{InFullRange}$ is the ADC input range.

- ADC

  The ADC converts the analogue signal into digital form and completes the signal conditioning circuitry. For the full bridge sensor, the algorithm selects the delta-sigma ADC due to its high common mode rejection. Also for force measurements, both the speed and the resolution of the conversion need to be taken into consideration. Of the three types of ADCs that PSoC supports, the delta-sigma ADC can provide high resolution with relatively high sample rate, 15.6ksps at 12-bit and 7.8ksps at 14-bit. In comparison, the incremental ADC is slower than the delta-sigma ADC at high resolution, 480sps at 12-bit and 121sps at 14-bit. The SAR ADC of the selected PSoC microcontroller only offers up to 6-bit. Also the delta-sigma and the incremental ADC feature an oversampling front-end SC circuit which significantly alleviates the anti-aliasing requirements. Based on the sensor characteristics and the ADC features, the algorithm builds a 12-bit delta-sigma ADC to provide a measurement accuracy of less than 0.5 Newton to meet the accuracy class of the connected sensor.

  The 12-bit delta sigma ADC consists of a second-order modulator producing a single bit data stream that represents the input voltage, a decimator which converts the bit stream into samples, and a timing generator which is basically a PWM module for generating clocks for the modulator and the decimator.

  The algorithm configures two SC blocks to assemble the second-order modulator and connects the modulator output to the decimator via the comparator bus. The decimator is a system resource of PSoC and can integrate the single bit data stream from the modulator and process the final output by itself so it can greatly reduce the CPU overhead requirement for conversion functions such as delta-sigma ADC and incremental ADC. As the decimator is available as a dedicated system hardware resource, the algorithm does not need to build it using analogue blocks but needs to set decimator parameters, and the key parameter is the clock frequency which decides the decimation rate. The decimation rate and the modulator type (i.e. first order or second

order) decide the delta-sigma ADC resolution. The decimator output is also the ADC raw data count output.

The main part of the hardware configuration for the algorithm here is to form the second-order modulator by configuring the two SC blocks into the circuitry shown in Figure 52. As previously explained in subsection 5.1.3, the SC blocks are essentially configured as integrators with the comparator output enabled in order to control the input reference for performing the 1-bit oversampling.

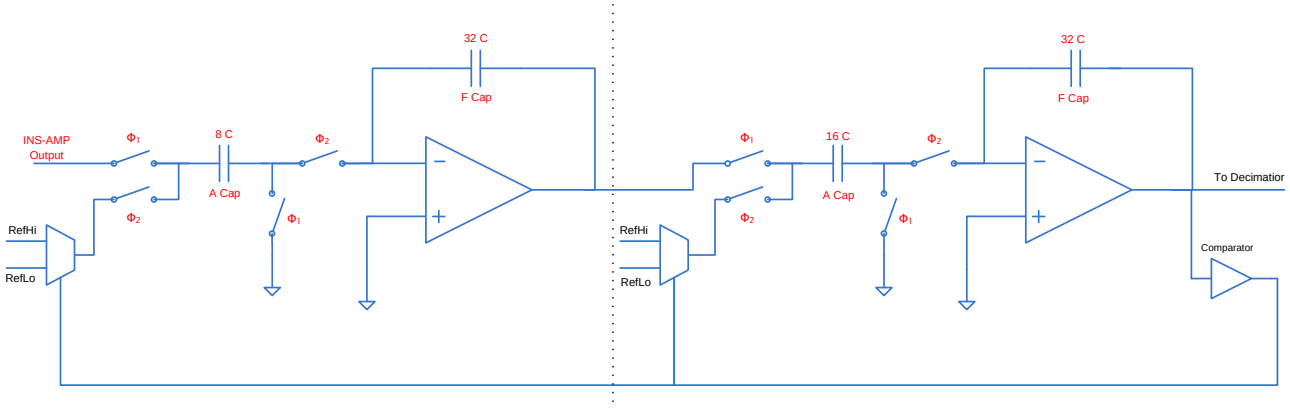


Figure 52 Second-order modulator of delta-sigma ADC using two SC blocks

- Sensor reading calculations

  Once the signal conditioning and processing chain is established, the algorithm autonomically converts the ADC raw data count and calculates meaningful physical measurement results based on the sensor characteristics and analogue function modules parameters set up above including the physical measurand and measurement range, mapping method, sensitivity, DAC output, and amplifier gain. For linear mapping sensors, the physical value calculation is expressed by Equation 5. The sensor readings then can be transmitted to the network gateway and stored in the data sink.

$$\begin{cases} \dfrac{PhysicalValue}{Phys_{FullRange}} = \dfrac{SensorOutput}{(C_{Nom} \times V_{Nom})} \\ V_{ADCIn} = Gain \times SensorOutput \\ \dfrac{RawCount}{2^n - 1} = \dfrac{V_{ADCIn}}{V_{InFullRange}} \end{cases}$$

$$\downarrow$$

$$PhysicalValue = \frac{RawCount_{ADC}}{2^n - 1} \times \frac{V_{InFullRange}}{Gain} \times \frac{Phys_{FullRange}}{(C_{Nom} \times V_{Nom})} \quad \text{Equation 5}$$

The adaptive reconfiguration process for accommodating the C9B full bridge force sensor is summarised in Figure 53 (the signal processing chain diagram was shown in Figure 49). In the reconfiguration process, the intelligent algorithm loads the analogue function modules one by one to build up the hardware configuration.

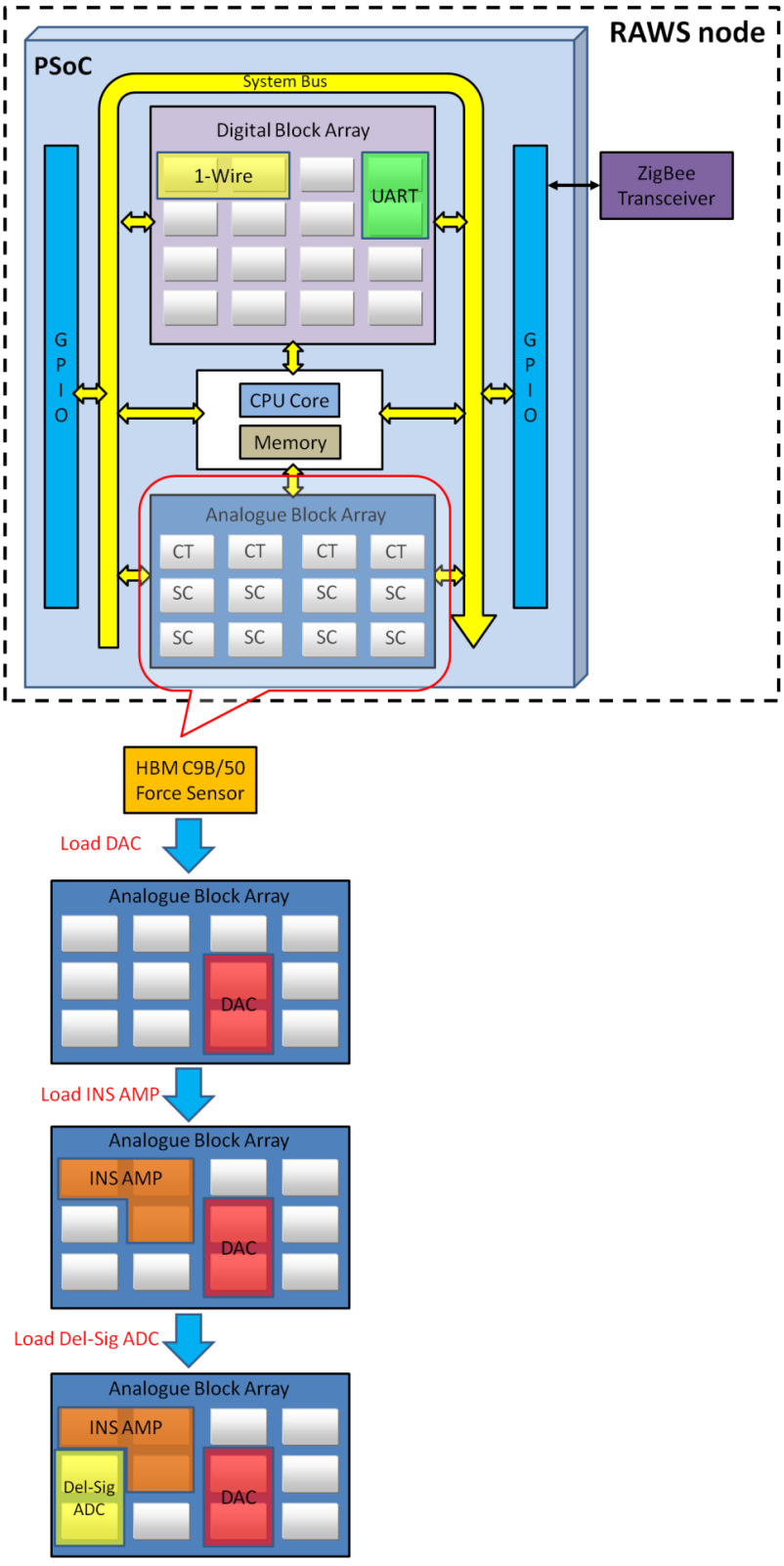


Figure 53 Reconfiguration process of the intelligent algorithm technique for C9B full bridge force sensor

If replacing the C9B force sensor with the 101-K thermocouple, the RAWS node detects that a different sensor is connected according to the information acquired by the sensor identification scheme, and the algorithm obtains the key characteristics information of the thermocouple:

- K type thermocouple
- Output voltage range: -6mV to 51mV
- Measurand: temperature, Measurement range: -200 to 1260 °C

Based on the sensor properties, the algorithm determines that an instrumentation amplifier and a delta-sigma ADC are needed to support the thermocouple and establish the signal chain shown in Figure 54.

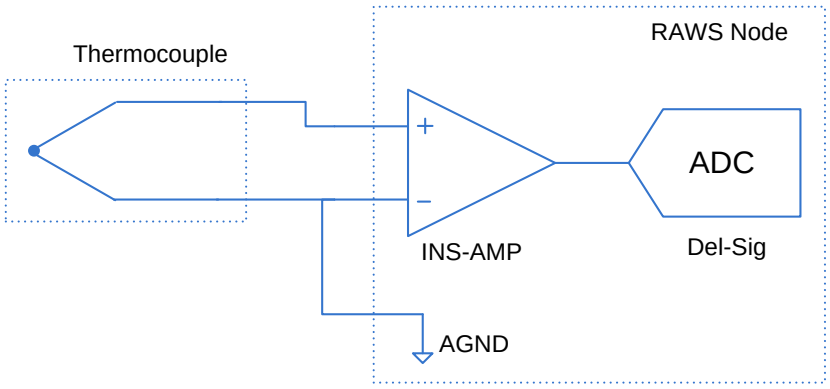


Figure 54 Signal conditioning and processing chain diagram for thermocouple

The analogue ground voltage is outputted to provide a reference level for the thermocouple differential signal and also act as an offset so that the single supply RAWS node system can handle the negative part of the thermocouple output. The thermocouple output voltage is at the microvolt level, so a high gain differential amplifier and a high resolution ADC are required. The INS-AMP and delta sigma ADC meet this requirement. Because these two modules are already loaded previously for the force sensor, the algorithm unloads the DAC and the hardware configuration for the thermocouple is then complete, as shown in Figure 55. The algorithm then reselects the parameters of the amplifier and the ADC to adapt to the thermocouple. The algorithm calculates and sets the highest possible amplifier gain based on the maximum output voltage of the sensor and the input range, and it reconfigures the resolution of the ADC to 14-bit because the output of the thermocouple is only tens of millivolts

per degree Celsius so higher resolution is required whereas the speed of the ADC is not the key consideration as temperature will not normally change at a fast rate. After the hardware reconfiguration is completed, the algorithm calls the thermoelectric coefficients corresponding to the thermocouple type to calculate the temperature readings.

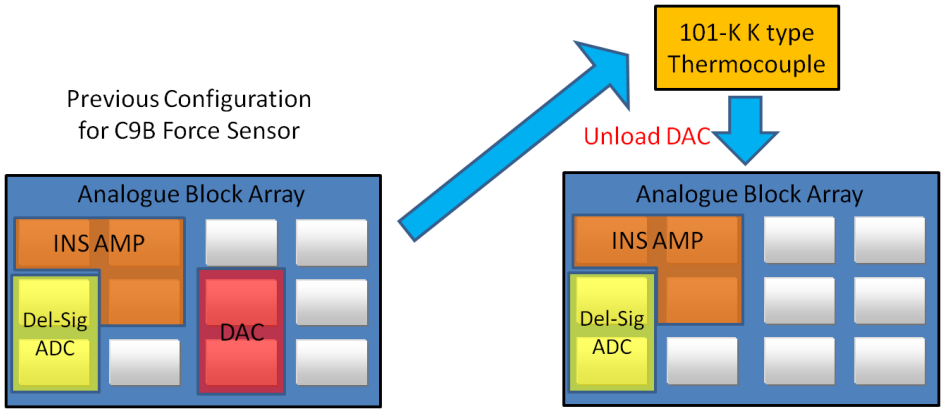


Figure 55 Reconfiguration process of the intelligent algorithm technique for 101-K thermocouple

When replacing the 101-K thermocouple with the 101-10B Pt1000 RTD, the algorithm detects that a resistive temperature sensor is connected, and it reconfigures the hardware to adapt to the sensor. The resistance of the RTD is measured against an onboard reference resistor as shown in Figure 56.

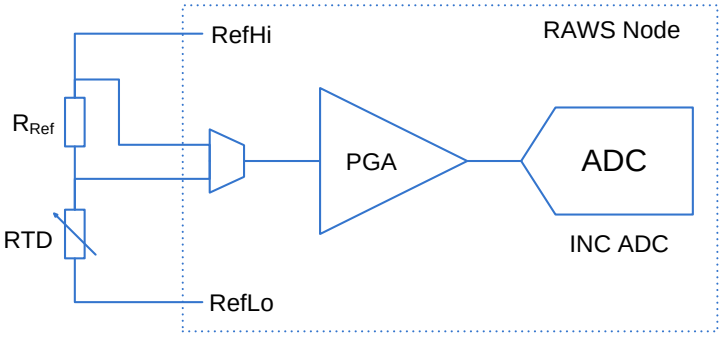


Figure 56 Signal conditioning and processing chain for RTD

The algorithm decides to employ a PGA and an incremental ADC to establish the signal chain. The incremental ADC is the more suitable option in this configuration, because two different voltages are to be measured through the analogue input multiplexer of the PSoC and the incremental ADC can adapt to the multiplexing inputs without delay. With regard to the

amplifier, the algorithm selects the single-ended input PGA because there is no differential signal with common mode voltage to be measured. In the reconfiguration process, the algorithm unloads the INS-AMP and the delta-sigma ADC first and then loads the PGA and incremental ADC one by one to build the configuration for the RTD, as shown in Figure 57. After setting up the signal chain, the algorithm calculates the temperature readings using the RTD temperature coefficients (Callendar-Van Dusen coefficients) and the RTD nominal resistance value acquired by the sensor identification scheme.

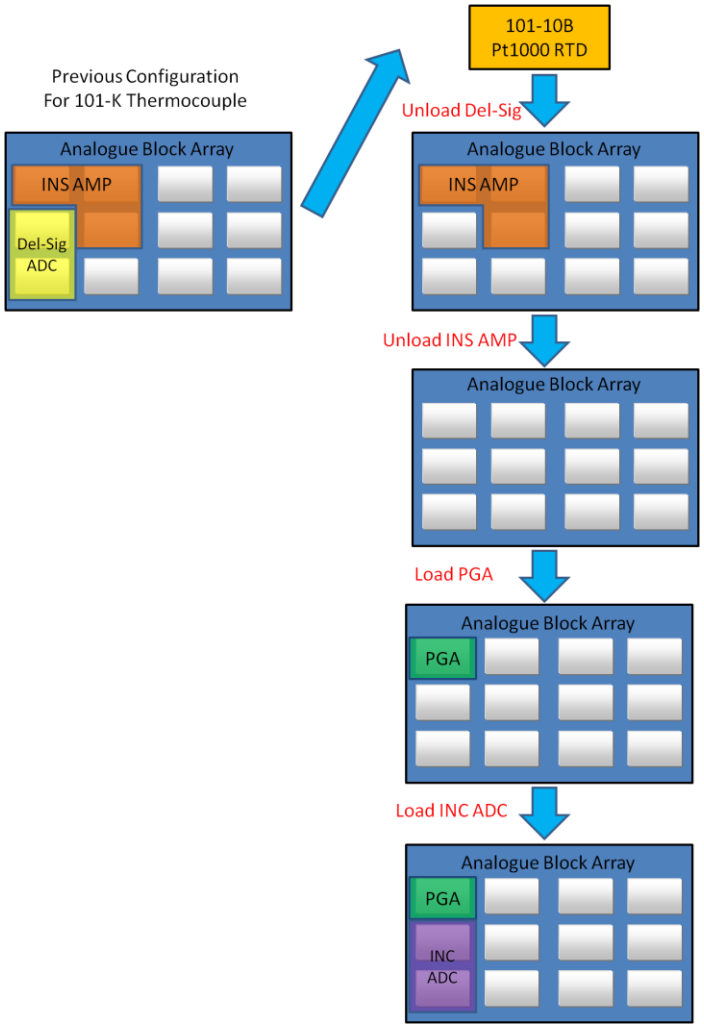


Figure 57 Reconfiguration process of intelligent algorithm technique for 101-10B RTD

The whole adaptive reconfiguration process of the intelligent algorithm technique to accommodate the three different sensors is shown in Figure 58. The P8AP pressure sensor is

also a full bridge type sensor and the RAWS node reconfiguration process for it is similar to the C9B force sensor so is not repeated here.

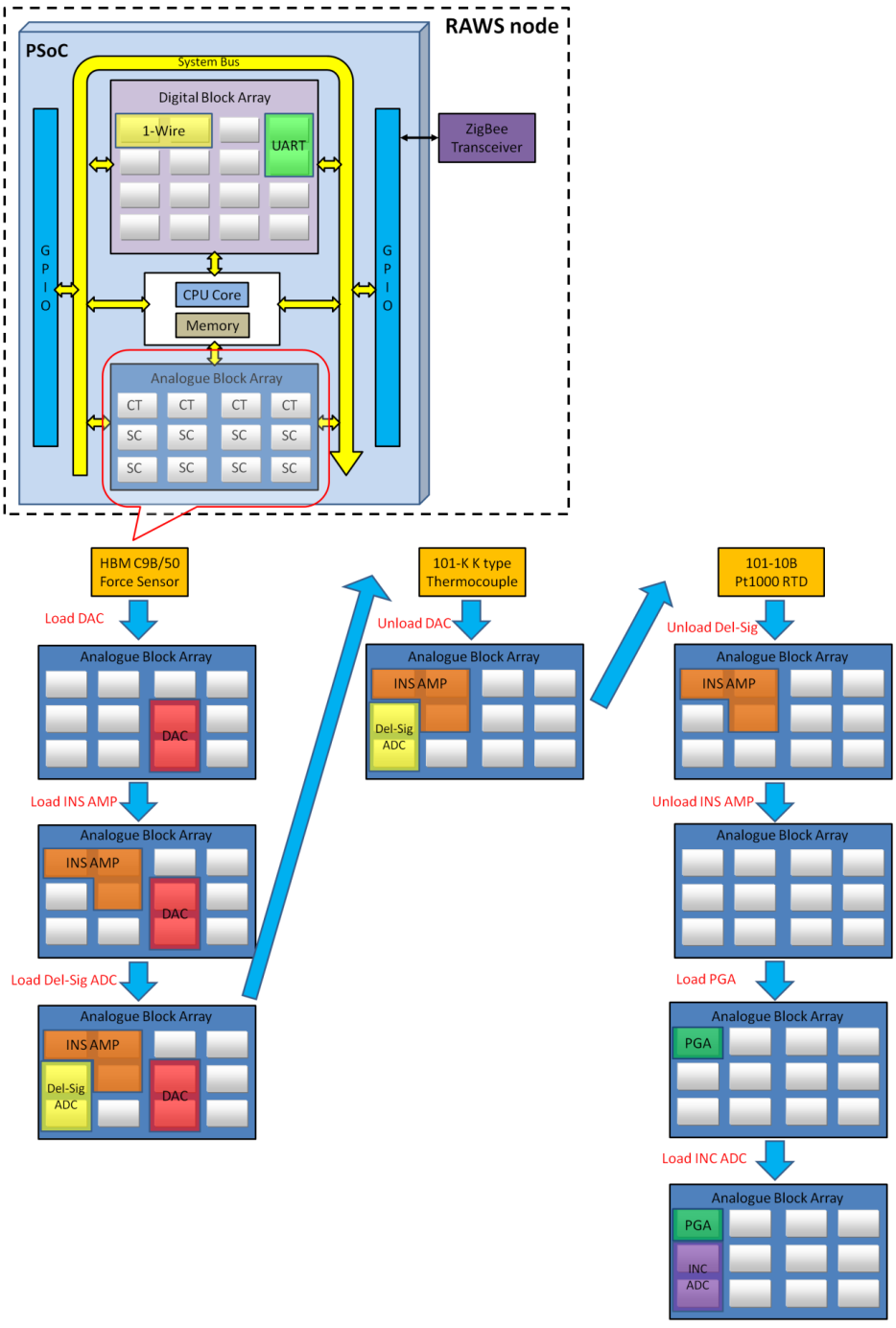


Figure 58 Overall reconfiguration process of intelligent algorithm technique

5.2.2 Local and Remote Configuration Settings Techniques

Both the local and remote configuration settings techniques utilise the sensor ID information obtained by sensor identification scheme to search for and load the hardware configuration settings corresponding to the connected sensor. The RAWS node searches its local configuration settings database first to avoid high power consumption wireless transmission, but the local configuration settings database is limited by the small embedded memory space. If no configuration is found locally, the node sends a remote configuration request to the network gateway to search the remote configuration settings database.

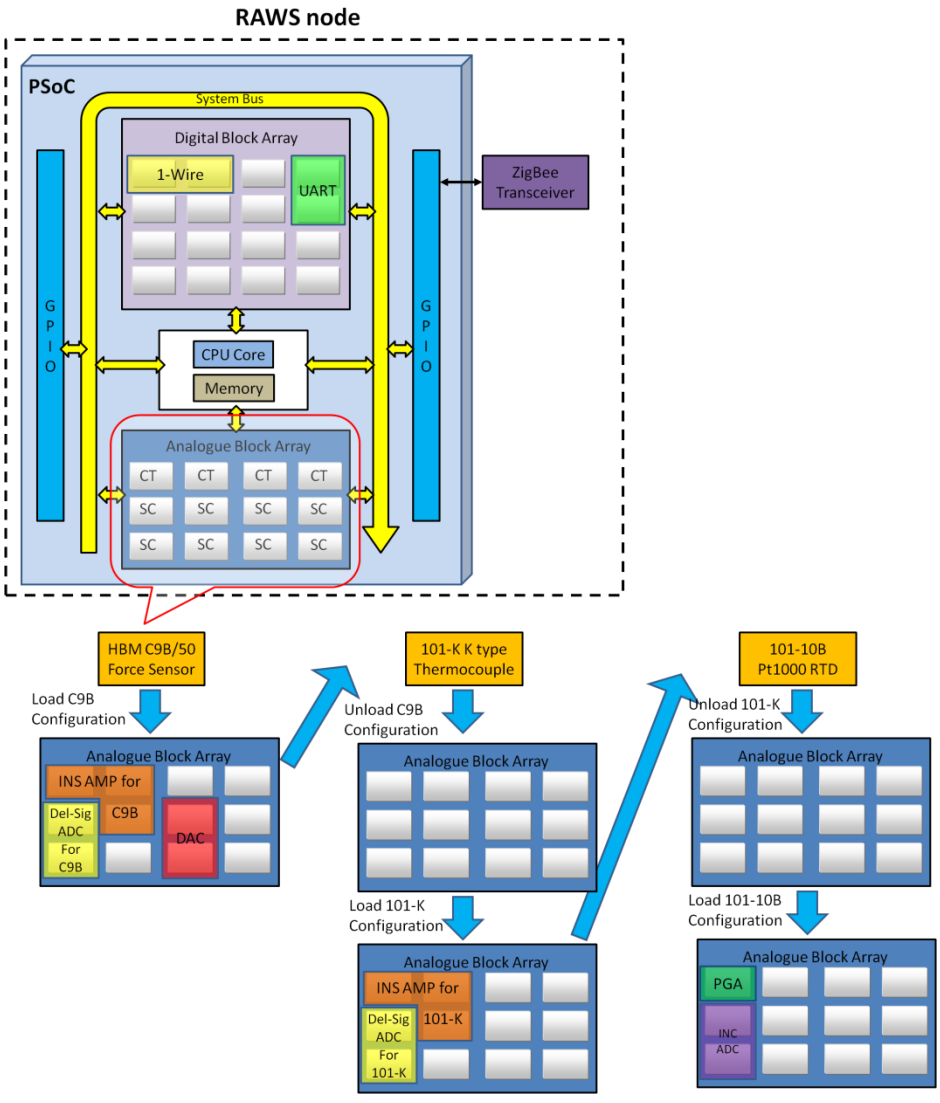


Figure 59 Overall reconfiguration process of local and remote configuration settings technique

The configuration settings pre-designed for a sensor contain the complete signal chain setups including the analogue function modules for building the signal chain as well as the preset parameters for each of the analogue modules, so the RAWS node only needs to load and unload configurations to adapt to different sensors. The configuration settings generated by the intelligent algorithm are also stored in the RAWS system and they can be used by the local and remote configuration setting techniques. The reconfiguration process of the local and remote configuration setting techniques for accommodating the three sensors discussed in the intelligent algorithm technique subsection is shown Figure 59. The reconfiguration process of the local and remote configuration setting techniques is simpler, but the intelligent algorithm technique is more flexible.

5.3 Adaptive Reconfiguration Techniques Summary

The adaptive reconfiguration is the key technique that combines the hardware reconfigurability and the sensor identification scheme together so that the RAWS system can flexibly utilise its hardware resources to support different sensors in an autonomic way.

The intelligent reconfiguration algorithm is the key adaptive technique developed in this research. It can synthesise sensor attributes and based on the important sensor features and hardware resources information it can autonomically and dynamically build a signal conditioning and processing chain to adapt to the connected sensor. To be more specific, it is able to consider different factors and make decisions to select and construct analogue function modules and set up parameters most suitable for the connected sensor. It is designed to build signal chains for sensors with the aim of achieving optimised system performance. It can also perform sensor data conversion calculations. This whole hardware reconfiguration and sensor signal conversion process are without any manual intervention. The intelligent algorithm equips the RAWS platform with the adaptive hardware reconfiguration capability in autonomic operations to support sensors, and as discussed before, it is the key novelty of the RAWS technology and the main difference which makes RAWS technology more flexible and adaptable compared to other similar research work.

Chapter 6 System Functionality Testing and Analysis

6.1 Testing Approach

The main aim of the research was to develop a new wireless sensor node technology that is able to perform hardware reconfiguration to adapt to different analogue sensors in an autonomic way. Therefore, the system testing mainly focused on validating the functionalities of the RAWS system to prove that the research concepts have been achieved.

The system development includes three main parts as described in Chapter 3, wireless communications, the sensor identification scheme, and adaptive reconfiguration techniques. So the system tests also included three main parts that concentrate on the three design parts respectively.

Wireless communication is a basic but necessary function of the system. The tests verify that the RAWS network can successfully establish RF transmissions between the RAWS nodes and the gateway to support high level system functions including sensor data storage and remote hardware configuration settings transfer.

The sensor identification scheme test verifies that the RAWS node platform is able to read and parse the TEDS data of the connected sensor in order to extract identification and characteristic information. This part of the test focuses on the TEDS parsing routine and validates that it can obtain key information for the platform to perform hardware reconfiguration.

The adaptive reconfiguration is the most important part of the design and the intelligent algorithm is the key reconfiguration technique. The test monitors the programmable analogue system to verify that the algorithm is able to properly perform reconfigurations according to the design logic and to build suitable signal conditioning and processing chains to support different sensors. The theoretical process of the hardware reconfiguration is analysed to help to verify the test results. The functionality of the complete signal chain built by the intelligent algorithm is also verified by the system measurement performance test. The measurement performance is an important and inseparable aspect of the system design. The measurement test can verify whether the system can meet the sensor accuracy requirement and achieve

optimised performance. Also the measurement performance test is a straightforward way to validate both the hardware and software functionalities of the signal chain.

The main purpose of the system tests is to validate the system functionalities and since these tests are mostly software tests, the debugging tools of the design environments for the system are the natural tools to monitor the system and collect test results. For the RAWS nodes, the core controller is the PSoC microcontroller. The PSoC hardware debugging system including an in-circuit emulator was set up for the tests. On the gateway side, the main testing component is the RAWS terminal software which is designed with Microsoft Visual C++. The Visual Studio debugging environment was utilised for the gateway tests.

Besides the system functionalities tests above, the power consumption test is also a part of the RAWS system testing procedure to verify whether the RAWS node can meet the low power requirement of WSNs. In the power consumption test and the system measurement performance test, necessary instruments such as high resolution source unit and digital multimeter are also included in the test setup.

The diagram in Figure 60 illustrates the general test setup. The detailed setups for different tests are described in each subsection.

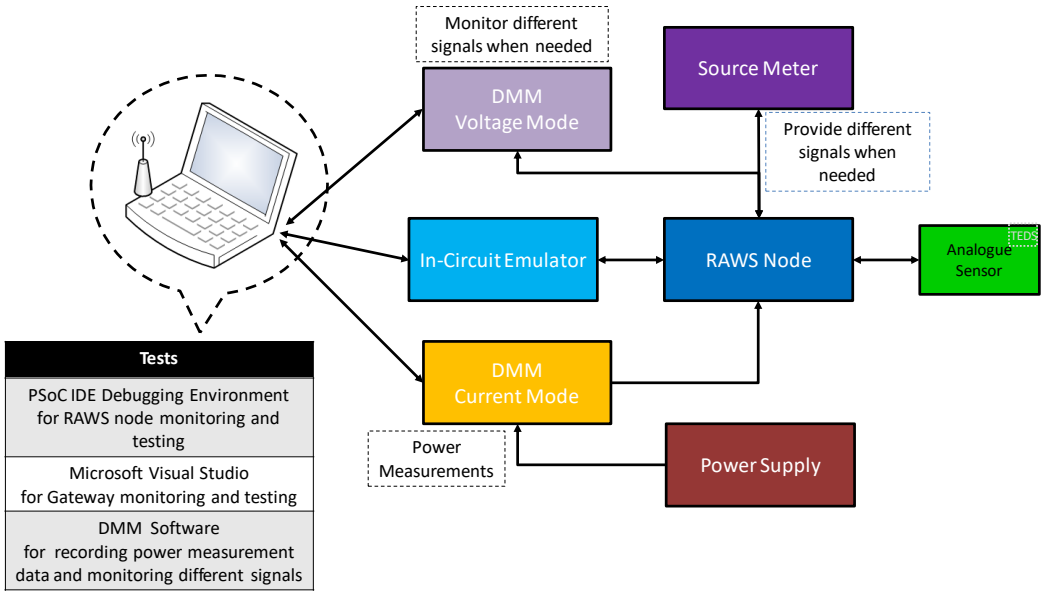


Figure 60 General diagram of system test setup

6.2 Wireless Communication Functionality Test

In the RAWS network, there are two types of network devices and they are the RAWS node and the network gateway. The following three types of communications within and between the two types of devices are the elements that enable the wireless transmissions of the network. Therefore, they were tested for validating the wireless functionality.

1. Serial communications between the RAWS node firmware (or the node controller, i.e. PSoC) and ZigBee end device transceiver module
2. Serial communications between RAWS node terminal application (or the network gateway controller, i.e. PC or laptop) and the ZigBee coordinator transceiver module
3. RF data packet communications between RAWS nodes and the network gateway

The serial communications between the controllers of the network devices and the ZigBee transceivers, i.e. the first two types described above, are the prerequisite for the actual wireless communications to be carried out between RAWS nodes and the gateway, i.e. the third type above.

In the tests, the communications are monitored to verify that all data packets are correctly constructed and sent or are successfully received and processed, and the aforementioned debugging tools are used to capture packets.

6.2.1.1 Serial Communications between Controllers and Transceiver modules

As discussed in Chapter 3, both the RAWS node terminal application and the firmware need to perform a series of parameter setups for the ZigBee transceiver module before the wireless network can be established. The terminal application and the node firmware complete the setup process by sending a series of AT Command API frames. In the tests, these AT command frames between the controller and the transceiver are captured and verified.

The data packets shown below are the captured AT Command frames for configuring the network ID ‘64-bit PAN ID’ and the channels to be scanned ‘ScanChannels’. Among the key network parameters, these are the two most important parameters for both the gateway and the end devices. They are also the first two parameters to be set up as the gateway and the

nodes need to have the same network ID and operate on the same channel to establish network connections. Therefore, the packets flow for them are analysed below to demonstrate the test result. AT Command frames sent from the terminal application and the PSoC to the coordinator and the end device ZigBee transceiver modules respectively are identical and are illustrated in Figure 61 and Figure 62.

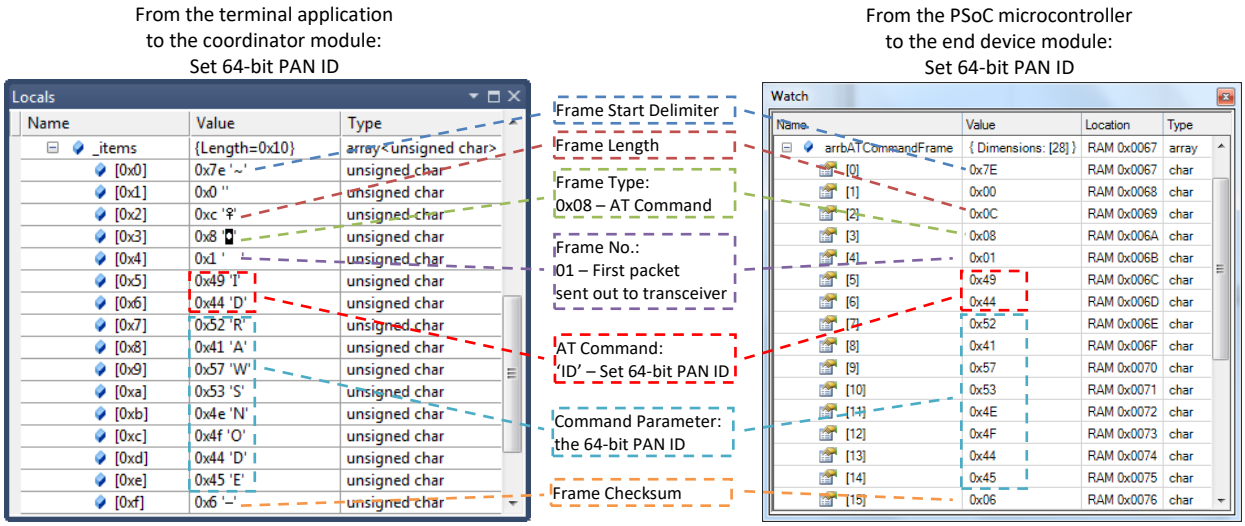


Figure 61 'Set 64-bit PAN ID' AT Command Frames

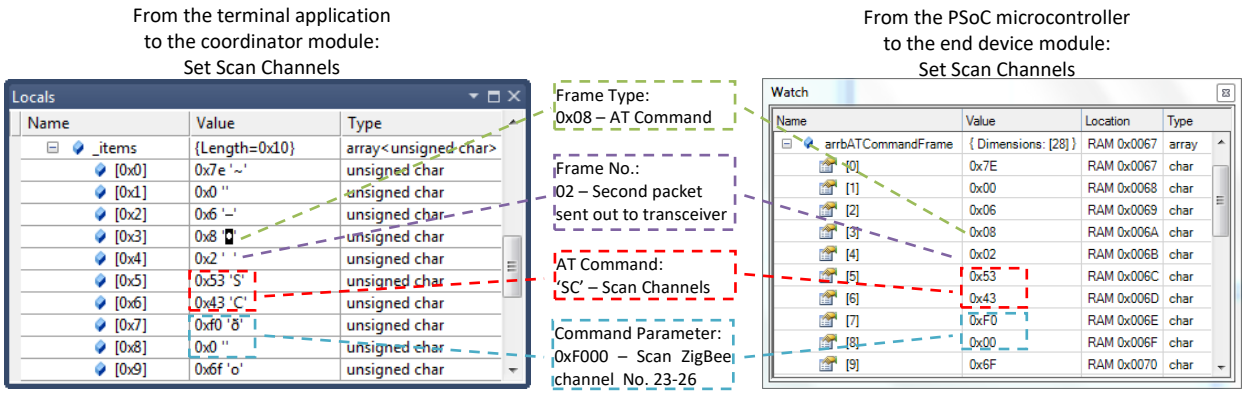


Figure 62 'Set Scan Channels' AT Command Frames

The values in each field of the above packets are shown to be identical to the theoretical values, where the theoretical AT Command frame for setting the 64-bit PAN ID has been previously

discussed (Table 9, Chapter 3 section 3.3.3.3) and the other theoretical frames such as the 'ScanChannels' frame can be found in Appendix I.

The transceiver modules reply to these commands with the following AT Command Response frames, which indicate that the parameter has been correctly set up. These responses also validate the serial communications between the controllers and the transceiver modules.

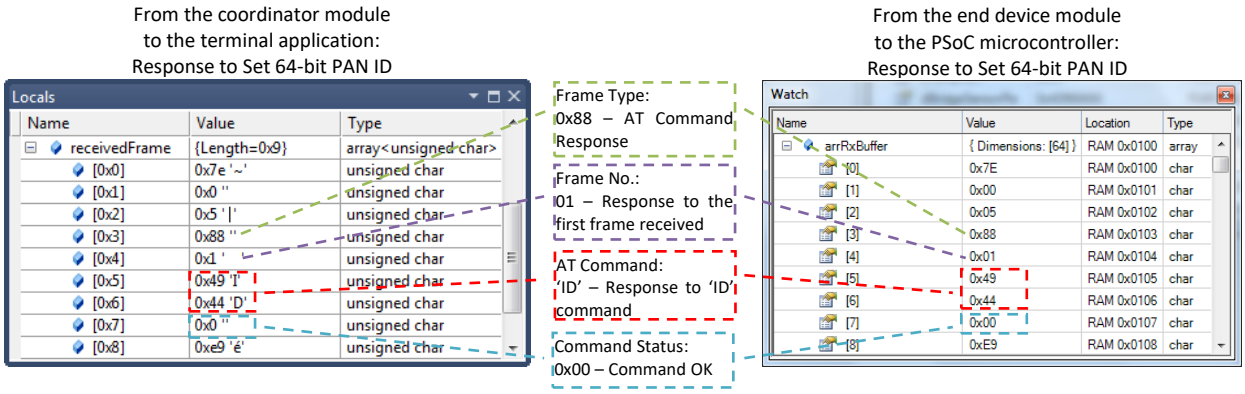


Figure 63 'Set 64-bit PAN ID' AT Command Response Frames

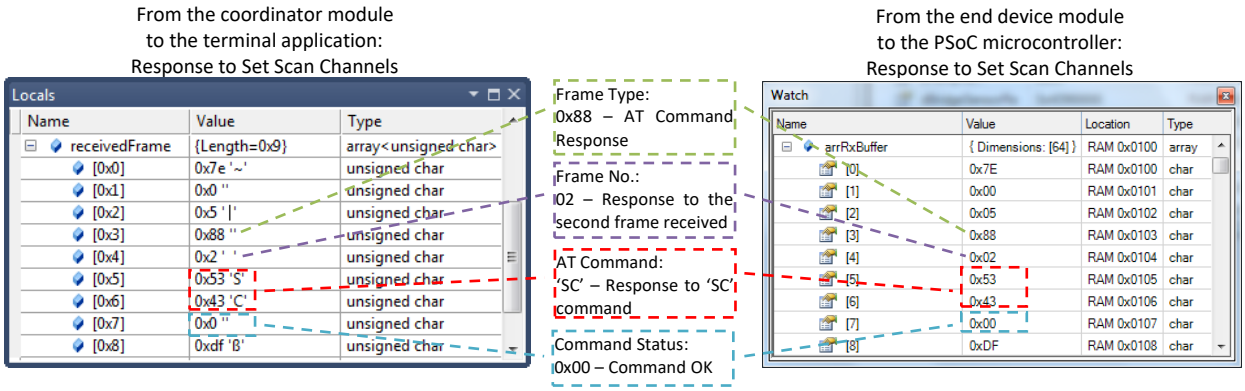


Figure 64 'Set Scan Channels' AT Command Response Frames

The data packets for setting up other network parameters are similar to the ones discussed above and their test results are presented in Appendix I.

The following two modern status message frames from the transceiver modules indicate that the coordinator has successfully started the ZigBee network and the end device has joined the network. Together they show that the network connection has been established.

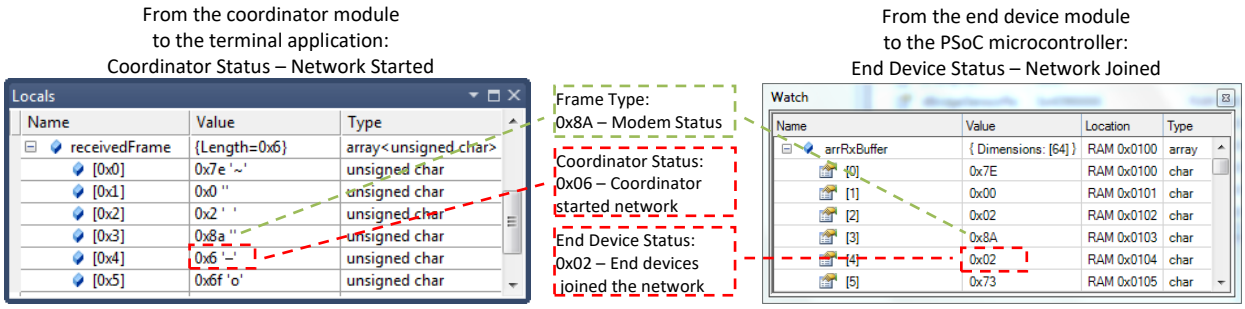


Figure 65 'Network-Success' Status Messages

6.2.1.2 Wireless Communications between RAWS nodes and the Gateway

The wireless communications between RAWS nodes and the network gateway include sensor readings transfer, TEDS information transfer, and hardware configurations requests and responses. Because the sensor reading transfer is the most common type, this subsection analyses the sensor reading transmission process to help illustrate the tests of the wireless communications.

Similar to the serial communication tests above, the data packets are also monitored in order to verify the wireless communications. Figure 66 illustrate the data flow monitoring of a C9B force sensor transmission process as an example of the tests.

The sensor reading is packed into a ZigBee Transmit Request frame constructed by the RAWS node firmware with the coordinator as the destination, and this packet is sent to the transceiver module for wireless transmitting.

From the PSoC microcontroller to the end device module:
ZigBee Transmit Request
(From the RAWS node to the gateway)

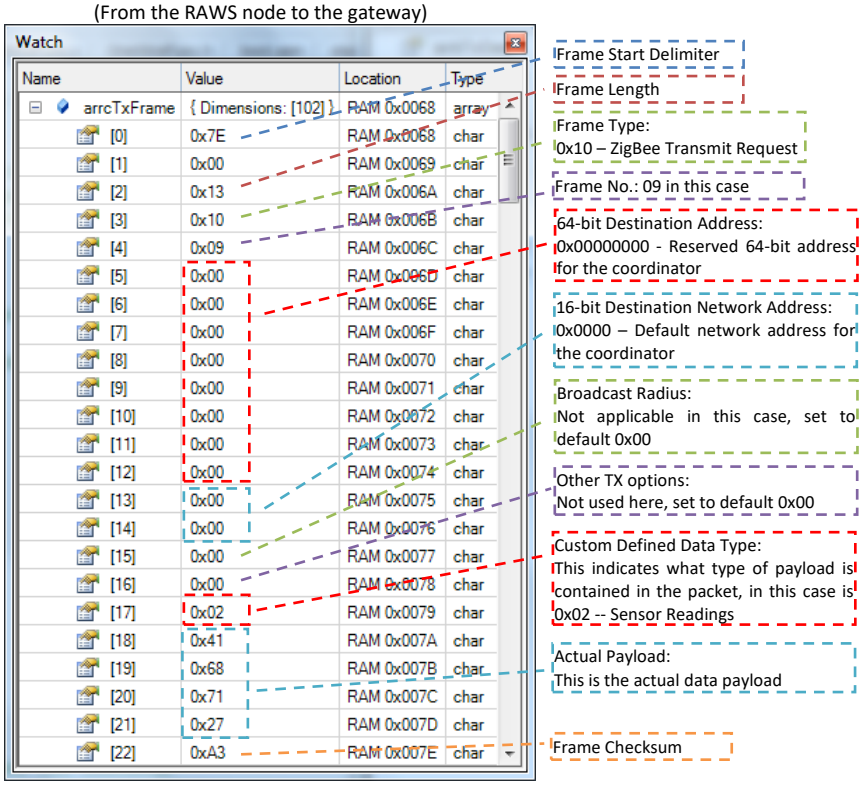


Figure 66 ZigBee Transmit Request Packet

When the end device transceiver module completes the wireless transmission attempt, it responds with a ZigBee Transmit Status frame indicating whether the transmission is successful or errors have occurred. In this test example, the transmission attempt is successful and the transceiver replies with the following status message.

From the end device module
to the PSoC microcontroller:
ZigBee Transmit Status

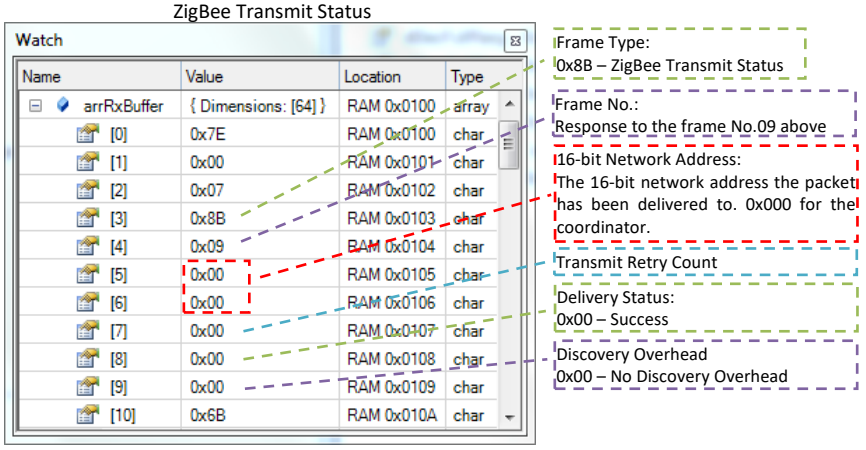


Figure 67 ZigBee Transmit Status Packet

The coordinator transceiver module receives the RF data and sends it out in a ZigBee Receive Packet via the serial interface. As illustrated below, the sensor reading data received at the gateway matches the one sent out from the RAWS node which also shows that the wireless transmission is successful.

From the coordinator module
to the terminal application:
ZigBee Receive Packet

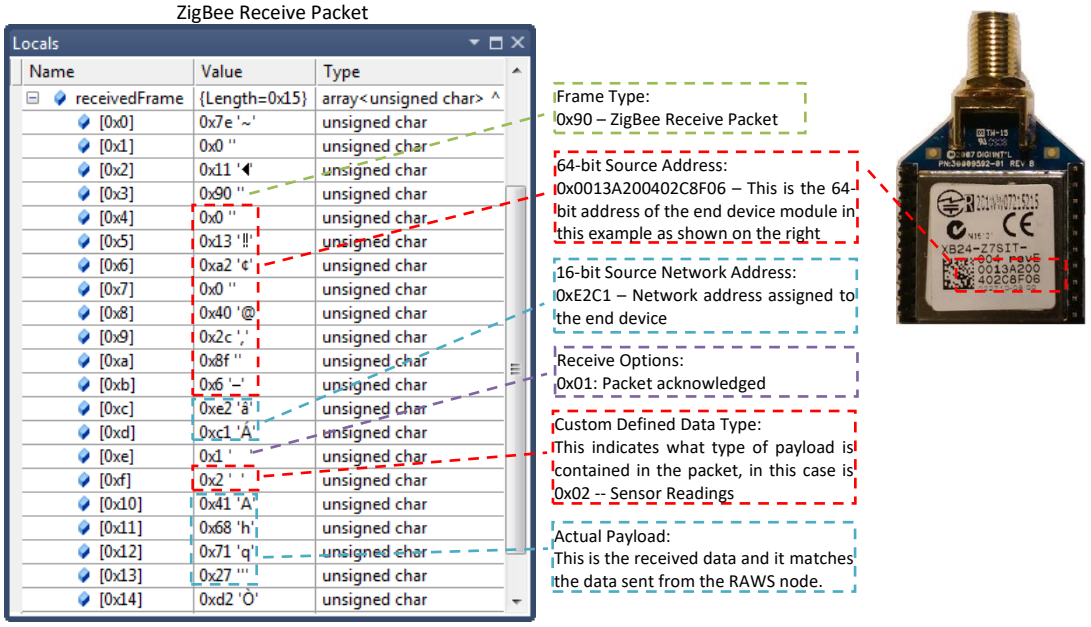


Figure 68 ZigBee Receive Packet

The RAWS node terminal application parses the ZigBee Receive Packet and then extracts, processes, displays, and stores the data.

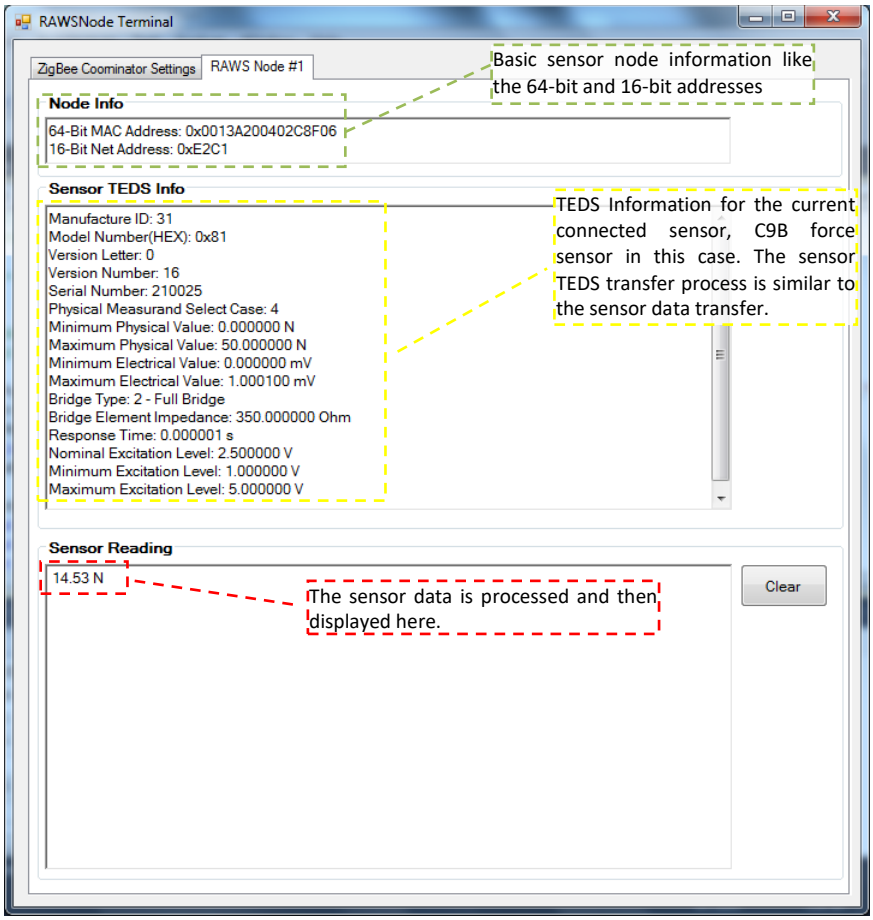


Figure 69 ZigBee Transmitted Data Displayed on RAWS Terminal Software

6.3 Sensor Identification Scheme Test

This section discusses the test strategy and results of the sensor identification scheme to verify that it is able to correctly extract sensor identification and characteristic information. The two main functions of the sensor identification scheme are:

1. Retrieve TEDS data from the memory chip of the connected sensor via 1-Wire communications
2. Parse TEDS raw data and extract the sensor identification and characteristic information

The PSoC microcontroller as the 1-Wire master device controls the TEDS data transmission, and the TEDS parsing routine which is a part of the firmware analyses the data and extracts the sensor information. In the tests, the TEDS transmitting and parsing processes are monitored with the PSoC IDE debugging tools, and the parsing results are verified against the theoretical values to confirm that the correct sensor information has been acquired. The tests focus on the Basic and the Standard TEDS sections because these two sections provide the identification and attribute information important for RAWS node hardware reconfigurations.

Figure 70 shows the raw TEDS data retrieved from the HBM C9B (50N) force sensor, including the Basic and the Standard TEDS sections.

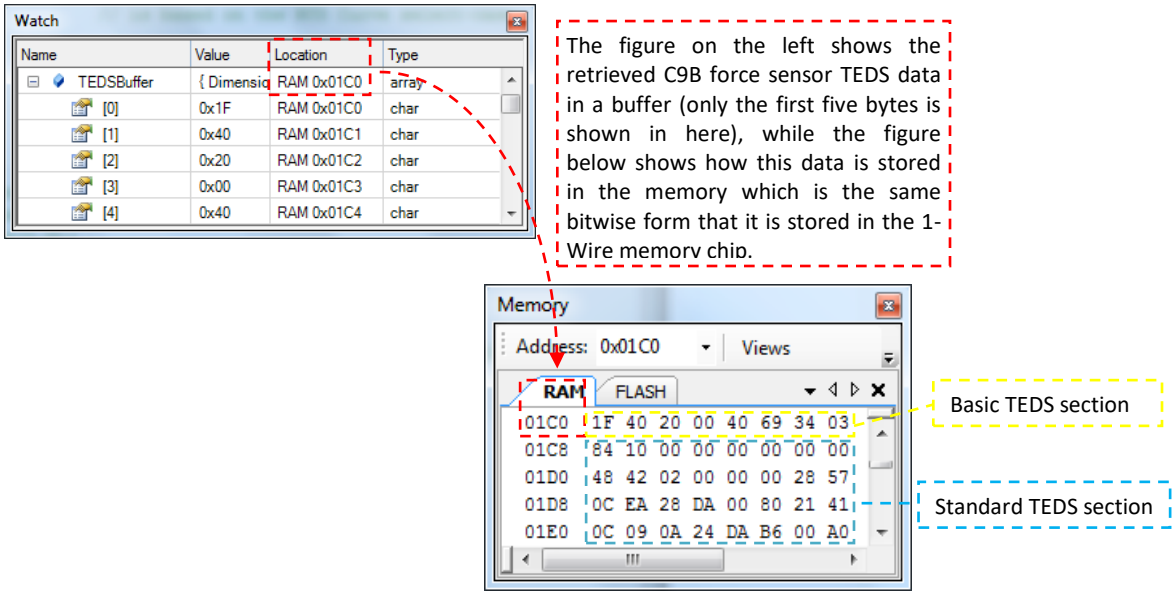


Figure 70 C9B Force Sensor TEDS data

The following figures illustrate the simplified process of parsing the raw data and extracting the theoretical values of sensor attributes from both of these TEDS sections. The TEDS parsing routine of the RAWS node firmware follows a similar process even though its algorithm structure is more complicated. Firstly, Figure 71 illustrates the parsing of the 8-byte Basic TEDS section. This section contains sensor identification information including manufacturer ID, model number, serial number, etc. This parsing process is the same for all sensors as the structure of the Basic TEDS section is fixed as previously discussed in Chapter 4.

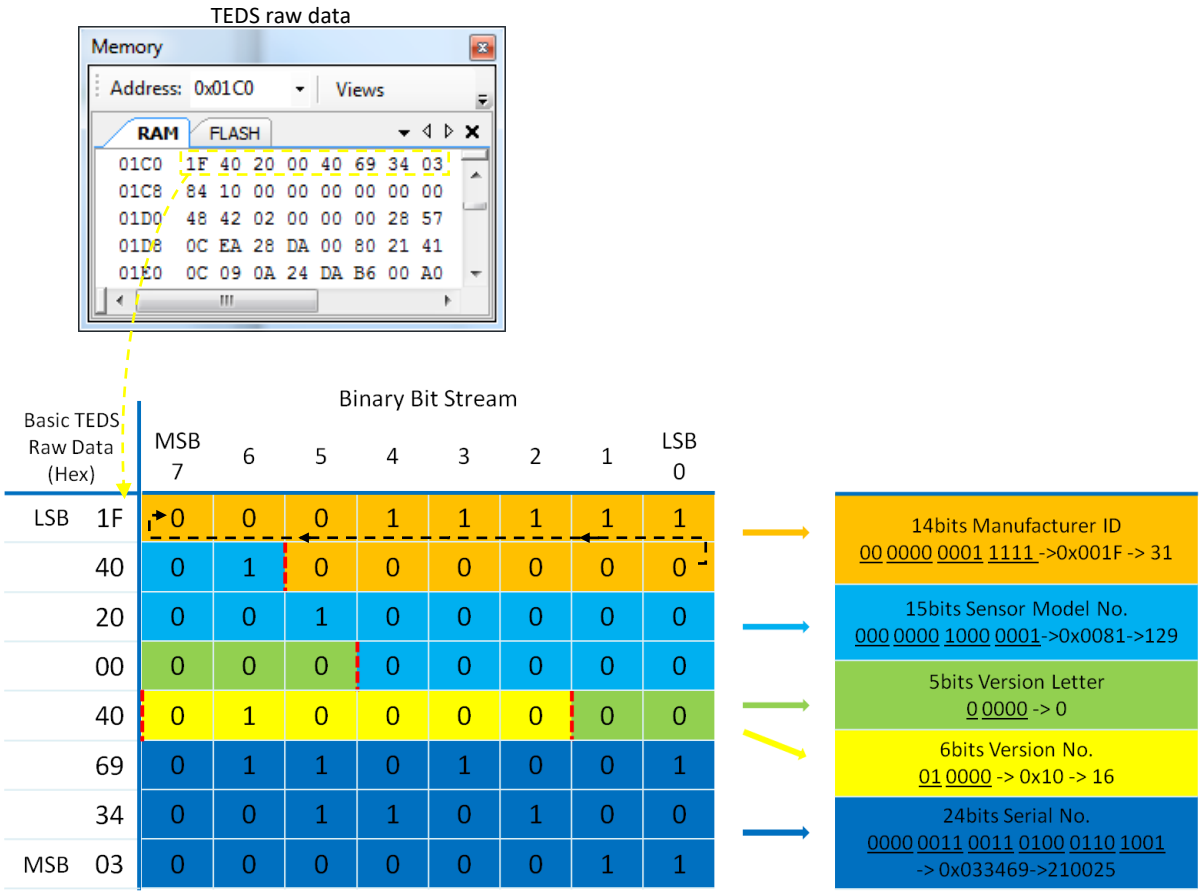


Figure 71 Parsing process of the C9B sensor Basic TEDS section

The parsing results of this Basic TEDS section from the RAWS firmware routine is shown in Figure 72 below. The identification information extracted by the routine is identical to the theoretical one above and it also matches the information provided by the IEEE registration authority [130] and the sensor's manufacturer [250] as shown in the figure.

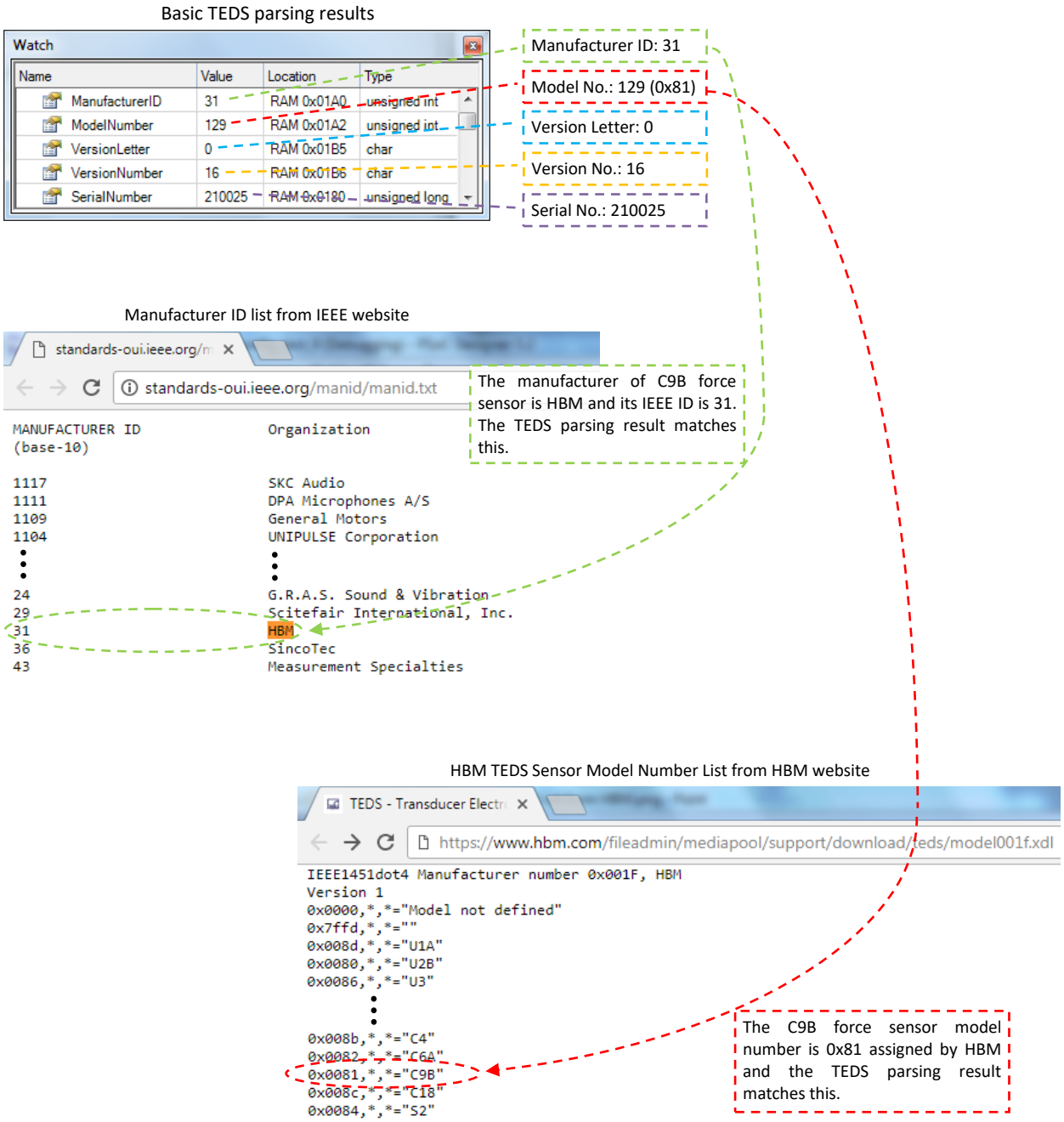


Figure 72 C9B sensor Basic TEDS section parsing results from the TEDS parsing routine

The parsing of the second TEDS section of the C9B sensor, the Standard TEDS section, is illustrated in Figure 73. This section includes important sensor attributes such as sensor type, physical measurand and measurement range, output signal range, etc.

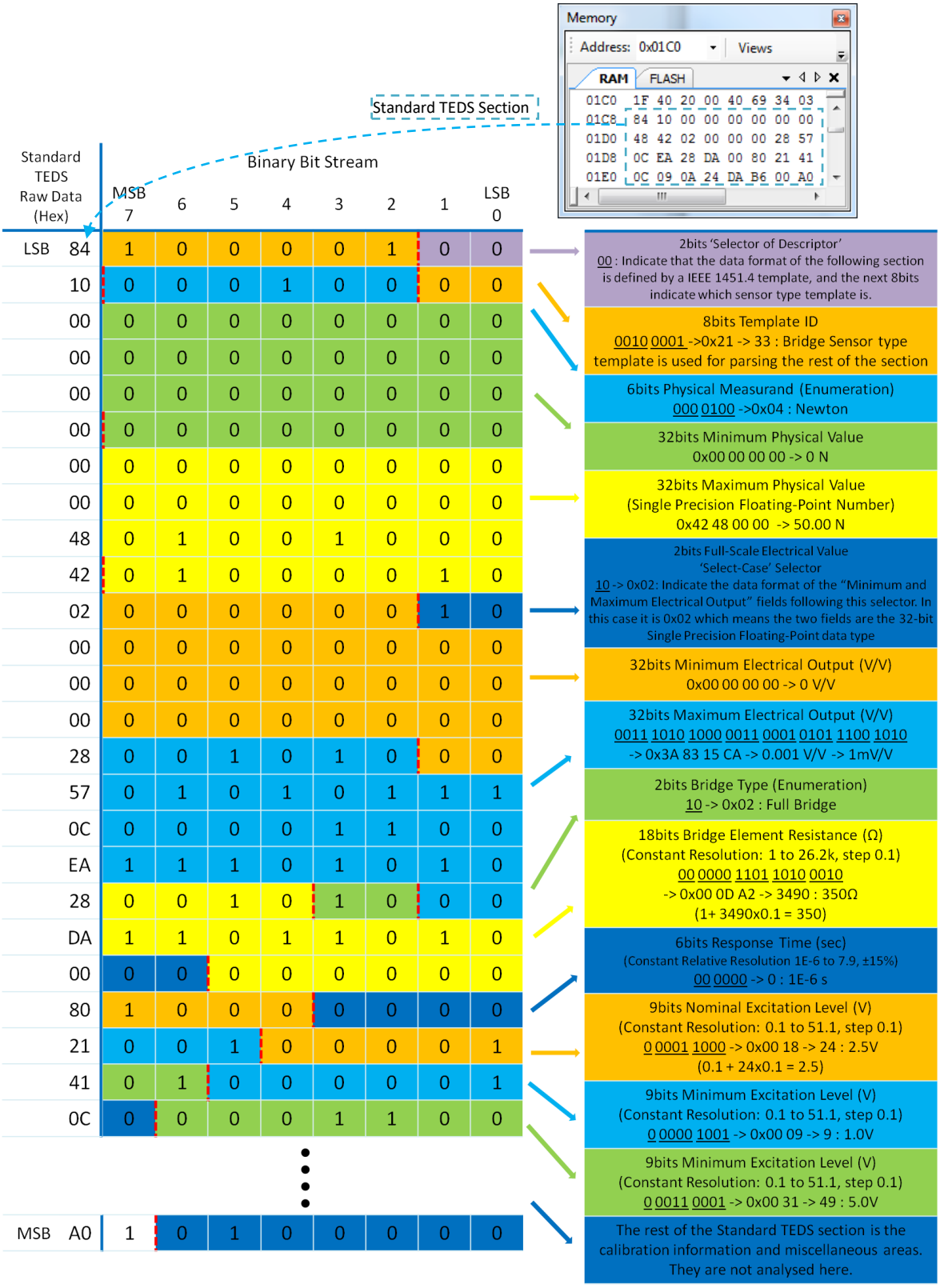


Figure 73 Parsing process of the C9B sensor Standard TEDS section

The structure of the Standard TEDS section is defined by different sensor type templates and the ‘select-case’ selectors in the template can also vary the format of this section as discussed in Chapter 4. These factors are taken into account and illustrated in the parsing process shown in Figure 73 above. The TEDS parsing routine also needs to handle this variable structure with branched subroutines. The parsing results of the Standard TEDS section of the parsing routine are illustrated in Figure 74. As demonstrated, they are identical to the results of the theoretical analysis above. The attributes also match the C9B sensor datasheet.

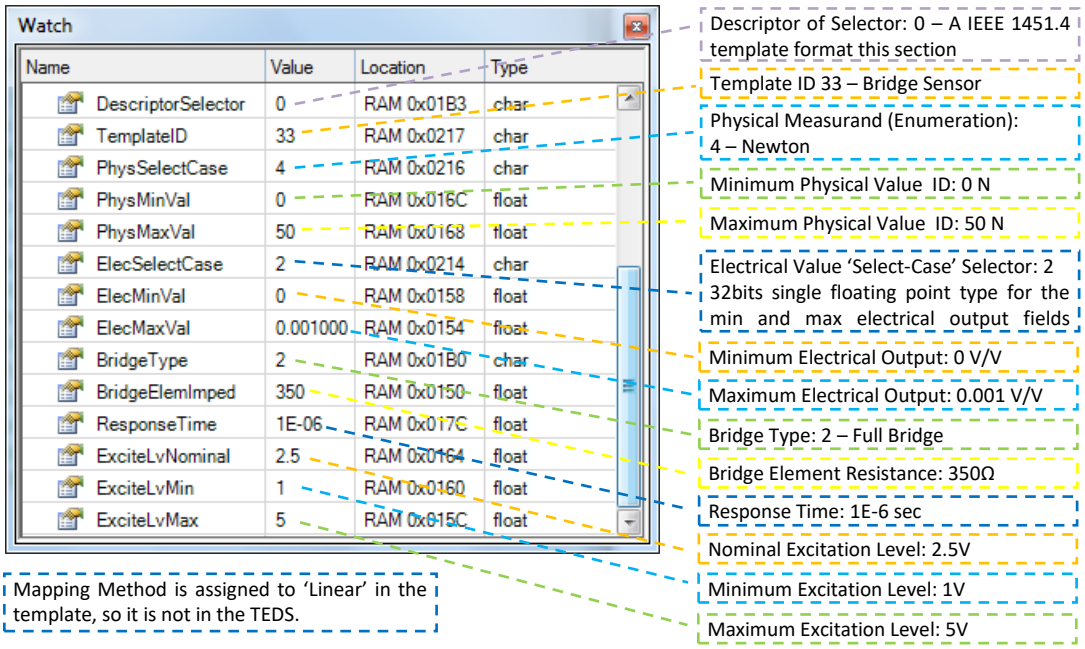


Figure 74 C9B Sensor Standard TEDS section parsing results by the TEDS parsing routine

The parsing results are also displayed on the RAWS node terminal application. The TEDS information for the C9B force sensor was shown in Figure 69 in the previous subsection so it is not repeated here.

Figure 75 to Figure 80 illustrate the test results of the reading and parsing of the Weed Instrument 101-K thermocouple and the 101-10B RTD. The IEEE 1451.4 thermocouple and RTD TEDS templates can be found in Appendix F.

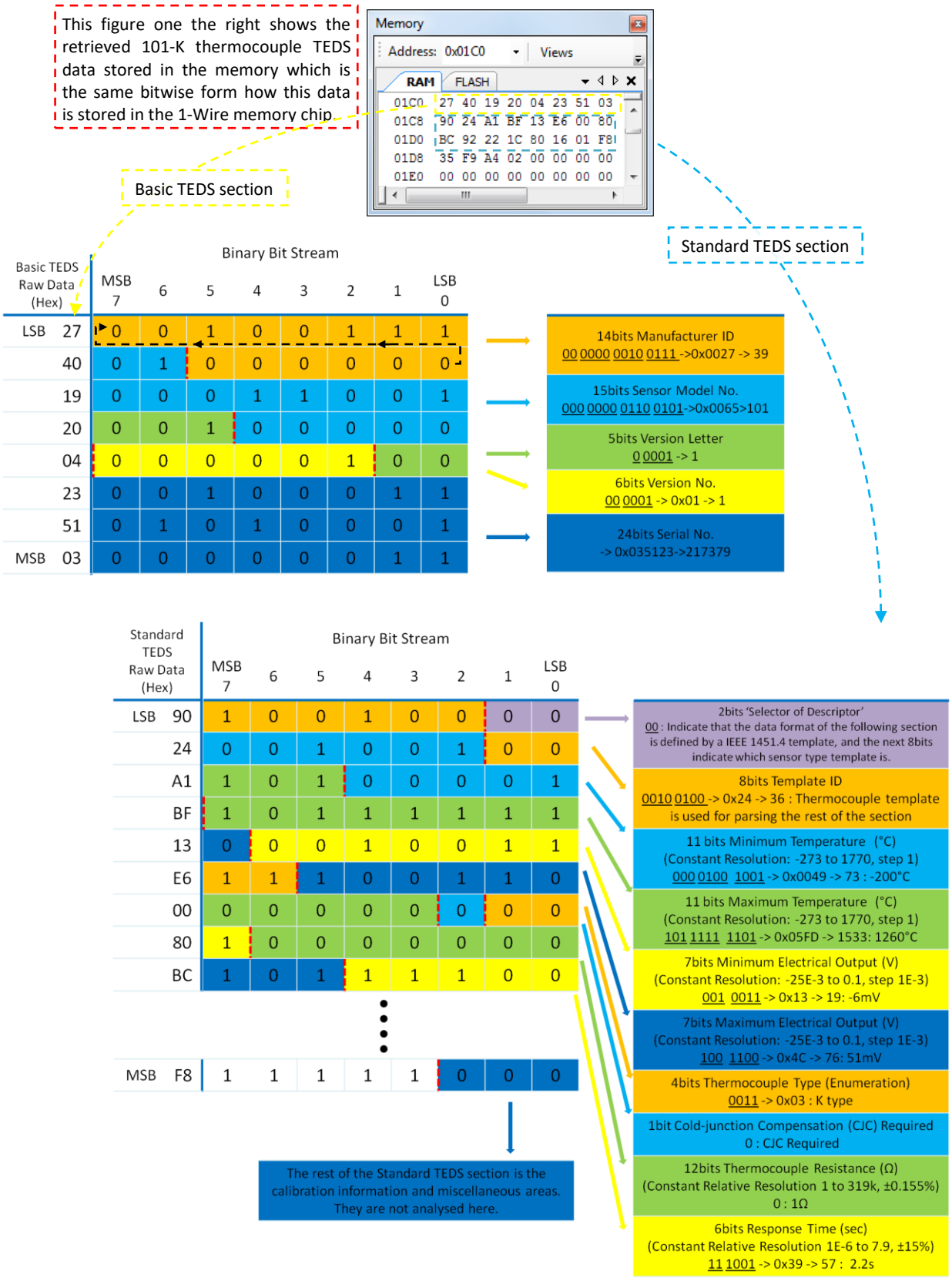


Figure 75 101-K Thermocouple TEDS data and parsing process

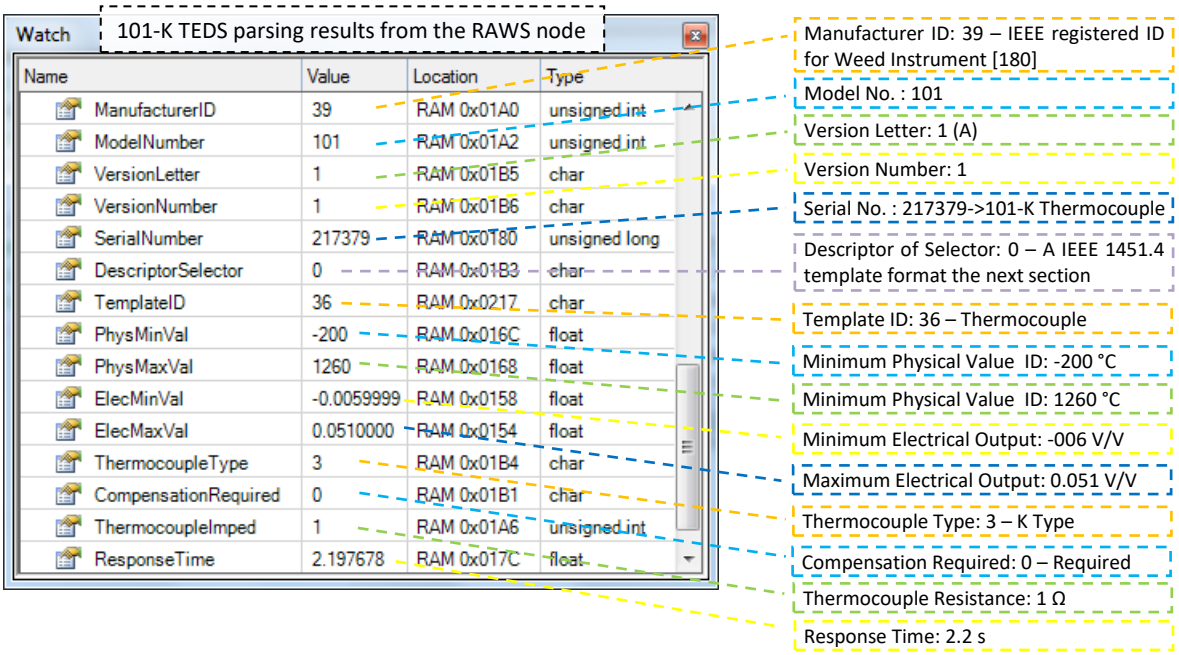


Figure 76 101-K Thermocouple TEDS parsing results by the TEDS parsing routine

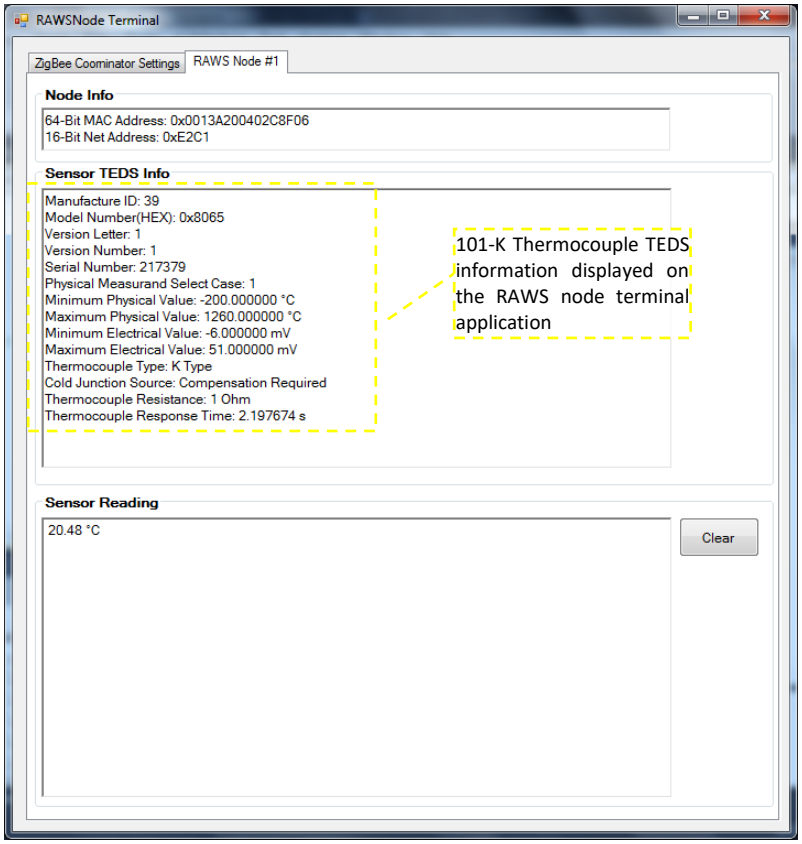


Figure 77 101-K Thermocouple TEDS parsing results on RAWS node terminal application

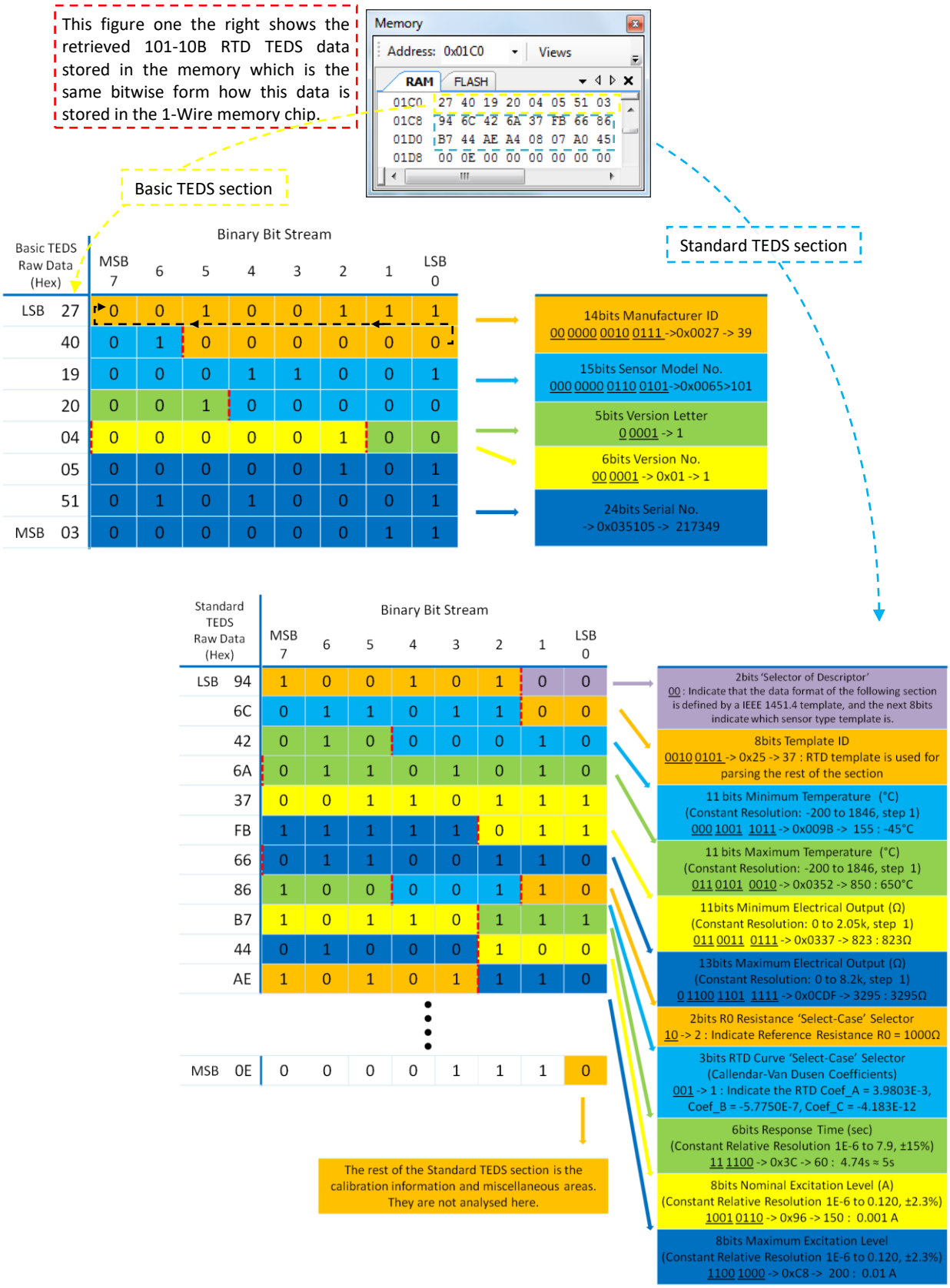


Figure 78 101-10B RTD TEDS data and parsing process

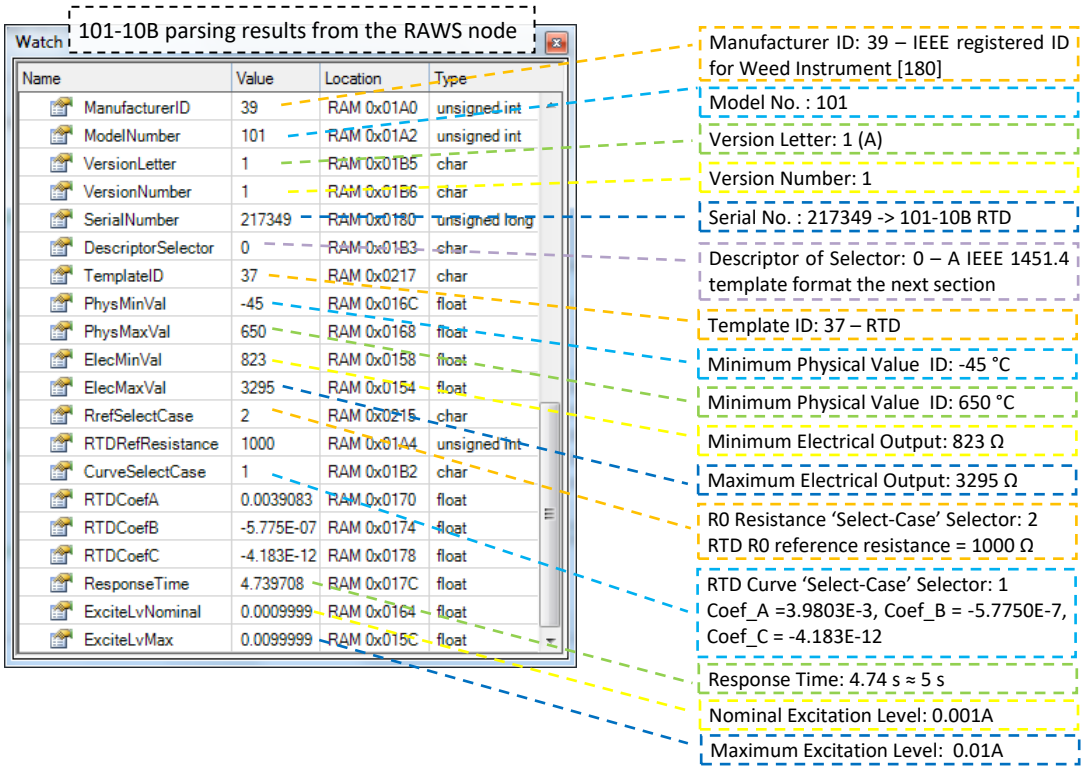


Figure 79 101-10B RTD TEDS parsing results by the TEDS parsing routine

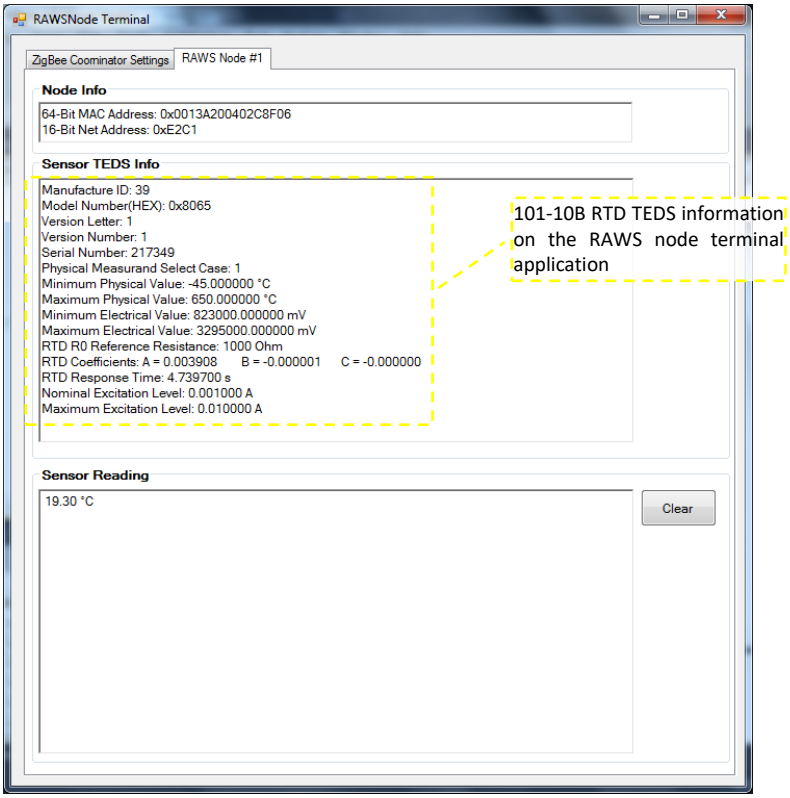


Figure 80 101-10B RTD TEDS parsing results on RAWS node terminal application

The figures above show that the parsing results of the 101-K thermocouple and the 101-10B RTD TEDSs from the RAWS node routine match the theoretical analyses. These results match the sensor's data sheet as well.

The HBM P8AP pressure sensor TEDS is also a bridge sensor and the parsing process of its TEDS is similar to the C9B force sensor, so it is not presented here but it can be found in Appendix I.

The test results show that the RAWS node can successfully retrieve TEDS data from the connected sensor and correctly extract sensor identification and attributes information.

As discussed in Chapter 4, the sensor information acquired by the identification scheme is then utilised to adaptively reconfigure the programmable hardware to support the sensor. The sensor key attributes are utilised by the intelligent algorithm to build the signal conditioning and processing chain. The identification information is used as the sensor ID by the local and remote configuration settings adaptive reconfiguration techniques to search for the corresponding hardware configuration settings. So the sensor identification scheme test results discussed in this subsection are important for the next key step of the system tests, the adaptive reconfiguration tests.

6.4 Adaptive Reconfiguration Test and Analysis

Three adaptive reconfiguration techniques were developed in the research and they are the key component of the RAWS technology that binds the reconfigurability of programmable hardware and the sensor identification scheme together to realise the autonomic multi-sensing capability.

The intelligent algorithm adaptive reconfiguration is the key technique of the three. Based on the sensor key attributes, the algorithm autonomically constructs analogue function modules and sets up suitable parameters to build the signal conditioning and processing chain for supporting the connected sensor. Plus the hardware configuration settings generated by it can be reused by the other two techniques. Therefore, the adaptive reconfiguration tests focus on the intelligent algorithm technique.

The control registers of the analogue blocks as well as other analogue resources of the PSoC programmable platform are monitored to verify whether the reconfiguration technique has functioned properly and to validate that the suitable signal processing chain has been successfully built. The theoretical and test results of the hardware reconfiguration process for the C9B force sensor (load cell), 101-K thermocouple and 101-10B RTD are discussed in detail here. The reconfiguration for the P8AP pressure sensor is similar to the process for C9B sensor and the test results are presented in Appendix I.

6.4.1 C9B Force Sensor Reconfiguration Process Analysis and Test Results

In the case where the C9B force sensor is connected to the RAWS node platform, the following key sensor attributes are acquired as discussed in the test results in the previous section.

Attribute	Value
Sensor Type	Bridge Sensor (Full Bridge)
Physical Measurand	Force in Newtons
Measurement Range	0 – 50 N
Electrical Output Range	0 – 1 mV/V (Sensitivity)
Mapping Method	Linear (Defined in the template, not in TEDS)
Nominal Excitation Voltage	2.5 V

Table 18 C9B Force Sensor Key Attributes

The theoretical process of how the intelligent algorithm performs the hardware reconfiguration with the above sensor information is analysed as follows. As discussed in Chapter 5, the intelligent algorithm is able to process the sensor characteristics and construct a signal chain which consists of three main analogue modules including a three opamp instrumentation amplifier (INS-AMP), a 12-bit delta sigma ADC, and a 9-bit DAC. The algorithm is designed to deploy the function modules starting from the analogue column 0 of the CAB array so the modules for C9B sensor should be located in the array as illustrated in Figure 81.

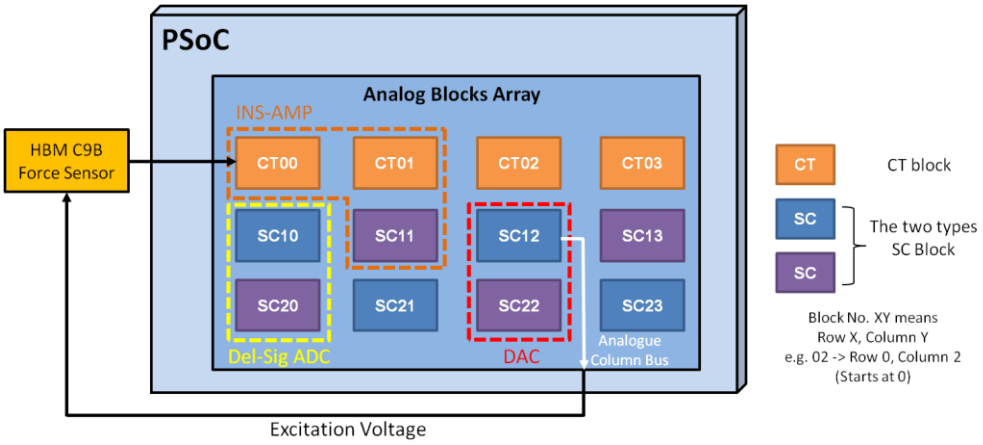


Figure 81 C9B signal process chain placement in PSoC

DAC Module

The DAC is built using two SC blocks with the configuration shown below, SC12 block as the MSB stage and the SC22 as the LSB stage.

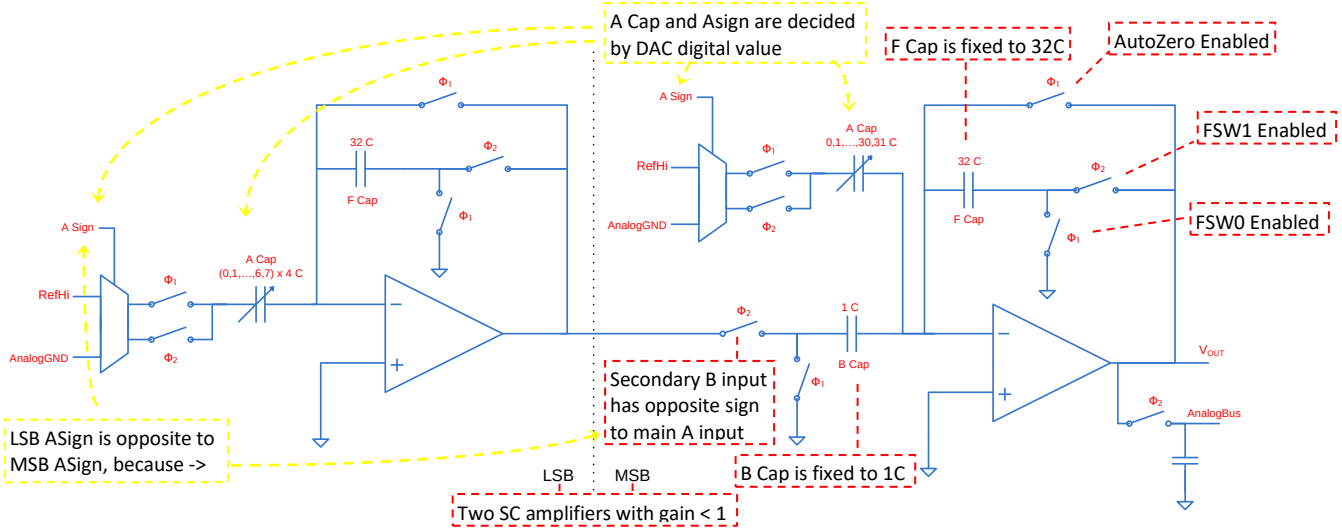


Figure 82 9-bit DAC Configuration

The configuration determines the following control register bits for the MSB and LSB blocks.

Register Bits Name	Register Bits	Value (Binary)	Description
AMux	SC12_CR1[7:5] (MSB Block)	010	The main A inputs for both MSB and LSB stage are the same and are connected to RefHi. The AMux values are different for them only because they take the two different types of SC blocks in one column which have different register values assigned for RefHi.
	SC22_CR1[7:5] (LSB Block)	100	
ARefMux	SC12_CR3[7:6] SC22_CR3[7:6]	00	The reference of the main A input is connected to AnalogGND for both blocks
BMux	SC12_CR3[3:2] (MSB Block)	11	The secondary B input of the MSB is connected the output of the LSB block, the block SC22.
	SC22_CR1[3:2] (LSB Block)	00	The secondary B input of the LSB is not used so it is left at the default setting.
AutoZero	SC12_CR2[5] SC22_CR2[5]	1	Shorting switch of the feedback path is controlled by Φ_1 for both blocks
FSW0	SC12_CR3[4] SC22_CR3[4]	1	FSW0 switch of the feedback path is controlled by Φ_1 for both blocks
FSW1	SC12_CR3[5] SC22_CR3[5]	1	FSW1 switch of the feedback path is controlled by Φ_2 for both blocks
AnalogBus	SC12_CR2[7]	1	Analogue bus output connection is enabled for the MSB block
	SC22_CR2[7]	0	Analogue bus output connection is disabled for the LSB block
CompBus	SC12_CR2[6] SC22_CR2[6]	0	Comparator output is disabled for both blocks

Table 19 Control Register Settings for 9-bit DAC Part 1

The algorithm also needs to set up the parameters for the function module which is the DAC digital value that represents the DAC output voltage in this case.

The DAC output voltage is controlled by configuring the capacitors in the SC circuitry. According to the SC DAC operation principle discussed in Chapter 5 Section 5.2, the ratio of the input capacitors and the feedback capacitors decides the DAC output. The feedback capacitor, F Cap, is fixed to 32 capacitor units (32C) in the configuration. The MSB stage secondary input capacitor, which is connected to the LSB stage output, is fixed to 1C. Therefore, the variables in the configuration are the two main A input capacitors (ACap bits at CR0[4:0]) and the sign of the main input (ASign bit at CR0[5], controls whether the input is sampled at Φ_1 for positive

output or Φ_2 for negative output with reference to the analogue ground). They specify the DAC value and vice versa.

The 9-bit DAC digital value is represented in the form of 2's complement numbers in the range of -255 to +255. The MSB of the DAC digital 2's complement value, i.e. the sign bit of the digital value, decides the sign of the main input. The next 5 bits specify the A capacitor of the MSB stage and the 3 least significant bits specify the A capacitor of the LSB stage. The algorithm sets up these parameters according to the DAC value.

The digital value is calculated based on the desired voltage, which is the sensor nominal excitation level and it is 2.5V for the C9B sensor, as well as the DAC output range. The algorithm is designed to choose the smallest DAC output range option that is still able to cover the required excitation level. In this case, it chooses the '$V_{BandGap} \pm V_{BandGap}$' reference option which means the DAC output is ranging from 0 to 2.6V. Therefore, the DAC value can be calculated:

$$DAC_{Value} = 510 \times \frac{2.5}{2.6} - 255 \approx 235 = 0:1110\ 1:011\ in\ Binary$$

Based on the analysis above, the capacitor and input sign settings of the SC circuitry can be determined and are summarised in Table 20.

Register Bits Name	Register Bits	Value (Binary)	Description
ACap	SC12_CR0[4:0] (MSB Block)	11101	The 5 bits (11101b in this case) after the MSB bit of the DAC digital value specify the ACap of the MSB block.
	SC22_CR0[4:0] (LSB Block)	01100	The 3 least significant bits (011b in this case) specify the ACap of LSB block. As the ACap is 5 bits, two 0s are appended.
ASign	SC12_CR0[5]	0	The ASign of the MSB block is specified by the MSB bit of the DAC value. In this case is 0 which means input is sampled at Φ_1 as the desired output above the AGND. The ASign of the LSB block is opposite to the MSB ASign because the LSB output will be inverted back at the MSB stage.
	SC22_CR0[5]	1	
BCap	SC12_CR1[4:0]	00001	The BCap of the MSB block is fixed to 1C, while the secondary B input of the LSB block is not used so the BCap is left at 0.
	SC22_CR1[4:0]	00000	
FCap	SC12_CR0[7] SC22_CR0[7]	1	Both feedback capacitors are fixed to 1C.

Table 20 Control Register Settings for 9-bit DAC Part 2

During the sensor measurement process, the power mode of the two blocks which is controlled by the PWR bits of CR3 is set to full-power as shown in Table 21.

Register Bits Name	Register Bits	Value (Binary)	Description
PWR	SC12_CR3[1:0] SC22_CR3[1:0]	11	PWR bits are set to 11b for full-power only during the sensor measurement process and will be set to 00b for off-mode for saving power.

Table 21 Control Register Settings for 9-bit DAC Part 3

The bits that are not used for this configuration are set to the default state which is 0. To conclude the DAC configuration settings analysed above, the control registers of the two SC blocks should have the values as summarised in Table 22 and Table 23.

SC12(MSB)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0x9D	FCap 1	ClockPhase 0	ASign 0	ACap[4:0] 1 1101				
CR1	0x41	AMux[7:5] 010			BCap[4:0] 0 0001				
CR2	0xA0	AnalogBus 1	CompBus 0	AutoZero 1	0 0000				
CR3	0x3F	ARefMux[7:6] 00		FSW1 1	FSW0 1	BMux[3:2] 11		PWR[1:0] 11	

Table 22 MSB Control Registers of 9-bit DAC

SC22(LSB)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0xAC	FCap 1	ClockPhase 0	ASign 1	ACap[4:0] 0 1100				
CR1	0x80	AMux[7:5] 100			BCap[4:0] 0 0000				
CR2	0x20	AnalogBus 0	CompBus 0	AutoZero 1	0 0000				
CR3	0x33	ARefMux[7:6] 00		FSW1 1	FSW0 1	BMux[3:2] 00		PWR[1:0] 11	

Table 23 LSB Control Registers of 9-bit DAC

Figure 83 shows the control registers monitor of the SC12 and SC22 blocks at runtime when the C9B sensor is connected. As illustrated in the figure, the test results match the theoretical analysis results discussed above.

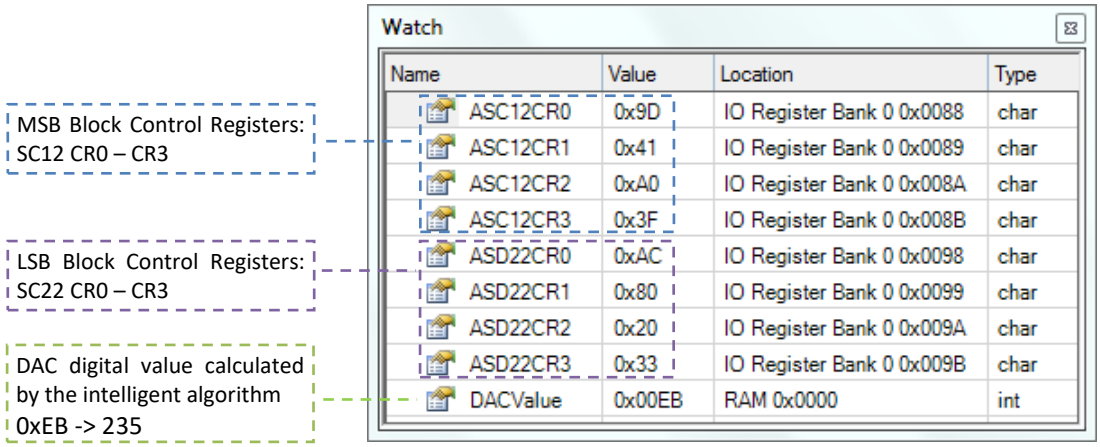


Figure 83 DAC control registers in PSoC

Instrumentation Amplifier Module

The three opamp instrumentation amplifier is constructed using two CT blocks and one SC block with the configuration shown in Figure 84.

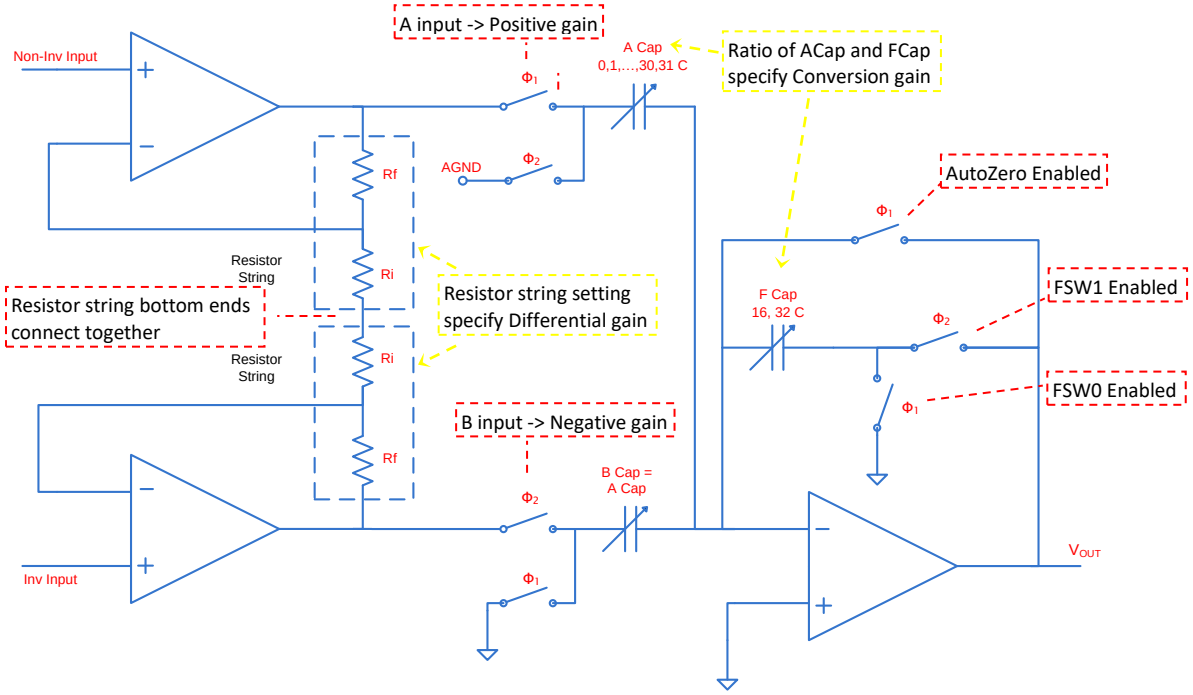


Figure 84 Three opamp instrumentation amplifier configuration

The configuration determines the following control register bits (Table 24 and Table 25) for the two CT blocks of the differential stage and the SC block of the conversion stage, with CT00 block as the inverting (Inv) input, CT01 block as non-inverting (NonInv) input, and SC11 block as the conversion stage.

CT Register Bits Name	Register Bits	Value (Binary)	Description
PMux	CT01_CR1[2:0] (NonInv Block) CT00_CR1[2:0] (Inv Block)	001	The positive inputs of the opamps of both CT blocks are connected to port inputs, i.e. input signals from GPIOs through the analogue subsystem's input multiplexer. Bits value 001b select the port inputs option.
NMux	CT01_CR1[2:0] CT00_CR1[2:0]	100	The negative inputs of the opamps of both CT blocks are connected to feedback point of the opamp circuit. 100b select the feedback option.
INSAMP	CT01_CR3[1] CT00_CR3[1]	1	The INSAMP bit is set and this specifies that the resistor string bottom ends of the two CT blocks are tied together to form INS-AMP, and this overrides the RBotMux bits below.
RBotMux	CT01_CR0[1:0] CT00_CR0[1:0]	00	In normal cases, RBotMux bits control the resistor string bottom multiplexer but for INS-AMP it is overridden by INSAMP bit and is left at the default value 00b.
CMOut	CT01_CR3[2] CT00_CR3[1]	0	Common mode output connection is not used so it is disabled for both blocks.
AnalogBus	CT01_CR1[7] CT00_CR1[7]	0	Analogue bus output connection is disabled for both blocks.
CompBus	CT01_CR1[6] CT00_CR1[6]	0	Comparator output connection is disabled for both blocks.

Table 24 CT Blocks Control Register Settings for INS-AMP Part1

SC Register Bits Name	Register Bits	Value (Binary)	Description
AMux	SC11_CR1[7:5]	000	The main A input of the conversion stage is connected to the output of the NovInv block output, i.e. CT01 block. CT01 is right above SC11. The register bits value for connecting to the block above is 000b.
ARefMux	SC11_CR3[7:6]	00	The reference of the main A input is connected to AnalogGND.
ASign	SC11_CR0[5]	0	The A input is sampled at Φ_1 for positive gain to handle the differential output, since the B input is fixed to be negative in the SC block topology.
BMux	SC11_CR3[3:2]	10	The secondary B input branch of the conversion stage is switched with sampling and is connected to the output of the Inv block, i.e. CT00 block.
AutoZero	SC11_CR2[5]	1	Shorting switch of the feedback path is controlled by Φ_1.
FSW0	SC11_CR3[4]	1	FSW0 switch of the feedback path is controlled by Φ_1.
FSW1	SC11_CR3[5]	1	FSW1 switch of the feedback path is controlled by Φ_2.
AnalogBus	SC11_CR2[7]	0	Analogue bus output connection is disabled.
CompBus	SC11_CR2[6]	0	Comparator output connection is disabled.

Table 25 SC Block Control Register Settings for INS-AMP Part1

The algorithm also sets the gain of the instrumentation amplifier. The three opamp INS-AMP supports gain settings up to 93. As discussed in Chapter 5, the algorithm selects the highest possible gain setting based on factors including the sensor output signal range and the ADC input range. In this case, the maximum output signal is equal to the product of the sensor sensitivity 1mV/V (the maximum electrical output attribute in the TEDS) and excitation voltage level 2.5V, which is 2.5mV. The ADC input range is also determined by the analogue system reference setting like the DAC output range, so it is from 0 to 2.6V. Therefore, the algorithm selects the maximum gain supported by the amplifier, i.e. 93.

The INS-AMP consists of two stages, the differential stage and the conversion (output) stage, so the gain of the INS-AMP is also the total gain of two stages and it is the product of the differential gain and the conversion gain. To get a total gain of 93, the differential gain has to be 48 while the conversion gain setting is 1.9375, which are the maximum gains supported by the CT and SC block respectively. The differential gain is determined by the symmetric resistor string settings of the two CT blocks and it is equal to $1+(R_f/R_i)$. Therefore the gain of 48 is achieved by dividing the resistor string of 48 unit resistors into R_f segment of 47 units and R_i segment of 1 unit. The conversion gain is decided by the ratio of the input capacitor and the

feedback capacitor, and the maximum ratio is achieved by setting the input capacitor to 31 capacitor units (31C) and the feedback capacitor to 16C resulting the gain of 1.9375. The register settings for these resistors and capacitors are illustrated in Table 26 and Table 27.

Register Bits Name	Register Bits	Value (Binary)	Description
Gain	CT01_CR0[3] (NonInv Block) CT00_CR0[3] (Inv Block)	1	The gain bit specifies whether the resistor string is connected around the opamp for gain or for loss. In this case, the bit is set to 1 for gain.
RTapMux	CT01_CR0[7:4] CT00_CR0[7:4]	0000	The combination of the RTapMux and EXGAIN specify the selection of the resistor taps from 18 taps. The 0000b and 1b combination selects the bottom tap which means R_f = 47 and R_i = 1. Therefore the block gain is equal to 1+47/1 = 48.
EXGAIN	CT01_CR0[7:4] CT00_CR0[7:4]	1	

Table 26 CT Blocks Control Register Settings for INS-AMP Part2

Register Bits Name	Register Bits	Value (Binary)	Description
ACap	SC11_CR0[4:0]	11111	A capacitor and B capacitor are set to 31C (11111b). F capacitor is set to 16C (0b). So the conversion stage gain is equal to 31/16 = 1.9375.
BCap	SC11_CR1[4:0]	11111	
FCap	SC12_CR0[7]	0	

Table 27 SC Block Control Register Settings for INS-AMP Part2

The power mode of the amplifier is set to full-power during the sensor measurement process as shown in Table 28.

Register Bits Name	Register Bits	Value (Binary)	Description
PWR	CT01_CR2[1:0] CT00_CR2[1:0] SC11_CR3[1:0]	11	PWR bits are set to 11b for full-power only during the sensor measurement process and will be set to 00b for off-mode for saving power.

Table 28 Control Register Settings for INS-AMP Part 3

The other register bits are set to the default states for amplifiers. To conclude the INS-AMP configuration settings analysed above, the control registers of the two CT blocks and one SC block should have the values as summarised in Table 29 and Table 30.

CT01&00	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0x0C	RTapMux[7:4] 0000				Gain 1	1	RBotMux[1:0] 00	
CR1	0x21	AnalogBus 0	CompBus 0	NMux[5:3] 100			PMux[2:0] 001		
CR2	0x23	0	0	1	0	00		PWR[1:0] 11	
CR3	0x03					0	CMOut 0	INSAMP 1	EXGAIN 1

Table 29 CT Blocks Control Registers of INS-AMP

SC11(Conv)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0x1F	FCap 0	ClockPhase 0	ASign 0	ACap[4:0] 1 1111				
CR1	0x1F	AMux[7:5] 000			BCap[4:0] 1 1111				
CR2	0x20	AnalogBus 0	CompBus 0	AutoZero 1	0 0000				
CR3	0x3B	ARefMux[7:6] 00		FSW1 1	FSW0 1	BMux[3:2] 10		PWR[1:0] 11	

Table 30 SC Block Control Registers of INS-AMP

Figure 85 shows the control registers monitor of the three blocks of INS-AMP at runtime and it shows that the test results match the theoretical analysis results discussed above.

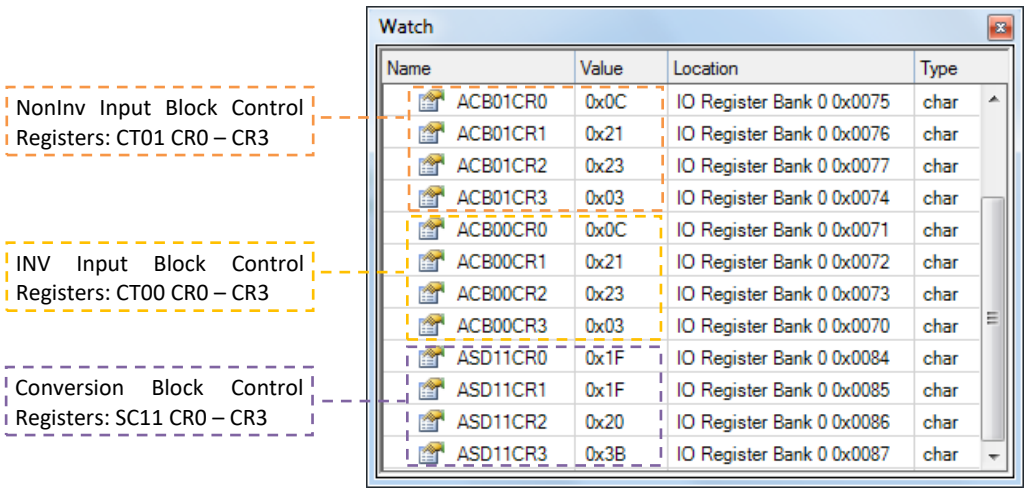


Figure 85 INS-AMP control registers in PSoC for C9B sensor

Delta Sigma ADC

As discussed in Chapter 5, the delta sigma ADC module is comprised of a modulator, a decimator and a timing generator. The main part of the hardware configuration is to construct the modulator. For the C9B sensor, the algorithm should select the 12-bit delta sigma ADC and build a second-order modulator with the configuration illustrated in Figure 86.

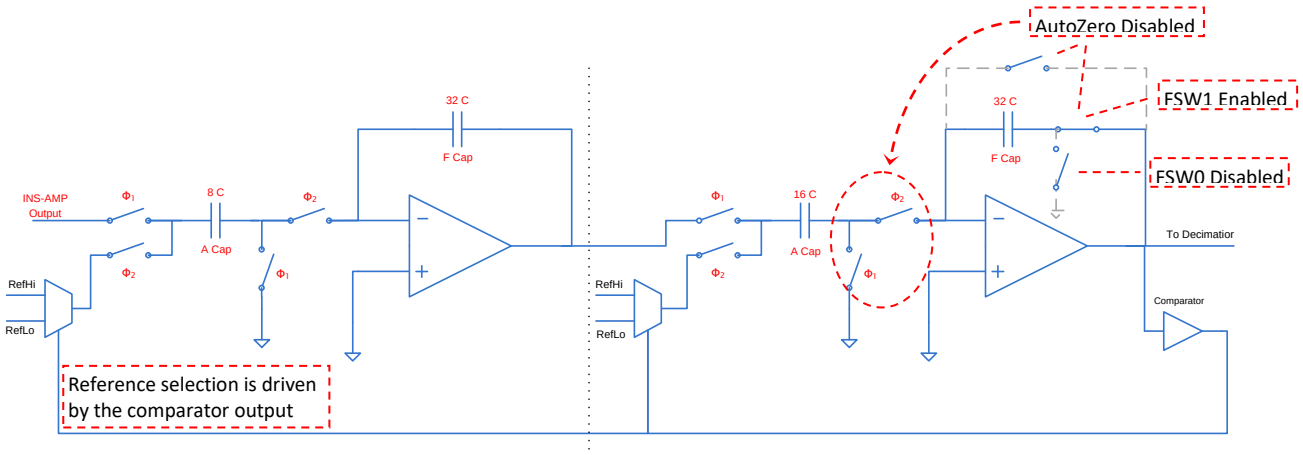


Figure 86 Second-order modulator configuration for Delta Sigma ADC

The two blocks are essentially configured into an integrator circuit with the comparator output for 1-bit oversampling. This configuration fully determines the control registers for the two SC blocks as shown in the following three tables (Table 31 to Table 33), with SC10 as the first stage (input) and the SC20 as the second stage (output).

Register Bits Name	Register Bits	Value (Binary)	Description
AMux	SC10_CR1[7:5] (Input Block)	001	The A input of the first stage is connected to the INS-AMP output which is SC11 block, and the register value is 001b.
	SC20_CR1[7:5] (Output Block)	000	The A input of the second stage is connected to the first stage which is SC10 block right above the SC 20 block and the register value is 000b for this.
ARefMux	SC10_CR3[7:6] SC20_CR3[7:6]	11	The reference of the main A input is controlled by the comparator output through the comparator bus of the analogue column for both blocks.
AutoZero	SC10_CR2[5] SC20_CR2[5]	0	AutoZero is disabled. The shorting switch of the feedback path is always open while the two switches behind the A cap is controlled by the internal clocks.
FSW0	SC10_CR3[4] SC20_CR3[4]	0	FSW0 switch of the feedback path is always open.
FSW1	SC10_CR3[5] SC20_CR3[5]	1	FSW1 is enabled and AutoZero is disabled, so the FSW1 switch of the feedback path is always closed.
AnalogBus	SC10_CR2[7]	0	Analogue bus output connection is disabled for both blocks.
	SC20_CR2[7]	0	
CompBus	SC10_CR2[6]	0	Comparator bus connection of the first stage is not used.
	SC20_CR2[6]	1	Comparator bus connection is enabled for the second stage to drive the reference multiplexers of the two blocks.
ClockPhase	SC10_CR0[6] SC20_CR0[6]	1	Swapped the internal clocks Φ_1 and Φ_2 of the ADC to work with the three opamp INS-AMP because the INS-AMP provides outputs at Φ_2.

Table 31 Control Register Settings for Delta Sigma ADC 2nd order Modulator Part 1

Register Bits Name	Register Bits	Value (Binary)	Description
ACap	SC10_CR0[4:0] (Input Block)	01000	The A capacitor of the first stage block is set to 8C (01000b).
	SC20_CR0[4:0] (Output Block)	10000	The A capacitor of the second stage block is set to 16C (10000b).
ASign	SC10_CR0[5] SC20_CR0[5]	0	The normal the switch sequencing is used for integrators, i.e. input is sampled at Φ_1 and reference is sampled at Φ_2.
FCap	SC10_CR0[7] SC20_CR0[7]	1	Both feedback capacitors are fixed to 32C.

Table 32 Control Register Settings for Delta Sigma ADC 2nd order Modulator Part 2

Register Bits Name	Register Bits	Value (Binary)	Description
PWR	SC10_CR3[1:0] SC20_CR3[1:0]	11	PWR bits are set to 11b for full-power only during the sensor measurement process and will be set to 00b for off-mode for saving power.

Table 33 Control Register Settings for Delta Sigma ADC 2nd order Modulator Part 3

The other register bits are set to the default state. The control registers of the two modulator SC blocks should have the values as summarised in Table 34 and Table 35.

SC10(Input)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0xC8	FCap 1	ClockPhase 1	ASign 0	ACap[4:0] 0 1000				
CR1	0x20	AMux[7:5] 001			BCap[4:0] 0 0000				
CR2	0x00	AnalogBus 0	CompBus 0	AutoZero 0	0 0000				
CR3	0xE3	ARefMux[7:6] 11		FSW1 1	FSW0 0	BMux[3:2] 00		PWR[1:0] 11	

Table 34 First Stage Control Registers of Delta Sigma ADC 2nd order Modulator

SC20(Output)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0xD0	FCap 1	ClockPhase 1	ASign 0	ACap[4:0] 1 0000				
CR1	0x00	AMux[7:5] 000			BCap[4:0] 0 0000				
CR2	0x40	AnalogBus 0	CompBus 1	AutoZero 0	0 0000				
CR3	0xE3	ARefMux[7:6] 11		FSW1 1	FSW0 0	BMux[3:2] 00		PWR[1:0] 11	

Table 35 Second Stage Control Registers of Delta Sigma ADC 2nd order Modulator

The algorithm also needs to set up the key parameters for the decimator including the operation mode and decimation rate which are specified in one of the decimator control registers as shown in Table 36.

SC20(Output)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR3	0xA8	Mode[7:6] 10 – 'Full Sinc^2 Mode'		10		1	Decimation Rate[2:0] 000 – 'External Clock Source'		

Table 36 Decimator Control Registers of Delta Sigma ADC

In the delta sigma ADC, the decimator is set to work in 'full mode' in order to fully implement the $Sinc^2$ decimation filter function in hardware. The decimation rate is 128x (128x oversampling) for the 12-bit configuration. Under the full mode, the decimation rate is specified by the external clock source which is a PWM module in the design (can also be a counter or a timer). The PWM period setting needs to be four times the decimation rate minus one, so the period value should be equal to 511 (0x01FF) in this case and be contained in the data registers (DR) of the PWM module in the form as shown in Table 37.

PWM DR	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB DR	0x01	0	0	0	0	0	0	0	1
LSB DR	0xFF	1	1	1	1	1	1	1	1

Table 37 PWM period setting of 12-bit Delta Sigma ADC

Figure 87 shows the control registers monitor of the 12-bit delta sigma ADC at runtime and the test results match the theoretical analysis results discussed above.

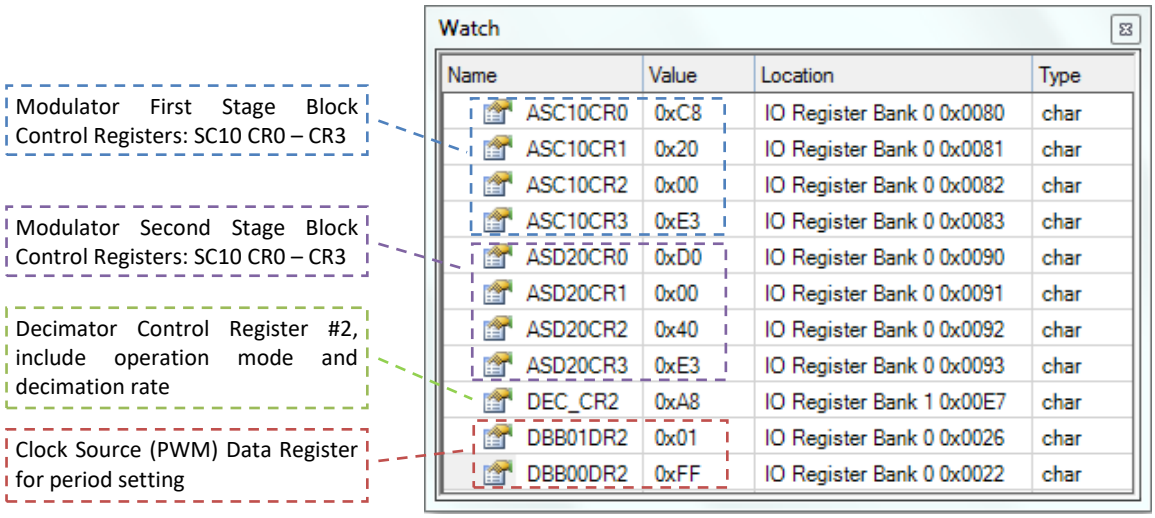


Figure 87 Delta Sigma ADC control registers in PSoC

Complete Signal Chain Tests

The analogue block control registers monitor results above show that the algorithm is able to construct and set up the function modules to build the signal chain circuit. The functionality of the complete signal chain including the data conversion routine can further be verified by the final outputs of the signal chain which are the sensor measurement results. The sensor

measurement results also show the performance of the RAWS node system. As discussed before even though the tests focused on validating the system functionalities and also the system performance can depend mainly on the selected programmable hardware platform, the performance is an inseparable aspect of the system design and it can better reflect the validity of the functionalities of the intelligent reconfiguration algorithm and the whole system. Therefore, sensor measurement results are also analysed to show the system performance.

In the C9B force sensor tests, because it is difficult to apply an accurate force to the sensor, simulated C9B sensor electric signals are fed into the RAWS system replacing the actual sensor output signals. Other sensor connections such as the TEDS output remain intact. The test setup is shown in Figure 88.

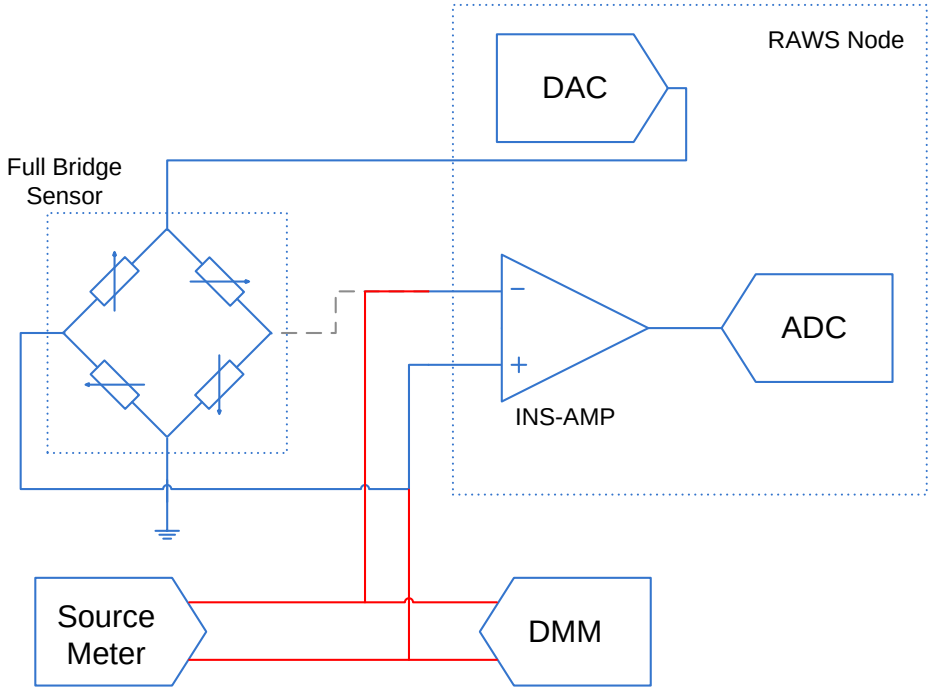


Figure 88 Compete Signal Chain Test Setup for C9B Force Sensor

The simulated sensor signal was generated by a source meter as shown in Figure 88. A 6.5-digit DMM was also included in the test setup for monitoring the simulated signal. Four different signals from 0.5mV to 2mV with a step of 0.5mV were generated to simulate forces from 10N to 40N with a step size of 10N. The system took 20 measurements of each signal. A group of the

measurement test results are presented in Table 38 (The rest of the results can be found in Appendix I). The results show that the algorithm successfully built a complete signal conditioning and processing chain to support the C9B force sensor, and with this signal chain the RAWS node is able to correctly read the sensor and acquire meaningful physical readings.

Signal (mV)	Theoretical Result (N)	RAWS Node Reading (N)									
0.516	10.32	10.13	10.28	10.39	10.61	10.26	10.13	10.52	10.36	10.14	10.46
1.009	20.18	20.33	19.82	20.07	20.25	20.37	20.22	19.96	20.31	20.22	20.13
1.508	30.16	30.26	30.38	30.22	29.85	30.13	30.32	29.99	30.36	30.14	30.10
2.012	40.24	40.43	40.07	40.14	40.12	40.25	40.38	40.21	40.29	40.18	40.31

Table 38 Complete Signal Chain Test Results of C9B Force Sensor

The measurement results show that the signal chain built by the algorithm for the C9B force sensor can reach an accuracy level of around 0.4 Newton (Max error = 0.36N) and it can meet the 0.5 Newton accuracy class of the sensor. The measurement accuracy can be affected by many factors, and the programmable hardware platform and the function modules built on it play a major role. In this case the three main function modules that comprise the signal chain are a DAC, an instrumentation amplifier, and a delta sigma ADC.

The DAC is used as the excitation source for the connected full bridge sensor. Based on the analogue subsystem and function module settings, the 9-bit DAC in the chain has a resolution of 5.1mV which is the highest resolution DAC the hardware platform can support, but its actual accuracy is less than the resolution due to converter's errors. According to the hardware specifications this DAC which is built with the generic analogue block can achieve an accuracy of around 8.98mV [245]. However, in this case the accuracy of the DAC has a very limited impact on the measurement performance of the signal chain. This is because the output of a full bridge sensor is a differential signal measured from the middle points of the two legs of the bridge circuit, whilst the excitation level affects the common-mode part of the sensor output. Therefore, in this case the amplifier and the ADC have the main influence on the measurement performance of the signal chain.

The 12-bit delta sigma ADC built by the intelligent algorithm has a 0.63mV resolution under the current analogue subsystem and function module settings. Based on the hardware settings and specifications, the ADC and three opamp instrumentation amplifier together could achieve an accuracy of around 1.28mV [245], corresponding to around 0.27 Newton of the C9B force sensor.

In this measurement test setup, the source meter that produces the simulated force signals is also a source that could introduce errors into the final results especially when the simulated signals are small. According to the instrument specifications [251], the source meter used in the tests is accurate to around 0.64mV, corresponding to around 0.13 Newton.

Based on the analysis above, the theoretical signal chain reconfiguration process and accuracy calculation results match the control registers monitor and measurement performance test results for the C9B force sensor. The results show that the RAWS node can autonomically build a complete signal conditioning and processing chain to support the C9B force sensor and the system performance can meet the sensor accuracy class requirement.

6.4.2 101-K Thermocouple Reconfiguration Process Analysis and Test Results

Replacing the C9B force sensor with the 101-K thermocouple, the sensor identification scheme detects the change and acquires the following key information for the sensor as discussed in the test results in Section 6.3.

Attribute	Value
Sensor Type	K Type Thermocouple
Physical Measurand	Temperature in degree Celsius
Measurement Range	-200 – 1260 °C
Electrical Output Range	-6mV – 51 mV (Sensitivity)

Table 39 Key Attributes of the 101-K Thermocouple

Based on the sensor information and the configuration settings already existing in the programmable hardware, the algorithm should keep the instrumentation amplifier and the delta sigma ADC and unload the DAC to form the signal chain as described and analysed in Chapter 5.

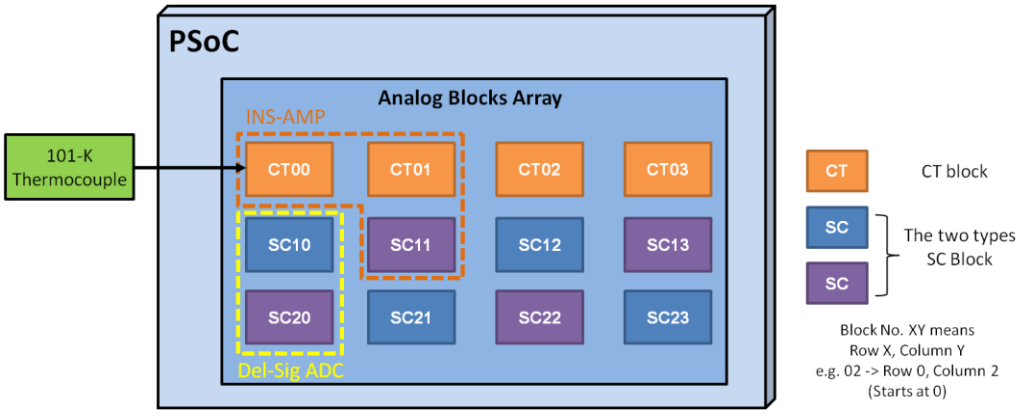


Figure 89 101-K Thermocouple signal process chain placement in PSoC

The algorithm then reselects suitable parameters for the thermocouple and the theoretical process is analysed as follows.

For the INS-AMP module, the algorithm calculates and reconfigures the gain setting. The gain setting is determined based on factors including the ADC input range and sensor output signal range. The 101-K thermocouple maximum output is 51mV. The ADC input range is decided by the analogue system voltage reference setting. In this case, the algorithm also chooses the setting that can minimise the ADC input range in order to increase the ADC measurement precision because the thermocouple generates such small signals. Therefore, the ADC input range is kept at 2.6V, the same setting as for the C9B sensor. Considering the sensor signal is offset from the analogue ground which is the halfway point of the full input range, the effective input range for the positive and negative part of the signal (relative to analogue ground) is also half of the full input range. Taking all the aforementioned factors into account, the product of the gain multiplied by the thermocouple maximum output should be less than half of the full input range. Based on this, the gain setting should be 24 in this case, where the differential gain is set to 24 and the conversion gain is set to 1. The corresponding control register settings are illustrated in Table 40 and Table 41.

Register Bits Name	Register Bits	Value (Binary)	Description
Gain	CT01_CR0[3] CT00_CR0[3]	1	The gain bit specifies whether the resistor string is connected around the opamp for gain or for loss. In this case, the bit is set to 1 for gain.
RTapMux	CT01_CR0[7:4] CT00_CR0[7:4]	0001	The combination of the RTapMux and EXGAIN specify the selection of the resistor taps from 18 taps. The 0001b and 1b combination selects the tap second to the bottom which means R_f = 46 and R_i = 2. Therefore the block gain is equal to 1+46/2 = 24.
EXGAIN	CT01_CR0[7:4] CT00_CR0[7:4]	1	

Table 40 INS-AMP CT Blocks Control Register Settings Change for 101-K Thermocouple

Register Bits Name	Register Bits	Value (Binary)	Description
ACap	SC11_CR0[4:0]	1 0000	A capacitor and B capacitor are set to 16C (10000b). F capacitor is set to 16C (0b). So the conversion stage gain is equal to 16/16 = 1.
BCap	SC11_CR1[4:0]	1 0000	
FCap	SC12_CR0[7]	0	

Table 41 INS-AMP SC Block Control Register Settings Change for 101-K Thermocouple

The algorithm should also reconfigure the ADC to 14-bit resolution for the thermocouple for better measurement accuracy. The main difference between the 14 and 12 bits delta sigma ADC is the decimation rate, which is 256x for the 14-bit configuration. This is reflected in the setting of the decimator external clock frequency, which is specified by the PWM period setting contained in the PWM data registers, and the value should be 1023(0x3FF) as shown in Table 42. The other configuration settings are the same as for the 12-bit ADC.

PWM DR	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MSB DR	0x03	0	0	0	0	0	0	1	1
LSB DR	0xFF	1	1	1	1	1	1	1	1

Table 42 PWM period setting of 14-bit Delta Sigma ADC

So for the 101-K thermocouple, the control registers for the function module in the signal processing chain should have the values shown below based on the theoretical analysis above.

Function Module	Block	Control Register	Value
INS-AMP	NonInv & INV input stage: Two CT blocks CT00 & CT01	CT00_CR0, CT01_CR0	0x1C
		CT00_CR1, CT01_CR1	0x21
		CT00_CR1, CT01_CR1	0x23
		CT00_CR1, CT01_CR1	0x03
	Conversion Stage: SC block SC11	SC11_CR0	0x10
		SC11_CR1	0x10
		SC11_CR2	0x20
		SC11_CR3	0x3B
14-bit Delta Sigma ADC	2nd order Modulator: Two SC block SC10 & SC20	SC10_CR0	0xC8
		SC10_CR1	0x20
		SC10_CR2	0x00
		SC10_CR3	0xE3
		SC10_CR0	0xD0
		SC10_CR1	0x00
		SC10_CR2	0x40
		SC10_CR3	0xE3
	Decimator	DEC_CR2	0xA8
	PWM (Clock Source) Period Setting	MSB	0x03
		LSB	0xFF

Table 43 Function Modules settings for 101-K Thermocouple

Figure 90 to Figure 92 show the control registers monitor of the function modules in the signal chain at runtime. As illustrated in the figures, the test results show the algorithm successfully reconfigured the hardware to adapt to the 101-K thermocouple.

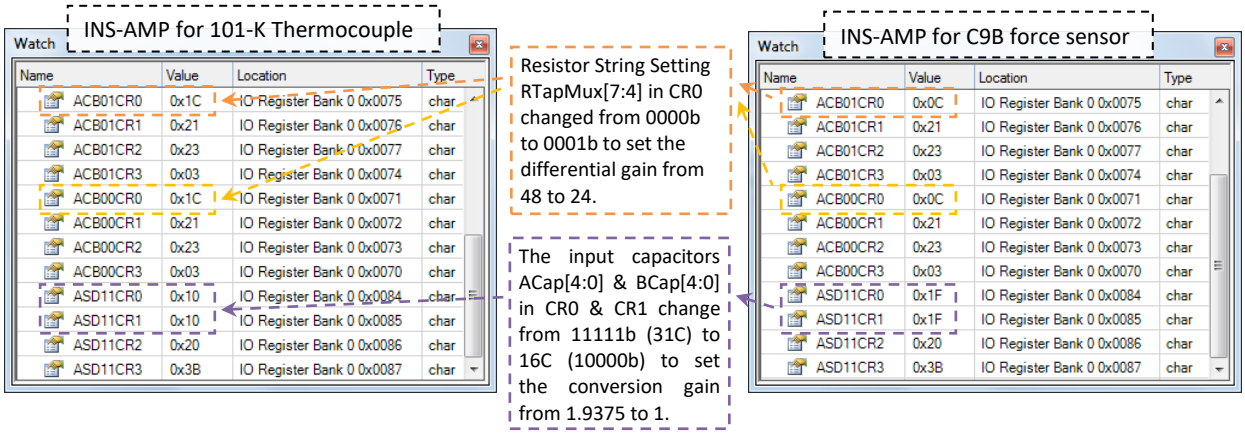


Figure 90 INS-AMP control registers change in PSoC for 101-K thermocouple

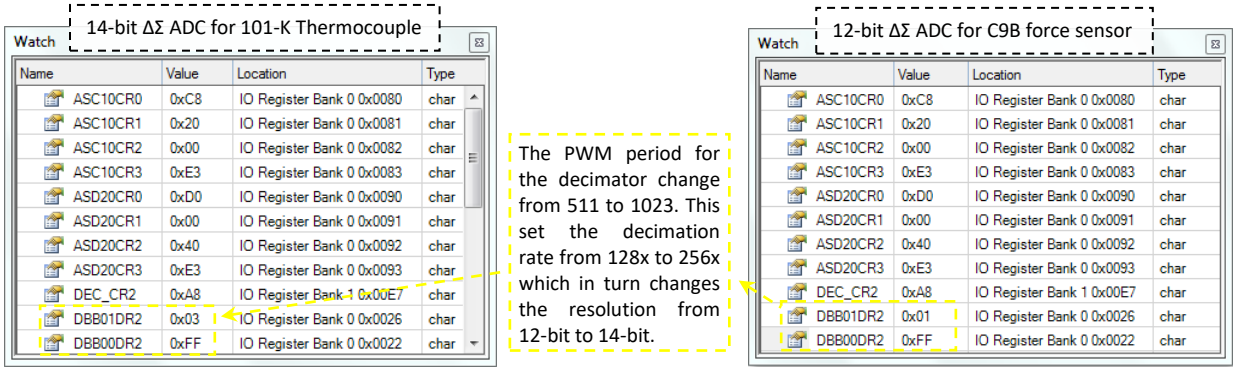


Figure 91 Delta Sigma ADC control registers change in PSoC for 101-K thermocouple

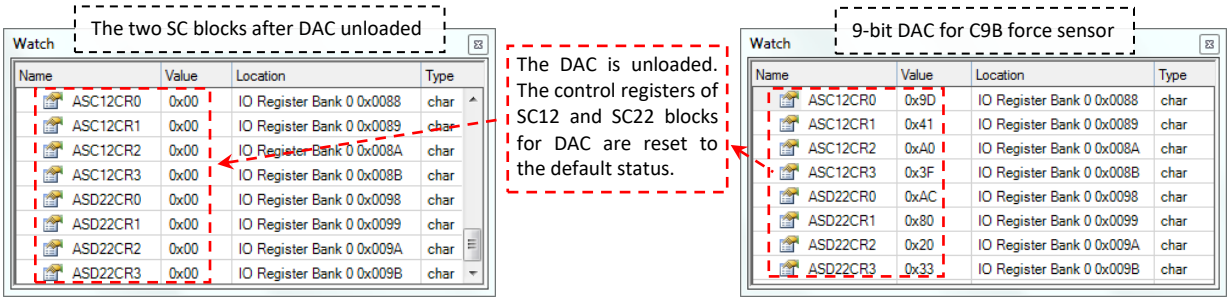


Figure 92 Delta Sigma ADC control registers change in PSoC for 101-K thermocouple

Similar to the tests with the C9B force sensor, the complete signal conditioning and processing chain built by the algorithm was further verified by the final data conversion results as shown in Table 44 (Table 44 shows a group of the results. The rest results are similar and can be found in

Appendix I). The system measured the room temperature during the tests, and a commercial digital thermometer of ±0.2°C accuracy was used as reference. The system took 30 measurements in around every 30s, and the readings of the reference thermometer were recorded at the same time.

101-K Thermocouple (°C)	Reference Thermometer (°C)
17.66	17.6
17.57	17.6
17.52	17.5
17.45	17.4
17.34	17.4
17.29	17.4
17.35	17.4
17.52	17.5
17.63	17.6
17.70	17.6

Table 44 Data Conversion Test Results of 101-K Thermocouple

The measurement results show that the RAWS system with the 101-K thermocouple is accurate to less than 0.15°C (Max Error = 0.11°C) with respect to the reference thermometer, and it is able to meet the 1.1°C accuracy class of the connected 101-K thermocouple [252] [253]. In this case, the two main function modules built by the intelligent algorithm are the three opamp instrumentation amplifier and 14-bit delta sigma ADC. Based on the system settings and hardware specifications, the 14-bit delta sigma ADC has a resolution of around 0.16mV and could achieve an accuracy of around 0.34mV [245], corresponding to around 0.1°C for the 101-K thermocouple. The theoretical measurement performance analysis matches the test results. The measurement results show that the algorithm successfully built the signal chain for the 101-K thermocouple and could correctly take measurements and convert the sensor outputs.

6.4.3 101-10B RTD Reconfiguration Process Analysis and Test Results

In the case when the 101-10B RTD is connected, the algorithm selects a PGA and an incremental ADC (INC ADC) to build the signal processing chain as discussed in Chapter 5. The placement of these two modules in the analogue block array is illustrated in Figure 93.

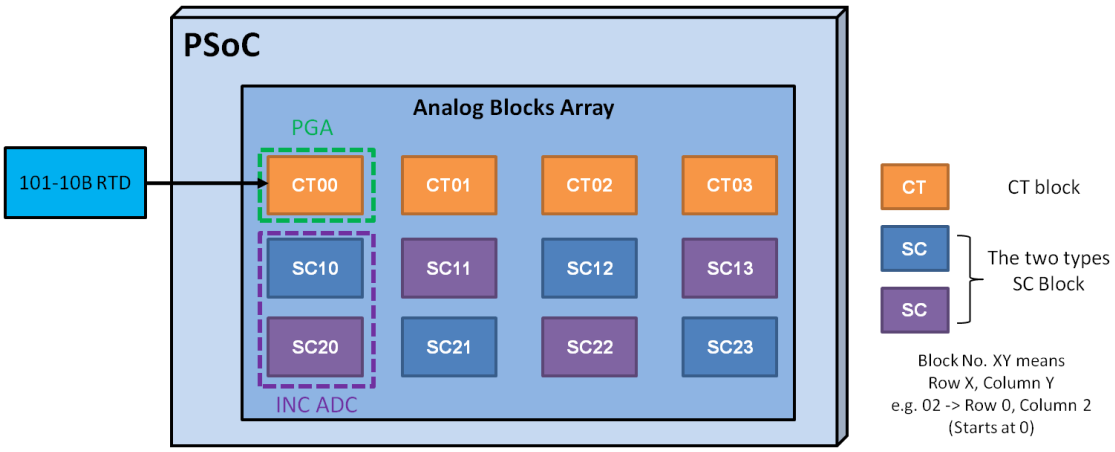


Figure 93 101-K Thermocouple signal process chain placement in PSoC

The PGA is constructed with one CT block with the configuration shown in Figure 94. It works as a buffer stage in the chain with a gain of 1. This is because the excitation voltages applied to the resistors are the reference voltage from PSoC so no signal amplification is needed.

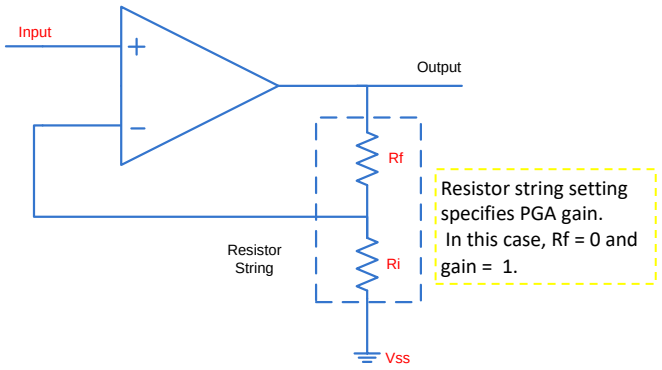


Figure 94 PGA Configuration

This configuration fully determines the control registers of the PGA block, CT00 block, as illustrated in Table 45 and Table 46.

CT Register Bits Name	Register Bits	Value (Binary)	Description
PMux	CT00_CR1[2:0]	001	The positive input of the opamp is connected to port inputs, i.e. input signals from GPIOs through the analogue subsystem's input multiplexer. Bits value 001b select the port inputs option.
NMux	CT00_CR1[2:0]	100	The negative input of the opamp is connected to feedback point of the opamp circuit. 100b select the feedback option.
INSAMP	CT00_CR3[1]	0	Not building an INSAMP so this bit is 0.
RBotMux	CT00_CR0[1:0]	10	RBotMux bits control the resistor string bottom multiplexer when INSAMP bit is disabled. In this case, the input reference is set to V_{SS} (10b).
CMOut	CT00_CR3[1]	0	Common mode output connection is not used so it is disabled for both blocks.
AnalogBus	CT00_CR1[7]	0	Analogue bus output connection is disabled for both blocks.
CompBus	CT00_CR1[6]	0	Comparator output connection is disabled for both blocks.

Table 45 CT Block Control Register Settings for PGA Part1

Register Bits Name	Register Bits	Value (Binary)	Description
Gain	CT00_CR0[3]	1	The gain bit specifies whether the resistor string is connected around the opamp for gain or for loss. In this case, the bit is set to 1 for gain.
RTapMux	CT00_CR0[7:4]	1111	The combination of the RTapMux and EXGAIN specify the selection of the resistor taps from 18 taps. The 1111b and 0b combination selects the top tap which means R_f = 0 and R_i = 48, and this set the block gain to 1 (1+0/48).
EXGAIN	CT00_CR0[7:4]	0	

Table 46 CT Blocks Control Register Settings for PGA Part2

Based on the analysis above, the control registers of the PGA block should have the values as summarised in Table 47.

CT00	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0xFE	RTapMux[7:4] 1111				Gain 1	1	RBotMux[1:0] 10	
CR1	0x21	AnalogBus 0	CompBus 0	NMux[5:3] 100		PMux[2:0] 001			
CR2	0x23	0	0	1	0	00		PWR[1:0] 11	
CR3	0x00					0	CMOut 0	INSAMP 0	EXGAIN 0

Table 47 PGA Control Registers

Figure 95 shows the control registers monitor of the PGA block (CT00 block) at runtime and the test results match the theoretical analysis results discussed above.

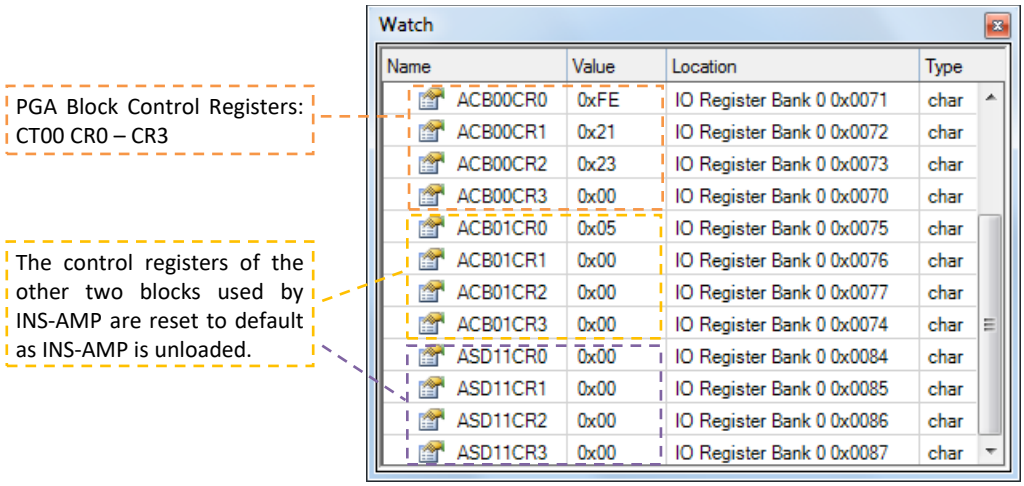


Figure 95 PGA control registers in PSoC for 101-10B RTD

The incremental ADC also includes a modulator and a decimator. The modulator can be implemented as first order or second order for the INC-ADC. While the second order type takes one more SC block, it can improve the signal-to-noise ratio (SNR) so the second order type is constructed. During the conversion process, the modulator of INC-ADC has the same configuration as the modulator of the delta sigma ADC shown in Figure 86 in subsection 6.4.1, i.e. it is essentially an integrator circuit. But the INC-ADC modulator is reset before each sample by controlling switches in the feedback path, and this is the main difference from the delta sigma ADC modulator. The modulator control registers for these two operating modes are illustrated as follows.

During the conversion process, the key settings are the same in the delta sigma ADC configuration as shown in Table 48 and Table 49. The difference here is the input of the modulator is routed to the PGA at CT00 block. Also, there is no need to swap the internal clocks Φ_1 and Φ_2 (ClockPhase bit) because the output of the PGA is continuous, while in the previous configurations the delta sigma ADC internal clocks are swapped because the output of the INS-AMP is sampled and held at Φ_2.

SC10(Input)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0x88	FCap 1	ClockPhase 0	ASign 0	ACap[4:0] 0 1000				
CR1	0x00	AMux[7:5] 000 (To CT00 of PGA)			BCap[4:0] 0 0000				
CR2	0x00	AnalogBus 0	CompBus 0	AutoZero 0	0 0000				
CR3	0xE3	ARefMux[7:6] 11		FSW1 1	FSW0 0	BMux[3:2] 00		PWR[1:0] 11	

Table 48 First Stage Control Registers of Incremental ADC 2nd order Modulator during Conversion

SC20(Output)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0x90	FCap 1	ClockPhase 0	ASign 0	ACap[4:0] 1 0000				
CR1	0x00	AMux[7:5] 000			BCap[4:0] 0 0000				
CR2	0x40	AnalogBus 0	CompBus 1	AutoZero 0	0 0000				
CR3	0xE3	ARefMux[7:6] 11		FSW1 1	FSW0 0	BMux[3:2] 00		PWR[1:0] 11	

Table 49 Second Stage Control Registers of Incremental ADC 2nd order Modulator during Conversion

Before conversion, the AutoZero and FSW0 bits are set to 1 to reset the integrator circuit, and the control registers settings are illustrated in Table 50 and Table 51.

SC10(Input)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0x88	FCap 1	ClockPhase 0	ASign 0	ACap[4:0] 0 1000				
CR1	0x00	AMux[7:5] 000 (To CT00 of PGA)			BCap[4:0] 0 0000				
CR2	0x20	AnalogBus 0	CompBus 0	AutoZero 1	0 0000				
CR3	0xF3	ARefMux[7:6] 11		FSW1 1	FSW0 1	BMux[3:2] 00		PWR[1:0] 11	

Table 50 First Stage Control Registers of Incremental ADC 2nd order Modulator before Conversion

SC20(Output)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR0	0x90	FCap 1	ClockPhase 0	ASign 0	ACap[4:0] 1 0000				
CR1	0x00	AMux[7:5] 000			BCap[4:0] 0 0000				
CR2	0x60	AnalogBus 0	CompBus 1	AutoZero 1	0 0000				
CR3	0xF3	ARefMux[7:6] 11		FSW1 1	FSW0 1	BMux[3:2] 00		PWR[1:0] 11	

Table 51 Second Stage Control Registers of Incremental ADC 2nd order Modulator before Conversion

Beside the modulator, the decimator is set to the incremental mode for the INC ADC and the control register is illustrated in Table 52.

SC20(Output)	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CR3	0x48	Mode[7:6] 01 – 'Incremental Mode'		00		1	000		

Table 52 Decimator Control Registers of INC ADC

Figure 96 shows the control registers monitor of the incremental ADC at the two different operating modes and the test results match the theoretical analysis results discussed above.

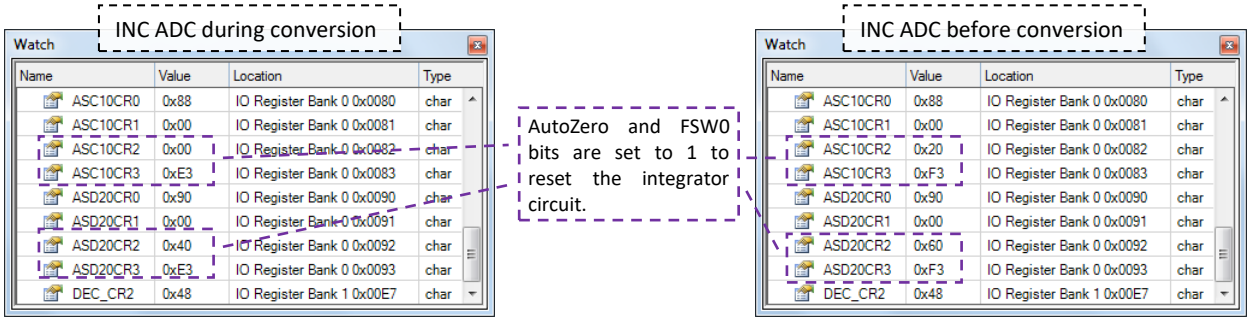


Figure 96 INC ADC control registers in PSoC for 101-10B thermocouple

The control registers monitor results show the algorithm is able to successfully reconfigure the hardware to adapt to the 101-10B RTD sensor.

The complete signal chain was also verified by the system measurement performance tests and the signal chain final output results are shown in Table 53 (Table 53 shows a group of the results. The rest results are similar and can be found in Appendix I). Same as for the 101-K thermocouple tests, the system also took the room temperature measurements 30 times approximately every 30s with the same commercial digital thermometer as reference.

The measurement test results show that the system with the 101-10B RTD sensor can be accurate to less than 0.15°C (Max Error = 0.12°C) with respect to the reference thermometer. It can meet the 0.3°C accuracy class of the connected 101-10B RTD sensor at 0 to 100°C range [254] [255]. In this case the two main function modules of the signal chain are the PGA and 14-bit incremental ADC. An accuracy level of around 0.24mV can be achieved based on the system settings and hardware specifications [245], corresponding to around 0.1°C for the 101-10B RTD sensor in the normal room temperature range. The accuracy classes of an RTD sensor differ at different temperature ranges. The accuracy classes for the 101-10B RTD reduce to 0.8°C, 1.3°C and 1.8°C for 100 to 200°C, 200 to 300°C, and over 300°C temperature ranges respectively [254] [255]. The system's equivalent accuracy levels also drop to around 0.24°C, 0.30°C, and bigger than 0.37°C respectively based on the theoretical analysis under the current hardware setup [245] [254]. But according to the theoretical analysis the system accuracy levels in these temperature ranges are still more than capable of meeting the RTD accuracy class requirements.

The 101-10B RTD sensor has a much slower response time than the reference thermometer, around 5s for the RTD versus less than 0.2s for the reference thermometer [254] [255] [256]. This is also reflected in the test results. When the room temperature is gradually increasing during the performance tests, it can be noticed that overall the RTD sensor's readings is behind the reference thermometer's.

In summary, the theoretical analysis described above matches the 101-10B RTD sensor test results. The results show that the algorithm successfully built the signal chain to support the RTD sensor and obtain sensor readings.

101B-10 RTD Sensor (°C)	Reference Thermometer (°C)
18.98	19.1
19.02	19.0
19.05	19.1
19.17	19.2
19.24	19.2
19.26	19.3
19.36	19.3
19.41	19.3
19.48	19.4
19.42	19.3

Table 53 Data Conversion Test Results of 101-B-10 RTD Sensor

6.4.4 Adaptive Reconfiguration Test Summary

The RAWS node intelligent algorithm was tested to verify that it is able to autonomically reconfigure the system hardware and build signal processing chains to support different sensors. The test mainly includes two parts for evaluating the intelligent algorithm, the hardware reconfiguration process monitoring and the complete signal chain performance validation.

In the tests, the hardware reconfiguration process was examined by monitoring the control registers of the programmable analogue system. The control registers monitor results show that the intelligent algorithm is able to perform hardware reconfigurations according to the design logic. It can select and construct analogue function modules and set up parameters to build a signal processing chain based on the connected sensor's attributes without any manual intervention.

The functionality and performance of the signal chains built by the algorithm were further verified by the measurement performance tests. The performance analysis and test results also show that the intelligent algorithm is able to successfully reconfigure the RAWS node hardware to build signal chains to adapt to the connected sensors, and the signal chains built by the algorithm can achieve performance levels that meet the accuracy requirements of the different

sensors used in the projects. However, it also became clear during the tests that the capability and performance of the RAWS node are limited by its programmable hardware platform. As discussed in Chapter 2, although the PSoC platform fits well for the proof-of-concept of the RAWS node research, it has limitations in terms of the analogue capabilities and specifications, which can compromise the applicability of the system in some applications such as the ones that have very high precision and performance requirements. Also, a small number of sensors are included and tested in the research. Even though the theoretical analysis shows that the system can generally handle different types of sensors, the only way to actually verify this is to test a large number of sensors.

In summary, the adaptive reconfiguration test results show that the RAWS system can autonomically reconfigure its hardware to adapt to different sensors and can correctly take the sensor measurements and acquire meaningful physical readings with adequate accuracy level. The key technique of the RAWS node research, the intelligent adaptive reconfiguration algorithm, is able to combine the reconfigurability of programmable hardware and the sensor identification scheme together to realise the autonomic multi-sensing capability.

6.5 Power Consumption Test

The system power consumption is tested by measuring the system current during different working modes and stages of the RAWS node. The system has two modes which are the sleep mode and active mode, and the active mode contains five main working stages including the system start-up stage (wireless initialisation stage), sensor identification stage, sensor reading stage, and wireless data transmission stage. The current measurement results of these modes and stages are discussed in this section. Dynamic energy and power management features have been incorporated into the RAWS firmware for each stage and these features are also reflected in the test results.

Sleep Current

The sleep current measurement results are shown in Table 54.

Overall System Sleep Current	PSoC Sleep Current	ZigBee Transceiver Sleep Current
32 ~ 34 uA	4.8 ~ 4.9 uA	28 ~ 29 uA

Table 54 RAWS Node System Sleep Current (at around 22°C and 3.3V)

As shown in the results, the main power consuming component during the sleep mode is the XBee ZigBee transceiver. This is the trade-off for the ease-of-use feature of the XBee module and it can be improved with a wireless transceiver with ultra low sleep current which could be as low as a few micro-amps.

Wireless Initialisation Stage Current

In the system start-up stage, the main task of the node is to initialise the wireless transceiver and join the RAWS network. Therefore, this stage is referred to as wireless initialisation stage here. The current measurement results of this stage are illustrated in Figure 97.

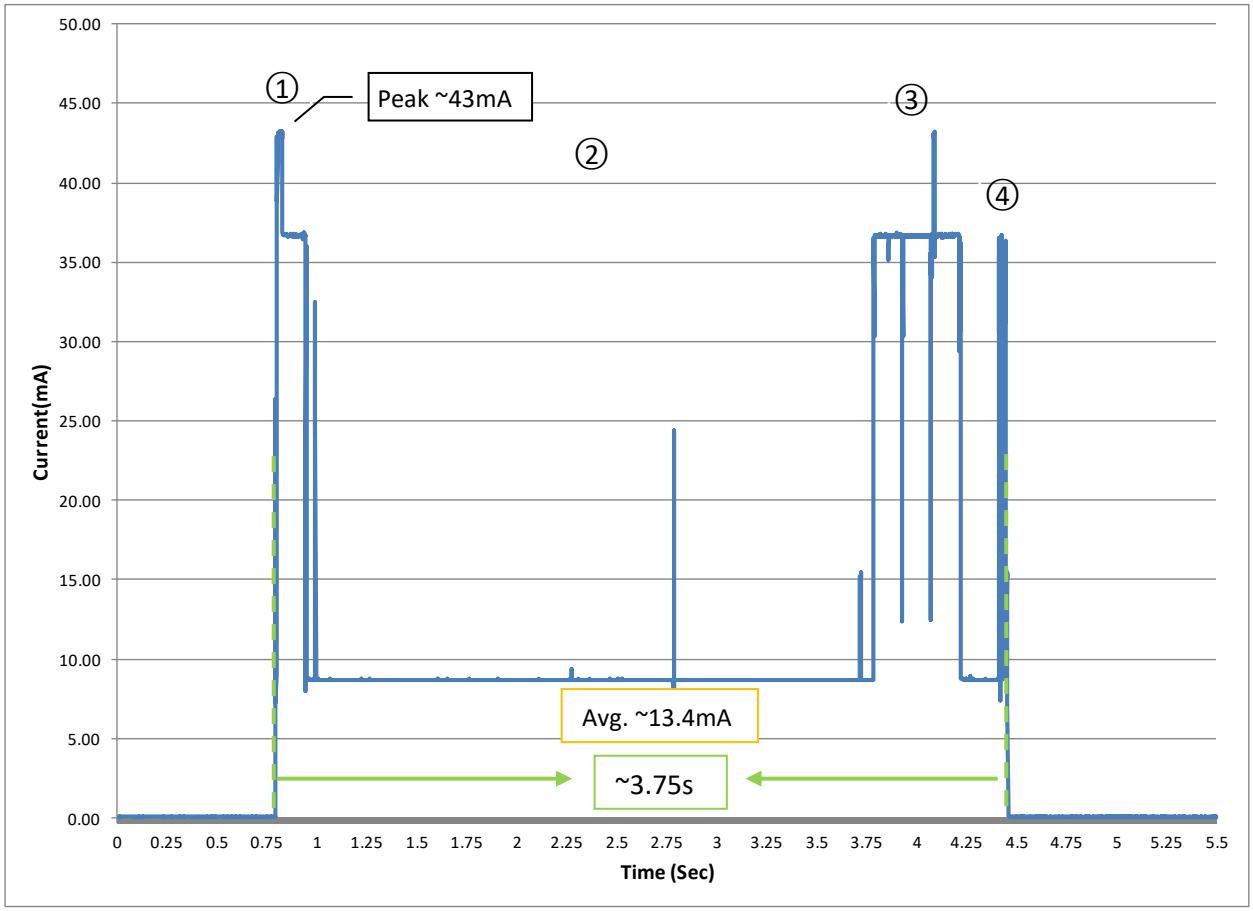


Figure 97 Wireless Initialisation Stage Current

The initialisation stage takes about 3.75 seconds and can be divided into four phases as shown in the figure above, and the current consumption of each phase is explained as follows:

①. The microcontroller starts the wireless initialisation procedure with sending the AT Command API frames to the ZigBee module to set up the network parameters. The PSoC microcontroller and the ZigBee transceiver module are both powered on at this phase. The radio of the wireless module is on at this phase because the radio will be turned on at power-up. Therefore, the current is around 43mA of which the PSoC consumes about 6mA and the ZigBee transceiver takes around 37mA. This is the also peak current of all the system active stages. This phase lasts around 26ms.

②. The PSoC goes into sleep mode once it successfully sends out the API frames to the transceiver module in phase ①. In this phase the PSoC is asleep for most of the time. The ZigBee module processes the frames with the new network parameters, and then it turns its radio off and prepares for the channel scan with the new parameters. This preparation period takes about 2.9s. The radio is off during this period, so the transceiver is working at a relatively low power mode.

③. The ZigBee module performs the channel scan in this phase and PSoC wakes up periodically to check whether the scan is finished and the network joining process has succeeded. The scan process takes about 0.5s with the network settings described in Chapter 3. During the scan, the transceiver current consumption is around 37mA. The PSoC is set to wake up in every 0.25s to check the scan status, but due to the inaccuracy of the internal low frequency sleep timer, this time interval can vary. In the test result shown in Figure 97, the PSoC wake-up period is about 0.38s.

④. Once the microcontroller detects that the node has successfully joined the network, it puts the transceiver into sleep. The system then goes into the next stage which is the sensor identification stage.

Sensor Identification Stage

In the sensor identification stage, the system retrieves and parses TEDS data from the connected sensor and extracts the sensor identification and characteristic information. Figure 98 shows an example of the current measurement results of this stage with the C9B force sensor connected to the system. Other sensors generate very similar results.

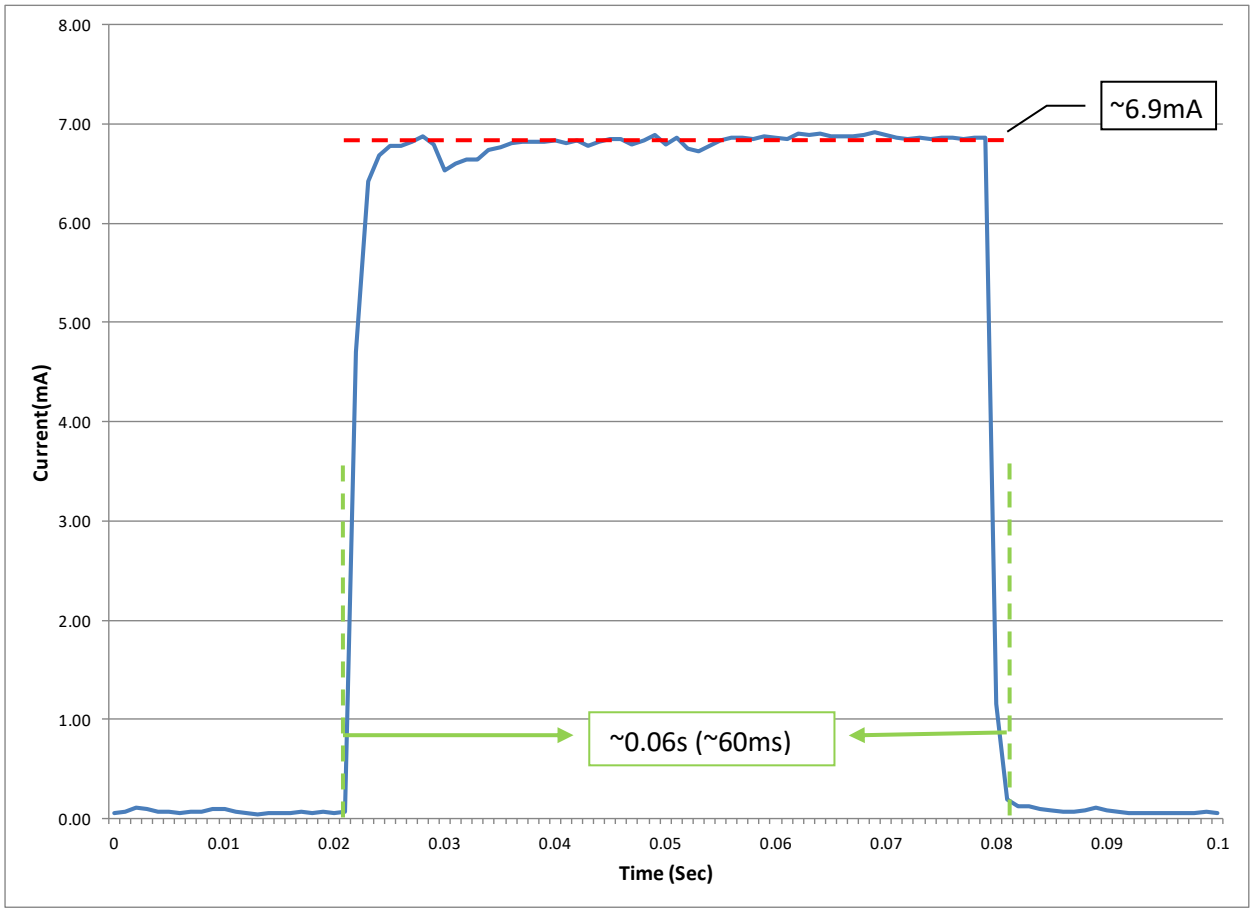


Figure 98 Sensor Identification Stage Current

In this stage, only the PSoC is active whilst the wireless transceiver is in sleep mode. The system current is less than 7mA. This stage takes around 60ms. Even though the TEDS parsing algorithm has a complicated structure and a lot of bitwise operations, most of this time is spent on 1-Wire communications which is not designed to be high data rate.

Sensor Reading Stage

In the sensor reading stage, the system firstly reconfigures itself to adapt to the connected sensor based on the acquired sensor information, and then it reads the sensor and converts the raw data into physical readings. In this stage, only the PSoC is active but the power consumption can vary greatly depending on the connected sensor. Figure 99 shows an example of the current waveform of the C9B force sensor measurement period.

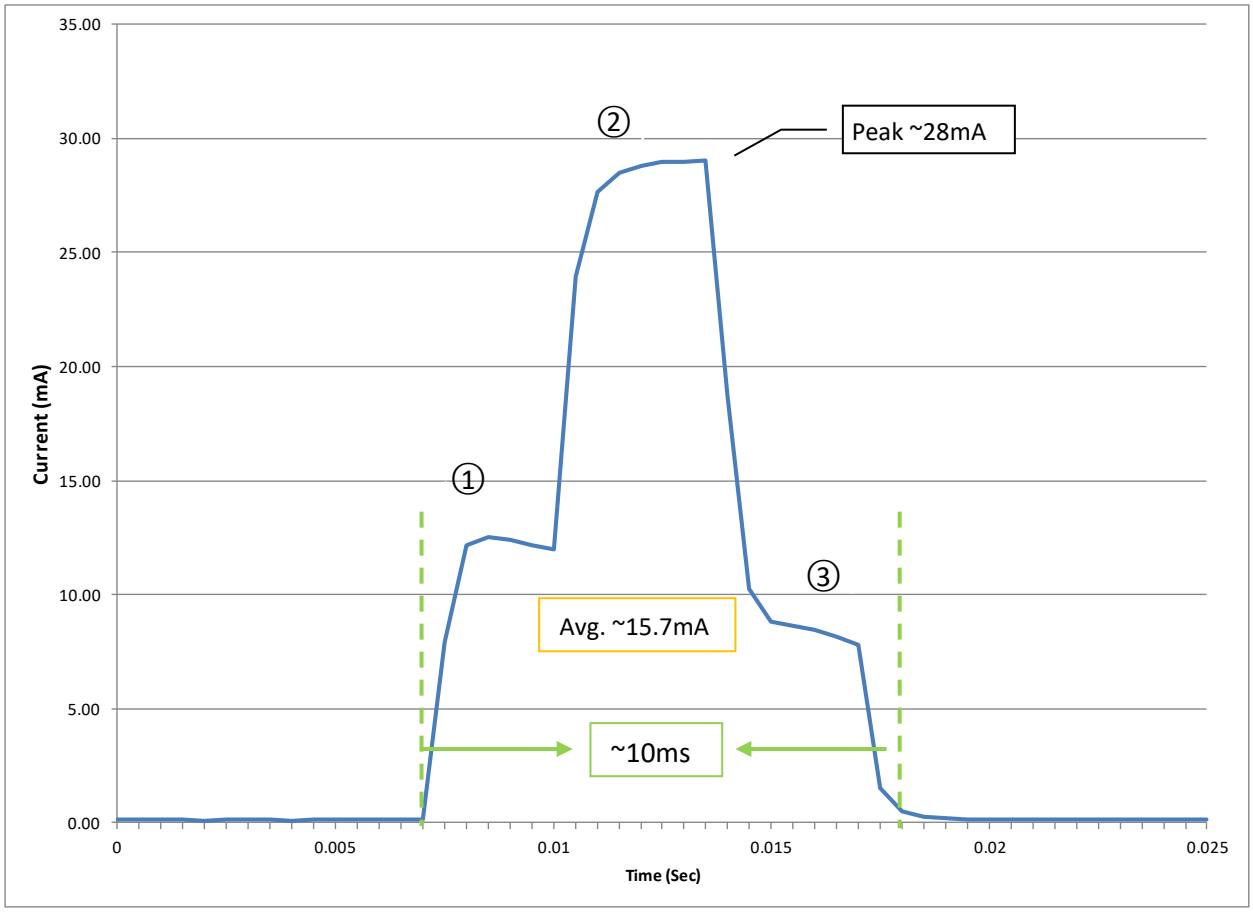


Figure 99 Sensor Reading Stage Current – C9B Force Sensor

As shown in the figure above, the whole C9B sensor measurement period can be divided into three phases. At phase ①, the system firstly builds function modules for signal conditioning and processing and sets parameters for these modules. For the bridge type of sensors, there are many characteristics to be considered, and the adaptive reconfiguration algorithm has to perform many different calculations for building the modules and setting up parameters. For

bridge sensor this phase takes about 2.5ms. Then at phase ② the function modules are all powered on to perform the actual sensor measurement procedure which takes about 4.5ms. The peak current of this stage occurs at this phase. As the system has to provide excitation voltage for bridge sensors, the current goes up to around 28mA. After reading the sensor, all the function modules are powered off in phase ③ and the CPU performs data conversions. Overall, the C9B sensor reading takes about 11ms and the average current is about 15.7mA.

Another example of the current consumption of this stage with 101-K thermocouple connected to the system is shown in Figure 100 below.

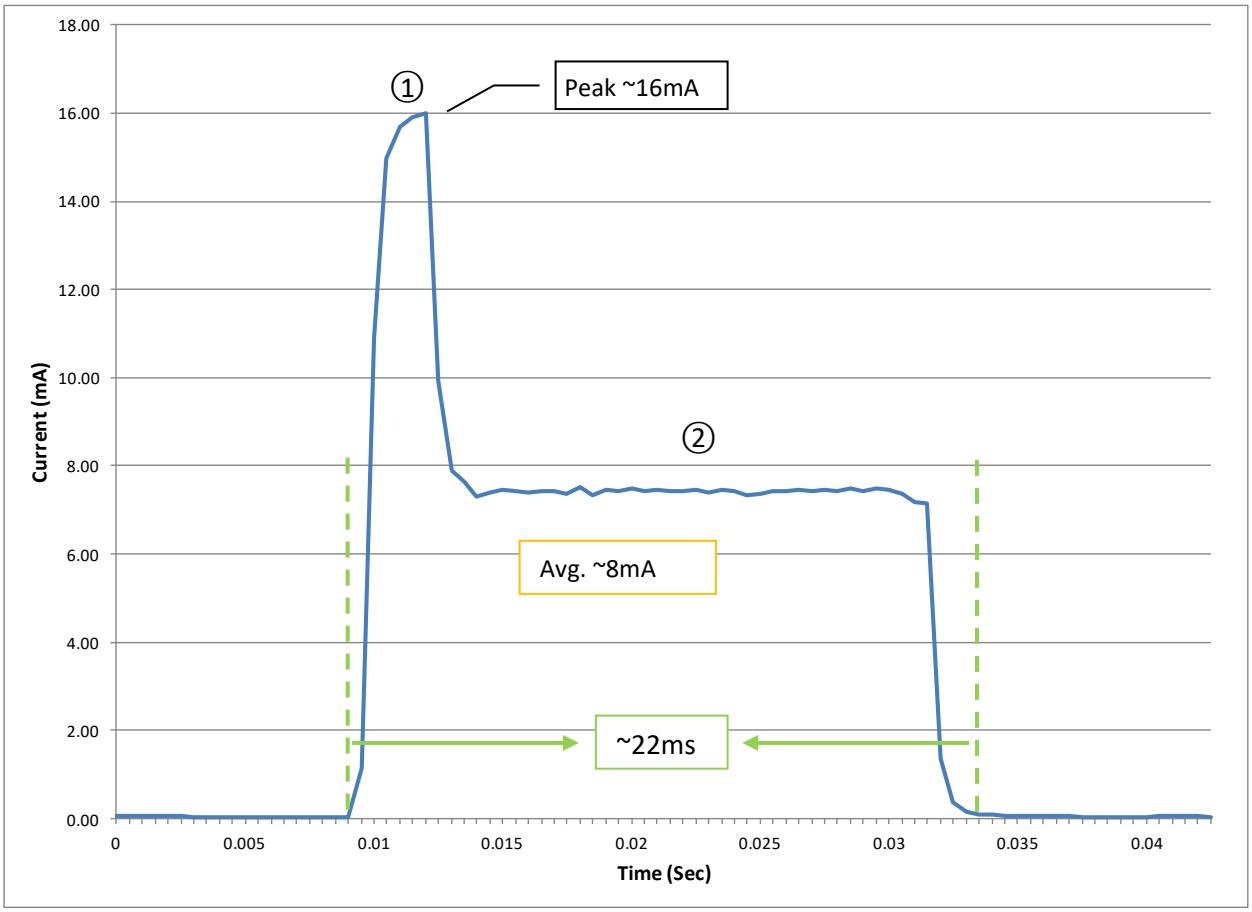


Figure 100 Sensor Reading Stage Current – 101-K Thermocouple

In this example, the hardware reconfiguration phase is much shorter compared to the same phase with the C9B force sensor discussed above, and is difficult to separate from the sensor measurement phase (Phase ① in Figure 100). This is because the algorithm does not need to

perform many calculations when preparing the signal processing chain for thermocouples because there are not too many factors needed to be taken into consideration as thermocouples are standardised. Also, the peak current is much lower because thermocouples do not require excitation power from the system. However, the data conversion phase (Phase ② in Figure 100) is much longer than the one for bridge sensors due to the voltage-temperature characteristics of thermocouples. So for thermocouples, the overall sensor measurement period is longer than the bridge sensors' but the average current is still lower.

Therefore, the system power consumption of the measurement stage is sensor dependent. In the tests, the length of this stage can last from several milliseconds to tens of milliseconds, and the average current is less than 17mA.

Wireless Transmission Stage

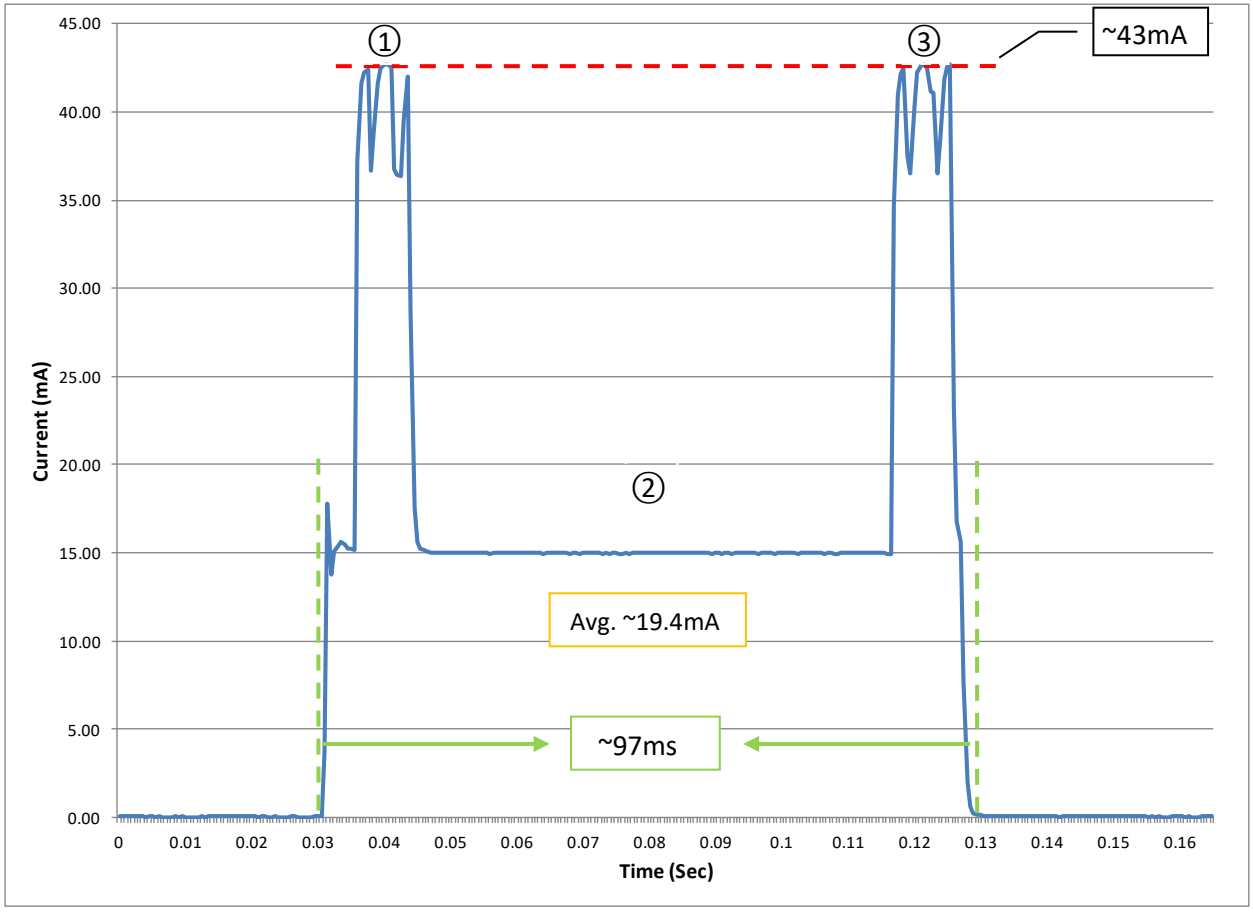


Figure 101 Wireless Transmission Stage Current

The wireless data transmission stage current waveform is shown in Figure 101. In this stage, the PSoC microcontroller and the ZigBee transceiver module are both active. At phase ①, the PSoC wakes up the ZigBee transceiver. After it wakes up, the ZigBee end device sends a poll transmission to its parent which is the network coordinator/gateway in the RAWS network. One of the purposes of these poll transmissions is to find out whether it is still in a valid network. The poll transmission takes about 8ms. In the meantime, the PSoC sends the ZigBee transmit request frame with sensor data to the transceiver for wireless transmission. At phase ②, the transceiver processes the request frame and prepares for ZigBee transmission which takes about 80ms in this example. The radio is off at phase ②. Then at phase ③, the data is wirelessly transferred to the network gateway, and this process takes about 9ms and the peak current is around 43mA. Once the transmission has completed, the transceiver module responds to the microcontroller with a transmit status frame and then the system goes back into sleep mode. Overall, the length of this stage is around 97ms and the average current is about 19.4mA.

Power Consumption Tests Summary

The test results show that during the normal operating stages, including the sensor identification stage, sensor reading stage, and wireless transmission stage, the system current is around 8 to 20mA. The power consumption of the RAWS system in active mode is in the comparable range of many other traditional WSN platforms featuring low power [28] [34] [41] [208] [202]. These low power WSN platforms also feature typical active system current of around 8-30mA, but they do not have the ability to adaptively and autonomically support different sensors. In terms of the sleep mode, the RAWS system current is around 32 to 34uA. This sleep current is also comparable to or lower than some of the typical traditional WSN platforms [28] [41] [208]. But as shown in Table 54, this sleep current can be greatly reduced because most of it is consumed by the XBee ZigBee module which can be replaced by a lower power transceiver module.

In summary, the RAWS node prototypes meet the low power requirement of WSNs. The design has achieved the main research goal without compromising the power consumption of wireless

sensor nodes. The overall system power consumption is at a comparable level to other low power WSN platforms that have been deployed in many different applications. However, the power consumption can be even further reduced.

Chapter 7 Conclusion

This research project sets out to solve the multi-sensing problem in the WSN area. As WSN applications were fast expanding, especially with the rapid growth of IoT technology, it is becoming more common that WSN node platforms are required to perform many different sensing tasks across different applications, i.e. multi-sensing capability. Also, it is common that WSN nodes need to support different analogue sensors as these sensors are still the dominant sensor kind in use. However, they vary greatly in characteristics and usually generate non-standardised output signals, making multi-sensing capability a challenge for WSNs.

The literature review found that traditional wireless sensor node technologies lack the multi-sensing capability required to flexibly support a large number of different types of analogue sensors. The traditional WSN technologies usually took the direct approach to support different analogue sensors by simply adding different signal processing chains for each individual sensor. This approach does not allow WSNs to flexibly support different analogue sensors and it also has significant shortcomings such as high system complexity and high power which are incompatible with the inherent requirements of WSNs. Therefore, there is a need for a new WSN node that has the infrastructure and ability to interchangeably work with different analogue sensors to support various WSN applications.

This brings out the key research question for the RAWS research project, i.e. how to develop a new WSN sensor node technology that is able to support different types of analogue sensors in a flexible and scalable way and more importantly in an autonomic manner, while minimising compromises to the inherent low power requirement.

This research has designed the RAWS technology to answer the main research question. The main feature of the RAWS technology is that it is able to adaptively and autonomically reconfigure itself to accommodate a wide range of analogue sensors. The main feature is incorporated into the following key research concepts.

- Programmable hardware, which can be reconfigured in real-time to realise different analogue functions for signal conditioning and processing, as the enabling technology for developing the RAWS system.
- A sensor identification scheme which enables the RAWS nodes to recognise the connected sensors and provides critical parameters about the sensors.
- Adaptive reconfiguration techniques, which integrate the hardware reconfigurability with the sensor identification scheme, to autonomically reconfigure the system hardware resources to adapt to different analogue sensors.

In order to translate the research concepts into reality, the research employed mixed-signal programmable hardware with dynamic reconfigurability as the foundation stone to build the node platform. Different programmable analogue technologies were assessed including continuous time and switched capacitor technologies, and they were analysed for their advantages and disadvantages as well as how they could be deployed for signal conditioning and processing. The research reviewed and compared different commercial programmable hardware platforms and selected the PSoC for the proof-of-concept due to its high reconfigurability, high integration, and relatively low power consumption features. More importantly, the analogue subsystem of the PSoC employs a heterogeneous design and contains both CT and SC analogue blocks that can be reconfigured at runtime. Therefore, the RAWS node system is able to reconfigure its hardware resources into the different analogue functions necessary for constructing the signal chains for different sensors, dynamically adapting to diverse application requirements.

The research investigated two potential approaches for developing a sensor identification scheme for enabling the RAWS platforms to acquire important sensor information. The first approach investigated the feasibility of developing a systematic characteristics classification process of sensor identification. A sensor characteristics analysis was carried out in order to determine the important characteristics that could be utilised by the RAWS platform to identify sensors and to further develop a series of systematic tests to acquire these characteristics. The characteristics of the two major types of temperature sensors, the thermistor and

thermocouple, were examined and the analysis results indicated that it is possible to distinguish a particular set of analogue sensors under the right conditions, and the characteristics classification process could be useful in applications where a specific limited number of sensors are to be supported. However, due to the limitations of this approach, this research project investigated a second approach to develop a more general sensor identification scheme built around the TEDS (transducer electronic data sheet) concept.

The TEDS based sensor identification scheme has been implemented based on the sensor self-identification and self-description capability provided by IEEE 1451.4 TEDS. This scheme allows the RAWS node to identify a large number of analogue sensors. The research developed TEDS interface circuits and data transmission API functions and routines based on the 1-Wire communication protocol to retrieve sensor TEDS data. A TEDS parsing routine which is the core of this scheme has been developed as part of the RAWS technology software system to extract the important sensor identification and attributes information.

Three adaptive reconfiguration techniques have been developed to integrate the programmable hardware and the sensor information to enable the RAWS node to reconfigure its hardware to adapt to different sensors without any manual intervention. The intelligent algorithm technique is able to decide and establish suitable signal conditioning and processing chains with optimised performance for different sensors based on the sensor attributes. It is able to reconfigure the hardware resources and construct different types of analogue function modules required in the signal chain. This technique can be deployed to support a large number of sensors. It can also generate and store hardware configuration settings which can be reused by the other two techniques, reducing computing overhead. The local and remote configuration settings techniques are able to search and load suitable hardware configurations based on the sensor identification information. The hardware configuration settings can be flexibly updated. These two techniques are straightforward ways to perform reconfiguration according to the connected sensors.

Four analogue sensors including a force sensor, a liquid and gas pressure sensor, a thermocouple and an RTD sensor have been used to assist and verify the system design. The

test results show that the identification scheme can successfully acquire information from different connected sensors. The tests results also indicate that the adaptive reconfiguration techniques are able to successfully carry out the hardware reconfigurations and adapt to different sensors as well as convert the sensor output signal into meaningful physical readings, and the system performance is able to sensor accuracy requirements. The results show that the main objective of the research has been achieved, i.e. the RAWS node platform is able to adaptively reconfigure itself to support different analogue sensors in a totally independent and autonomic manner.

The wireless capability of the RAWS node has been developed based on the ZigBee communication protocol. The RAWS network with star topology as the wireless testbed has also been established. A network gateway including the terminal software has been developed to support the wireless functions of the RAWS node. It serves as a data sink that collects the sensor readings from RAWS nodes. It also works as a hardware configuration settings database that supports the remote configuration settings transmission. The test results show the wireless functionalities of the RAWS nodes have been successfully implemented.

The test results also show that, by utilising techniques including dynamic power management, the overall RAWS system power consumption has been kept at a comparable level to other low power WSN.

Overall, the research concepts have been demonstrated through the design, development and testing of the RAWS node technology.

The original contribution of this research is a new wireless sensor node technology capable of autonomically adapting itself to support different analogue sensors with optimised performance in a flexible and scalable way while remaining low power. The RAWS node technology offers a number of novel features and advantages to wireless sensor networks including:

- High Flexibility – enable WSNs to adaptively support different sensors thus handle a variety of sensing tasks

- Improved Applicability – enable WSNs to be deployed in a range of diverse applications and to be easily reused in different applications without redesigning the system
- High Scalability – allow WSNs to be upgraded and expanded easily to meet new requirements

These features, as well as the low power, small size and low cost features, make RAWS technology widely applicable with the potential to benefit many WSN applications such as structural health monitoring, industrial monitoring, sustainable energy systems, medical applications, IoT applications, to name but a few.

However, there are some limitations to the RAWS technology. At the moment the capability and performance of the RAWS node is limited by the PSoC platform. Even though the PSoC was a suitable hardware for the proof-of-concept, the types of analogue function modules that can be built on it are still limited. For example, with the analogue hardware resources of the PSoC device currently used for prototyping, a transimpedance amplifier module can be difficult to realise so this device is not an ideal platform to handle sensors with current outputs. Also, for some of the function modules the specifications are limited, such as the ADCs as previously discussed in Chapter 2.

The system in theory can support a large number of sensors. However, the number of sensors included in the research is still small. More sensor types are needed to be supported and tested. Also, the signal chain built by the adaptive reconfiguration algorithm currently only contains the necessary function modules and the algorithm is not refined to cope with the different specific characteristics of different sensors.

The wireless network testbed built in the research is a simple star topology network. More complicated network topologies are not included in the system. Also, the wireless communication functions implemented are basic and only based on one wireless protocol.

For the proof-of-concept, the power consumption was compromised in some parts of the design to trade for other features such as ease-of-use, especially the wireless part. This should be improved to achieve greater energy efficiency.

Therefore, the aspects of the system discussed above can be further improved and they can be tasks for future work. A more powerful hardware platform with better specification and flexibility can be built to improve the system's ability to handle more diverse types of sensor and further increase the applicability of the system. Designing a highly reconfigurable custom IC suited for the research objectives could be a potential solution. This custom IC platform ideally should integrate a highly flexible analogue system with both CT and SC blocks, which allow different functions to be flexibly built to provide support for a large number of different sensors, i.e. a programmable system tailored for adaptive and autonomic multi-sensing capability. This custom IC could also feature other important components to meet the key requirements of WSNs, such as a low power CPU subsystem, common digital peripheral functions or a small array of digital configurable blocks, and an RF communication subsystem.

The adaptive reconfiguration techniques can be further improved, especially the intelligent algorithm. With the support of a more powerful hardware platform, the algorithm can be enhanced in order to be able to create more complicated signal conditioning and processing chains to improve the measurement precision and meet the diverse characteristics and requirements of different analogue sensors.

Advanced networking and communication functions can be developed in the future to improve the wireless capabilities of the RAWS platform to achieve features such as more complicated network topologies, larger network coverage. Also, different wireless protocols, such as 6LoWPAN and Bluetooth Low Energy with mesh networking, can also be employed to further extend the wireless specifications and the applicability of the RAWS technology.

The power efficiency of the system can be further improved. The current transceiver module used for prototyping should be replaced with a lower power ZigBee module, and other low power wireless protocols can be used as discussed above. Also, other techniques that can improve the lifetime of the system can be incorporated such as energy harvesting techniques.

In conclusion, this research has realised a reconfigurable adaptive wireless sensor node technology which is capable of supporting different analogue sensors by being able to

reconfigure itself in an autonomic, flexible and scalable way, and has achieved an analogue sensor *plug & play* concept in the WSN area.

Appendices

Appendix A – WSN Node Platforms Analogue Capabilities

Microcontrollers Used by WSN platforms	**Platform Analogue Capability**	**WSN platforms**
Atmel ATMega 128x family and related models	10-bit SAR ADC (Not for MiniMote), Analogue Comparator	BTnode [219], Dot , DSYS25 [222], Ember node [185], FireFly [187], Fleck [227], Indriya_DP_03A Series[189], Indriya_DP_03B Series[189], Iris/XM2110CB [192], MANTIS Nymph[230], Mica [180]/Mica2/MicaZ/Mica2Dot[231], n-Core[196], panStamp AVR[198], T-Nodes[257], Tyndall 25mm Mote[38], Waspmote [212], weC/Rene [208], WisMote Mini[258], MiniMote [232]
TI MSP430F1x family and related models Two major models: MSP430F149 and MSP430F1611	12-bit SAR ADC Analogue Comparator	BSN node [183], EYES [225] EyesIFX v1 [226], M1010[229], Pluto [199], ScatterWeb[237], Telos [208], OpenMote[197], panStamp NRG[235]
	12-bit SAR ADC 12-bit DAC Analogue Comparator	BEAN [220], EPIC Mote [186], EyesIFX v2 [226], G-Node G301 [228], Indriya_DP_01A Series [190], PowWow[200], SenseNode[203], SensiNode[204], Shimmer [205], Smartmote[207], SNoW5[238], TelosB/Tmote Sky [209] CM3000/CM4000/CM5000[210] Kmote[211], Tinynode[240], uNode[242], XM1000[216], Zolertia Z1[218]
	12-bit SAR ADC	MeshScape [195], WisMote[214],
	10-bit SAR ADC Analogue Comparator	WiSense[213]

Microcontrollers **Used by WSN platforms**	**Platform** **Analogue Capability**	**WSN platforms**
Microchip PIC16 and PIC18	10-bit ADC Analogue Comparator (Not for TCM120)	CIT Sensor Node [221], Everlast[224], i-Bean[229], Particle[236], TCM120[239], WMSN.P-2812 [215],
NXP LPC175x and LPC176x	12-bit SAR ADC 10-bit DAC	Indriya_DP_04A11 [191], Lotus [194]
Nordic nRF24L & nRF9	10-bit ADC	Eco [223], MITes[233], Tyndall 10mm Mote[241]
NXP Jennic JN514x	12-bit ADC 12-bit DAC Analogue Comparators	Jennic JN5148[193]
NXP LPC2106	None for LPC2106 10-bit SAR ADC from an external TI TLV1544 Op-Amp from an external Maxim MAX4061	eWatch [29],
ARM 920T	10-bit SAR ADC from an external Analog Devices ADT7411	SunSPOT[30]
Others with similar analogue capabilities as ATmega128x or MSP430F1x Such as: Analog Device ADuC841, Atmel SAM3U, Renesas M16C/62P, NXP PCH7970	8, or 10, or 12 bit ADC and/or 8, or 10, or 12 bit DAC	Cookie [73], Egs mote [184], Muller[234], Preon32[201], SAND[202], SENTIOF [39] ,XYZ[217]
Others with no analogue peripherals	None	Imote2[188], SmartMesh PM2511/LM2650 / Linear LTC58xx / LTP59xx Series [206]

Table A1 Existing WSN platforms' analogue capability

WSN controller options

The choice of the controller is important for developing the node platform. Generally speaking, the options include general purpose processors, digital signal processors (DSPs), field-programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), and microcontrollers. There are trade-offs between flexibility, performance, energy efficiency and costs when choosing from these different types of controllers.

General purpose processors such as CPUs usually have way too high power consumption for WSNs. Also their data processing capability is usually an overkill for WSN nodes.

Digital signal processors (DSPs) can be the right choice for the applications which require large amounts of data processing and intensive mathematical operations such as broadband wireless communications, video and audio processing. But the advantages of DSPs are not always beneficial for WSN applications as WSNs are often low data rate and the nodes' local data processing tasks are often not very complicated either. DSPs can be equipped as a co-processor when nodes do need to perform computation-intensive tasks.

Field-programmable gate arrays (FGPAs) can offer a high flexibility because their logic blocks (gate arrays) can be reprogrammed for different applications. Because of this feature, FPGAs are widely employed for prototyping, proof-of-concept design, and rapid or low cost product development in various applications. Some of the new high speed FPGA devices can be used to handle high performance computing and act as DSPs. So there are some wireless sensor nodes that employ FGPAs as the controller to handle heavy computing tasks. But for general WSN node designs, the FPGA option has several issues. The main one is the energy consumption. The energy budget for WSN nodes usually is only several hundreds of milliamp-hours, while the typical FPGA current consumptions are in the several hundred milliamps to few amps range. So the FPGA power consumption is high considering the power budget of the WSN nodes. Also, as the name suggests, the reprogramming operations are manually performed in-field. But this reprogramming method is not practical for many WSN applications and it is a high power operation as well.

Application-specific integrated circuits (ASICs) can have good performance in the application areas for which they are designed, but the main problem is that they normally require a very long design and development period and are usually subject to very high development cost.

Microcontrollers are the common choice as the WSN platform controller for the reasons such as low power, small size, flexible switching between different programmes as already discussed in Chapter 1. It is worth mentioning that besides the commercial off-the-shelf microcontrollers or SoCs, another option of the WSN node controller would be the custom designed IC. A big advantage of custom ICs is that they can be optimised for their application scenarios during the chip design phase. From the beginning of the design, they can be equipped with all the subsystems and function modules with specifications which will fit in well with their target scenarios. The main disadvantage for them is that the design period is long and costly, and in addition it requires significant volumes to keep the unit cost down.

Microcontroller selections in WSN platforms

Microcontrollers are the common choice as the WSN platform controller according to the results shown in Table A1. However, major microcontrollers being used on most of the WSN platforms have limited analogue capabilities and cannot provide flexible support for different analogue sensors. Atmel ATmega128x series and Texas Instruments (TI) MSP430F1x series are the two most common microcontroller series used in WSN platforms and they are very important in the WSN history. The reasons that many WSN platforms employ Atmel ATmega128x and TI MSP430F1x or the related microcontroller families link strongly with the most influential wireless sensor node platforms and operating system at the early stage of WSN history, which are the Mica series and Telos/TelosB series of WSN platforms and the TinyOS WSN operating system developed by University of California, Berkeley. The Mica series, including Mica, Mica2Dot, Mica2, MicaZ, and their predecessors, WeC, Rene and Dot all used Atmel's AVR 8-bit microcontrollers. The WeC WSN platform was the first TinyOS hardware platform developed in 1999. The WeC platform chose an early member of the Atmel AVR microcontroller family, the AT90LS8535. One main reason for this choice is the power consumption. Atmel AT90LS8535 was released in 1998 and has a 5mA typical active current and

15uA sleep current. Compared to many other microcontrollers often used at that time such as the Intel's 8051 and Motorola's HC08, Atmel AT90LS8535 has much lower power consumption [208]. The successors of the WeC platform, including the Rene and the Mica series, also chose Atmel AVR microcontrollers. One main reason is the software compatibility. Obviously it is easier to keep these hardware platforms compatible with the TinyOS operating system by continuing to use Atmel AVR microcontrollers. From 2001 to 2004, a series of Mica family WSN platforms, including Mica, Mica2, MicaZ, Mica2Dot, were designed and implemented using the Atmel ATmega128(L) microcontroller which is the one of the widely used device among the AVR 8-bit family and is still widely used in many electronic designs and systems nowadays due to its low power and low cost features such as the Arduino platforms. Atmel ATmega128 has an 8mA typical active current and 20uA sleep current which is higher than Atmel AT90LS8535, but still better than many other microcontrollers at the time. Plus it has 4kB RAM and 128kB Flash ROM. Compared to the Atmel AT90LS8535's 0.5kB RAM and 8kB Flash ROM and many other microcontrollers' memory capacities at that time, Atmel ATmega128's big memory space can benefit the software system in many ways. Mica series platforms were very successful in WSN area. Till 2004, Mica2 and Mica2Dot were the *de facto* standard platforms in WSN research [208]. In 2005, UC Berkeley released their new wireless sensor node platform, the Telos [208]. The Telos platform follows Mica platform's low power principle. After the release of Mica2 in 2002, UC Berkeley compared many microcontrollers that can offer lower power consumption and more on-chip peripherals, and chose the TI MSP430F149 as the controller for the original version Telos, and then later chose the TI MSP430F1611 for the Telos revision B, a.k.a. TelosB/Tmote Sky. These two TI MSP430 members can offer around 1~2 mA active current and 1uA sleep current which are much lower power consumption compared to the Mica series. TinyOS was also transplanted to support Telos/TelosB platform which means TI MSP430 series microcontrollers are supported by TinyOS as well. With the support of TinyOS, the TelosB platform was another big success after the Mica series, and lots of WSN research works have been conducted using TelosB or TI MSP430-based platform ever since [208].

Because of the huge influence of the Mica and Telos series WSN platforms and the TinyOS operating system, as well as their open source feature, a large number of wireless sensor nodes

also employed Atmel ATmega128x or TI MSP430F1x or their related families as the controller. Many new WSN research works are still based on these microcontrollers. At the early stage of the WSN technology, the main focus of UC Berkeley was to build low power hardware platforms for TinyOS. The on-chip peripheral functions of the Microcontrollers at that time were usually limited. So even though ATmega128x and MSP430F1x both feature low power especially the MSP430F1x series, their analogue capabilities are limited. Atmel ATmega128x series only features a 10-bit SAR ADC and analogue comparators which are inadequate for many analogue signal processing applications. Texas Instruments MSP430F1x series supports a 12-bit ADC, a 10-bit or 12-bit DAC and analogue comparators. TI MSP430F1x series has better analogue capabilities than ATmega128x but can still be inadequate in many applications. One other microcontroller family that also have been relatively widely used by WSN platforms is the Microchip PIC1x (mainly PIC16 and PIC18). Some research projects also tried to implement a version of TinyOS for PIC family microcontrollers [221] [239]. The Microchip PIC1x microcontrollers also feature low power. They have around 1~4 mA active current and 1uA sleep current. However, PIC16 and PIC18 family microcontrollers have similar analogue capabilities as ATmega128. They only support a 10-bit ADC and analogue comparators. (Microchip released some new PIC16 models with operational amplifiers in 2013).

With the WSN technology developing, WSN node platforms also start to employ new microcontrollers. Some WSN platforms (e.g. [184], [191]) choose the burgeoning ARM-based microcontrollers, such as Atmel SAM3x series, NXP LPC176x series, for the faster CPU core and better computing capabilities or more memory space; some WSN platforms (e.g. [241] [193], [258]) employ the so called 'Network-on-a-Chip' solution where the microcontroller and wireless transceiver are integrated into one chip, such as TI CC2430, TI CC430, Atmel ATmega128RFA1, Nordic nRF9E5, for further reducing the complexity and power consumption of the whole system. However, as shown in Table A1, these microcontrollers do not offer better analogue capabilities. For example, the Atmel ATmega128RFA1 microcontroller has the same analogue peripherals as ATmega128x series, i.e. a 10-bit SAR ADC and analogue comparators. TI CC430 series also have similar analogue capabilities as the TI MSP430F1x. Some other microcontrollers that have been used on WSN platforms such as Analog Device ADuCRF101,

NXP Jennic JN514x series, Nordic nRF9 and nRF24L series, NXP LPC175x and LPC176x, also have similar analogue capabilities as ATmega128x or MSP430F1x. Other controllers such as Intel PXA27x series, Linear Dust Networks SmartMesh series, NXP LPC210x series do not have any built-in analogue peripheral functions for signal conditioning and processing.

A WSN node's ability to support analogue sensors mainly relies on the controller of the node. This is because a WSN platform is built around its controller, and unnecessary external circuits and ICs are not reserved in the system to reduce the system complexity, size and overall power consumption. However, when the connected sensors require analogue functions that their controllers do not support, these WSN platforms need to resort to external circuits and peripheral ICs to support sensors as the traditional approach does. Some WSN platforms do take this approach, for example, the eWatch platform [29] employ an external Maxim MAX4061 differential amplifier chip and a TI TLV1544 10-bit SAR ADC chip to support analogue sensors as its on-board microcontroller does not provide analogue peripheral functions. However, this traditional approach lacks flexibility and scalability, and it can bring problems in terms of the circuitry complexity, size, cost and power consumption of the sensor node. In summary, most of the existing wireless sensor node platforms do not have the proper hardware infrastructure to solve the multi-sensing challenge.

Appendix B – Switched Capacitor Circuit Basic Theory

The idea behind switched capacitor technology is to use capacitors and alternately opened and closed switches to replace the resistors in analogue circuits and control the signal flow paths. The basic working theory of the SC circuit is shown in Figure B1.

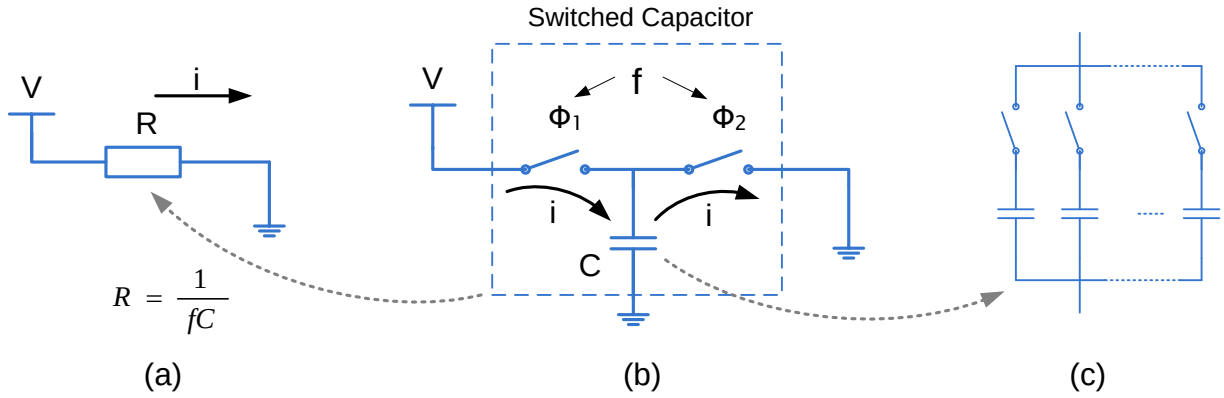


Figure B1 Basic principle of the switches capacitor circuit

In the basic continuous time circuit shown in Figure B1 (a), the current *i* flows through the resistor *R* to ground and the current can be determined by Ohm's law:

$$i = \frac{V}{R}$$

In the simplest form of the switched capacitor circuit shown in Figure B1 (b), a clock signal of frequency *f* controls the two switches alternated open and close to repeatedly fully charge and discharge the capacitor *C*. The amount of charge *q* that can be stored in the capacitor in fully charged condition is:

$$q = CV$$

The current flow in this switched capacitor circuit is discrete and it is caused by the repeated charge and discharge of the capacitor. The current can be determined by the rate of charge *q* travel across the switched capacitor:

$$i = fq = fCV$$

In actual IC circuits, the switches within the switched capacitor are MOSFET switches and the switching frequency can be varied. The capacitance of the capacitor in the SC circuit is also configurable, and this can be achieved through using an array of the capacitors which can be connected to or disconnected from the circuit by an array of switches, as shown in the rightmost of Figure B1. Choosing suitable values for f and C can make the current that flows through the SC circuit equals to the current in the basic continuous time resistive circuit, and effectively make these two circuits equivalent, i.e.:

$$i = \frac{V}{R} = fCV$$

This means that a switched capacitor can be used to reproduce the functionality of resistors and can behave as an equivalence of a resistor of a certain value, and the equivalent resistance is determined by the capacitance and the switching frequency.

$$R = \frac{1}{fC}$$

Appendix C – IEEE 1451 Standards

The IEEE 1451 standards family categorises the elements of a complete transducer system into two types of devices by introducing the concepts of the network capable application processors (NCAPs) and the transducer interface modules (TIMs). The TIM and NCAP concepts are original introduced in the first two members of the 1451 family, the IEEE 1451.1 and IEEE 1451.2. A TIM is a module that generally contains one or many sensors and/or actuators with the signal conditioning and analog-to-digital and/or digital-to-analog conversion circuitries, and an interface to the NCAP devices [122]. Different forms of TIMs with different types of interfaces to NCAPs are defined in different members of the 1451 family, but in general a TIM is a digital module with sensing and/or actuating functions.

A network capable application processor mainly contains a controller and two interfaces. One interface is to equip the NCAP device with the network capability and connect the NCAP and the transducer system into an external network. The 1451 family is aimed for network independent and it does not specify how physically the NCAP should connect to the external network or what kind of external networks the NCAP should connect to, so the external network for the IEEE 1451 transducer system is decided by the user and could be a WLAN or an Ethernet or the Internet. In the 1451 family there are no physical specifications for the external user network interface of the NCAP. The IEEE 1451.1 does define a logical object model for the external network interface of the NCAP and the purpose of this logical object model is to provide an abstraction layer that isolates high level applications from the details of network communications [259]. Another interface of the NCAP device is for communicating with the TIMs. With these two interfaces, the NCAP device works like a bridge between the TIMs and the external user network. IEEE 1451.1 provides a logical object model for the TIM-to-NCAP interface as well, but the IEEE 1451.2 is the actual standard for this interface. IEEE 1451.2 was the first official member of the 1451 family and was the first member to provide a specification for the TIM-to-NCAP interface. The 1451.2 defines a standard digital point-to-point communication interface to connect a TIM to an NCAP. The physical and electrical implementation of this interface with low level details including lines, protocol and timing definitions were specified in the 1451.2 standard [122].

The first form of the TIM concept which was referred to as smart transducer interface module (STIM) was also defined in the dot2 standard. The standard describes the specification of the design and implementation of the STIMs. As the TIM-to-NCAP interface is part of the standards as well as the STIM concept, the implementation of a STIM module have to include the TIM-to-NCAP interface and have to conform to the interface specification defined in the standard. The NCAP devices also have to conform to the physical and electrical interface specification defined in 1451.2 to work with the STIM, even though the NCAP side of the specification is covered in the 1451.1 and the 1451.1 only defines the common logical model which mainly acts as a software programming paradigm.

In addition to the TIM-to-NCAP interface part of the STIM, the dot2 standard includes the specification of the required functions for the overall STIM module and the compulsory and optional functions for each transducer element within the STIM module. These required functions include low level functions such as addressing, interface data transport, interrupt, interrupt mask, and they are described with low level detail, e.g. data format and data transport rate and pacing are defined for interface data transport function [122].

The key feature of the IEEE 1451 family, the transducer electronic datasheet (TEDS), was originally introduced by the 1451.2 standard. The general idea of the transducer electronic datasheet concept is to provide identification, calibration, correction, and manufacture-related information of the transducers. The 1451.2 standard defines the first version of the logical format and the content of the TEDS for STIM which describes the type, operation and attributes of the transducer elements within STIM module. The 1451.2 standard uses the TEDS as a mechanism for specifying the combination of the transducers, signal conditioning and conversion to the rest of the STIM system since the standard does not include a specification for signal conditioning and data conversion [122]. The 1451.2 requires memory device for containing the TEDS information remaining physically associated with the transducer and the STIM. The NCAP can access the 1451.2 TEDS through the TIM-to-NCAP interface defined by 1451.2.

The implementation of a TIM has to meet all of the requirements in the TIM-to-NCAP interface specification, the required functions specification and the TEDS specification defined by the 1451.2 to be deemed in conformance of the standard and become a 1451.2 STIM.

Overall, the 1451.2 tries to create a standard and a unified form for the development and implementation of the transducer modules and define a universal digital interface and protocol for transducers to connect to microprocessor-based NCAP devices, while the 1451.1 specify the NCAPs and tries to make NCAP devices as bridges that can connect to transducers to different kinds of user networks for transducer systems.

The later members of 1451 family progressively expand the general concept and functionality of the TIM and NCAP and broaden the options for implementing the TIM-to-NCAP interface. IEEE 1451.3 defines a new form of the TIM module, the transducer bus interface module (TBIM). A multi-drop bus interface is specified in the standard to interconnect TBIMs and NCAPs. The NCAP have to incorporate a transducer bus controller (TBC) to interface with TBIMs. Like the 1451.2, the 1451.3 defines specifications for the physical implementation and communication protocols of the interface, the required functions and commands, and the transducer electronic datasheet format for TBIMs. It is worth mentioning that the TEDS format defined in 1451.3 for the TBIM is not compatible with the 1451.2 TEDS for the STIM. The 1451.3 TEDS is not required to be physically associated with the TBIMs. IEEE 1451.5 defines the specification for the wireless transducer interface module (WTIM) and the wireless TIM-to-NCAP interface standard. IEEE 1451.0 released at the same time with the 1451.5 defines a set of common functions, commands and APIs to be performed by the TIMs and NCAPs for the purpose of facilitating the communications between TIMs and NCAPs as well as the communications between NCAPs and the external user network. The 1451.0 also define a general TEDS format for the 1451.5 and onward standards including 1451.6, 1451.7. The 1451.0 also proposes to solve the format compatibility issue between the 1451.2 and 1451.3 TEDS and the new general format. IEEE 1451.6 defines the TIM-to-NCAP interface using CANopen network, but it has not been officially released yet. IEEE 1451.7 specifies RFID based communications between TIMs and NCAPs.

The aforementioned 1451 standards in essence are specifications for the communication interfaces and protocols between TIMs (WSN nodes) and NCAPs (sinks or gateways) rather than the specifications for the smart sensors. The TEDS defined in these standards are also for describing the TIM module for the NCAP, for assisting the NCAP to communicate with the TIM module and to establish the TIM-to-NCAP interface, rather than for facilitating the TIMs (WSN nodes) to interface with the actual sensors.

Also the 1451 standards discussed above are trying to encapsulate sensors into certain forms of digital modules and standardise sensors into digital forms. However, the analogue sensor is still the dominant sensor type in use. Analogue sensors are reliable and robust and can survive harsh environments. Because of these advantages, analogue sensors cannot be replaced by the digital ones in a lot of cases. But the various characteristics and non-standardised outputs make it difficult to communicate with traditional analogue sensors, and the lack of smart features makes the ease-of-use and interoperability of the analogue sensors worse. The standards discussed above try to encapsulate the analogue sensing elements into a digital structure but this effectively tries to turn all sensors into digital and also essentially takes out the advantages of the analogue sensors.

Appendix D – Wireless Communication Protocols Review

IEEE 802.15.1 and Bluetooth

Bluetooth is a wireless technology standard designed for short range communications. The intent behind the development of Bluetooth is to create a single digital wireless protocol to replace the cables connecting electronic devices and to provide a way to exchange information between multiple devices such as mobile phones, laptops, personal computers, printers, digital cameras, and video game consoles, creating wireless personal area networks (WPANs) [260]. Bluetooth is designed to support the low power consumption, low cost and robustness key features. Bluetooth is adopted by IEEE as 802.15.1 standard but IEEE only specified the PHY and MAC layers in its standard.

Bluetooth operates at the 2.4GHz globally unlicensed Industrial, Scientific, and Medical (ISM) band. Bluetooth devices are classified into three power classes and the transmission range depends on the power class. The three Bluetooth power classes and the corresponding transmission ranges are shown in Table D1. The most commonly used radio power class is the Class 2 with a max transmission power of 2.5mW (4dBm) and it provides less than 10 meters transmission range in most cases. But some Class 2 devices can achieve up to 30 meters range at outdoor line-of-sign conditions [261]. The transmit current for Class 2 devices (chips) is usually around 30mA to 50mA, and receive current is around the same values. Deep sleep current can be 50uA or less but some devices do not have a sleep mode. For Class 1 devices the transmit current can go up to between 100mA and 200mA and the range is extended 20 to 30 meters indoors and can be more than 100 meters outdoors.

Class	Maximum Transmission power	Range (approximate)
Class 1	100 mW (20 dBm)	~100 meters
Class 2	2.5 mW (4 dBm)	~10 meters
Class 3	1 mW (0 dBm)	~1 meter

Table D1 Bluetooth Power Classes

Bluetooth have different operation modes for data transmitting using different techniques, and the data transfer rate depends on the operation mode. The practical data transfer rate (net throughput) in the Basic Rate (BR) mode is 721.2 kbps, and 2.1 Mbps for the Enhanced Data

Rate (EDR) mode. Since the version 3.0 in 2009, Bluetooth introduces a third and optional operation mode called Alternate MAC/PHYs (AMP) mode which allows the Bluetooth devices to use different MAC and PHY layer protocols for faster large size data transportation rather than the ones used in original Bluetooth BR/EDR operation modes. At the moment the Bluetooth over 802.11, where the 802.11 is the IEEE MAC/PHYs standard for Wi-Fi technology, can offer up to 24Mbps practical data rate. The AMP operations need the support from additional controllers in the Bluetooth systems, so using this mode will increase the power consumption of the Bluetooth.

The basic communication topology in Bluetooth standards is called piconet. A piconet is formed anytime a Bluetooth link is established. Because the piconet has a 3-bit address space, a maximum of 8 Bluetooth devices can exist in one piconet. One of the devices must act as the master and this master device can connect a maximum of 7 slave devices. The slave devices communicate with the master device but they cannot directly communicate with each other. The piconet is virtually a star topology.

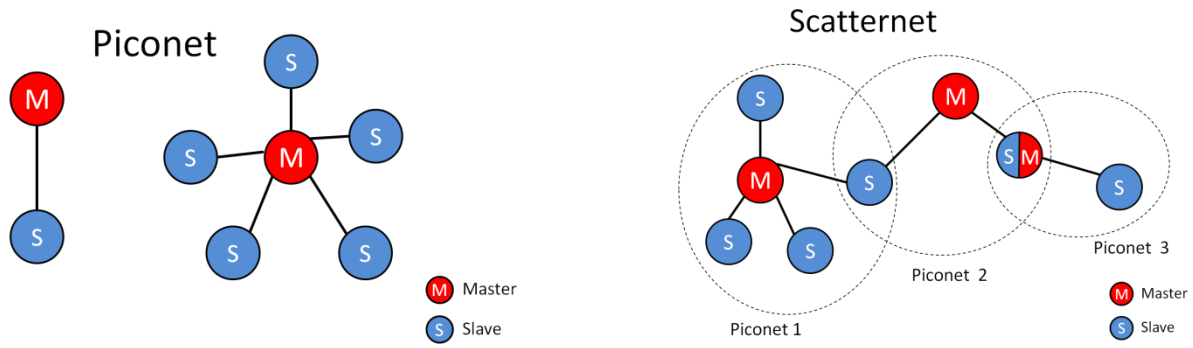


Figure D1 Bluetooth Topologies

Bluetooth allows a number of piconets exist within a common location, and it also allows one device to be the member of two or more piconets, which means the Bluetooth device that joined two or more different piconets can act as a joint node and connect piconets together. These interconnected piconets form a larger network named scatternet in Bluetooth standard which is similar to a cluster tree topology. These joint nodes shared by different piconets can relay data between the members of these networks. By founding a scatternet the size of the Bluetooth network can be expanded. However, Bluetooth protocols do not provide any

network routing capability or function for the joint nodes that participate in different piconets, so for making the data actually flow among different piconets within the scatternet, some kind of high level network routing protocols or management software have to be implemented on these joint node devices by the user.

Bluetooth Low Energy (Bluetooth Smart, Wibree)

Bluetooth Low Energy (BLE) was the former Wibree wireless technology. Wibree was originally designed for applications where ultra low power consumption, small size and low cost are the critical requirements and it was seen as Bluetooth enhancer, and later Wibree was included in the Bluetooth specification version 4.0 in 2010 as BLE [262].

Bluetooth Low Energy uses the same ISM 2.4GHz frequency band as the previous Bluetooth BR/EDR protocols which now is often referred to as classic Bluetooth. This allows BLE to share the same radio hardware as the classic Bluetooth so that they can coexist in one chip which is called dual mode Bluetooth device. However, BLE uses different radio techniques and a different set of protocol stack to make sure BLE devices can operate at low power consumption, so BLE is not back-compatible with the classic Bluetooth and the BLE devices cannot interconnect with the classic Bluetooth ones. The coexisted BLE and classic Bluetooth can work separately in one dual mode Bluetooth device but not at the same time.

The key difference between Bluetooth Low Energy and classic Bluetooth is the power consumption as the name already suggested and this is due to the different design objectives. Classic Bluetooth was originally designed for providing a continuous robust wireless connection between electronic devices such as mobile phones, laptops and PCs to replace the physical cables between them. Therefore classic Bluetooth devices usually have an ‘always on’ power profile to maximise their networking functionalities and some of them do not even support low power sleep mode. As a result the life time of the battery powered classic Bluetooth device is normally measured by days or even less.

On the other hand, Bluetooth Low Energy intends to extend the use of Bluetooth wireless technology to devices which are powered by small, coin-cell batteries [263]. BLE was designed for low power operation from the beginning. According to the BLE specification, BLE includes

features designed to enable products that require lower current consumption, lower complexity and BLE is also designed for use cases and applications with lower data rates and has lower duty cycles [260]. These features include lightweight protocols for providing ultra-low power idle mode operation, more efficient device discoveries, easier connection setup, short data packages, etc. As aimed for low duty cycles applications, BLE devices will try to stay in the sleep mode as long as possible unlike the 'always on' classic Bluetooth devices. BLE radio will only wake up to send or receive data, and because the size of BLE data packets is small and BLE connections can be set up quickly, the radio active time can be minimised. The maximum transmit power defined in the specification is 10mW (+10dBm). The actual transmit power of the BLE devices is normally configurable over the range of 0.01mW (-20dBm) to 2mW (3dBm) and the typical transmit power of the BLE device is 1mW (0dBm) or lower. The transmit and receive currents of some early BLE devices can be more than 20mA, and few newest models can already reduce these currents to less than 10mA, but overall BLE devices usually have 12mA to 20mA transmit current, 14mA to 18mA receive current, and less than 1uA deep sleep current (usually 0.2 to 0.5uA). In some low duty cycle applications, it is possible to operate BLE devices for more than a year with a coin cell battery.

The typical operating range is less than 10 meters indoor and can be around 30 to 50 outdoor at 1mW (0dBm), and the range can be optimised to above 60 meters by using higher transmit power and radios. The maximum practical data transfer rate of BLE is around 300kbps but the typical number usually is much less than 100kbps.

Bluetooth low energy has similar piconet topology as the classic Bluetooth. BLE is with a 32 bit address space, so the master device in the BLE piconet can support more than 2 billion slaves in theory while a classic Bluetooth piconet can only have 7 slaves. But BLE devices in the piconet are only allowed to connect to one device at a time so scatternets cannot be formed and the scatternet topology does not exist in BLE technology. BLE introduces a broadcast group topology which allows BLE devices sent out data one-way to multi devices at the same time without establishing any connections.

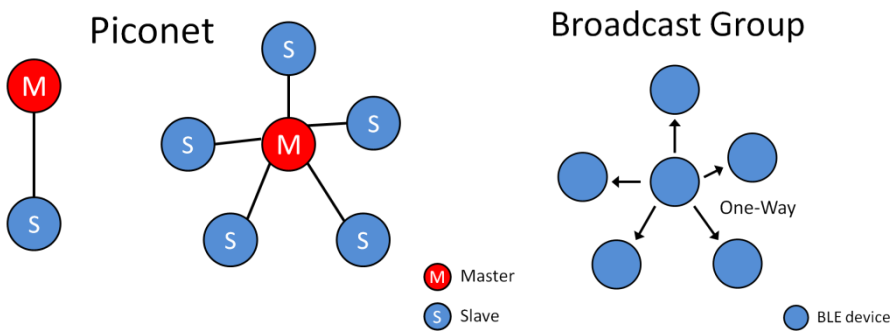


Figure D2 BLE Network Topologies

UWB (Ultra Wideband) over IEEE 802.15.3

UWB wireless technology is designed to provide high speed short range wireless data transmissions. Although mainly targeted at wireless multimedia applications such as audio and video delivery in WPANs and works as a replacement for high speed short range wired communications such as USB or FireWire (IEEE 1394), UWB over IEEE 802.15.3 was also intended for wireless sensor network applications [264, 265]. The IEEE 802.15.3 standard for high rate WPANs defines the MAC and PHY layers of the UWB technology. As already stated in the name and the technology objectives, the key feature of UWB is the high bandwidth. The 2.4GHz physical layer defined in the first version of 802.15.3 in 2003 can already support a maximum data rate of 55 Mbps and the draft of the second version (802.15.3a) supports up to 100 Mbps to 500Mbps data rate in the range of 1 to 10 meters [266]. In 2004 UWB over IEEE 802.15.3 devices is already capable of 114Mbps maximum data rate with 10 meter transmission range [267], while at this time another high rate short range wireless standard IEEE 802.11 (a.k.a. Wi-Fi) was only capable of 54Mbps (802.11g), and the version of 802.11 (802.11n) which supports more than 100Mbps data rate had not been drafted. The transmit and receive currents of the UWB devices is usually above 100mA [267]. The basic topology of UWB is a piconet (star) topology which can support maximum 8 devices in one network. It is similar to the piconet network in classic Bluetooth standard, but unlike Bluetooth piconets where any device can act as the master, UWB piconets have to have a specific network controller as the master device. UWB also support scatternet topology similar to the classic Bluetooth scatternet.

UWB over IEEE 802.15.3 once attracted much attention in WSN area for the applications that may require very high data rate. However, because the two major UWB industry groups cannot reach agreement on the RF technologies in the version of IEEE 802.15.3a, UWB over IEEE 802.15.3 did not have much support from manufacturers. Part of the IEEE 802.15.3a proposal was then incorporated by IEEE 802.15.4 standard as an optional PHY layer and the 802.15.3c was released in late 2009 but overall UWB over 802.15.3 still do not have much support from industry.

IEEE 802.15.4

IEEE 802.15.4 standard is important in wireless sensor network area as it is intended for low data rate wireless personal area networks. From the beginning, the goal of 802.15.4 was to produce a standard that enabled low cost, low power wireless communications [268]. IEEE 802.15.4 defined the PHY and MAC layers specifications for data communication devices with no battery or limited battery consumption requirements, and the standard intends to provide ultra low complexity, ultra low cost, ultra low power consumption, and low data rate RF wireless connectivity among inexpensive devices [268]. The objectives and features of the 802.15.4 match well with the low data rate and low power characteristics of wireless communications in most WSN applications and the small size, low cost features of the WSN nodes.

IEEE 802.15.4 operates in unlicensed ISM bands: 2.4 GHz for worldwide use, 868 MHz in Europe, 902 to 928MHz in North America. The data rate is 250kbps maximum and can be scalable down to 20kbps or below based on the application requirements. The maximum practical data rate is around 130kpbs [269]. The minimum transmit power defined in the standard is 0.5mW (-3dBm), but many 802.15.4 devices offer transmit power options lower than this limit, e.g. down to 0.016mW (-18dBm) or even 0.003mW (-25dBm), for the applications which require very low power or very small transmission ranges. The maximum transmit power is not defined in the standard but limited by local regulatory bodies. In Europe this limit is 100mW (20dBm). The typical transmit power of the 802.15.4 devices is in the range of 1mW (0dBm) to 2mW (3dbm) with a 10 to 30 meters indoor and 70 to 100 meters outdoor range. The transmit current of

802.15.4 devices is usually around 24mA to 31mA for the 1mW (0dBm) to 2mW (3dbm) transmit power but few models can achieve less than 20mA and even less than 15mA. The receive current of 802.15.4 devices is usually around 19mA to 24mA. The deep sleep current can be less than 1uA.

An IEEE 802.15.4 network can contain more than 60,000 devices as the standard defines a 16-bit network address space. IEEE 802.15.4 defines two types of network nodes: full-function devices (FFDs) and reduced-function devices (RFDs). The main difference between the two types is that full-function devices can provide networking and routing functions like relaying communications and/or establishing an 802.15.4 network while reduced-function devices cannot. IEEE 802.15.4 networks support two basic topologies, the star topology and the peer-to-peer topology. In the star topology, a network central controller called network coordinator takes charge of creating, initiating, and terminating an 802.15.4 network and routing communications within the network. The network coordinator role has to be performed by a full-function device. After the network coordinator created a star network, both types of devices FFDs and RFDs can join the network by connecting to the network coordinator directly. These nodes can communicate with the network coordinator and with any other devices in the network through the coordinator but they cannot directly communicate with each other.

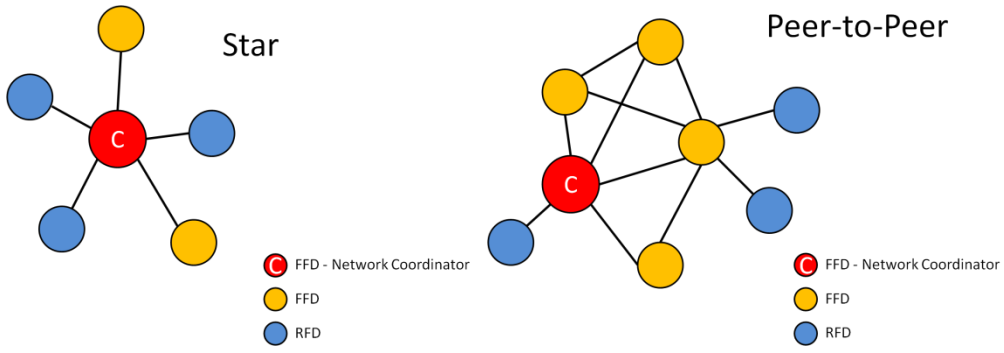


Figure D3 IEEE 802.15.4 Network Topologies

Compared to star network, peer-to-peer topology in 802.15.4 is much more flexible. A peer-to-peer topology network also has a network coordinator which creates the network, but the network coordinator is no longer the centre point of the network communication. The peer-to-peer topology allows the devices in the network to communicate with any other devices

directly if they are in the range of each other or through multiple hops if their transmission range is inadequate, but one exception is that when two RFDs want to communicate with each other they still have to go through at least one FFD as RFDs do not support any routing functions. Also in peer-to-peer topology one RFD has to connect to an FFD and can only connect to one FFD at a time whereas an FFD can create one-to-multiple links. The peer-to-peer topology offers flexible connectivity. Different network formations can be implemented based on the peer-to-peer topology such as tree topology or mesh networking topology, but these complex topologies and the relevant networking and routing functions such as multihop transmissions are not included in the 802.15.4 standard because it only specifies the two bottom layers, PHY layer and MAC sublayer, while these functions are defined at the higher layers.

IEEE 802.15.4 is the base for many other WPAN or WLAN standards including many important ones in WSN area such as ZigBee, 6LoWPAN, etc.

ZigBee

ZigBee is a set of wireless communication protocols designed for the low-power, low-cost, and low-data-rate wireless communication. ZigBee is built on the foundation of the two lower layer specified by IEEE 802.15.4 standard. Based on 802.15.4, ZigBee defined the network (NWK) layer and the framework for the application layer as shown in Figure D4. The NWK layer of ZigBee provides networking and routing functions, security services, etc, and the application layer framework consists of the application support sub-layer (APS) and other objects and profiles for application development and communication [270]. Like 802.15.4, ZigBee is also optimised for low power consumption operations through protocol layers.

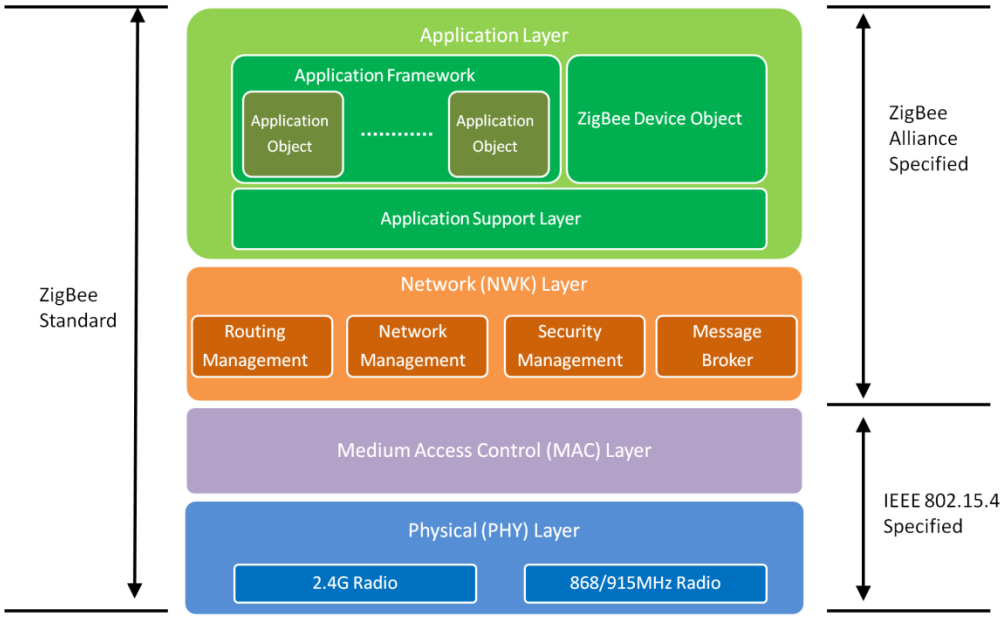


Figure D4 ZigBee and IEEE 802.15.4 Protocol Layers

ZigBee uses the PHY and MAC layers defined by 802.15.4, so it also operates at 868/915MHz or 2.4GHz ISM bands, and can support more than 60000 devices in one network. The RF parameters and power consumption performance of ZigBee devices are also very similar to the 802.15.4 devices. New 802.15.4 and ZigBee devices are not separated and are based on the same hardware foundation as ZigBee is built on 802.15.4. The maximum raw data rate of ZigBee is also 250kbps but the maximum practical data rate will be normally less than the 130kpbs that 802.15.4 can offer because of the higher layer overheads of ZigBee.

The ZigBee standard supports star, tree, and mesh topologies in the network layer based on the connectivity modes provided by 802.15.4 [270]. Same as 805.14.2 networks, there is always one network coordinator in any ZigBee networks and this coordinator is responsible for initiating the network and setting up the parameters for operating the network. The ZigBee star topology is the same as defined in 802.15.4. In the star topology, the ZigBee coordinator manages all the end devices that joined the network and routing the communications. All the end devices directly communicate with the ZigBee coordinator and any communications between end devices have to go through the coordinator.

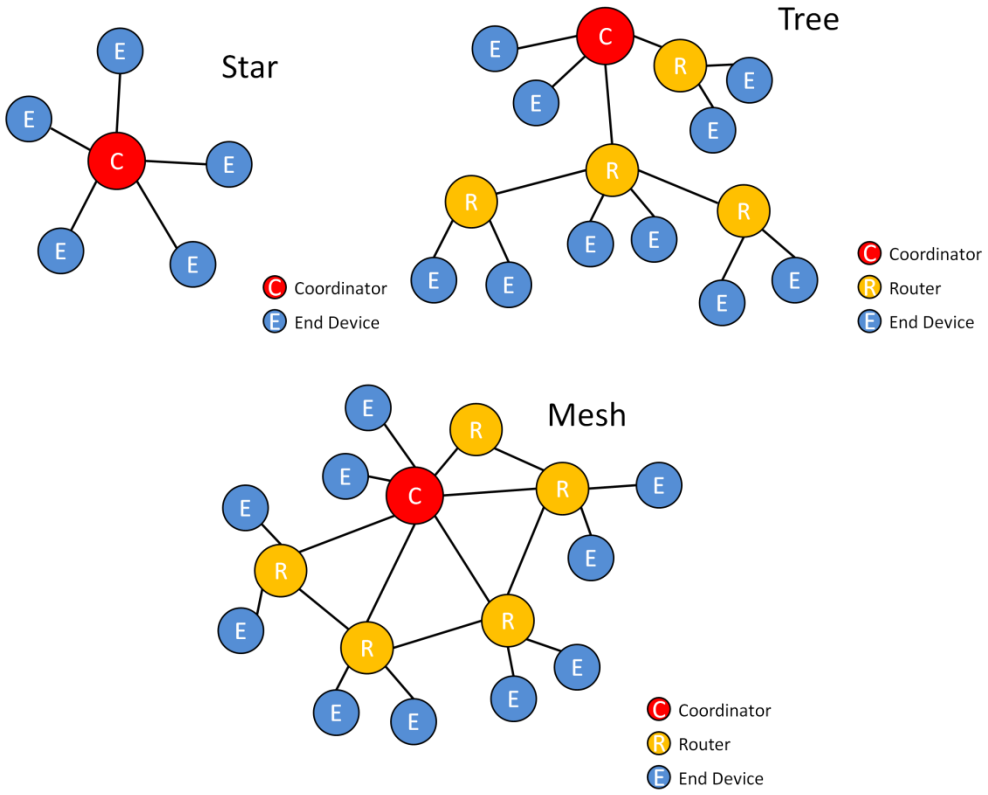


Figure D5 ZigBee Network Topologies

For the tree and mesh topology, ZigBee introduces an intermediate managing device called ZigBee router. ZigBee routers are full-function devices (FFDs), i.e. they have networking and routing capacities. ZigBee network can be easily extended through the use of ZigBee routers and the peer-to-peer connection features based on the 802.15.4 standard. In tree networks, end devices can join the network by directly connect to the coordinator or through the routers. The routers maintain the devices under their sub-network and control the communications through their sub-network, and routers report to the ZigBee coordinator or a higher level router. The communications of the whole network are managed via this hierarchical routing strategy.

ZigBee supports mesh topology but it is not a complete mesh topology as the mesh networking ability is only applied to router and coordinator level devices because they are FFDs and are capable of routing communications but not applied to the end devices as they are reduced-function devices. The routers in ZigBee mesh networks are allowed full peer-to-peer

communication and they do not have to go through the coordinator for exchanging information. The communications between end devices still have to go through the coordinator or routers. ZigBee end devices are normally battery powered and they can and should go into sleep mode when idle to conserve power. The ZigBee coordinator and routers administrate the network and provide routing functions, so unlike end devices they are not designed to sleep and are normally powered by a reliable energy source to remain on to maintain the network.

6LoWPAN

6LoWPAN is short for Internet Protocol version 6 (IPv6) over Low power Wireless Personal Area Networks. IPv6 is the next generation of the Internet Protocol (IP) standard which is the key communications protocol in the network layer of the Internet model for providing routing function and delivering data on the Internet. IPv6 is intended to supplement and eventually replace the current version of Internet Protocol version 4 (IPv4). The Internet Engineering Task Force (IETF), which is a standards organization defines many important Internet standards including the Internet protocol suite (a.k.a. TCP/IP), defines the 6LoWPAN. 6LoWPAN is a set of standards designed to enable efficient IPv6 communications on top of low power wireless network protocols, specifically the IEEE 802.15.4 standard. The most important feature of 6LoWPAN is that it provides the direct interconnectivity between the representative low power wireless networks, the 802.15.4-based networks and the universal Internet. 6LoWPAN together with the Cloud technologies can facilitate the remote collection and distribution of the information gathered by the low power IEEE 802.15.4 based WSNs. 6LoWPAN defines an adaptation layer which interfaces the 802.15.4 physical and MAC layers with the IPv6 network layer and upper layers of the Internet. This 6LoWPAN adaption layer employs mechanisms such as protocol header compression, package fragmentation and resembling to allow big size Internet data packages to be transmitted over the 802.15.4-based devices. This means 6LoWPAN wireless devices and networks can access the Internet while still maintain the low power and other important features from IEEE 802.15.4. 6LoWPAN can be used to provide direct end-to-end Internet connectivity to low power wireless networks where IPv6 addresses can be allocated to each network nodes. In this case these low power wireless networks can be considered as an extension of the Internet. 6LoWPAN can also be utilised in the closed wireless

networks, or in some cases it can be implemented only on the gateways of non-6LoWPAN networks to provide Internet connections.

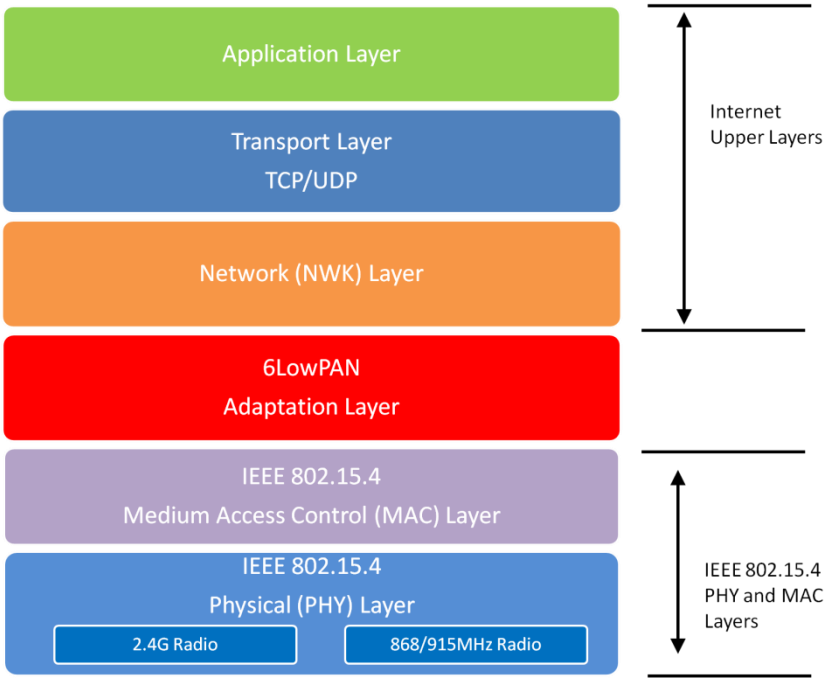


Figure D6 6LoWPAN Layer Structure

Since 6LoWPAN is also based on the PHY and MAC layers defined by 802.15.4, 6LoWPAN have many identical or similar parameters to 802.15.4 and ZigBee such as operation frequency bands, raw data rate, transmit and receive power performance, range, network capacities, and network topologies and so on. 6LoWPAN devices can share a lot of common hardware resources with 802.15.4 and ZigBee devices and in fact some manufacturers such as Texas Instruments offer single-chip solutions which can support all these three wireless communication protocols.

Other 802.15.4 Based Protocols

WirelessHART and ISA100.11a are another two protocols based on 802.15.4, and they both aim for automation and industrial control applications. WirelessHART is designed as a wireless version of the wired HART (highway addressable remote transducer) protocol standard which is a widely used digital industrial automation protocol. WirelessHART is intended to be back compatible with the HART protocol in the application level to protect industrial investments.

WirelessHART is based on the PHY layer specified by the 802.15.4 but it does not use the 802.15.4 MAC sublayer and defined its own MAC and higher layers with features such as frequency hopping to sustain the harsh industrial environments. A qualitative analysis indicates that these features can make WirelessHART more robust and secure than the cost-effective and low-complexity 802.15.4/ZigBee in industrial environments [271]. As WirelessHART still bases on 802.15.4 PHY layer, it can share the radio of the 802.15.4 devices so in general the WirelessHART devices have the same or similar RF parameters and performance as 802.15.4 or ZigBee devices in terms of operation frequency bands, raw data rate, transmission range, etc. WirelessHART devices normally use 6.3mW (8dBm) to 10mW (10dBm) transmit power which is much higher than the typical 1mW (0dBm) for 802.15.4/ZigBee devices, thus the transmit current is normally higher than 802.15.4/ZigBee devices but few new models with low power radio technology can achieve around less than 10mA transmit current at 8dBm which is even lower than the transmit current at around 0dBm for the most older WirelessHART devices. WirelessHART supports star, tree, and mesh topologies. But unlike 802.15.4 there are no reduced-function devices in WirelessHART networks and every device has the routing capability, so WirelessHART can support full mesh topology. About 30,000 devices can be contained in one WirelessHART network.

ISA100.11a is established by International Society of Automation (ISA) as the standard for wireless systems in industrial automation process control and related applications. Like WirelessHART, ISA100.11a also uses the 802.15.4 PHY layer as the basis and adds its own features. There are many similarities between ISA100.11a and WirelessHART as they are both aimed at industrial control and automation and applications, and there is also an ISA group investigating the possibility the convergence of these two industrial wireless standards around 2012 although at the end this did not happen. There are also many differences between these two standards. In general ISA100.11a is more flexible as its application layer can be compatible with more well-established industrial wired communication protocols such as Profibus, Fieldbus Foundation, and Modbus as well as the wired HART, while WirelessHART can be simpler in terms of implementations.

Wi-Fi

Wi-Fi is one of the most popular wireless networks all around the world as Wi-Fi technology has been fit into all kinds of devices in people's daily life such as smart phones, tablets, laptops, desktops, portable music players, video game consoles, smart TVs and stereo systems. Generally speaking, Wi-Fi is a well known wireless networking technology that allows electronic devices to connect Internet or communicate with each other. From a technical point of view, Wi-Fi is the brand name for the wireless technology based on the IEEE 802.11 standards family which defines the MAC and PHY layer specifications for wireless connectivity within a local area, i.e. for wireless local area networks (WLANs). The family of 802.11 standards provides the basis for the wireless devices certified for the Wi-Fi trademark. The 802.11 standards mainly operate at 2.4GHz ISM frequency bands especially the versions with widespread adoption: 802.11b, 802.11g and 802.11n. 802.11n which is the current most common version can also operates at 5GHz ISM to avoid the heavily used 2.4GHz band. The first amendment to the original standard, 802.11a, is operating at 5GHz as well.

IEEE 802.11 family is intended for high rate wireless communications. 802.11b has a maximum raw data rate of 11 Mbps and the maximum practical data rate (net throughput) is about 5.9 to 7.1 Mbps. For 802.11a/g the maximum raw data rate is 54 Mbps and maximum practical data rate is about 24 Mbps. 802.11n can achieve 600 Mbps maximum raw data rate and roughly 300 Mbps practical data rates. The data rates of the 802.11 family protocols can be scaled down depending on the signal quality or for achieving longer range. The typical transmission range of the 802.11 standards is around 30m indoor and 100m outdoor. More details about the data rates and transmission ranges of different versions of 802.11 are shown in Table D2.

	Max Data Rate (Mbps) (Scalable Data Rate Option) (Mbps)	Max Throughput (Mbps)	Approx Max Range	
			Indoor (Meters)	Outdoor (Meters)
802.11a	54 (6, 9, 12, 18, 24, 36, 48)	~24	~10 to 30	~40 to 80
802.11b	11 (1, 2, 5.5)	~5.9 to 7.1	~30 to 50	~90 to 120
802.11g	54 (6, 9, 12, 18, 24, 36, 48)	~24	~30 to 50	~90 to 120
802.11n	150, 300, 450, 600 (Max rate is Device depended) (15, 30, 45, 60, 90, 120, 135)	~100 to 300	~30 to 90	~150 to 250

Table D2 802.11 Standards Data Rate and Range

Wi-Fi (802.11) devices were often used in consumer or commercial products such as Wi-Fi routers, Wi-Fi adapters and laptops where the power consumption is not a vital concern. Many manufacturers also offer Wi-Fi transceiver modules or Wi-Fi SoC solutions which are suitable for research projects and the design and development of embedded applications. These Wi-Fi modules or SoCs feature low power active mode and provide idle and sleep modes for power saving. They normally have the following features: 802.11b/g/n compatibility, up to 100Mbps, 1mW (0dBm) to 63mW (18dBm) transmit power options, 130mA to 300mA transmit current, 30mA to 90mA receive current, less than 15mA idle current and less than 1mA for some newer models, less than 6uA deep sleep current.

IEEE 802.11 standards support two basic connectivity modes: access point network which is called basic service set (BSS) in the specification and ad-hoc network which is called independent basic service set (IBSS). The ad-hoc network is similar to the peer-to-peer topology network where all the devices can directly communicate with each other and no central device is required as the communication controller. In the access point network, one central device called access point sets up and administrates the network. The access point is normally a router. Other devices join the network by connecting to the access point and communicate with or through the access point. Generally the access point configuration is similar to the star topology. Access points can be linked together by wired or wireless networks and form multiple basic

service sets into an extended service set (ESS) which is similar to a cluster tree topology. Access point level mesh networks for are also supported after the 802.11s is defined.

Other Protocols

ANT

ANT is a proprietary low power, low data rate, short range wireless technology targeted at sensor applications or similar applications. The ANT technology has a number of similarities with Bluetooth Low Energy as shown in Table D3. ANT operates at the 2.4GHz ISM band as well. It also designed for low data rate, low duty cycle applications. The maximum raw data rate of ANT is 1Mbps, the same as BLE. The practical data rate is 20kpbs, less than BLE. ANT devices also have less than 1uA deep sleep current (usually around 0.5uA) and ANT also supports techniques such as lightweight protocols, short data packages to ensure low power consumption and allow ANT devices to operate on one coin cell for more than a year. The typical transmit power of ANT devices is around 1mW (0dBm). The transmit current of the ANT devices is usually around 15mA to 28mA. The receive current is around 17mA to 24mA. The typical operating range can be from less than 10 meters to 30 meters. ANT is more flexible than BLE in terms of network configurations. ANT supports star, tree, and mesh topologies, while BLE only supports star configuration. ANT also supports 32 bits address space.

	Bluetooth Low Energy	**ANT**
Frequency	2.4GHz	2.4GHz
Topologies	Star	Star, Tree, Mesh
Raw Data Rate	1Mpbs	1Mbps
Practical Data Rate	Typical < 100kpbs	Up to 20kbps
Range	Typical < 30	Typical < 30
Tx Power	Typical 1mW (0dBm)	Typical 1mW (0dBm)
Tx Current	~ 12 to 20mA	~ 15 to 28mA
Rx Current	~ 14 to 18mA	~ 17 to 24mA
Deep Sleep Current	~ 0.2 to 0.5uA	~ 0.5uA

Table D3 BLE vs. ANT

Z-Wave

Z-Wave is a proprietary wireless communication protocol often compared with ZigBee because they have overlapping targets at small area network applications especially the home sensing and automation field. Z-Wave also features low power, low data rate as ZigBee. Z-Wave also works at 868 MHz band in Europe, 908MHz and 916MHz band in North America, but it avoids the 2.4GHz band. The data rate is up to 100kbps. The typical range of Z-Wave devices is also less than 30 meters indoor and around 100 meters outdoor. The transmit power is configurable and is typically around -5dBm to 2dBm and transmit currents are around 24mA to 40mA. The receive current is about 23mA. The deep sleep current is around 1uA to 2.5uA. These parameters are similar to the ones of 802.15.4/ZigBee. A maximum of 232 nodes can be included in one Z-Wave network while 802.15.4/ZigBee can support more than 60000. Z-Wave supports mesh topology.

Appendix E – Principles of ZigBee Networking and Communications

ZigBee standard is built on the top of IEEE 802.15.4 standard. The IEEE 802.15.4 specifies the physical (PHY) and medium access control (MAC) layers standard which defines the operating frequency bands, modulation techniques, transmission power range, data rates, etc. The IEEE 802.15.4 defines two device types, full-function devices (FFDs) and reduced-function devices (RFDs), where full-function devices have the ability to perform networking and routing functions while reduced-function devices do not. IEEE 802.15.4 also defines two basic network topologies, star network and peer-to-peer network. Based on the IEEE 802.15.4 standard, ZigBee defines the upper layers standard including the network (NWK) layer and the framework for the application layer, and many other specifications.

Device Types, Roles and Basic Working Principles

ZigBee supports star, tree and mesh topologies. ZigBee standard defines three types of ZigBee network devices: coordinator, router and end device. The three device types have different networking capabilities and characteristics. ZigBee coordinators and routers have to be full-function devices whereas the ZigBee end devices normally are reduced-function devices. The coordinator of a ZigBee network is the initiator of the network. Only the coordinator type device can start and initial a network. One ZigBee network can have many FFDs but only has one single coordinator regardless of the network topology. The ZigBee coordinator selects and initialises key network parameters and starts the network. After network initialisation, the coordinator can allow routers and end devices to join the network, and it becomes the parent of the devices with a direct link and routes communication among its device children. In a star topology, the coordinator is the centre of the network and it is the parent of all devices. It manages the network and directly communicates all the devices joined the network. All the communications between children have to go through the coordinator. In tree and mesh topologies, the coordinator starts the network but after this it basically just helps to route messages and behaves like a router. ZigBee routers are FFDs and as the name suggested they can route communication among the network but they have to join a ZigBee network first before they can function. Beyond routing functions, routers can also be installed with the functions supported by end devices. In tree and mesh topologies, after joining a network

routers can allow end devices and other routers to join them so the network can be expanded. In a star topology, there is no need for routers because the star topology only has one level in its hierarchy and all the devices in this topology will join the coordinator directly. So even though routers can join the coordinator in a star network, they still act as end devices and their routing functions are not utilised. ZigBee end devices can join a network through the coordinator or routers. Like routers, end devices also have to join a network first before they can transmit and receive data. End devices are RFDs so they are not able to route messages and they cannot be joined by other devices. However, end devices are designed to be low power so they can enter sleep modes to conserve energy. On contrary, because the coordinator and routers are responsible for routing data within the network, they have to be always on.

Creating Network

To start a ZigBee network the coordinator firstly selects key network parameters including the operating channel, network identifier, security policy, and stack profile.

Operating Channel

ZigBee/IEEE 802.15.4 operates at the unlicensed 2.4GHz ISM band globally. ZigBee/IEEE 802.15.4 assigns 16 channels within the 2.4GHz band as shown in Figure E1. The 16 channels start from 2.405GHz and are spaced 5MHz apart. Each channel covers 2MHz bandwidth so there is no overlap between channels. The channels in the 2.4GHz band are numbered from 11(0x0B) to 26(0x1A).

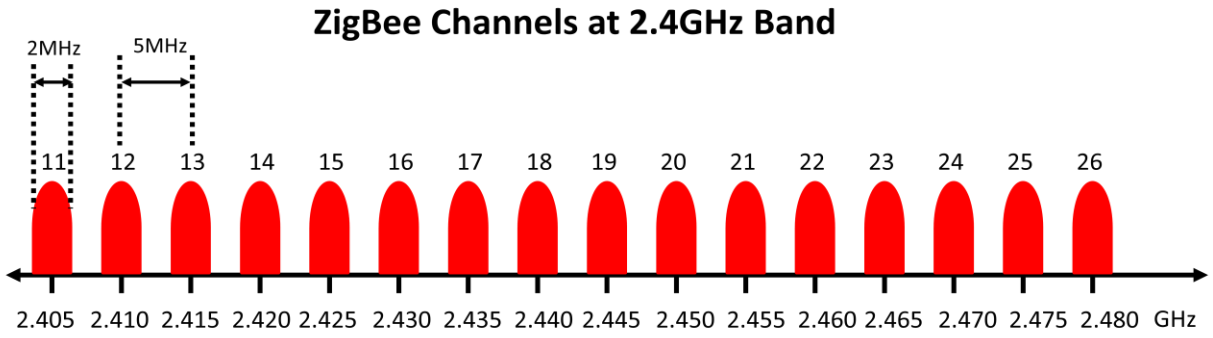


Figure E1 ZigBee channels

ZigBee/IEEE 802.15.4 employs direct-sequence spread spectrum technique and a ZigBee network operates on a fixed channel once it formed. Therefore, when starting a network the coordinator needs to select a channel for the network to operate on. To determine which channel to work on, the coordinator performs a type of channel scan called energy scan on all the 16 channels or a subset of the 16 channels specified by users and detects the energy level on each channel, and then it excludes the channels with energy levels over a certain threshold and keeps the rest as the potential channels to start the network on. The channel selection scheme is defined in the application level profile, so how many channels and what channels to be scanned also depends on the application level profile. Generally the coordinator scans 16 channels by default, but it is also common to allow users to specify a set of channels for the channel scans and in this case the coordinator only scans the user preset channels. The list of channels to scan is defined as ScanChannels parameter in ZigBee. The ZigBee coordinator transceiver devices are available that can perform channel scans automatically without manual initialling and micromanaging the scans and processing the scan results, but the ScanChannels and the attributes discussed later in this section are still important parameters for the users to decide when developing the ZigBee network. Because the scanning process is time consuming, and also the 2.4GHz band is extensively utilised by many wireless protocols especially Wi-Fi which is almost everywhere nowadays, specifying the ScanChannels to limit the choice of channels available for the energy scan can save time and avoid some heavily occupied frequency bands such as the channel 1, 6, and 11 of Wi-Fi locating at 2.412GHz, 2.437GHz, 2.462GHz respectively. Channel 25 and 26 of ZigBee are outside the frequency bands normally used by Wi-Fi so ZigBee networks operate at these two channels are likely to have less interference. But reducing the available channels for the energy scan also decreases the number of potential operating channels and this could increase the possibility of frequency collision with other RF communications.

Network ID

As ZigBee networks are classified as personal area networks (PAN), the network ID is usually referred to as PAN ID. One ZigBee network has two PAN IDs, a 64-bit ID and a 16-bit ID. Both IDs are used to identify ZigBee networks and all the devices in the same ZigBee network have the

same 64-bit and 16-bit IDs. Originally the earlier version of ZigBee standard before 2007 only defined the 16-bit PAN ID for ZigBee networks and it is also used as a MAC layer addressing field for the transmissions of a network. Because the limited addressing space of this short ID could cause PAN ID conflicts, ZigBee standard defines the long 64-bit ID to uniquely identify a ZigBee network so the 64-bit ID is intended to be unique for different networks. The 64-bit PAN ID is used for coordinators to create a ZigBee network and for other devices to join the network. The 64-bit PAN ID is another important parameter for the developers to set when establishing the ZigBee network. The 64-bit PAN ID can be manually assigned to the ZigBee coordinator, or the coordinator can generate a random 64-bit PAN ID if it is not a pre-assigned. In the former case, when the coordinator starts a network it has to use the preconfigured 64-bit ID as the PAN ID, so in this case the developers are responsible for avoiding 64-bit PAN ID conflicts which cannot be resolved at runtime. In the latter case, the coordinator performs another type of channel scan called active scan or PAN scan on all the potential operating channels derived from the energy scan. The PAN scan can acquire the PAN IDs of other networks within the operating range among other useful information such as the operating channels. The coordinator then selects a 64-bit ID not currently used by the overlapping networks. The 16-bit ID is normally selected by coordinators and cannot be specified by users.

Scan Duration

The channel scan processes take time. The scan duration (Scan Duration) parameter determines how long one scan process can spend on one channel. The scan duration parameter can be set as an integer between 0 and 14 and the scan time for each channel is defined by the equation below:

$$\text{Scan Time} = 15.36\text{ms} \times \left(2^{ScanDuration} + 1\right)$$

Longer scan time can increase the possibility of detecting other networks and their activities on the current scanning channel but will consume more power. As shown in Table E1, at ScanDuration setting of 6, the scan time for just one channel is already about 1 second. For battery power devices, scanning a number of channels with high scan duration can reduce the lifetime of the device.

Scan Duration	Scan Time (ms)	Scan Duration	Scan Time (s)
0	30.72	7	1.981
1	46.08	8	3.948
2	76.8	9	7.880
3	138.4	10	15.744
4	261.12	11	31.473
5	506.98	12	62.930
6	998.4	13	125.844
		14	251.674

Table E1 ZigBee Scan Duration Settings vs. Scan Time for one Channel

After the energy scan and PAN scan, the coordinator can select a suitable channel for the network based on the channel selection scheme and by default it selects the channel with fewest networks and lowest energy level.

Security Policy

The coordinator is also usually responsible for setting the security policy when starting a network. ZigBee supports different layers of security mechanisms including the network (NWK) layer security and application support (APS) layer security which are responsible for secure data payload of their respective layer field within the ZigBee packet. A network key and APS link keys can be used to encrypt and authenticate data via 128-bit AES (Advanced Encryption Standard) encryption. If the security is enabled, NWK layer security is applied to all data transmissions originated from NWK or higher layers. NWK layer security employs the hop-by-hop authentication scheme, i.e. the data packets with NWK security are decrypted, authenticated and then re-encrypted and re-authenticated at every hop en route. Because of the NWK layer hop-by-hop authentication scheme, secure networks have higher latency. NWK layer security uses a network key which is shared by all devices in the secure ZigBee network to encrypt the NWK payload packets. The network key may also be used at APS layer. Generally the network key can be pre-installed to the trust centre of the secure network or randomly generated by the trust centre and it can be passed to the devices trying to join the network advisedly through link key encrypted transmissions. Normally the trust centre is the coordinator. The network key can be changed and updated periodically. The security mode decides the network key update policy and the key distribution approach. APS layer security can provide end-to-end security between

two ZigBee devices and other security services such as key establishment. APS layer security authenticates packets at the source and destination devices that shared a link key, i.e. APS payload secured by APS layer security are encrypted at the source and only decrypted at the destination with the common link key. Link keys are used at APS layer only. Link keys can be generated using the key establishment services defined in the ZigBee standard or can be pre-installed in the devices. The latest ZigBee standard defined a default global link key that must be supported by the new ZigBee device if there is no other specified link key at the time of joining a network. Enabling security reduces the maximum data payload because of the security overhead. The security settings and security modes can be configured based on the application requirements but should be the same for all devices in a network. In RAWS node research the ZigBee security feature is implemented, but as the aim is to build a simple ZigBee network as a simple testbed for the proof-of-concept so the security is not enabled for the RAWS node prototypes.

Stack Profile

The stack profile setting of the ZigBee devices is a result of the ZigBee Pro specification release in 2007. Compared to the previous ZigBee 2006 specification, the ZigBee Pro or ZigBee 2007 specification is enhanced and provides more features in different aspects. ZigBee 2007 is fully backwards compatible with ZigBee 2006 and it includes two stack profiles: the ZigBee Pro stack that offers all the new features defined in the 2007 specification but consumes more memory space, and the ZigBee stack which is the stack defined in the 2006 specification and requires less memory to operate. ZigBee devices are normally able to select which stack profile to use. However, because of the differences in routing operations, when the devices running on ZigBee Pro stack join the network operating on ZigBee stack they have to act as the end devices and vice versa. So when establishing a network it is necessary to determine the stack profile for all devices to join.

Permit Joining

The coordinator (and routers) can control whether other devices are allowed to join it at a given time by configuring the permit joining (PermitDuration) parameter. The joining

permission can be permanently enabled or disabled, or only enabled for a specific length of time based on the parameter value. The permit joining can be set at runtime. Also, even if the permit joining is enabled when a device is trying to join, the parent device still need to check if it still has capacity available for a new child device.

Joining Network

Once the coordinator formed the network, it allows other devices to join the network. ZigBee routers and end devices have to join a network and they are not able to start networks. The parameters discussed above are important and have to be properly configured for successful network joining. First of all, routers and end devices have to find a network to join. Typically routers and end devices are designated to join a target network and they are assigned with the 64-bit PAN ID of the network. They will try to find the network with this 64-bit ID and join the network. If no pre-assigned 64-bit PAN ID, routers and end devices will try to find a joinable ZigBee network. No matter having a pre-assigned 64-bit PAN ID or not, when joining a network the PAN scan is needed to find the target or suitable network. The ScanChannels parameter of ZigBee routers and end devices defines the channels on which PAN ID scans are performed. If intended to join a particular network, routers and end devices should scan all the channels supported by the network coordinator, otherwise they may not be able to find the network. The PAN scan is performed on each channel in the ScanChannels list in sequence. For each channel, the PAN scan discovers all the ZigBee networks on the channel and returns the information of these networks. The joining device then checks the following parameters retrieved from an operating network to determine whether the network is the target network or a suitable network to join:

- 64-bit PAN ID (if the joining device has been pre-assigned one)
- Stack Profile
- Permit Joining
- Security Policy

In the network design, these parameters need to be properly set up not just for the coordinator but all the joining devices as well. If the target network or a suitable network was not

discovered on the current scan channel the PAN scan is performed on the next channel based on the ScanChannels parameter. After a device successfully joins a network, it can start communications.

Addressing

ZigBee devices have two addresses, a 64-bit MAC address and a 16-bit network address. The 64-bit address is an ID assigned to a device at manufacturing. The 64-bit address is unique for each device and cannot be changed. The 16-bit address is assigned when a device joins a network. The 16-bit address of a device is unique only within a network. The 16-bit address can change when a device leaves and rejoined a network. The 64-bit address of 0x0000000000000000 and the 16bit address of 0x0000 are reserved for the coordinator. For the RAWS node research concept proofing and prototype testing, the requirements for wireless communications are simple single-hop unicast transmissions between the RAWS nodes and the coordinator, also all communications are commenced by the RAWS node end. Knowing the coordinator address is adequate for the RAWS nodes to transmit packets to the coordinator and the coordinator can obtain the addresses of RAWS nodes from the incoming packet thus bi-directional links have been established, so advanced addressing features as well as advanced transmission functions and routing functions are not required and not discussed in this thesis.

Appendix F – IEEE 1451.4 Relevant Information

IEEE 1451.4 Sensor Type Templates

Function	Select	Property	Description	Bits	Data Type	Units
ID	—	TEMPLATE	Template ID	8	Integer (Value=36 in this case)	—
Measurement	—	%MinPhysVal	Minimum Temperature	11	ConRes (–273 to 1,770, step 1)	°C
	—	%MaxPhysVal	Maximum Temperature	11	ConRes (–273 to 1,770, step 1)	
Electrical Signal Output	—	%ElecSigType	Electrical Signal Type	—	Assign = 0, “Voltage Sensor”	—
	—	%MinElecVal	Minimum Electrical output	7	ConRes (–25E-3 to 0.1 step 1E-3)	V
	—	%MaxElecVal	Maximum Electrical output	7	ConRes (–25E-3 to 0.1 step 1E-3)	V
	—	%MapMeth	Mapping Method		Assign = 4, “Thermocouple”	—
	—	%TCType	Thermocouple Type	4	Enumeration: B | E | J | K | N | R | S | T | Non-std.	—
	—	%CJSource	Cold-junction Compensation	1	Enumeration: CJC required | Compensated	—
	—	%SensorImped	Thermocouple resistance	12	ConRelRes (1 to 319k, ±0.155%)	Ω
	—	%RespTime	Response Time	6	ConRelRes (1E-6 to 7.9, ±15%)	s
Calibration Information	—	%CalDate	Calibration Date	16	DATE	—
	—	%CalInitials	Calibration initials	15	CHR5	—
	—	%CalPeriod	Calibration period	12	UNINT	days
Misc.	—	%MeasID	Measurement location ID	11	UNINT	—
			Total bits:	121 bits		

Table F1 IEEE 1451.4 Thermocouple Template

Function	Select	Property	Description	Bits	Data Type	Units
ID	—	TEMPLATE	Template ID	8	Integer (Value=37 in this case)	—
Measurement	—	%MinPhysVal	Minimum Temperature	11	ConRes (–200 to 1,846, step 1)	°C
	—	%MaxPhysVal	Maximum Temperature	11	ConRes (–200 to 1,846, step 1)	
Electrical Signal Output	—	%ElecSigType	Electrical Signal Type	—	Assign = 2, “Resistance Sensor”	—
	—	%MinElecVal	Minimum Electrical output	11	ConRes (0 to 2.05 k, step 1)	Ω
	—	%MaxElecVal	Maximum Electrical output	13	ConRes (0 to 8.2 k, step 1)	Ω
	—	%MapMeth	Mapping Method	—	Assign = 5, “RTD”	—
	Select Case—R0 Resistance			2	Select Case	—
	Case 0	%RTDCoef_R0	Resistance R0	—	Assign = 100.0	Ω
	Case 1	%RTDCoef_R0	Resistance R0	—	Assign = 120.0	Ω
	Case 2	%RTDCoef_R0	Resistance R0	—	Assign = 1000.0	Ω
	Case 3	%RTDCoef_R0	Resistance R0	20	ConRelRes (1 to 12.5k, ±4.5 ppm)	Ω
	Select Case—RTD Curve (Callendar-Van Dusen Coefficients)			3	Select Case	—
	Case 0	%RTDCoef_A	CVD Coefficient A	—	Assign = 3.8100E-3	1/C
		%RTDCoef_B	CVD Coefficient B	—	Assign = –6.0200E-7	$1/C^2$
		%RTDCoef_C	CVD Coefficient C	—	Assign = –6.000E-12	$1/C^3$
	Case 1	%RTDCoef_A	CVD Coefficient A	—	Assign = 3.9083E-3	1/C
		%RTDCoef_B	CVD Coefficient B	—	Assign = –5.7750E-7	$1/C^2$
		%RTDCoef_C	CVD Coefficient C	—	Assign = –4.183E-12	$1/C^3$
	Case 2	%RTDCoef_A	CVD Coefficient A	—	Assign = 3.9692E-3	1/C
		%RTDCoef_B	CVD Coefficient B	—	Assign = –5.8495E-7	$1/C^2$
		%RTDCoef_C	CVD Coefficient C	—	Assign = –4.229E-12	$1/C^3$
	Case 3	%RTDCoef_A	CVD Coefficient A	—	Assign = 3.9739E-3	1/C
		%RTDCoef_B	CVD Coefficient B	—	Assign = –5.8700E-7	$1/C^2$
		%RTDCoef_C	CVD Coefficient C	—	Assign = –4.39E-12	$1/C^3$
	Case 4	%RTDCoef_A	CVD Coefficient A	—	Assign = 3.9787E-3	1/C
		%RTDCoef_B	CVD Coefficient B	—	Assign = –5.8685E-7	$1/C^2$
		%RTDCoef_C	CVD Coefficient C	—	Assign = –4.160E-12	$1/C^3$
	Case 5	%RTDCoef_A	CVD Coefficient A	—	Assign = 3.9888E-3	1/C
		%RTDCoef_B	CVD Coefficient B	—	Assign = –5.915E-7	$1/C^2$
		%RTDCoef_C	CVD Coefficient C	—	Assign = –3.816E-12	$1/C^3$
	Case 6	%RTDCoef_A	CVD Coefficient A	13	ConRes (3.8E-3 to 4E-3, step 2.5E-8)	1/C
		%RTDCoef_B	CVD Coefficient B	10	ConRes (–6.1E-7 to –5.6E-7, step 5E-11)	$1/C^2$
		%RTDCoef_C	CVD Coefficient C	7	ConRes (–6E-12 to –3E-12, step 2.3E-14)	$1/C^3$
	Case 7	%RTDCoef_A	CVD Coefficient A	32	Single	1/C
		%RTDCoef_B	CVD Coefficient B	32	Single	$1/C^2$
		%RTDCoef_C	CVD Coefficient C	32	Single	$1/C^3$
	—	%RespTime	Sensor Response Time	6	ConRelRes (1E-6 to 7.9, ±15%)	s
Excitation Supply	—	%ExciteAmplNom	Excitation current, nom	8	ConRelRes (1E-6 to 120E-3, ±2.3%)	A
	—	%ExciteAmplMax	Excitation current, max	8	ConRelRes (1E-6 to 120E-3, ±2.3%)	A
Calibration	—	%CalDate	Calibration Date	16	DATE	—

Information	—	%CalInitials	Calibration initials	15	CHR5	—
	—	%CalPeriod	Calibration period	12	UNINT	days
Misc.	—	%MeasID	Measurement location ID	11	UNINT	—
			Total bits:	135 to251 bits		

Table F2 IEEE 1451.4 RTD Sensor Template

Enumeration of 'Select Case' values for Physical Measurand

Case	Physical Units	Case	Physical Units	Case	Physical Units
0	K	16	m	32	l/l
1	°C	17	mm	33	kg/s
2	strain	18	in	34	m3/s
3	microstrain	19	m/s	35	m3/hr
4	Newton	20	mph	36	gpm
5	lb	21	fps	37	cfm
6	kgf	22	radians	38	l/min
7	m/s2	23	degrees	39	RH
8	ga	24	radian/s	40	%
9	Nm/radian	25	rpm	41	V
10	Nm	26	Hz	42	V(rms)
11	oz-in	27	g/l	43	Amperes
12	Pa	28	kg/m3	44	Amperes(rms)
13	psi	29	mole/m3	45	Watts
14	Kg	30	mole/l		
15	G	31	m3/m3		

Table F3 Enumeration of 'Select Case' values for Physical Measurand

IEEE 1451.4 Templates and Template ID numbers

Type	Name of Template	Template ID
Transducer Type templates	Accelerometer/Force transducer w. const. curr. ampl.	25
	Charge amplifier (incl. attached accelerometer)	26
	Microphones w. built-in preamp.	27
	Microphone preamps. w. attached micr. or system	28
	Microphones (capacitive)	29
	High-level voltage output sensors	30
	Current loop output sensors	31
	Resistance sensors	32
	Bridge sensors	33
	AC linear/rotary variable differential transformer (LVDT/RVDT) sensors	34
	Strain gage	35
	Thermocouple	36
	Resistance temperature detectors (RTDs)	37
	Thermistor	38
	Potentiometric voltage divider	39
	Charge amplifier (incl. attached force transducer)	43
Calibration templates	Calibration table	40
	Calibration curve (polynomial)	41
	Frequency response table	42

Table F4 IEEE 1451.4 Templates and Template IDs

Appendix G – 1-Wire Signalling

The four types of 1-Wire signalling are explains as follows.

- Reset Sequence with Reset Pulse and Presence Pulse

  The reset sequence is also called initialisation sequence as it is required to start any communication with the slave devices. The initialisation sequence consists of a reset pulse and a presence pulse. The master issues the initialisation sequence by sending a reset pulse onto the 1-Wire signal line to set the slave device into the state ready for transmission, and then monitors the line for the presence pulse responded by the slave device indicating to the master that it is on the line and ready to operate. The master device sends the Reset Pulse by pulling the signal line low for at least 480µs and then releases the line and goes into receive mode and checks for the presence pulse. The slave device will be reset by the reset pulse, and when it detects the signal line is released it waits for 15 to 60µs and send a presence pulse by pulling the signal line low for 60 to 240µs to respond to the master.

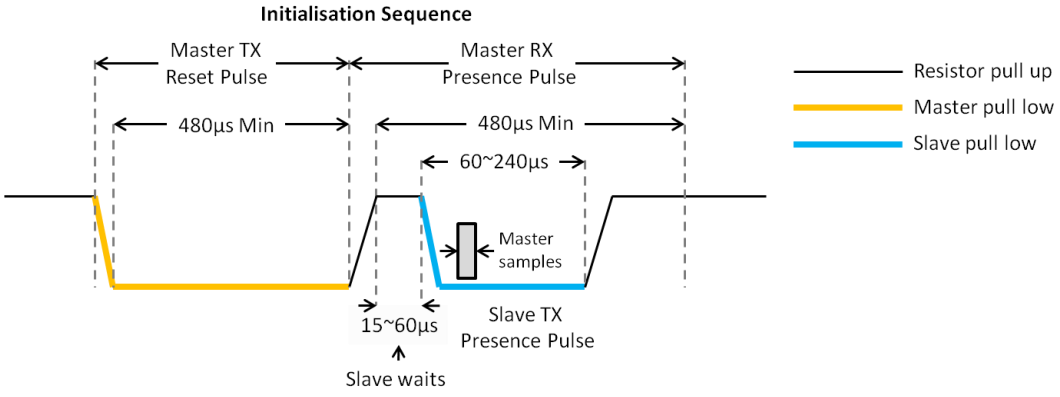


Figure G1 1-Wire Initialisation Sequence

- Write Time Slots (Write Signals)

The ‘Write 0’ and ‘Write 1’ time slots are used by the master to write bit ‘0’ and ‘1’ respectively to the slave device. The 1-Wire function commands and data sent from the master to the slave device consist of a serial of write time slots. The master device initiates both types of write signals by pulling data line low indicating to the slave device that a signal is started, then it keeps holding the line low for at least 60 µs to generate a ‘Write 0’ signal or it releases the line within 15 µs to produce a ‘Write 1’. The slave device samples the data line within the 15 to 60 µs window after the signal started and based on the state of the data line to decide whether a ‘0’ or a ‘1’ is transmitted by the master. The duration for both write signals has to be 60 µs minimum and 120 µs maximum, and between two sequential time slots there has to be at least 1 µs recovery time.

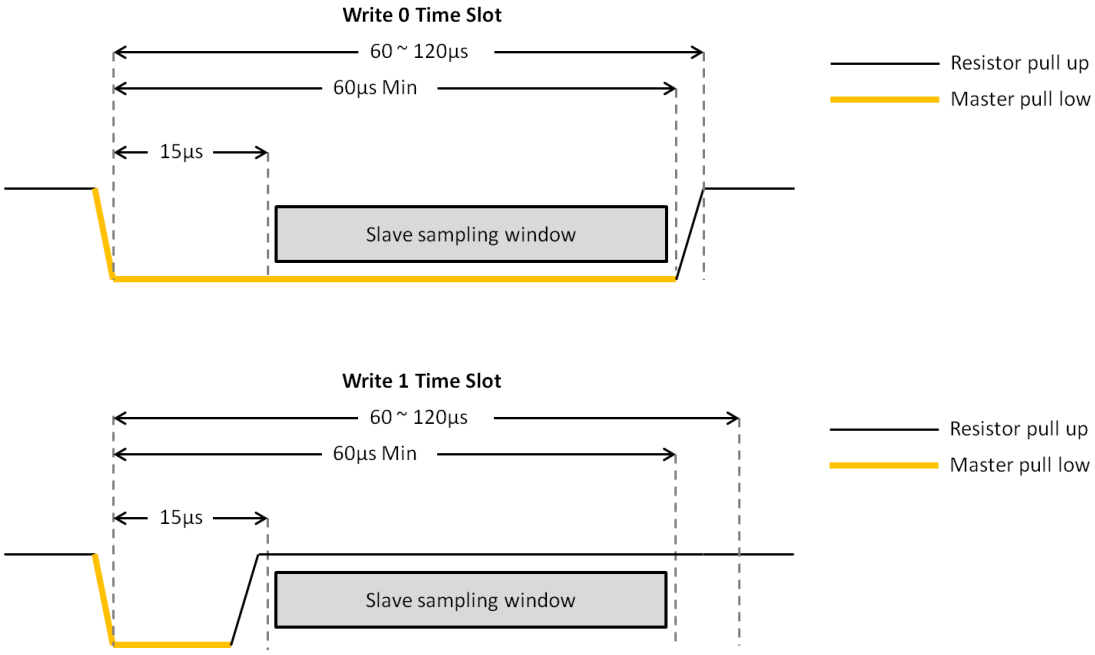


Figure G2 1-Wire Write Time Slots

- Read Data Time Slot (Read Signals)

  The ‘Read Data’ time slot is sent by the master device when the master needs to transmit data from the slave device to the master. The read signals are issued right after the master write a certain function command that needs a reply from the slave device. Slave devices cannot initiate transmissions, so only when the master issues the ‘Read

Data' time slot the slave device can respond and transmit data back to the master. Similar to write signals, the 'Read Data' time slot is also initiated by pulling the 1-Wire signal line low, and after 1 µs the master releases the line and the slave transmits one bit by pulling the line low as a '0' or leaving the line high to send a '1'. The master samples the state of the line to read the one bit data within in 15 µs after it initiates the read signal. The duration of the read signal also has to be in the range of 60 to 120 µs with 1 µs minimum recovery time.

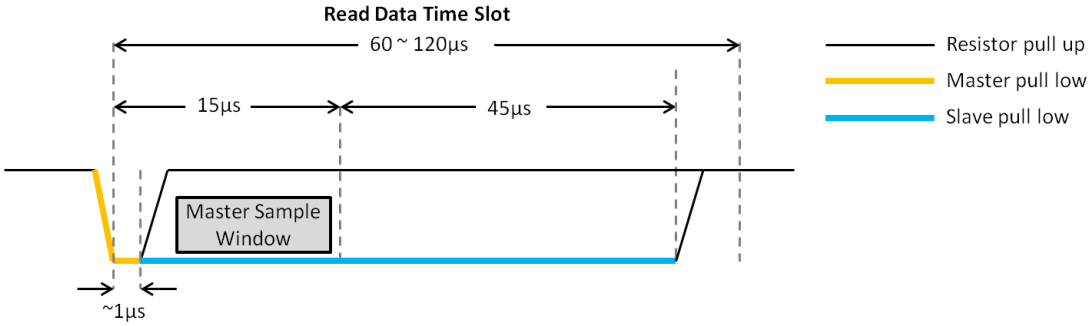


Figure G3 1-Wire Read Data Time Slot

Appendix H – PSoC Blocks Complete Circuit Diagrams

Continuous time (CT) block complete circuit diagram

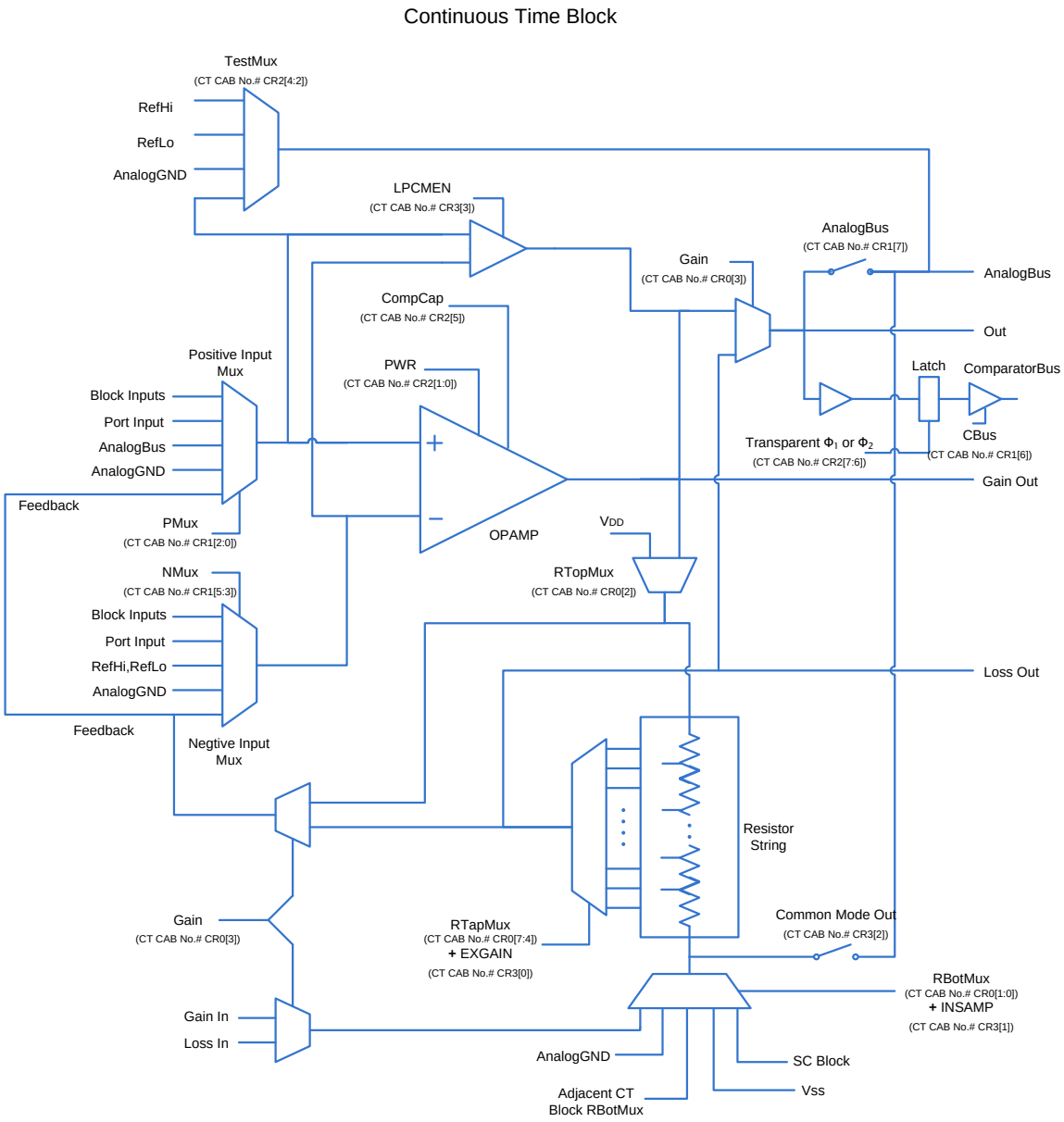


Figure H1 Continuous time (CT) block complete circuit diagram

Switch Capacitor (SC) ‘Type-C’ block complete circuit diagram

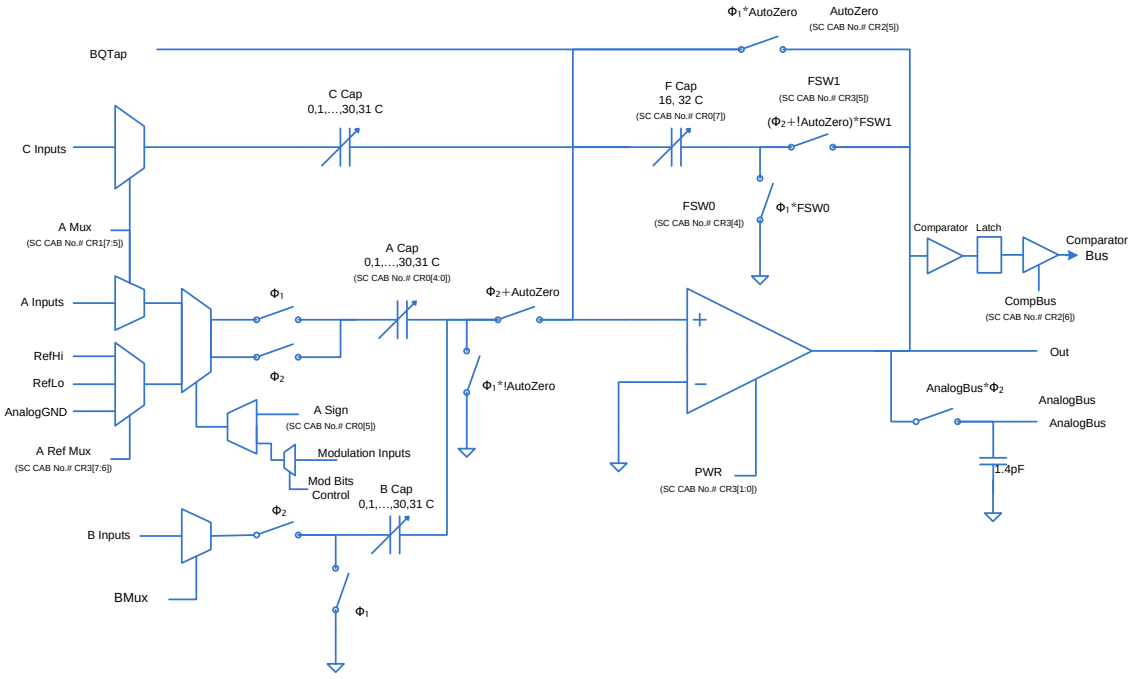


Figure H2 Switch Capacitor (SC) ‘Type-C’ block complete circuit diagram

Switch Capacitor (SC) ‘Type-D’ block complete circuit diagram

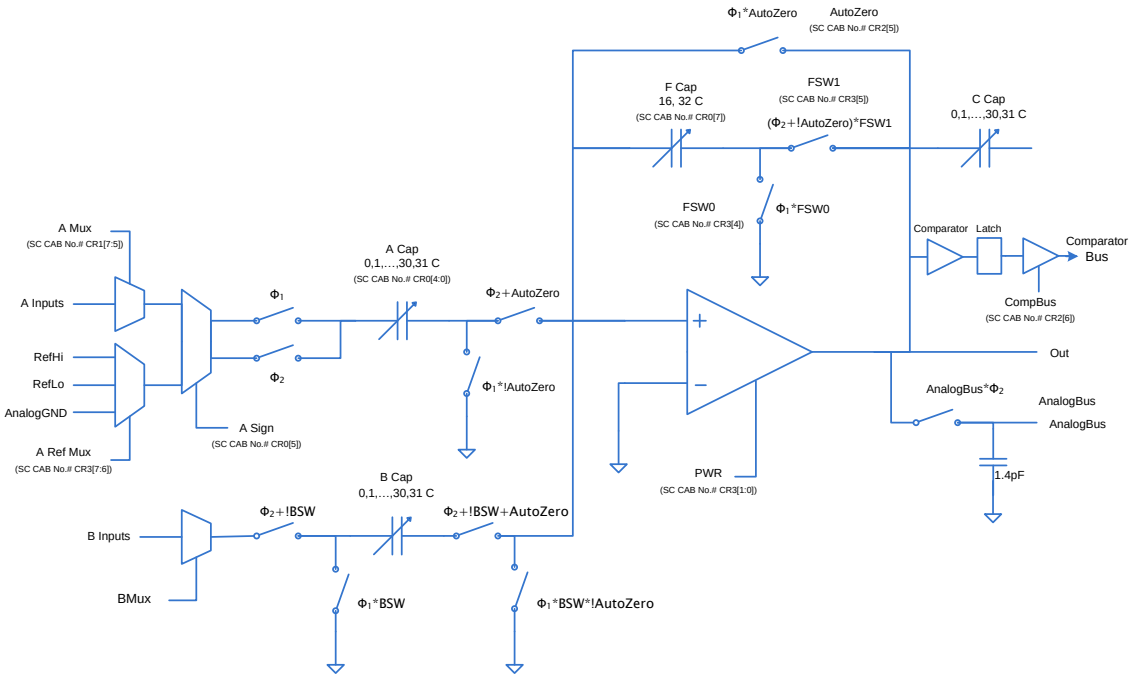


Figure H3 Switch Capacitor (SC) ‘Type-D’ block complete circuit diagram

Appendix I – Other Test Results

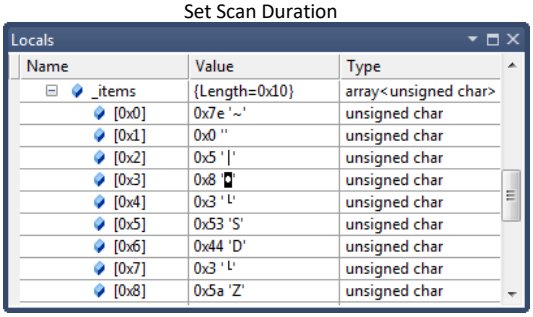
Theoretical AT Command frame for setting 'ScanChannels'

Frame Field		Offset	Value	Description
Start Delimiter		0	0x7E	
Length	MSB	1	0x00	Number of bytes between the length field and checksum, 12 in this example
	LSB	2	0x06	
Frame Type ID		3	0x08	AT Command frame type
Frame Number		4	0x02	Number of the frame, 0x02 in this example
Frame Payload	**AT Command**	5	0x53 (S)	AT command field. AT command "SC" is used to set or read Scan Channels.
		6	0x43 (C)	
	AT Command Parameter Value	7	0xF0	0xF000 – Scan ZigBee channel No. 23 to 26
		8	0x00	
Checksum		9	0x6F	0xFF minus the 8-bit sum of the bytes from offset 3 to the one before checksum

Table I1 Theoretical AT Command API Frame for Setting 'ScanChannel'

The data packets monitor for setting up other network parameters:

From the terminal application
to the coordinator module:
Set Scan Duration



From the PSoC microcontroller
to the end device module:
Set Scan Duration

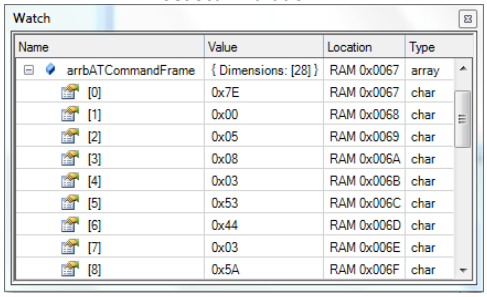


Figure I1 'Set Scan Duration' AT Command Frames

From the coordinator module
to the terminal application:
Response to Set Scan Duration

Locals

Name	Value	Type
receivedFrame	{Length=0x9}	array<unsigned char>
[0x0]	0x7e '~'	unsigned char
[0x1]	0x0 ''	unsigned char
[0x2]	0x5 '|'	unsigned char
[0x3]	0x88 ''	unsigned char
[0x4]	0x3 'L'	unsigned char
[0x5]	0x53 'S'	unsigned char
[0x6]	0x44 'D'	unsigned char
[0x7]	0x0 ''	unsigned char
[0x8]	0xdd 'Ý'	unsigned char

From the end device module
to the PSoC microcontroller:
Response to Set Scan Duration

Watch

Name	Value	Location	Type
arrRxBuffer	{ Dimensions: [64] }	RAM 0x0100	array
[0]	0x7E	RAM 0x0100	char
[1]	0x00	RAM 0x0101	char
[2]	0x05	RAM 0x0102	char
[3]	0x88	RAM 0x0103	char
[4]	0x03	RAM 0x0104	char
[5]	0x53	RAM 0x0105	char
[6]	0x44	RAM 0x0106	char
[7]	0x00	RAM 0x0107	char
[8]	0xDD	RAM 0x0108	char

Figure I2 'Set Scan Duration' AT Command Response Frames

From the terminal application
to the coordinator module:
Set Stack Profile

Locals

Name	Value	Type
_items	{Length=0x10}	array<unsigned char>
[0x0]	0x7e '~'	unsigned char
[0x1]	0x0 ''	unsigned char
[0x2]	0x5 '|'	unsigned char
[0x3]	0x8 ''	unsigned char
[0x4]	0x4 'J'	unsigned char
[0x5]	0x5a 'Z'	unsigned char
[0x6]	0x53 'S'	unsigned char
[0x7]	0x0 ''	unsigned char
[0x8]	0x46 'F'	unsigned char

From the PSoC microcontroller
to the end device module:
Set Stack Profile

Watch

Name	Value	Location	Type
arrbATCommandFrame	{ Dimensions: [28] }	RAM 0x0067	array
[0]	0x7E	RAM 0x0067	char
[1]	0x00	RAM 0x0068	char
[2]	0x05	RAM 0x0069	char
[3]	0x08	RAM 0x006A	char
[4]	0x04	RAM 0x006B	char
[5]	0x5A	RAM 0x006C	char
[6]	0x53	RAM 0x006D	char
[7]	0x00	RAM 0x006E	char
[8]	0x46	RAM 0x006F	char

Figure I3 'Set Stack Profile' AT Command Frames

From the coordinator module
to the terminal application:
Response to Set Stack Profile

Locals

Name	Value	Type
receivedFrame	{Length=0x9}	array<unsigned char>
[0x0]	0x7e '~'	unsigned char
[0x1]	0x0 ''	unsigned char
[0x2]	0x5 '|'	unsigned char
[0x3]	0x88 ''	unsigned char
[0x4]	0x4 'J'	unsigned char
[0x5]	0x5a 'Z'	unsigned char
[0x6]	0x53 'S'	unsigned char
[0x7]	0x0 ''	unsigned char
[0x8]	0xc6 'Æ'	unsigned char

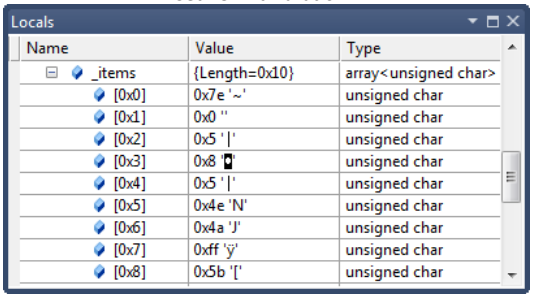
From the end device module
to the PSoC microcontroller:
Response to Set Stack Profile

Watch

Name	Value	Location	Type
arrRxBuffer	{ Dimensions: [64] }	RAM 0x0100	array
[0]	0x7E	RAM 0x0100	char
[1]	0x00	RAM 0x0101	char
[2]	0x05	RAM 0x0102	char
[3]	0x88	RAM 0x0103	char
[4]	0x04	RAM 0x0104	char
[5]	0x5A	RAM 0x0105	char
[6]	0x53	RAM 0x0106	char
[7]	0x00	RAM 0x0107	char
[8]	0xC6	RAM 0x0108	char

Figure I4 'Set Stack Profile' AT Command Response Frames

From the terminal application
to the coordinator module:
Set Permit Duration



From the PSoC microcontroller
to the end device module:
Set Permit Duration

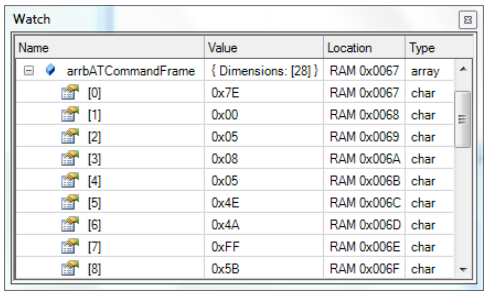
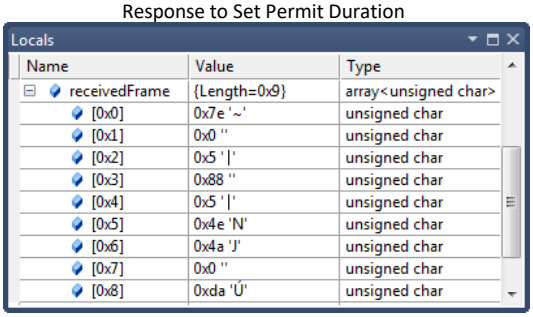


Figure I5 'Set Permit Duration' AT Command Frames

From the coordinator module
to the terminal application:
Response to Set Permit Duration



From the end device module
to the PSoC microcontroller:
Response to Set Permit Duration

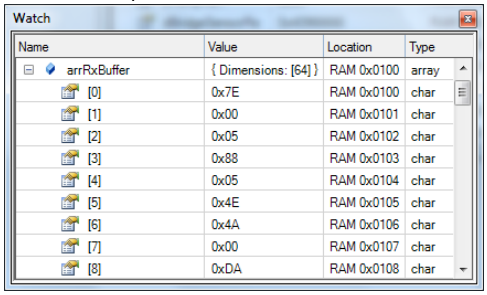


Figure I6 'Set Permit Duration' AT Command Response Frames

The HBM P8AP pressure sensor TEDS parsing results:

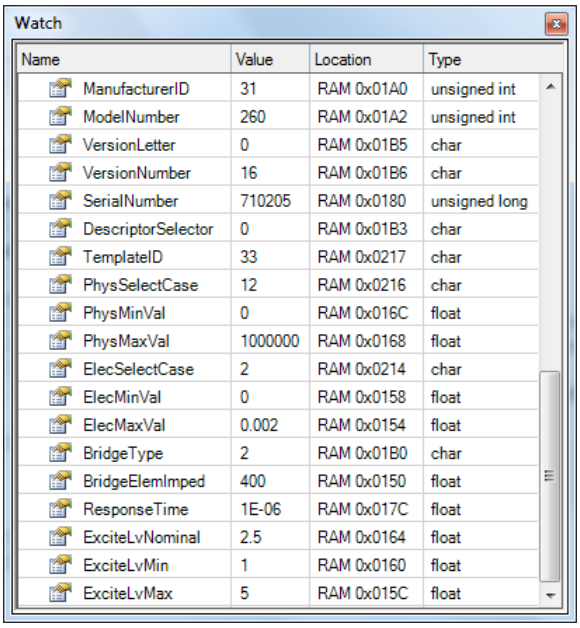


Figure I7 P8AP Pressure Sensor TEDS parsing results by the TEDS parsing routine

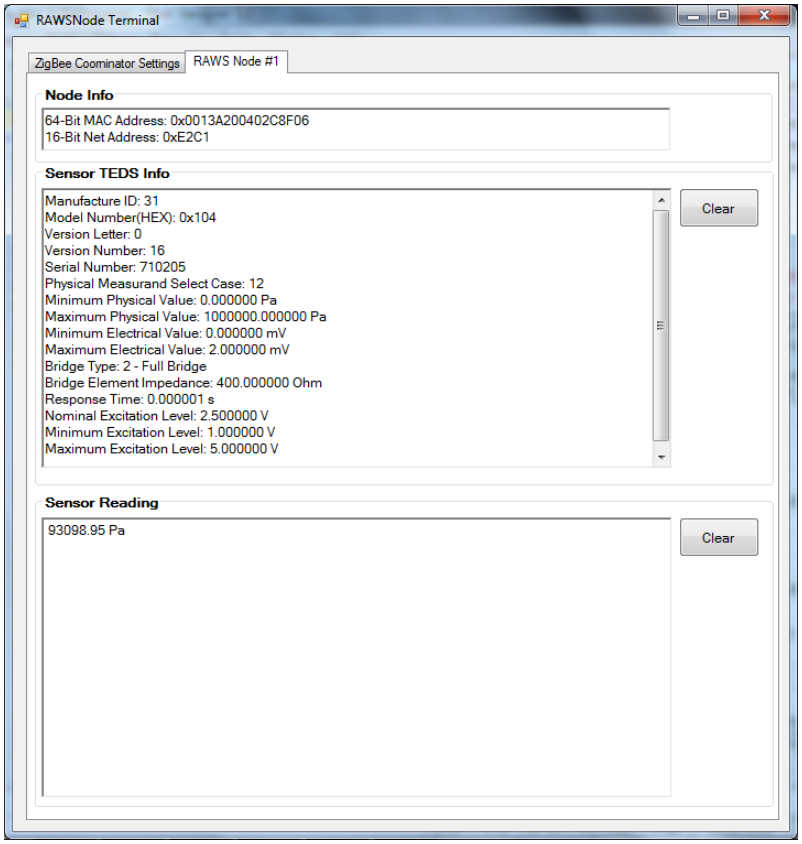


Figure I8 P8AP Pressure Sensor TEDS parsing results on RAWS node terminal application

The reconfiguration test results for the P8AP pressure sensor:

Watch

Name	Value	Location	Type
ASC12CR0	0x9D	IO Register Bank 0 0x0088	char
ASC12CR1	0x41	IO Register Bank 0 0x0089	char
ASC12CR2	0xA0	IO Register Bank 0 0x008A	char
ASC12CR3	0x3F	IO Register Bank 0 0x008B	char
ASD22CR0	0xAC	IO Register Bank 0 0x0098	char
ASD22CR1	0x80	IO Register Bank 0 0x0099	char
ASD22CR2	0x20	IO Register Bank 0 0x009A	char
ASD22CR3	0x33	IO Register Bank 0 0x009B	char
DACValue	0x00EB	RAM 0x0000	int

Figure I9 DAC control registers for P8AP pressure sensor in PSoC

Watch

Name	Value	Location	Type
ACB01CR0	0x0C	IO Register Bank 0 0x0075	char
ACB01CR1	0x21	IO Register Bank 0 0x0076	char
ACB01CR2	0x23	IO Register Bank 0 0x0077	char
ACB01CR3	0x03	IO Register Bank 0 0x0074	char
ACB00CR0	0x0C	IO Register Bank 0 0x0071	char
ACB00CR1	0x21	IO Register Bank 0 0x0072	char
ACB00CR2	0x23	IO Register Bank 0 0x0073	char
ACB00CR3	0x03	IO Register Bank 0 0x0070	char
ASD11CR0	0x1F	IO Register Bank 0 0x0084	char
ASD11CR1	0x1F	IO Register Bank 0 0x0085	char
ASD11CR2	0x20	IO Register Bank 0 0x0086	char
ASD11CR3	0x3B	IO Register Bank 0 0x0087	char

Figure I10 INS-AMP control registers for P8AP pressure sensor in PSoC

Watch

Name	Value	Location	Type
ASC10CR0	0xC8	IO Register Bank 0 0x0080	char
ASC10CR1	0x20	IO Register Bank 0 0x0081	char
ASC10CR2	0x00	IO Register Bank 0 0x0082	char
ASC10CR3	0xE3	IO Register Bank 0 0x0083	char
ASD20CR0	0xD0	IO Register Bank 0 0x0090	char
ASD20CR1	0x00	IO Register Bank 0 0x0091	char
ASD20CR2	0x40	IO Register Bank 0 0x0092	char
ASD20CR3	0xE3	IO Register Bank 0 0x0093	char
DEC_CR2	0xA8	IO Register Bank 1 0x00E7	char
DBB01DR2	0x01	IO Register Bank 0 0x0026	char
DBB00DR2	0xFF	IO Register Bank 0 0x0022	char

Figure I11 Delta Sigma ADC control registers for P8AP pressure sensor in PSoC

Signal (mV)	Theoretical Result (N)	RAWS Node Reading (N)									
0.516	10.32	10.40	10.38	10.01	10.04	10.11	10.34	10.19	10.27	10.47	10.21
1.009	20.18	20.22	20.01	20.28	20.41	20.30	20.09	19.94	19.84	20.09	20.26
1.508	30.16	29.91	30.11	30.24	30.19	30.01	30.22	30.33	30.30	30.22	30.13
2.012	40.24	40.11	40.10	40.35	40.42	40.34	40.18	40.02	40.09	40.18	40.22

Table I2 Complete Signal Chain Test Results of C9B Force Sensor (Part 2)

101-K Thermocouple (°C)	Reference Thermometer (°C)	101-K Thermocouple (°C)	Reference Thermometer (°C)
17.63	17.6	17.76	17.8
17.69	17.6	17.74	17.7
17.65	17.6	17.79	17.8
17.76	17.7	17.86	17.8
17.72	17.7	17.86	17.8
17.66	17.7	17.88	17.9
17.60	17.6	17.96	17.9
17.66	17.7	17.91	17.8
17.74	17.7	17.91	17.9
17.71	17.8	17.98	18.0

Table I3 Data Conversion Test Results of 101-K Thermocouple (Part 2)

101B-10 RTD Sensor (°C)	Reference Thermometer (°C)	101B-10 RTD Sensor (°C)	Reference Thermometer (°C)
19.41	19.3	19.52	19.5
19.32	19.4	19.49	19.5
19.31	19.4	19.55	19.5
19.39	19.4	19.51	19.6
19.44	19.4	19.59	19.6
19.41	19.5	19.61	19.6
19.52	19.5	19.66	19.6
19.48	19.5	19.65	19.7
19.48	19.4	19.75	19.7
19.44	19.5	19.68	19.7

Table I5 Data Conversion Test Results of 101-10B RTD Sensor (Part 2)

Appendix J – RAWS Research Publication

Reconfigurable Adaptive Wireless Sensor Node Technology using IEEE 1451.4 Standard

Rentao Wang, Ken Deevy
Microelectronics & Sensors Group
Waterford Institute of Technology
Waterford, Ireland
rwang@wit.ie, kdeevy@wit.ie

***Abstract*—This research paper describes the development of a highly flexible autonomic wireless sensor node technology that allows sensor nodes of wireless sensor networks (WSN) to adaptively accommodate different analogue sensors in a flexible and scalable way by utilising the reconfigurability of mixed-signal programmable hardware technology in combination with a sensor identification scheme inspired by IEEE1451.4 standard and the adaptive reconfiguration techniques developed in the research. The paper will illustrate the proposed technology and present the development of prototype sensor nodes that validate the key research concepts.**

Keywords—WSN; reconfigurable; adaptive; autonomic; IEEE1451.4

I. INTRODUCTION

Wireless sensor network (WSN) technology has been playing an increasingly important role in various applications involved in all parts of people's life in recent years. Having the ability to interact with the physical environment in addition to wireless communication capability, WSNs are powerful platforms for supporting many different applications and they have been employed in a wide range of real-world scenarios including scientific research, health care, environmental monitoring, public utilities, industry, security and surveillance, sustainable technologies, and military applications [1] [2].

Many WSN applications require the sensor node to support various sensors for different measurements as the sensors on each sensor node are the actual interface between the physical environment and the WSN system, and in some cases new sensors must be added to the existing system when additional measurement functions are required. For example, in a basic environmental monitoring experiment, the sensor node may need to support temperature sensing, relative humidity sensing, ambient light sensing, and to expand this system to a comprehensive monitoring system other sensing functions such as atmospheric pressure sensing, ultraviolet radiation sensing, and CO_2 sensing may be required also.

In most cases the inherent characteristics of WSNs require sensor nodes to be low power, low cost, and small size. Nowadays, analogue sensors are still the most common kind in use because they are low cost, robust and reliable, and they can be deployed in harsh environments where digital sensors cannot survive. However, analogue sensors vary greatly in their characteristics and provide non-standardised output signals. The traditional approach to support a number of sensors (generally the sensors discussed for the rest of this paper are analogue sensors except the ones explicitly stated as digital sensors) is to employ different types of analogue signal conditioning circuits and ADCs, as shown in Fig.1. But this approach will significantly increase the complexity and power consumption as well as the size and the cost of the sensor nodes, and this type of system can lack flexibility and scalability when new or additional sensors are required.

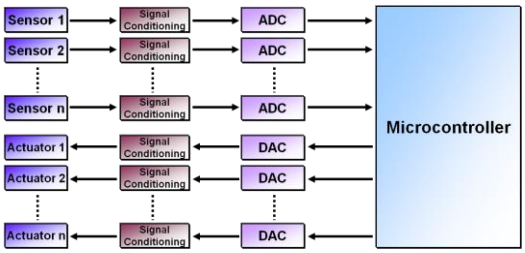


Fig. 1. Traditional approach for supporting multiple sensors.

In this context, the reconfigurable adaptive wireless sensor (RAWS) node research takes the reconfigurability and adaptability as the key considerations to solve this multi-sensing challenge, and develops a technology that can support or accommodate different types of sensors within each sensor node of the WSN in a flexible, autonomic and scalable way while not compromising power consumption, size and cost.

II. RECONFIGURABLE ADAPTIVE WIRELESS SENSOR NODE

The reconfigurable adaptive wireless sensor node research has devised a wireless sensor node technology that can adaptively reconfigure itself to accommodate different analogue sensors. The research employs programmable mixed-signal hardware as the enabling technology for implementing the reconfigurable adaptive wireless sensor node concept. The research combines the reconfigurability of the programmable hardware technology with the sensor identification scheme which is based on the IEEE 1454.4 standard, and has developed three adaptive reconfiguration techniques which utilise the sensor identification information to realise a WSN sensor node technology with autonomic sensor adaptation capability i.e. autonomic multi-sensing capability. The detailed research concept design and development is described in the following three subsections.

A. *Programmable Mixed-Signal Hardware Technologies*

Programmable mixed-signal (or custom CMOS) hardware with on-the-fly reconfigurability is the enabling technology for implementing the research concept. Compared to the traditional way that needs different types of hardware for different sensors, programmable hardware can be reconfigured in real-time to adapt to a number of different sensors for realising the corresponding functions so that the hardware resources can be optimally utilised.

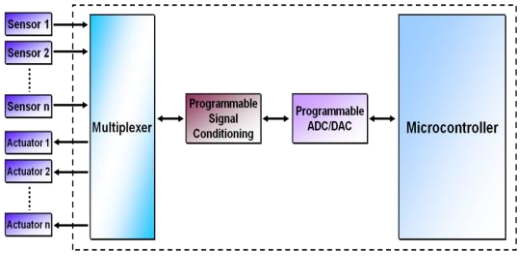


Fig. 1. Programmable mixed signal hardware technology for supporting multiple sensors.

As shown in Fig.2, by reconfiguring the hardware structures, the same programmable mixed signal circuits can be used for supporting different types of sensors. They also have the potential to support new sensors that are not included in the original design, avoiding redesigning the whole system from scratch when new sensors are needed. This can greatly increase the flexibility and scalability of the system, while reducing the complexity, size and cost.

For the purpose of research concept proving and prototyping, the two major commercially available programmable hardware technologies in the analogue domain, the field programmable analogue arrays (FPAA) and programmable mixed signal system on chip technology (e.g. PSoC – Programmable System on Chip) have been considered. The new FPGA technology incorporating mixed-signal capability has also been considered but its analogue capability is limited and not adequate for the purpose of this research. Both FPAA and PSoC technologies are capable of providing all the common components for supporting different analogue sensors, but compared to FPAA, PSoC possesses attributes which are ideally suitable for WSN applications. PSoC technology has been therefore chosen for this research for the following reasons:

1. Far lower power consumption and cost than FPAA.
2. PSoC has a built-in CPU subsystem with SRAM, EEPROM, and flash memory in addition to configurable analogue blocks while FPAA requires an external microcontroller. This should significantly reduce the complexity of the sensor node as well as the overall cost and power consumption.
3. PSoC contains programmable digital blocks that can be reconfigured into different digital and communication peripheral functions such as timers, PWMs, I2C, SPI, UART, etc. This enables better flexibility, scalability and applicability of the system, and digital sensors can also be easily accommodated if needed.

B. *Sensor Identification using IEEE 1451.4 Standard*

The RAWS node research further exploits the reconfigurability of the programmable hardware to develop a WSN sensor node technology that can autonomically perform hardware reconfiguration to adapt to different sensors. To achieve this adaptability, a sensor identification scheme must be first developed so that the RAWS node can reconfigure itself accordingly after the connected sensor is recognised. Although there are alternative methods, the most practical approach to realise a scheme that can identify a large number of sensor types and models is to utilise the sensor self-identification and self-description capability provided by the IEEE 1451.4 standard.

IEEE 1451.4 standard is a member of the IEEE 1451 family of open, industry consensus standards which aims to bring smart features into transducers. The standard defines the transducer electronic data sheet (TEDS) format and a mixed-mode interface (MMI) [3]. An IEEE 1451.4 TEDS contains the manufacturer ID, model number, version number, and serial number for the transducer, and also describes the important attributes of the transducer, such as transducer type, physical measurand, measurement range, sensitivity, electrical output range, etc. The mixed-mode interface described by IEEE 1451.4 retains the traditional analogue interface which provides a signal representing a physical phenomenon and contains a low-cost digital interface which can be used to read the TEDS information embedded in a memory device, for example an EEPROM, attached to the sensor.

IEEE 1451.4 standard has been supported by many sensor and DAQ system manufacturers and is becoming an important standard in the smart sensor area because while other smart sensor technologies do exist, IEEE 1451.4 is unique as it maintains the analogue output of the sensor [4]. The concept of MMI adapts the standard for seamless use with the existing analogue sensors and allows all the advantages of the analogue sensor to be kept, while the TEDS information which describes the identity, type, operation, and attributes of the transducer [3] provides a straightforward way for bringing in smart features and adding self-description and self-identification capability to traditional analogue sensors. This means TEDS features or virtual TEDS features can be easily added to existing *non-smart* sensors allowing them to be identified. All these features indicate that the IEEE 1451.4 standard ideally fits into the context of this research to provide the necessary sensor identification information that the RAWS node needs for adaptive reconfiguration.

The combination of reconfigurability and the IEEE standard has the potential to produce a highly capable autonomic sensor node technology, ideal for mass deployment in wireless sensor networks. Therefore, the RAWS research exploits the self-description and self-identification capability provided by the IEEE 1451.4 standard in combination with the reconfigurability of new programmable mixed signal hardware technologies. By reading the TEDS, the RAWS sensor node is able to determine the connected IEEE 1451.4-compatible sensors or even virtual IEEE 1451.4 sensors and adaptively

sensors or even virtual IEEE 1451.4 sensors and adaptively perform the reconfiguration via either an intelligent algorithm or a configuration file from a local or remote database to support different sensors as illustrated in the figure below.

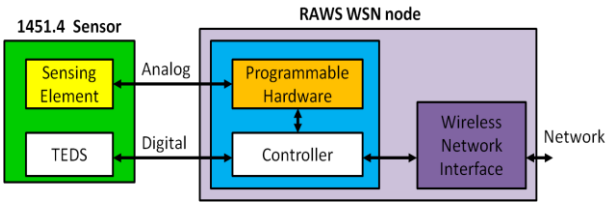


Fig. 1. Basic RAWS node block diagram.

A. *Adaptive Reconfiguration Techniques*

Once the attached sensor has been identified, how to reconfigure the hardware is the key to realising the autonomic multi-sensing capability. The RAWS research has developed three techniques to perform the adaptive reconfiguration.

1. An intelligent algorithm based on the TEDS information. The software algorithm intelligently selects suitable peripheral function modules and decides suitable parameters for the function modules for different sensors based on the important TEDS attributes such as sensor type, electrical output range, etc. The intelligent algorithm method can generally support a large number of sensors, and the configuration created by the algorithm can be stored for later use.

2. Local configuration settings or files. The RAWS node based on the manufacturer ID, model number, and maybe also the version letter and version number in the TEDS, loads the corresponding configuration setting or file which is predesigned for the sensor or backed-up from the previous knowledge of the same sensor from the local memory.

3. Remote configuration settings or files. Due to the limited memory space, normally the WSN sensor node cannot preserve a large number of configuration settings locally, and this could limit the number of sensors supported by the node platform. Utilising the wireless communication, the configuration settings or files can be stored in a remote database, for example a configuration file database located on the computer connected to the WSN gateway, and transferred to the WSN sensor node when needed. This method can avoid the local memory limitation problem and the remote configuration database has more flexible upgradability than the local memory to support sensors for which there is no corresponding configuration setting. The disadvantage of this method is, because wireless communications require high power consumption, frequent wireless communications can greatly reduce the life time of the sensor node.

In the actual design the three methods can be combined together to achieve a highly adaptive reconfiguration scheme.

B. *RAWS Node Research Concept Conclusion*

In conclusion, the reconfigurable adaptive wireless sensor node research develops a WSN sensor node technology that not only enables the sensor node to support different analogue sensors by adaptively reconfiguring itself utilising the reconfigurability of programmable mixed-signal hardware, but also realises an autonomic multi-sensing capability, a type of sensor *plug & play* vision in the context of wireless sensor node technology. The illustrations of the key research concepts and the RAWS node autonomic adaptive reconfiguration operation sequence are shown in the Fig.4 and Fig.5 respectively.

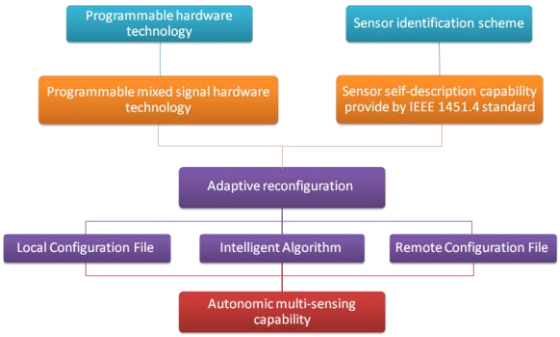


Fig. 2. Illustration of the key research concepts.

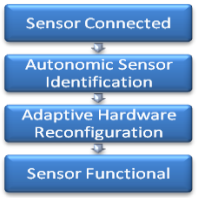


Fig. 3. Illustration of the adaptive reconfiguration sequence.

II. STATE-OF-THE-ART REVIEW

Programmable hardware technologies have been used in some WSN applications. However, there is a limited amount of research exploiting programmable hardware to adaptively support different sensors especially analogue sensors in the WSN area. In [5] a WSN node platform called *Cookie* has been developed using field programmable gate array (FPGA) technology, as a result the *Cookie* platform is only able to support digital sensors. Similar FPGA-based sensor node platforms which do not have analogue capabilities are presented in [6] [7]. In [8], the *SWAMs* sensor node uses FPAA to support analogue sensors, but the reconfiguration process is executed in manual operation i.e. the *SWAMs* platform is not able to autonomically reconfigure itself to adapt to different sensors. Also as discussed in section II.A, FPAA has high power consumption which will greatly reduce the lifetime of the sensor node and is not ideal for WSN applications. [9] and [10] present two similar reconfigurable interface devices for converting traditional analogue sensors to digital formats with custom TEDS inspired by IEEE1451 standard for use with a network capable processor unit designed for working with the interfaces. Both devices are interposed between the analogue

converting traditional analogue sensors to digital formats with custom TEDS inspired by IEEE1451 standard for use with a network capable processor unit designed for working with the interfaces. Both devices are interposed between the analogue sensor and the processor unit, to produce a type of programmable plug & play capability. A significant shortcoming of this kind of system however, is the fact that the interface device must be manually pre-programmed depending on the type of analogue sensor that will be connected to it. The plug & play capability only applies to the specific designed processor unit which loses the standardisation goal pursued by the RAWS solution.

In terms of exploring the merging of programmable hardware and the IEEE 1451.4 standard, this review only identifies one piece of relevant research [11]. But the system presented in [11] does not support wireless communications because it uses an FPAA as an analogue signal conditioning device and an FPGA as an FGAA controller, and this combination leads to very high power consumption and makes the system not practical for WSN applications. Also in [11] no intelligent adaptive reconfiguration technique is presented. No research work has been reported that combines the reconfigurability of programmable hardware technologies and the self-identification feature of IEEE 1451.4 to realise an autonomic and adaptive sensor multi-sensing capability in low power wireless sensor nodes as the RAWS research contributed.

I. RAWS Node Architechrue, Prototype Design and Implementation

The design and implementation of the RAWS sensor node platform with the autonomic adaptive reconfigurability is ongoing and several prototypes have been developed. The architecture of the RAWS node platform prototype and the diagram of the overall network system are shown in the Fig.6.

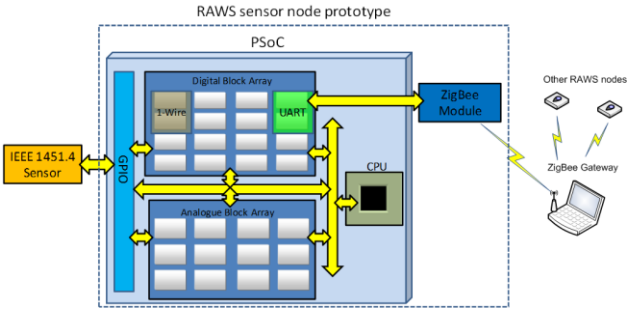


Fig. 1. RAWS node prototype architecture.

A. Hardware Architecture

As described in Section II-A, a PSoC device contains not only a reconfigurable analogue block array, but also a CPU subsystem with memory, and a digital block array. So as shown in the diagram the PSoC covers all the controller, processing, analogue and digital peripheral functions in one single chip. The prototypes use the Cypress CY8C29466 PSoC chip that features 16 digital blocks, and 4 continuous-time (CT) and 8 switched-capacitor (SC) analogue blocks. This PSoC chip has low power consumption with typical supply current of 2mA in active mode and 3µA in sleep mode.

Because the 1-Wire protocol is specified as the digital communication standard for transferring TEDS data in the IEEE1451.4 standard, a 1-Wire communication module is always configured in the design consisting of two digital blocks to read the TEDS information from the sensors.

In terms of wireless communications, while the focus of RAWS research is not on the networking aspect of WSNs, wireless communications is the foundation of a WSN. And to establish a robust and reliable WSN while achieving important features such as good energy efficiency and long life time, and network flexibility, the wireless communication technology plays a significant role. ZigBee is selected for the research as it is designed for low power, low cost, and low data rates wireless networks and its other inherent features such as mesh network topology are important for establishing a flexible and scalable wireless sensor network infrastructure. The ZigBee transceivers used in the RAWS node prototypes are the Digi XBeeZB modules. The PSoC microcontroller interfaces with the ZigBee module via an UART module which consists of two digital blocks.

The analogue blocks and the remaining digital blocks can be adaptively configured and reconfigured as different function modules depending on the connected sensors. Four IEEE 1451.4-enabled sensors, HBM C9B/50 force transducer, HBM P8AP/218 pressure sensor, Weed Instruments 101-K K type thermocouple and 101-10B Pt1000 RTD, have been used to help to design and test the RAWS node prototype. At the moment, the first two adaptive reconfiguration techniques describe in section II-C, the TEDS-based intelligent algorithm and local configuration settings, have been implemented in two prototypes, RAWS node #1 and #2, which have the same hardware architecture described above. The following two subsections explain the design of these two adaptive reconfiguration techniques using the C9B/50 force sensor, the 101-K thermocouple, and the 101-10B RTD as examples.

B. TEDS-based Intelligent Algorithm Technique

The C9B/50 is a full bridge type sensor and has a measurement range of 0 to 50N. The sensitivity is 1mV/V. The mapping method from the physical value to the electrical output is linear mapping. The minimum, nominal and maximum excitation levels are 1.0V, 2.5V and 5.0V respectively. For the RAWS node, this information is available in the TEDS.

RAWS node #1 reads the C9B/50 sensor's TEDS information, and the TEDS-based intelligent algorithm will be able to decide that the following analogue modules are needed for supporting the sensor and load them one by one to build-up the configuration for the C9B/50 sensor:

1. A DAC module consisting of two SC blocks for supplying the excitation voltage;
2. An instrumentation amplifier consisting of two CT blocks and one SC block for amplifying the differential voltage output;

1. A Delta-Sigma ADC consisting two SC blocks to convert the sensor output.

When the necessary modules are loaded, the algorithm selects the proper DAC output voltage and amplifier gain value based on the nominal excitation level and the sensitivity parameters in TEDS. Then the physical reading is determined and calculated based on parameters including the physical measurand and measurement range, mapping method, sensitivity, DAC output, and amplifier gain. Finally the measurement results are sent via the ZigBee transceiver to the gateway and displayed on the computer.

If replacing the C9B/50 force sensor with the 101-K K type thermocouple to the RAWS node, the algorithm identifies that a different sensor is connected by reading the TEDS and then decides that an instrumentation amplifier and a Delta-Sigma ADC are needed for this thermocouple. As these two modules are already loaded due to the previous force sensor, the algorithm will only unload the DAC. The Algorithm will then re-select the amplifier and ADC parameters for the thermocouple and use the corresponding coefficients for K type thermocouple to calculate the temperature readings.

When replacing the 101-K thermocouple with 101-10B Pt1000 RTD, because the resistance of the RTD is measured against a reference resistor, the ADC needs to measure two different voltages through the internal multiplexer in the PSoC. The Delta Sigma ADC is an integrating converter, so when changing multiplexed inputs it needs two sample delays to adapt to the new input and generate the correct reading. So for the RTD the intelligent algorithm will pick the incremental ADC. And because no differential voltage is needed to be measured, a single-input programmable gain amplifier (PGA) is adequate. However the algorithm has to unload the Delta-Sigma ADC and instrumentation amplifier first, and then load the PGA and incremental ADC one by one. This will take four steps to accomplish the final configuration for the RTD.

The whole adaptive reconfiguration process using the TEDS-based intelligent algorithm technique to accommodate the three different sensors is shown in Fig.7.

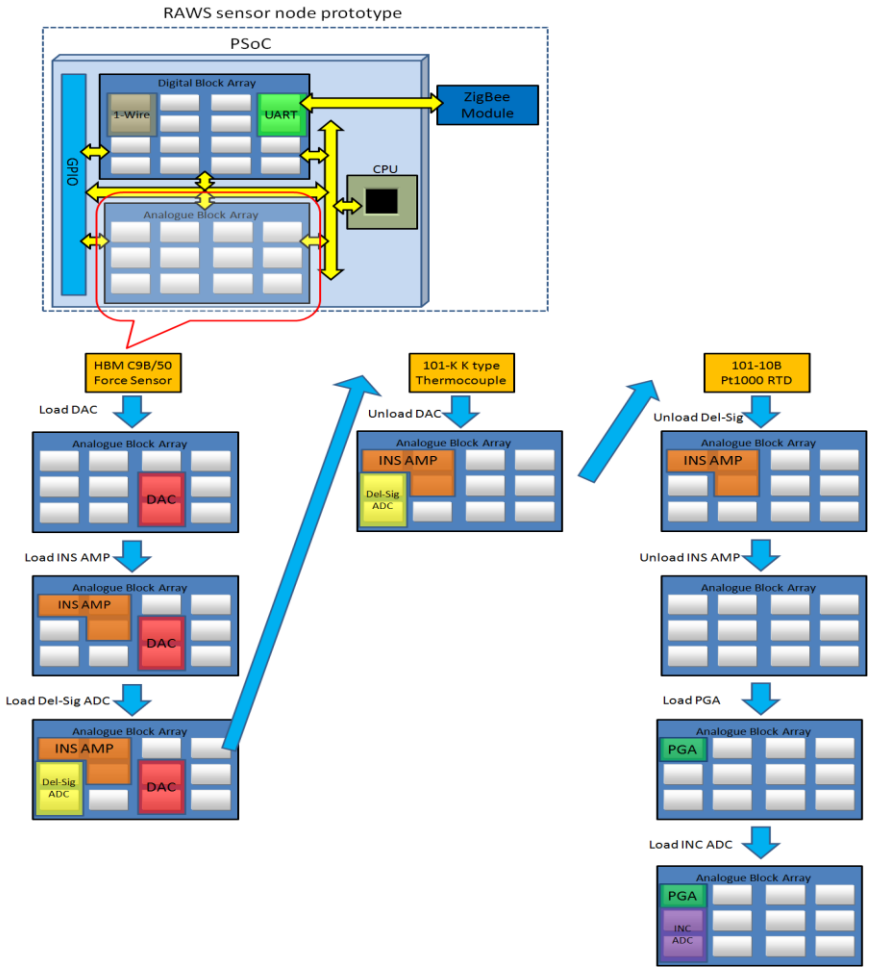


Fig. 1. Apative reconfiguration using TEDS-based intelligent algorithm.

A. *Local Configuration Settings Technique*

The RAWS node #2 equipped with the local configuration settings technique utilises the manufacturer ID, model number and version number in the TEDS to search and load the corresponding configuration setting for the connected sensor. In this case, the configuration settings for the C9B/50 force sensor, the 101-K thermocouple, and the 101-10B RTD sensors are already pre-designed and stored in PSoC, and the parameters for each of the analogue modules are also preset within each configuration setting. So the RAWS node #2 only needs to load and unload the configurations to adapt to different sensors. The adaptive reconfiguration process using the local configuration setting technique to accommodate the three sensors is shown in Fig.8.

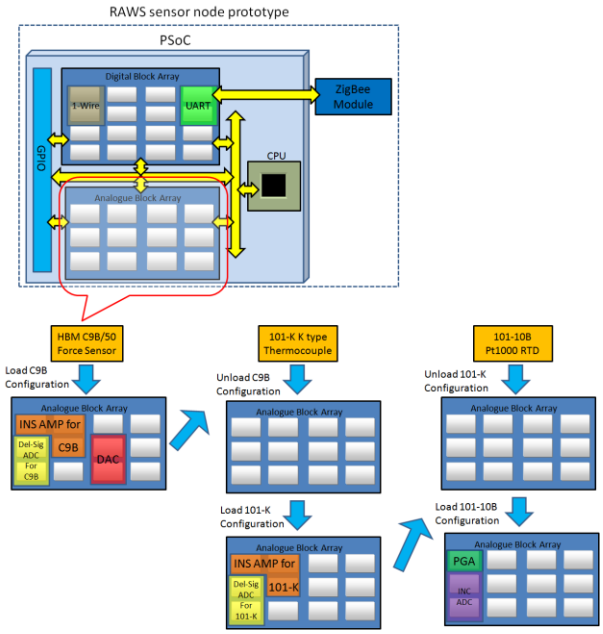


Fig. 1. Apative reconfiguration using local configuraion settings.

B. *Test Results and Comparison of the Two Techniques*

The test results show that the two RAWS node prototypes can successfully identify all four sensors and autonomically perform the adaptive reconfiguration to accommodate the sensor. The wireless networking and communications are fully functional.

The TEDS-based intelligent algorithm technique is more flexible as it is able to autonomically select and set suitable modules and parameters for different sensors, but its reconfiguration time can vary greatly depending on the current configuration and the subsequent sensor, and this could lead to high power consumption. In contrast the local configuration setting technique's behaviour is more predictable as all the configurations are pre-designed but for the same reason it lacks flexibility. Future work will try to combine the advantages of the two techniques to achieve a highly adaptive reconfiguration scheme.

II. Conclusion

The RAWS node research has developed a highly flexible autonomic WSN sensor node technology that can reconfigure its hardware in an autonomic way to interface with a large range of sensors to solve the multi-sensing challenge in many WSN applications where different types of sensors need to be supported or integrated within the sensor node in a flexible and scalable way. RAWS node prototypes have been developed to realise and prove the concept of RAWS node technology. For the future, more sensors will be included and tested to further develop the RAWS node platform.

Acknowledgment

The authors wish to thank the Irish Research Council for Science, Engineering & Technology for providing the Embark Initiative scholarship and making this research possible.

References

[1] J. Yick, *et al.*, "Wireless sensor network survey," *Computer Networks*, vol. 52, pp. 2292-2330, 2008.

[2] T. Arampatzis, *et al.*, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," in *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, 2005, pp. 719-724.

[3] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats," *IEEE Std 1451.4-2004*, pp. 0_1-430, 2004.

[4] *IEEE 1451.4 Standard Overview*. Available: http://zone.ni.com/devzone/cda/tut/p/id/3469

[5] Y. E. Krasteva, *et al.*, "Remote HW-SW reconfigurable Wireless Sensor nodes," in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, 2008, pp. 2483-2488.

[6] J. Portilla, *et al.*, "A Reconfigurable Fpga-Based Architecture for Modular Nodes in Wireless Sensor Networks," in *Programmable Logic, 2007. SPL '07. 2007 3rd Southern Conference on*, 2007, pp. 203-206.

[7] Y. Krasteva, *et al.*, "Embedded Run-time Reconfigurable Nodes for Wireless Sensor Networks Applications," *Sensors Journal, IEEE*, vol. PP, pp. 1-1, 2011.

[8] E. Mackensen, *et al.*, "Smart wireless autonomous microsystems (SWAMs) for sensor actuator networks," in *Sensors for Industry Conference, 2004. Proceedings the ISA/IEEE*, 2004, pp. 72-78.

[9] S. Vera, *et al.*, "A programmable plug&play interface for WSN applications," in *Sensors, 2011 IEEE*, 2011, pp. 1808-1811.

[10] V. Mattoli, *et al.*, "A Universal Intelligent System-on-Chip Based Sensor Interface," *Sensors*, vol. 10, pp. 7716-7747, 2010.

[11] D. P. Morales, *et al.*, "Merging FPGA and FPAA Reconfiguration Capabilities for IEEE 1451.4 Compliant Smart Sensor Applications," in *Programmable Logic, 2007. SPL '07. 2007 3rd Southern Conference on*, 2007, pp. 217-220.

References

[1] I. F. Akyildiz, *et al.*, "Wireless sensor networks: a survey," *Computer Networks,* vol. 38, pp. 393-422, 2002.

[2] T. Arampatzis, *et al.*, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," in *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, 2005, pp. 719-724.

[3] D. Culler, *et al.*, "Guest Editors' Introduction: Overview of Sensor Networks," *Computer,* vol. 37, pp. 41-49, 2004.

[4] IEC, "Internet of Things: Wireless Sensor Networks," 2014.

[5] ESSC, "The European Sensor Systems Cluster Roadmap Towards European Leadership in Sensor Systems - Survey of Industrial Needs," 2016.

[6] B. Betts. Smart Sensors - New standard could save lives and money. *IEEE Spectrum*.

[7] IFSA, "International Frequency Sensor Association Study 2016," 2016.

[8] K. Romer and F. Mattern, "The design space of wireless sensor networks," *Wireless Communications, IEEE,* vol. 11, pp. 54-61, 2004.

[9] J. Yick, *et al.*, "Wireless sensor network survey," *Computer Networks,* vol. 52, pp. 2292-2330, 2008.

[10] G. Tolle, *et al.*, "A macroscope in the redwoods," presented at the Proceedings of the 3rd international conference on Embedded networked sensor systems, San Diego, California, USA, 2005.

[11] P. Juang, *et al.*, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," presented at the Proceedings of the 10th international conference on Architectural support for programming languages and operating systems, San Jose, California, 2002.

[12] G. Werner-Allen, *et al.*, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE,* vol. 10, pp. 18-25, 2006.

[13] C. R. Baker, *et al.*, "Wireless Sensor Networks for Home Health Care," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, 2007, pp. 832-837.

[14] T. Naumowicz, *et al.*, "Wireless Sensor Network for habitat monitoring on Skomer Island," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, 2010, pp. 882-889.

[15] A. Mainwaring, *et al.*, "Wireless sensor networks for habitat monitoring," presented at the Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, USA, 2002.

[16] P. Bonnet, *et al.*, "Querying the physical world," *Personal Communications, IEEE,* vol. 7, pp. 10-15, 2000.

[17] P. Padhy, *et al.*, "Glacial Environment Monitoring using Sensor Networks," presented at the Real-World Wireless Sensor Networks, Stockholm, Sweden, 2005.

[18] *Building for the future*. Available: http://www.nextgenpe.com/article/Building-for-the-future/

[19] C. Kappler and G. Riegel, "A Real-World, Simple Wireless Sensor Network for Monitoring Electrical Energy Consumption," in *Wireless Sensor Networks*. vol. 2920, ed: Springer Berlin / Heidelberg, 2004, pp. 339-352.

[20] S. Tilak, *et al.*, "A taxonomy of wireless micro-sensor network models," *SIGMOBILE Mob. Comput. Commun. Rev.,* vol. 6, pp. 28-36, 2002.

[21] L. Krishnamurthy, *et al.*, "Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea," presented at the Proceedings of the 3rd

international conference on Embedded networked sensor systems, San Diego, California, USA, 2005.
[22] M. R. Akhondi*, et al.*, "Applications of Wireless Sensor Networks in the Oil, Gas and Resources Industries," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, 2010, pp. 941-948.
[23] I. A. Homles, "Making Maintenance Invisible," *AB Journal,* pp. 49-53, Sep 2004.
[24] F. Michahelles*, et al.*, "Applying wearable sensors to avalanche rescue," *Computers & Graphics,* vol. 27, pp. 839-847, 2003.
[25] K.-K. Yap*, et al.*, "MAX: human-centric search of the physical world," presented at the Proceedings of the 3rd international conference on Embedded networked sensor systems, San Diego, California, USA, 2005.
[26] P. Hung*, et al.*, "Wireless sensor networks for activity monitoring using multi-sensor multi-modal node architecture," in *IET Irish Signals and Systems Conference (ISSC 2009)*, 2009, pp. 1-6.
[27] OECD, "Smart Sensor Networks: Technologies and Applications for Green Growth," 2009.
[28] G. Simon*, et al.*, "Sensor network-based countersniper system," presented at the Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, 2004.
[29] U. Maurer*, et al.*, "eWatch: a wearable sensor and notification platform," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, 2006, pp. 4 pp.-145.
[30] *Sun SPOT*. Available: http://www.sunspotworld.com/products/
[31] E. A. Vittoz, "The Design of High-Performance Analog Circuits on Digital CMOS Chips," *IEEE Journal of Solid-State Circuits,* vol. 20, pp. 657-665, 1985.
[32] I. A. Grout, *Integrated Circuit Test Engineering: Modern Techniques*: Springer-Verlag, 2006.
[33] B. Dobkin and J. Williams, *Analog circuit design: A tutorial guide to applications and solutions*, 2011.
[34] *Cypress Programmable System-on-Chip*. Available: http://www.cypress.com/
[35] A. Gupta*, et al.*, "Rapid prototyping of AES encryption for wireless communication system on FPGA," in *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*, 2011, pp. 571-575.
[36] B. Benamrouche*, et al.*, "Ultra-low power IR-UWB transceiver for wireless sensors network," in *2017 International Semiconductor Conference (CAS)*, 2017, pp. 285-288.
[37] W. M. El-Medany, "FPGA implementation of RDR Manchester and D-Manchester CODEC design for Wireless Transceiver," in *Radio Science Conference, 2008. NRSC 2008. National*, 2008, pp. 1-5.
[38] *Wireless Sensor Networks Prototyping Platform 25mm*. Available: https://www.tyndall.ie/content/wireless-sensor-networks-prototyping-platform25mm
[39] K. Shahzad*, et al.*, "SENTIOF: An FPGA based high-performance and low-power wireless embedded platform," in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*, 2013, pp. 901-906.
[40] A. Hayek*, et al.*, "FPGA-based wireless sensor network platform for safety systems," in *Telecommunications (ICT), 2012 19th International Conference on*, 2012, pp. 1-6.
[41] H. Hinkelmann*, et al.*, "A Reconfigurable Prototyping Platform for Smart Sensor Networks," in *Programmable Logic, 2008 4th Southern Conference on*, 2008, pp. 125-130.
[42] G. Leelavathi*, et al.*, "RSA processor design with vedic multiplier for nodes in wireless sensor networks," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 1254-1257.

[43] T. T. O. Kwok and K. Yu-Kwong, "Computation and energy efficient image processing in wireless sensor networks based on reconfigurable computing," in *Parallel Processing Workshops, 2006. ICPP 2006 Workshops. 2006 International Conference on*, 2006, pp. 8 pp.-50.

[44] L. Fucai*, et al.*, "A Novel Partial Dynamic Reconfiguration Image Sensor Node for Wireless Multimedia Sensor Networks," in *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on*, 2012, pp. 1368-1374.

[45] B. d. Silva*, et al.*, "A Multimode SoC FPGA-Based Acoustic Camera for Wireless Sensor Networks," in *2018 13th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, 2018, pp. 1-8.

[46] G. G. Mplemenos and I. Papaefstathiou, "Fast and power-efficient hardware implementation of a routing scheme for WSNs," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, 2012, pp. 1710-1714.

[47] M. Turcanik, "Network routing by artificial neural network," in *Communications and Information Systems Conference (MCC), 2012 Military*, 2012, pp. 1-5.

[48] K. Parane*, et al.*, "Design of an Adaptive and Reliable Network on Chip Router Architecture Using FPGA," in *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2019, pp. 1-4.

[49] P. Hamalainen*, et al.*, "Configurable hardware implementation of triple-DES encryption algorithm for wireless local area network," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, 2001, pp. 1221-1224 vol.2.

[50] H. Hinkelmann*, et al.*, "A Domain-Specific Dynamically Reconfigurable Hardware Platform for Wireless Sensor Networks," in *Field-Programmable Technology, 2007. ICFPT 2007. International Conference on*, 2007, pp. 313-316.

[51] K. Ravi*, et al.*, "Implementation of ECC on FPGA for WSN- A Survey," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, 2019, pp. 0022-0026.

[52] Y. Sun*, et al.*, "Design of FPGA-Based Multimedia Node for WSN," in *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, 2011, pp. 1-5.

[53] T. Matić*, et al.*, "The receiver circuit for ultra-wideband integral pulse frequency modulated wireless sensor," in *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2016, pp. 1-4.

[54] A. Abbaszadeh and K. D. Sadeghipour, "A new hardware efficient reconfigurable fir filter architecture suitable for FPGA applications," in *Digital Signal Processing (DSP), 2011 17th International Conference on*, 2011, pp. 1-4.

[55] A. Shiva*, et al.*, "FPGA implementation of reconfigurable adaptive filters," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 2544-2547.

[56] L. Prono*, et al.*, "A High-level Implementation Framework for Non-Recurrent Artificial Neural Networks on FPGA," in *2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME)*, 2019, pp. 77-80.

[57] A. Jafari*, et al.*, "A Scalable and Low Power DCNN for Multimodal Data Classification," in *2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, 2018, pp. 1-6.

[58] H. Hinkelmann*, et al.*, "Design Concepts for a Dynamically ReconfigurableWireless Sensor Node," in *Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on*, 2006, pp. 436-441.

[59] S. Tanaka*, et al.*, "Reconfigurable hardware architecture for saving power consumption on a sensor node," in *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*, 2008, pp. 405-410.

[60] H. Saidi, *et al.*, "Embedded FPGA accelerator for Wireless Sensor Network nodes," in *2016 11th International Design & Test Symposium (IDT)*, 2016, pp. 313-318.

[61] Y. Li, *et al.*, "Using Energy-Aware Scheduling Weather Forecast Based Harvesting for Reconfigurable Hardware," *IEEE Transactions on Sustainable Computing,* vol. 4, pp. 109-117, 2019.

[62] Y. Li, *et al.*, "A low-power mixed-signal baseband system design for wireless sensor networks," in *Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005*, 2005, pp. 55-58.

[63] L. Wang, *et al.*, "A Field Programmable Analogue ASIC towards Low Power Processing-on-Node BSNs," in *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, 2009, pp. 213-216.

[64] P. Eswaran, *et al.*, "Design for 1µbar resolution pressure measurement and display ASIC for MEMS capacitive pressure sensor," in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2014, pp. 637-641.

[65] P. Zarkesh-Ha, *et al.*, "A reconfigurable ROIC for integrated infrared spectral sensing," in *IEEE Photonics Society, 2010 23rd Annual Meeting of the*, 2010, pp. 714-715.

[66] B. M. Kelly, *et al.*, "Reconfigurable analog signal processing for wireless sensor networks," in *Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium on*, 2013, pp. 221-224.

[67] S. Shah, *et al.*, "Reconfigurable analog classifier for knee-joint rehabilitation," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 4784-4787.

[68] B. Rumberg and D. W. Graham, "A low-power field-programmable analog array for wireless sensing," in *Quality Electronic Design (ISQED), 2015 16th International Symposium on*, 2015, pp. 542-546.

[69] B. Rumberg and D. W. Graham, "A low-power field-programmable analog array for wireless sensing," in *Sixteenth International Symposium on Quality Electronic Design*, 2015, pp. 542-546.

[70] C. Xiaoyan, *et al.*, "A CMOS Field Programmable Analog Array for intelligent sensory application," in *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, 2013, pp. 1-4.

[71] T. Yin, *et al.*, "A smart sensory platform based on field programmable analog array," in *2015 IEEE SENSORS*, 2015, pp. 1-4.

[72] L. Liu, *et al.*, "A novel mixed reconfigurable system on chip based on field programmable analog array," in *2017 IEEE 12th International Conference on ASIC (ASICON)*, 2017, pp. 948-952.

[73] Y. E. Krasteva, *et al.*, "Remote HW-SW reconfigurable Wireless Sensor nodes," in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, 2008, pp. 2483-2488.

[74] J. Portilla, *et al.*, "A Reconfigurable Fpga-Based Architecture for Modular Nodes in Wireless Sensor Networks," in *Programmable Logic, 2007. SPL '07. 2007 3rd Southern Conference on*, 2007, pp. 203-206.

[75] Y. Krasteva, *et al.*, "Embedded Run-time Reconfigurable Nodes for Wireless Sensor Networks Applications," *Sensors Journal, IEEE,* vol. PP, pp. 1-1, 2011.

[76] S. Yamaguchi, *et al.*, "Programmable wireless sensor node featuring low-power FPGA and microcontroller," in *Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference on*, 2013, pp. 596-601.

[77] S. Meena and N. K. Prakash, "Runtime reconfiguration of wireless sensor node using FPGA," in *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*, 2014, pp. 1-5.

[78] S. Bao, *et al.*, "FPGA-Based Reconfigurable Data Acquisition System for Industrial Sensors," *IEEE Transactions on Industrial Informatics,* vol. 13, pp. 1503-1512, 2017.

[79] K. Vinayakamoorthy, *et al.*, "Design and implementation of ASIC for weather monitoring application using FPAA and FPGA," in *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on*, 2014, pp. 365-369.
[80] V. F. Zavorotnyi, *et al.*, "Smart pressure sensor based on PSoC," in *Electronics Technology, 2008. ISSE '08. 31st International Spring Seminar on*, 2008, pp. 308-311.
[81] V. F. Zavorotnyi and Y. I. Yakimenko, "Based on PSoC Smart Sensor of gas leakage," in *Electronics Technology, 2009. ISSE 2009. 32nd International Spring Seminar on*, 2009, pp. 1-3.
[82] T. M. F. ElShafiey, "Design and implementation of a museum and banks security system using antenna as IR proximity sensor and PSoC technology," in *Wireless Technology and Applications (ISWTA), 2011 IEEE Symposium on*, 2011, pp. 156-161.
[83] R. Mohiddin, *et al.*, "Building a sensor network with PSoC," in *Sensing Technology (ICST), 2011 Fifth International Conference on*, 2011, pp. 353-357.
[84] K. Sonoda, *et al.*, "Wearable Photoplethysmographic Sensor System with PSoC Microcontroller," in *Emerging Trends in Engineering and Technology (ICETET), 2012 Fifth International Conference on*, 2012, pp. 61-65.
[85] E. A. V. Cabral and I. Valdez, "Airborne ultrasonic sensor node for distance measurement," in *SENSORS, 2013 IEEE*, 2013, pp. 1-4.
[86] W. T. Sung and C. C. Hsu, "Intelligent environment monitoring system based on innovative integration technology via programmable system on chip platform and ZigBee network," *Communications, IET,* vol. 7, pp. 1789-1801, 2013.
[87] P. N. Patil and S. D. Sawant, "Estimating multiple health parameters using PSoC controller," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 384-387.
[88] W. Xue-fen, *et al.*, "Wireless Sensor Node with Lightning and Atmospheric Pressure Detection for Severe Convective Weather Warning Networks," in *2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI)*, 2018, pp. 1-6.
[89] Y. W. Prakash, *et al.*, "Smart bluetooth low energy security system," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 2141-2146.
[90] D. H. Gawali and V. M. Wadhai, "Mixed signal SoC based Bio-Sensor Node for long term health monitoring," in *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2016, pp. 194-198.
[91] I. B. Cioc, *et al.*, "FPAA implementation of signal processing circuits for radiation sensors," in *Electronics Technology, 2009. ISSE 2009. 32nd International Spring Seminar on*, 2009, pp. 1-4.
[92] A. Umbarkar, *et al.*, "Improved sound-based localization through a network of reconfigurable mixed-signal nodes," in *Robotic and Sensors Environments (ROSE), 2010 IEEE International Workshop on*, 2010, pp. 1-6.
[93] D. P. Morales, *et al.*, "Enhancing ADC resolution through Field Programmable Analog Array dynamic reconfiguration," in *Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on*, 2008, pp. 635-638.
[94] G. Merendino, *et al.*, "Signal conditioning for capacitive sensors with field programmable analog arrays," in *Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on*, 2005, pp. 139-142 Vol. 1.
[95] C. De Marziani, *et al.*, "A low cost reconfigurable sensor network for coastal monitoring," in *OCEANS, 2011 IEEE - Spain*, 2011, pp. 1-6.
[96] G. Jiaqi, *et al.*, "A reconfigurable hardware platform for cognitive sensor networks towards behavioral biometrics," in *Sensors, 2012 IEEE*, 2012, pp. 1-4.

[97] M. Rui, *et al.*, "Active sensing of indoor human scenarios through mobile pyroelectric infrared sensors," in *SENSORS, 2013 IEEE*, 2013, pp. 1-4.
[98] L. Jiang, *et al.*, "Multi-agent based wireless pyroelectric infrared sensor networks for multi-human tracking and self-calibration," in *SENSORS, 2013 IEEE*, 2013, pp. 1-4.
[99] M. S. Kay and F. Iaione, "Reconfigurable Embedded System for ECG Signal Acquisition," in *2015 IEEE 28th International Symposium on Computer-Based Medical Systems*, 2015, pp. 25-26.
[100] S. Shah, *et al.*, "A proof-of-concept classifier for acoustic signals from the knee joint on a FPAA," in *2016 IEEE SENSORS*, 2016, pp. 1-3.
[101] S. Shah, *et al.*, "Temperature Sensitivity and Compensation on a Reconfigurable Platform," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 26, pp. 604-607, 2018.
[102] J. J. Cabrera-López and C. A. Romero-Beltrán, "Auto-adjustable low-signal processing technique based on programmable mixed-signal SoCs," in *2014 IEEE 9th IberoAmerican Congress on Sensors*, 2014, pp. 1-4.
[103] J. Hereford and C. Pruitt, "Robust sensor systems using evolvable hardware," in *Evolvable Hardware, 2004. Proceedings. 2004 NASA/DoD Conference on*, 2004, pp. 161-168.
[104] B. Massot, *et al.*, "A wearable, low-power, health-monitoring instrumentation based on a programmable system-on-chip^{TM}," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, 2009, pp. 4852-4855.
[105] V. Zavorotnyi and Y. Yakimenko, "Dynamic reconfiguration in smart sensors' design," in *Electronics Technology (ISSE), 2011 34th International Spring Seminar on*, 2011, pp. 608-611.
[106] E. Mackensen, *et al.*, "Smart wireless autonomous microsystems (SWAMs) for sensor actuator networks," in *Sensors for Industry Conference, 2004. Proceedings the ISA/IEEE*, 2004, pp. 72-78.
[107] E. Mackensen and C. Muller, "Implementation of Reconfigurable Micro-Sensor Interfaces Utilizing FPAAs," in *Sensors, 2005 IEEE*, 2005, pp. 1064-1067.
[108] D. P. Morales, *et al.*, "Merging FPGA and FPAA Reconfiguration Capabilities for IEEE 1451.4 Compliant Smart Sensor Applications," in *Programmable Logic, 2007. SPL '07. 2007 3rd Southern Conference on*, 2007, pp. 217-220.
[109] A. Kalupahana, *et al.*, "FPAA and FPGA based universal sensor node design," in *2017 Eleventh International Conference on Sensing Technology (ICST)*, 2017, pp. 1-6.
[110] I. Adly, *et al.*, "Over-The-Air Programming of PSoC sensor interface in wireless sensor networks," in *MELECON 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, 2010, pp. 997-1002.
[111] J. Guevara, *et al.*, "A reconfigurable WSN node based on ISO/IEC/IEEE 21451 standard," in *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2014, pp. 873-877.
[112] J. A. Guevara, *et al.*, "Dynamically Reconfigurable WSN Node Based on ISO/IEC/IEEE 21451 TEDS," *IEEE Sensors Journal,* vol. 15, pp. 2567-2576, 2015.
[113] M. Paselli, *et al.*, "A High-Performance Wireless Sensor Node for Industrial Control Applications," in *Systems, 2008. ICONS 08. Third International Conference on*, 2008, pp. 235-240.
[114] A. Lin, *et al.*, "Utilization of PSoC-5 for sensor data acquisition in a semi-autonomous robotic platform," in *2015 SAI Intelligent Systems Conference (IntelliSys)*, 2015, pp. 254-257.
[115] M. Altayeb, *et al.*, "The Internet-of-Things and Integration with Wireless Sensor Network Comprehensive Survey and System Implementation," in *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2018, pp. 1-6.
[116] R. J. M. Mercado, "Design of wireless sensor networks using embedded Programmable System-on-Chip (PSoC) as applied to community-based flood early warning systems (CBFEWS)," in *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, 2016, pp. 214-223.

[117] S. Vera, *et al.*, "A programmable plug&play interface for WSN applications," in *Sensors, 2011 IEEE*, 2011, pp. 1808-1811.
[118] V. Mattoli, *et al.*, "A Universal Intelligent System-on-Chip Based Sensor Interface," *Sensors,* vol. 10, pp. 7716-7747, 2010.
[119] Y. J. Huang, *et al.*, "A Self-Powered CMOS Reconfigurable Multi-Sensor SoC for Biomedical Applications," *IEEE Journal of Solid-State Circuits,* vol. 49, pp. 851-866, 2014.
[120] X. Zou, *et al.*, "A 1-V 450-nW Fully Integrated Programmable Biomedical Sensor Interface Chip," *IEEE Journal of Solid-State Circuits,* vol. 44, pp. 1067-1077, 2009.
[121] D. F. Lemmerhirt, *et al.*, "A Multitransducer Microsystem for Insect Monitoring and Control," *IEEE Transactions on Biomedical Engineering,* vol. 53, pp. 2084-2091, 2006.
[122] "IEEE Std 1451.2-1997, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats," ed: The Institute of Electrical and Electronics Engineers.
[123] *DS2430A 256-Bit 1-Wire EEPROM*. Available: https://www.maximintegrated.com/en/products/digital/memory-products/DS2430A.html
[124] *DS2433 4Kb 1-Wire EEPROM*. Available: https://www.maximintegrated.com/en/products/digital/memory-products/DS2433.html
[125] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats," *IEEE Std 1451.4-2004,* pp. 0_1-430, 2004.
[126] *TEDS - Plug and Play Sensor Configuraiton*. Available: http://www.sensotec.com/pnpterms.shtml
[127] D. Wobschall, "An implementation of IEEE 1451 NCAP for Internet access of serial port-based sensors," in *Sensors for Industry Conference, 2002. 2nd ISA/IEEE*, 2002, pp. 157-160.
[128] *Current Status of IEEE 1451.4*. Available: http://www.sem.org/PDF/Peres_Current%20Status%20of%20IEEE%201451.4.pdf
[129] *IEEE 1451.4 Standard Overview*. Available: http://zone.ni.com/devzone/cda/tut/p/id/3469
[130] (2019, *IEEE Registration Authority - Manufacturer ID List*. Available: http://standards-oui.ieee.org/manid/manid.txt
[131] (2019, *Honeywell - Sensors for Test & Measurement - Plug and Play Sensors*. Available: https://measurementsensors.honeywell.com/partners/honeywell_sensors/Pages/default.aspx
[132] (2019, *FUTEK Advanced Sensor Technology - TEDS Sensors*. Available: http://www.futek.com/search.aspx?ws=0&q=teds
[133] (2019, *Endevco Test and Measurement Product Catalog 2019*. Available: https://endevco.com/wp-content/uploads/2019/01/Meggitt_TM_Product_Catalog.pdf
[134] (2019, *HBM Improving Test & Measurement with TEDS*. Available: https://www.hbm.com/en/0547/teds-transducer-electronic-data-sheet/
[135] (2019, *National Instruments Data Acquisition and Control - C Series Sound and Vibration Input Module*. Available: https://www.ni.com/en-ie/shop/select/c-series-sound-and-vibration-input-module?modelId=122198
[136] (2019, *B&K Data Acquisition Systems and Hardware - General Purpose Analyzer Systems*. Available: https://www.bksv.com/en/products/data-acquisition-systems-and-hardware/general-purpose-analyzer-system/
[137] (2019, *B&K Data Acquisition Systems and Hardware - LAN-XI data acquisition hardware*. Available: https://www.bksv.com/en/products/data-acquisition-systems-and-hardware/general-purpose-analyzer-system/
[138] (2019, *VTI Instruments (Ametek) Data Acquisition*. Available: https://www.powerandtest.com/searchresult?userinput=teds

[139] (2019, *m+p International VibRunner Measurement Hardware Platform*. Available: https://www.mpihome.com/en/products/measurement-hardware/mp-vibrunner-measurement-hardware.html
[140] (2019, *The Modal Shop - TEDS Sensor Interface Kit*. Available: http://www.modalshop.com/TEDS-Sensor-Interface-Kit?ID=20
[141] (2019, *North Instruments - NID-1001/NID-1003 TEDS Interface Kit*. Available: https://www.interfaceforce.com/products/accessories/transducer-electronic-data-sheet-teds/
[142] (2013, *Tedsdongle reduces time required for configuring Teds sensor systems*. Available: https://www.engineerlive.com/content/tedsdongle-reduces-time-required-configuring-teds-sensor-systems
[143] (2019, *B&K TEDS Editor Software*. Available: https://www.bksv.com/en/Service/downloads
[144] (2019, *Honeywell - Sensors for Test & Measurement - Enhancing Existing Honeywell Sensors*. Available: https://measurementsensors.honeywell.com/partners/honeywell_sensors/Pages/EnhancingExistingHoneywellSensors.aspx
[145] (2019, *Interface Inc - TEDS option*. Available: https://www.interfaceforce.com/products/accessories/transducer-electronic-data-sheet-teds/
[146] (2019, *Dewesoft X3 Data acquisition, recording and analysis software*. Available: https://dewesoft.com/products/daq-software/dewesoft-x
[147] (2019, *EVIDAS Data Acquisition Software*. Available: https://www.hbm.com/en/6968/evidas-data-acquisition-software/
[148] N. Ulivieri and S. Rocchi, "LabWeb1451: Remotely Accessible Virtual Laboratory for Transducer Electronic Data Sheets Creation and Testing," in *Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE*, 2006, pp. 857-860.
[149] J.-D. Kim*, et al.*, "The definition of basic TEDS of IEEE 1451.4 for sensors for an electronic tongue and the proposal of new template TEDS for electrochemical devices," *Talanta,* vol. 71, pp. 1642-1651, 2007.
[150] N. Ulivieri*, et al.*, "IEEE1451.4: A way to standardize gas sensor," *Sensors and Actuators B: Chemical,* vol. 114, pp. 141-151, 2006.
[151] V. Viegas*, et al.*, "Transducer Electronic Data Sheets: Why not print them?," in *Electrical and Power Engineering (EPE), 2014 International Conference and Exposition on*, 2014, pp. 897-900.
[152] E. Song*, et al.*, "A prototype IEEE 1451.4 smart transducer interface for sensors and actuators," in *Electronic Measurement & Instruments (ICEMI), 2011 10th International Conference on*, 2011, pp. 1-6.
[153] N. Jevtic and V. Drndarevic, "Development of smart transducers compliant with the IEEE 1451.4 standard," in *Instrumentation and Control Technology (ISICT), 2012 8th IEEE International Symposium on*, 2012, pp. 126-131.
[154] K. Malar and N. Kamaraj, "Development of smart transducers with IEEE 1451.4 standard for Industrial automation," in *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, 2014, pp. 111-114.
[155] A. Cherian*, et al.*, "An IoT interface for industrial analog sensor with IEEE 21451 protocol," in *2017 IEEE Sensors Applications Symposium (SAS)*, 2017, pp. 1-5.
[156] Y. Ma*, et al.*, "A combined ISO/IEC/IEEE 21451-4 and -2 data acquisition module," in *2017 IEEE Sensors Applications Symposium (SAS)*, 2017, pp. 1-6.
[157] G. Da-li and H. Xiao, "Integration Research of Intelligent Traffic Monitoring System for Urban Intersection Based on IEEE and CAN," in *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*, 2008, pp. 606-610.
[158] S. Y. Yurish, "Low-cost, dual-axis smart inclinometer," in *Sensors, 2008 IEEE*, 2008, pp. 1488-1491.

[159] R. Wang, *et al.*, "Wireless intelligent sensors supporting integrated systems health management architecture," in *Sensors Applications Symposium, 2008. SAS 2008. IEEE*, 2008, pp. 215-219.
[160] Z. Li, *et al.*, "The prototype IEEE 1451.4 applied in the IOT," in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, 2012, pp. 4241-4244.
[161] K. S. E. Phala, *et al.*, "Air Quality Monitoring System Based on ISO/IEC/IEEE 21451 Standards," *IEEE Sensors Journal,* vol. 16, pp. 5037-5045, 2016.
[162] A. Singh and A. Kumar, "Network controlled distributed energy management system for smart cities," in *2016 2nd International Conference on Communication Control and Intelligent Systems (CCIS)*, 2016, pp. 235-238.
[163] F. P. Jooste, *et al.*, "Energy efficient irrigation scheduling system based on the ISO/IEC/IEEE 21451 standards," in *2017 IEEE International Conference on Industrial Technology (ICIT)*, 2017, pp. 1441-1446.
[164] A. Kumar and A. Singh, "Energy efficient and low cost air quality sensor for smart buildings," in *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 2017, pp. 1-4.
[165] R. Wyżgolik, *et al.*, "RT/FPGA implementation of the IEEE 1451 standard in sensors for machine conditioning systems," in *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2016, pp. 794-799.
[166] N. J. Jevtič and V. R. Drndarević, "Smart sensors for environmental radiation monitoring networks," in *2015 23rd Telecommunications Forum Telfor (TELFOR)*, 2015, pp. 607-614.
[167] D. Wobschall and S. Mupparaju, "Low-power wireless sensor with SNAP and IEEE 1451 protocol," in *Sensors Applications Symposium, 2008. SAS 2008. IEEE*, 2008, pp. 225-227.
[168] O. Postolache, *et al.*, "Smart sensor architecture for vital signs and motor activity monitoring of wheelchair' users," in *Sensing Technology (ICST), 2012 Sixth International Conference on*, 2012, pp. 167-172.
[169] O. Postolache, *et al.*, "IEEE 1451.4 embedded smart sensors architecture for wheelchair user monitoring," in *Medical Measurements and Applications Proceedings (MeMeA), 2012 IEEE International Symposium on*, 2012, pp. 1-5.
[170] J. Freire, *et al.*, "IEEE1451.4 multi-sensing platform for wheelchair users assessment," in *Telecommunications (ICT), 2014 21st International Conference on*, 2014, pp. 27-31.
[171] V. Viegas, *et al.*, "Smart transducer interfaces applied to health monitoring," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2015, pp. 1640-1645.
[172] H. G. Ramos, *et al.*, "Embedding IEEE 1451.4 smart sensing nodes in a Wireless Air Quality Monitoring Network," in *Circuits and Systems, 2006. MWSCAS '06. 49th IEEE International Midwest Symposium on*, 2006, pp. 177-181.
[173] T. Gulrez and M. Kavakli, "Sensor Relevance Establishment Problem in SharedInformation Gathering Sensor Networks," in *Networking, Sensing and Control, 2007 IEEE International Conference on*, 2007, pp. 650-655.
[174] T. Gulrez and M. Kavakli, "Precision Position Tracking in Virtual Reality Environments using Sensor Networks," in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, 2007, pp. 1997-2003.
[175] K. Konte, *et al.*, "Implementing XEDS for a CubeSat communication subsystem," in *2018 IEEE Sensors Applications Symposium (SAS)*, 2018, pp. 1-5.
[176] A. Fatecha, *et al.*, "Reconfigurable architecture for smart sensor node based on IEEE 1451 standard," in *SENSORS, 2013 IEEE*, 2013, pp. 1-4.

[177] J. Guevara, *et al.*, "A reconfigurable WSN node based on ISO/IEC/IEEE 21451 standard," in *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, 2014, pp. 873-877.
[178] A. C. Ranasinghe, *et al.*, "Development of an intelligent reconfigurable NCAP for Smart Transducer applications," in *Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on*, 2013, pp. 1-6.
[179] W. Rentao and K. Deevy, "Reconfigurable adaptive wireless sensor node," in *Signals and Systems Conference (ISSC 2012), IET Irish*, 2012, pp. 1-6.
[180] J. L. Hill and D. E. Culler, "Mica: a wireless platform for deeply embedded networks," *Micro, IEEE,* vol. 22, pp. 12-24, 2002.
[181] V. Raghunathan, *et al.*, "Energy-aware wireless microsensor networks," *Signal Processing Magazine, IEEE,* vol. 19, pp. 40-50, 2002.
[182] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM,* vol. 43, pp. 51-58, 2000.
[183] *BSN node*. Available: http://ubimon.doc.ic.ac.uk/bsn/index.php?article=167
[184] K. JeongGil, *et al.*, "Egs: A Cortex M3-Based Mote Platform," in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, 2010, pp. 1-3.
[185] C. Chih-Chun, *et al.*, "Measurement of Energy Costs of Security in Wireless Sensor Nodes," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, 2007, pp. 95-102.
[186] *Epic: An Open Mote Platform for Application-Driven Design*. Available: http://www.cs.berkeley.edu/~prabal/projects/epic/
[187] R. Mangharam, *et al.*, "FireFly: a cross-layer platform for real-time embedded wireless networks," *Real-Time Syst.,* vol. 37, pp. 183-231, 2007.
[188] *Imote2*. Available: http://wsn.cse.wustl.edu/images/e/e3/Imote2_Datasheet.pdf
[189] *Indriya_DP_03A14 - XBEE based wireless sensor network development platform*. Available: http://www.indrion.co.in/datasheet/Indriya_DP_03A14.pdf
[190] *Indriya_DP_01A11 - Zigbee based ultra-low power wireless sensor network development platform*. Available: http://www.indrion.co.in/datasheet/Indriya_DP_01A11.pdf
[191] "Indriya_DP_04A11 - Zigbee based ultra-low power wireless sensor network development platform."
[192] *Iris*. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0124-01_B_IRIS.pdf
[193] *JN5148 Wireless Microcontroller Modules*. Available: http://www.jennic.com/products/modules/
[194] *Lotus - High performance wireless sensor network platform*. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0705-01_A_LOTUS.pdf
[195] *MeshScape GO OEM Developers Kit*. Available: http://millennialnet.com/Wireless-Technology/Products/MeshScape-GO-KIT.aspx
[196] *n-Core Sirius IOn D-E-M User Manuel*. Available: http://resources.nebusens.com/n-core/Manual_n-Core_Sirius-IOn-D-E-M_EN.pdf
[197] *OpenMote-CC2538*. Available: http://www.openmote.com/shop/openmote-cc2538.html
[198] *panStamp AVR*. Available: http://www.panstamp.com/product/panstamp-avr/
[199] V. Shnayder, *et al.*, "Sensor networks for medical care," presented at the Proceedings of the 3rd international conference on Embedded networked sensor systems, San Diego, California, USA, 2005.
[200] *PowWow Hardware Platform*. Available: http://powwow.gforge.inria.fr/

[201] *Preon32 - Innovative 2.4 GHz radio module for Preon32-Series by Virtenio*. Available: http://www.virtenio.com/en/products/radio-module.html
[202] *Philips Small Autonomous Network Devices*. Available: http://www.futuretechnologycenter.eu/downloads/sand.pdf
[203] *Forest Fire Detection with Wireless Sensor Networks*. Available: http://das.ege.edu.tr/~sinanyil/yayinlar/icast2010/forestfire.pdf
[204] T. Ahonen*, et al.*, "Greenhouse Monitoring with Wireless Sensor Network," in *Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*, 2008, pp. 403-408.
[205] A. Burns*, et al.*, "SHIMMER - A Wireless Sensor Platform for Noninvasive Biomedical Research," *Sensors Journal, IEEE,* vol. 10, pp. 1527-1534, 2010.
[206] *Wireless Sensor Networks - Dust Networks*. Available: http://www.linear.com/products/Wireless_Sensor_Networks_-_Dust_Networks
[207] K. Nanda*, et al.*, "Smartmote - An innovative autonomous Wireless Sensor Node architecture," in *Electronics, Computing and Communication Technologies (IEEE CONECCT), 2014 IEEE International Conference on*, 2014, pp. 1-6.
[208] J. Polastre*, et al.*, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, 2005, pp. 364-369.
[209] *Telos Ultra low power IEEE 802.15.4 compliant wireless sensor module*. Available: http://www4.ncsu.edu/~kkolla/CSC714/datasheet.pdf
[210] *CM5000 802.15.4 TelosB mote Module*. Available: http://www.advanticsys.com/shop/mtmcm5000msp-p-14.html
[211] *Kmote Overview*. Available: http://tinyos.stanford.edu/tinyos-wiki/index.php/Kmote
[212] *Waspmote*. Available: http://www.libelium.com/products/waspmote/
[213] *WiSense Overview*. Available: http://www.wisense.in/api/html/SystemOverview.html
[214] *WiSMote - Sensor/Actuator wireless and low consumption module for WSN application*. Available: http://www.aragosystems.com/en/wisnet-item/wisnet-wismote-item.html
[215] A.-B. García-Hernando*, et al.*, *Problem Solving for Wireless Sensor Networks*: Springer, 2009.
[216] *AS-XM1000 802.15.4 Mote Module*. Available: http://www.advanticsys.com/shop/asxm1000-p-24.html
[217] *XYZ Sensor Node*. Available: http://www.eng.yale.edu/enalab/xyz/
[218] *Zolertia Z1 Low-Power WSN Platform*. Available: http://zolertia.com/sites/default/files/Zolertia-Z1-Brochure.pdf
[219] *BTnodes - A Distributed Environment for Prototyping Ad Hoc Networks*. Available: http://www.btnode.ethz.ch/
[220] M. M. Vieira*, et al.*, "Designing Wireless Sensor Nodes," in *Embedded Computer Systems: Architectures, Modeling, and Simulation*. vol. 4017, S. Vassiliadis*, et al.*, Eds., ed: Springer Berlin Heidelberg, 2006, pp. 99-108.
[221] C. Lynch and F. O'Reilly, "PIC-based TinyOS implementation," in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, 2005, pp. 378-385.
[222] A. Barroso*, et al.*, "The DSYS25 sensor platform," presented at the Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, 2004.
[223] P. Chulsung*, et al.*, "Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, 2005, pp. 398-403.
[224] F. Simjee and P. H. Chou, "Everlast: Long-life, Supercapacitor-operated Wireless Sensor Node," in *Low Power Electronics and Design, 2006. ISLPED'06. Proceedings of the 2006 International Symposium on*, 2006, pp. 197-202.

[225] *EYES project overview*. Available: ftp://ftp.cordis.europa.eu/pub/ist/docs/dir_c/ems/meratnia_wireless.pdf
[226] *The eyesIFX platform*. Available: http://www.tinyos.net/ttx-02-2005/platforms/ttx2005-eyesIFX.ppt
[227] P. Sitka*, et al.*, "Fleck - A platform for real-world outdoor sensor networks," in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, 2007, pp. 709-714.
[228] *SOWNet G-Node G301 Wireless Sensor Node*. Available: http://www.sownet.nl/index.php/products/gnode
[229] S. M. Chih-Chun Chang, David J. Nagel, "Cryptographic Algorithms for Wireless Sensor Nodes," presented at the ACM Conference on Wireless Network Security, 2008.
[230] S. Bhatti*, et al.*, "MANTIS OS: an embedded multithreaded operating system for wireless micro sensor platforms," *Mob. Netw. Appl.,* vol. 10, pp. 563-579, 2005.
[231] *MICA2Dot*. Available: http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2dot.pdf
[232] *BSAC Wireless Sensor Workshop*. Available: https://www-bsac.eecs.berkeley.edu
[233] E. M. Tapia*, et al.*, "The design of a portable kit of wireless sensors for naturalistic data collection," presented at the Proceedings of the 4th international conference on Pervasive Computing, Dublin, Ireland, 2006.
[234] *Mulle Hardware Guide*. Available: http://www.eistec.se/downloads/legacy/Mulle_HW_Guide.pdf
[235] *panStamp NRG*. Available: http://www.panstamp.com/product/197/
[236] *PARTiCLE Web Site - Particle Computer - the wireless core*. Available: http://particle.teco.edu/documentation/index.html
[237] J. Schiller*, et al.*, "ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing," in *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*, 2005, pp. 286c-286c.
[238] M. B. Reiner Kolla, Clemens Muhlberger, "SNoW5: A versatile ultra low power modular node for wireless ad hoc sensor networking," *Marrón PJ (ed) 5. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Stuttgart,* pp. 55-59, 2006.
[239] H.-J. Krber*, et al.*, "Embedding a Microchip PIC18F452 based commercial platform into TinyOS," *Real-World Wireless Sensor Networks(REALWSN'05),* June 2005.
[240] H. Dubois-Ferriere*, et al.*, "TinyNode: a comprehensive platform for wireless sensor network applications," in *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, 2006, pp. 358-365.
[241] *Tyndall 10mm Miniaturised Wireless Sensor Node*. Available: https://www.tyndall.ie/content/tyndall-10mm-miniaturised-wireless-sensor-node
[242] *In-Network Data Aggregation in Wireless Sensor Networks*. Available: http://www.gleonrcn.org/media/SensorNetworkSystems.pps
[243] J. Akerberg*, et al.*, "Integration of WirelessHART networks in Distributed Control Systems using PROFINET IO," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, 2010, pp. 154-159.
[244] I. Muller*, et al.*, "Development of a WirelessHART compatible field device," in *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, 2010, pp. 1430-1434.
[245] *Cypress Programmable System-on-Chip Device Datasheet CY8C29X66*. Available: http://www.cypress.com/
[246] *Using Thermistors - The Thermistor*. Available: http://www.omega.com/temperature/z/pdf/z036-040.pdf

[247] *NTC Thermistor Engineering Notes*. Available: http://www.specsensors.com/pdfs/ntc_engineering_notes.pdf
[248] *NTC Thermistors*. Available: http://www.phy.syr.edu/courses/PHY351.05Spring/ntcnotes.pdf
[249] *Thermocouples - An Introduction*. Available: http://www.omega.com/prodinfo/thermocouples.html
[250] *HBM TEDS Sensor Modle Number List*. Available: https://www.hbm.com/fileadmin/mediapool/support/download/teds/model001f.xdl
[251] *Keithley 2400 Series SourceMeter SMU Instruments Datasheet*. Available: https://www.tek.com/keithley-source-measure-units/keithley-smu-2400-series-sourcemeter
[252] *Ultra Electronics (Weed Instruments) Models 101 & 103 General Purpose Thermocouple Datasheet*. Available: https://www.ultra-nspi.com
[253] *Ultra Electronics (Weed Instruments) Thermocouple General Specifications*. Available: https://www.ultra-nspi.com
[254] *Ultra Electronics (Weed Instruments) Models 101, 102, & 103 General Purpose RTDs Datasheet*. Available: https://www.ultra-nspi.com
[255] *Ultra Electronics (Weed Instruments) RTD General Specifications*. Available: https://www.ultra-nspi.com
[256] *Digitron 2000 Series Thermometer Technical Datasheet*. Available: https://www.digitron.com/
[257] *T-Nodes*. Available: https://www.sownet.nl/index.php/en/technology/t-nodes
[258] *WiSMote mini - Sensor/Actuator wireless and low consumption module for WSN application*. Available: http://www.aragosystems.com/en/wisnet-item/wismotemini.html
[259] "IEEE Std 1451.1-1999, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor (NCAP) Information Model," ed: The Institute of Electrical and Electronics Engineers.
[260] "Specification of the Bluetooth System," ed: Bluetooth Special Insterest Group.
[261] *A Look at the Basics of Bluetooth Technology* Available: http://www.bluetooth.com/Pages/Basics.aspx
[262] *Wibree*. Available: http://developer.nokia.com/community/wiki/Wibree
[263] *The Low Energy Technology Behind Bluetooth Smart*. Available: http://www.bluetooth.com/Pages/low-energy-tech-info.aspx
[264] B. Allen*, et al.*, "Ultra wideband: applications, technology and future perspectives."
[265] I. Oppermann*, et al.*, "UWB wireless sensor networks: UWEN - a practical example," *Communications Magazine, IEEE,* vol. 42, pp. S27-S32, 2004.
[266] D. Porcino and W. Hirt, "Ultra-wideband radio technology: potential and challenges ahead," *Communications Magazine, IEEE,* vol. 41, pp. 66-74, 2003.
[267] *XS110 UWB Solution for Media-Rich Wireless Applications*. Available: www.uta.edu/rfmems/Telemetry/UWB/UWBFS.pdf
[268] "IEEE Std 802.15.4," ed: IEEE Standards Association.
[269] *Calculating 802.15.4 Data Rates*. Available: http://www.nxp.com/documents/application_note/JN-AN-1035.pdf
[270] "ZigBee Specification," ed: ZigBee Standards Organization.
[271] T. Lennvall*, et al.*, "A comparison of WirelessHART and ZigBee for industrial applications," in *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*, 2008, pp. 85-88.