



Low Drift, Wireless Temperature Sensor for Harsh Industrial Applications

By

Anupama Kulkarni

Supervisors: Prof. Dr. Stoyan Nihtianov, TU Delft

Dr. Ir. Mihai Patrascu, ASML. B.V

A thesis submitted in partial fulfillment for the
degree of Master of Science

at

The Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology



September 2015

Abstract

In this thesis, a stand-alone, battery powered, wireless temperature sensor that can be used for diagnostic purposes in harsh industrial environments, including magnetic interferences and very low pressure (vacuum), is presented.

Being easily mountable and removable during machine stoppage, this autonomous temperature sensor can be used at any location in a machine to gather temperature data over a period of a few months. The proposed sensor is realized with easily available, off-the-shelf components. It outperforms wireless temperature sensors available on the market in terms of the resolution achieved: 0.2 mK (3 sigma/ 1 reading per second), the power consumption: <1 mW, and long-term stability: <1 mK/year.

Acknowledgements

Two years of Masters at TU Delft has brought out the best in me in terms of gaining in-depth technical knowledge, confidence, brain-storming skills and questioning ability. The training I received in the Microelectronics department is unparalleled with any of my previous experiences. From learning every aspect of Microelectronics, to thought provoking assignments and examinations, these two years have facilitated not only my professional growth, but also made me a better person. My first heartfelt acknowledgement goes out to this colossal institution.

The courses offered during the first year helped me improve my analytical and problem-solving skills. I would like to thank all the professors of my courses.

During the second year, I was fortunate to have gotten an opportunity to carry out my thesis at ASML. B.V., Veldhoven. Under the expert guidance of Prof. Stoyan Nihtianov, I could broaden my scope, inching towards becoming a thorough professional. His wide-spread knowledge in Industrial electronics was an inspiration to me throughout my thesis. The discussions were thought-provoking and made me challenge my own limits every time. I am very thankful to him for being so kind and supportive.

I would sincerely like to thank my supervisor at ASML, Mihai Patrascu, for guiding me through my thesis, along with providing me an excellent environment to work. His guidance and support has been paramount at every stage of my thesis.

The wireless sensor team at ASML including Rik Naulaerts, Evert Westerhuis and Daniel Kaarls were real fun to work with. They did well to keep my progress in track with meetings every week. Also, a warm thanks to my other colleagues at ASML – Jie Fu, Avanti Vyas, Roshan Gupta, Parag Sankhavara, Vittal Rao, Rick van Puinenbroek, Ton Schuwer, Ugur Bagci and Yang Chang.

I would like to extend my acknowledgements to Mr. Jaap van Wensveen, TempControl B.V., and Mr. Alexander Kerezov, Sofia University, Bulgaria, who have assisted me at a few points during my thesis. Thanks to Zu-yao Chang and Lukasz Pakula, from TU Delft, for assisting me with experiments conducted in the lab.

Special thanks to the two Groups leads at ASML, Robert Pijnenburg and Koen Tijskens for taking care of all the administrative responsibilities at ASML for me and keeping up the motivation.

I would also like thank Prof. Dineshkumar Harursampath and Mr. Ajay Bangalore Harish, NMCAD Lab, IISc, Bangalore. They were my first mentors in research during my bachelor's study and did very well to lay a solid base in me.

A special mention to Prof. Ying-dar Lin, High Speed Networks Laboratory, NCTU, Taiwan, for providing an extensive training during my internship.

Heartfelt acknowledgement to Prof. Dr. Vijaya. C, SDMCET – my mentor during bachelors, for her guidance and support throughout.

I would like to extend warm wishes to my friends – Shailja Shukla, Vijayakumar Purushothaman, Sotiris Thomas, Revanth Bellamkonda, for making my stay in the Netherlands wonderful. I have great memories with all of them, which I will cherish for the rest of my life.

My family – father, Dr. Manohar Kulkarni, brother, Raghavendra, and sister-in-law, Smita, my grandparents and relatives have always been a great support to me in every sense. I consider myself extremely lucky to have them with me.

My close friends – Aditya, Anundhara and Neeti have been with me through thick and thin and kept me composed. I take this opportunity to thank all of them for their love and faith in me, always.

This thesis is dedicated to my mother, Late. Dr. Geeta Kulkarni. She has and will be a constant source of inspiration in all my endeavors.

Table of Contents

Abstract	3
Acknowledgements.....	5
List of Figures.....	10
List of Tables.....	12
Chapter I. Introduction	15
1.1 Introduction	15
1.2 Motivation.....	15
1.3 Challenges	16
1.4 Scope and organization of the thesis.....	16
Chapter 2: An overview of existing wireless temperature sensors	18
2.1 Industrial applications	18
2.2 Available products	21
2.2.1 Wireless temperature sensors	21
2.2.2 High-performance temperature sensors	22
2.3 Introduction to low-power, wireless communication.....	23
2.4 Conclusions.....	26
References	27
Chapter 3: Wireless Sensor Design.....	Error! Bookmark not defined.
3.1 Specifications.....	31
3.2 System design	32
3.2.1 Principle of operation.....	32
3.2.2 Choice of components:	33
3.2.3 Block diagram:.....	36
3.3 PCB and Layout.....	38
3.4 Bluetooth Communication Program.....	39

3.4.1 CY8Ckit-042 BLE	39
3.4.2 Program (flow chart).....	39
3.4.3 CySmart.....	40
References:.....	42
Chapter 4: Experiments/tests.....	43
4.1 Long term drift of NTC temperature sensors.....	43
4.1.1 Motivation.....	43
4.1.2 Experimental setup	44
4.1.3 Methodology	45
4.1.4 Error minimization.....	47
4.1.5 Conclusion.....	48
4.2 Wireless sensor qualification tests	49
4.2.1 Noise measurements	49
4.2.2 Transfer Characteristics	49
4.2.3 Power consumption	50
4.2.4 Power supply rejection ratio (PSRR).....	51
4.2.5 Reliability of communication.....	51
4.2.6 Life-time tests of batteries in vacuum.....	51
4.2.7 . Long-term drift of electronics	53
4.2.8 Conclusion.....	53
4.3 BJT-based temperature sensor stability tests.....	53
4.3.1 Drift measurement	54
4.4 Conclusion.....	55
Chapter 5: Experimental results	Error! Bookmark not defined.
5.1 Long-term drift of NTC temperature sensors	58
5.2 Wireless sensor prototype qualifying tests	63
5.2.1 Noise measurements	63
5.2.2 Transfer characteristics.....	64

5.2.3 Power consumption	66
5.2.4 Power supply rejection ratio (PSRR).....	67
5.2.5 Reliability of communication.....	68
5.2.6 Life-time tests of batteries in vacuum.....	68
5.2.7 Long-term drift of electronics	69
Chapter 6: Conclusion	71
6.1 Contributions.....	71
6.2 Summary	72
6.3 Future work.....	73
Appendix I.....	74
Appendix II	Error! Bookmark not defined.
Appendix III.....	88
Appendix IV.....	89
Appendix V.....	92

List of Figures

Figure 1.1a Simplified block diagram of the wireless system designed using IQMESH protocol

Figure 1.1b Structure of IQRF transceiver module

Figure 1.2: The complete system for the temperature measurement reported in

Figure 1.3: Wireless temperature sensor block diagram used in

Figure 3.1: Functional block diagram of the wireless temperature sensor prototype

Figure 3.2: circuit diagram of full (half) bridge configuration of the wireless sensor

Figure 3.3 (a, b, c): Layout and component arrangement of the half bridge configuration.

Figure 3.4: Snapshots of the wireless sensor prototype along with the BLE module.

Figure 3.5: Flow chart describing the various events taking place to transmit one data of temperature measure by the sensor

Figure 4.1: a) IPRT sensors; b) Triple point of water cell; c) Readout set-up used to measure drift of NTCs, which includes AC ratiometric bridge and Keithley DMM.

Figure 4.2: NTCs enclosed in tubes along with three Pt100 sensors.

Figure 4.3: (a) Temperature chamber used to measure the transfer characteristics of the sensor (b) PT100 sensor

Figure 4.4: Aluminum block inside the chamber, connected to reference and wireless sensor.

Figure 4.5: vacuum vessel used for testing the effects of vacuum on life-time of batteries

Figure 4.6: a) batteries listed in table 1 soldered to a PCB for vacuum tests b) The batteries connected to DAQ and feedthrough of the vacuum vessel

Figure 4.7: long-term drift test setup (wireless prototype, reference sensor and Al block)

Figure 4.8 : a) The integrated temperature sensor (8 BJT sensors), with one of them inserted in the bath b) The flat wire connection between the pcb and MCU.

Figure 5.1 (a) – (i) Drift measurement results of various NTCs and fixed resistors, mentioned in Table 1

Figure 5.2: Example case showing the pre-aging phase

Figure 5.3: Graph representing the average of each set of NTC

Figure 5.4: Deterministic error in each set of NTCs introduced during measurements

Figure 5.5: Final drift results after the removal of deterministic error

Figure 5.6: Measured noise performance of full bridge configuration

Figure 5.7: Noise performance of half bridge configuration

Figure 5.8 Transfer characteristics of wireless sensor with half bridge

Figure 5.9 Transfer characteristics of wireless sensor with full bridge

Figure 5.10: Difference between the ideal and measured transfer characteristics for full bridge

Figure 5.11: Difference between the ideal and measured transfer characteristics for half bridge

Figure 5.12. Behavior of full bridge configuration with respect to variations in the power supply

Figure 5.13. Behavior of half bridge configuration with respect to variations in the power supply

Figure 5.14 Distance vs. percentage loss of packets

Figure 5.15: Life time of batteries in vacuum test results for the two LiSOCl₂ batteries

Figure 5.16: Drift results of electronics part of the wireless temperature sensor prototype

Figure 5.17: Drift results of BJT-based temperature sensor

Figure A.1: Different components of the CY8C042-kit BLE from Cypress

Figure A.2: Displays the component catalogue from PSoC creator through which different components can be selected for the design

Figure A.3: The PSoC creator workspace where in the components can be dragged and dropped.

Figure A.4: The source and header files of the desired components created by the PSoC creator

Figure A.5: Screen shot of the device selector dialogue box from PSoC creator

Figure A.6: Screen shot of CySmart GUI showing the available devices for connection

Figure A.7: The screen shot of Cysmart GUI highlighting the steps: pair, discover all attributes, and enable notifications.

Figure A.8: Screen shot of the “log window” on Cysmart where the temperature data is displayed

Figure A.9: Plot of average value of 8 samples of NCP15XH103D03RC, along with the linear approximation

Figure A.10: Example plot showing the difference of average value of the sensors and the linear trend line

Figure A.11: Screen shot of the excel file for the error calculation of NCP15XH103D03RC

List of Tables

Table 2.1: Comparison of different wireless temperature sensors available in the market

Table 2.2: A few sub-mK resolution, low power and small foot print temperature sensors reported in the literature.

Table 2.3: Comparison of the four short-range wireless protocols

Table 2.4: Comparison between classic Bluetooth and Bluetooth Low Energy

Table 2.5: Comparison of current and power consumptions of ZigBee and BLE

Table 3.1: List of specifications for the design of the wireless temperature sensor prototype.

Table 3.2. Survey of best suitable battery technology.

Table 3.3. Comparison of ADCs regarding noise, power and CMRR.

Table 3.4: Comparison of BLE chipsets available in the market suitable for this design

Table 3.5: Comparison of most suitable SRAM and FRAM memory chips for this design, available in the market

Table 4.1: Details of the type, manufacturers and excitation voltages applied to the sensors being tested for drift performance

Table 6.1: Long-term drift results of NTC temperature sensors, with drift rates before and after error minimization.

Chapter I. Introduction

1.1 Introduction

Numerous sensors are used in advanced industrial machines to measure various parameters like temperature, pressure, humidity. Typical requirements for sensor systems in hi-tech industrial equipment are: low power, high stability, high resolution and high accuracy. Additionally, wireless sensor networks (WSN) are preferred, to avoid using cables, especially in harsh working environment, i.e. vacuum, where special (vacuum compatible low-outgassing) materials are only allowed. Using numerous heavy cables may have a negative impact on the dynamics of the machine, and can deteriorate its reliability. Autonomous wireless sensors could prove useful for such applications. They are easily deployable at a freely accessible spot in the machine. A combination of sensor systems – a wireless sensor network, could be used to gather data from different parts of the machine and significantly increase the availability of the machine.

1.2 Motivation

Continuous, accurate temperature measurement is challenging when used in industrial machines operating in harsh environments. The traditional application of temperature measurement is to connect the sensor to the read-out electronics using wires. However, in harsh environments with magnetic fields and vacuum tight parts, the electronics is exposed to interferences from the field, which also couple through the wires connecting the sensors and the electronics. Merely twisting the wires would not be an efficient solution, as the fields could be non-uniform. A simple solution for this would be to make the sensor system wireless! It is to be noted that making the sensor wireless can be useful in avoiding magnetic field coupling along the wires. However, the coupling at the sensor node itself may not be avoided. There has been a great interest among researchers to build wireless sensor networks in the past few years. Wireless temperature sensors are used for various applications like: health monitoring, ambient temperature monitoring, smart homes and so on. Particularly, for advanced industrial applications, they can be used for diagnostics: to measure temperature at different parts, thus effectively avoiding its breakdown and increasing availability (more detailed explanation along with references in chapter II).

Unfortunately, the wireless temperature sensors available on the market are bulky, power hungry and with limited data rate, resolution and accuracy. Usually, the commercial wireless sensors available in market are designed for wide temperature ranges (-40°C to $+125^{\circ}\text{C}$, 0°C to $+60^{\circ}\text{C}$) with a trade-off between accuracy and resolution. They have limited resolution of $\times 10$ milli-Kelvin and accuracy ranging from 0.2°C to 2°C . Apart from that, they are power hungry with consumptions in the order $\times 10$ mW. Long-term drift of these sensors is not usually specified in the datasheet. When measured, the drift appears to be unacceptably high, requiring frequent calibration. A

detailed comparison of the existing wireless temperature sensors available in the market is provided in section 2.2 of chapter 2.

Contrastingly, some of the advanced industrial applications require lower operating temperature ranges (10°C to 30 °C), resolution in the sub-mK range, along with low power, high accuracy, small foot-print, and low drift over time. However, the commercially available wireless sensors fail to cater the needs of these applications. In this work, a wireless temperature sensor suitable for high-end industrial applications is designed and tested. For high resolution, a negative temperature coefficient (NTC) thermistor is used as the sensing element. Since long-term stability is one of the main targets, in the chosen architecture, components like pre-amplifiers which introduce drift, are avoided.

1.3 Challenges

In advanced industrial applications, temperature sensing can be used for either measuring the temperature of specific machine parts, or for diagnostic purposes. Particularly for diagnostic purposes, the sensor needs to be stand-alone and deployable at any point of interest. Sometimes, the temperature variation to be measured is in the sub-mK range. In order to save space, small dimensions of the sensor systems are preferred. To avoid temperature measurement errors due to self-heating, the power dissipation of the sensing element has to be as low as possible (in μW range). The interface electronics, when close to the sensing element, also needs to be very low power, which contradicts with the requirements of high resolution and stability. Presence of high power electrical motors creates magnetic fields which can interfere with the data signals, by coupling through the wires connecting the sensors and the electronics. Apart from that, a harsh working environment (vacuum) can affect the performance of the sensor module, can reduce its lifetime, and significantly limit the choice of components and materials when designing the sensor.

Another important feature required for high-end industrial applications is long-term stability. Systems with low drift over time, of the order of a few mK/year, are preferred as they are stable, avoiding repeated calibration and thus increasing the overall availability of the machine (in other words, avoiding downtime of the machine). All these issues form the primary specifications for the wireless temperature sensor presented in this work.

To summarize, the main challenges for designing a wireless temperature sensor for advanced industrial applications are: autonomy, low drift, small size, low power, sub-mK resolution, and high accuracy, which have to be realized in harsh environments.

1.4 Scope and organization of the thesis

The scope of the thesis is: to design, implement and qualify a low drift, low power small foot print wireless temperature sensor for vacuum applications. The organization of each chapter is explained in detail below:

Chapter 2: A comparison of the existing wireless temperature sensors in the market is presented. Next, a few sensors reported in the literature are discussed, focusing on the most important performance parameters. Next, a few high-performance temperature sensing solutions (BJT-based sensors/NTC based sensors) which could be used in wireless configurations are discussed. Finally, a comparison of different wireless protocols (Zigbee, WiFi, Bluetooth, UWB) is presented, including the comparison between classic Bluetooth and the more recent, Bluetooth Low Energy.

Chapter 3: In this chapter, the wireless temperature sensor design is discussed in detail, including specifications, design principle, choice of components, PCB layout and the communication program.

Chapter 4: This part of the thesis presents the various experiments/tests done with the sensor. The long-term stability of the NTC thermistor is measured separately, and is discussed in detail. This is followed by the qualification tests of the wireless temperature sensor. Next, a comparison of the long-term stability of BJT-based sensor and the wireless sensor is presented. Finally, the experimental setup for the long-term drift measurement of one of the BJT-sensors available on the market is presented. The results of these series of experiments are included in Chapter 5.

Chapter 5: Complete set of results of all tests discussed in chapter 4 are presented here and discussed in detail.

Chapter 6: This is the concluding chapter, in which some important inferences drawn from this thesis are presented.

The references are included at the end of each chapter.

Chapter 2: An overview of existing wireless temperature sensors

Wireless temperature sensors are used in various applications. Section 2.1 provides an overview specifically about some of the industrial applications. Section 2.2 gives a comparison of the wireless products available on the market. Further, a few high-performance temperature sensors reported in the literature are also discussed which could be made wireless. The chapter ends with an introduction to low-power wireless communication, comprising of comparison between different wireless protocols, deducing which among them is most suitable for the present application (section 2.3).

2.1 Industrial applications

In industrial machines operating in harsh environments, wired sensing systems sometimes increase infrastructure complexity and costs. Owing to the severe magnetic fields, wires serve as a medium for the interference to couple with the data. If the machines are hermetically sealed, the use of wires needs creation of extra paths and feedthroughs. Wireless solutions serve to be robust solutions for such environments since it is able to penetrate through many materials. Also, the frequency of operation of wireless protocols is much higher than the frequency of the electromagnetic interferences.

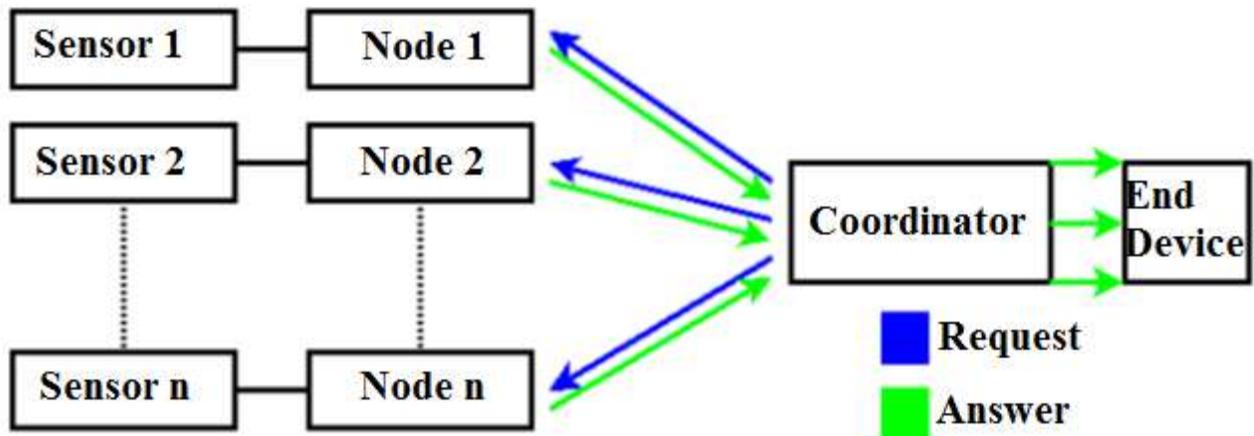


Figure 1.1a Simplified block diagram of the wireless system designed using IQMESH protocol [1]

Lately, many wireless temperature sensing solutions for industrial applications have been reported [1-6]. Most of them use the ISM (Industrial, Scientific and Medical) band of frequencies for the wireless communication, which include different wireless protocols. The solution in [1] is dedicated to industrial systems that need to be upgraded often. In such a case, RF (radio frequency) is suitable which allows fast manipulation of the RF parameters. The work is based on the evolving IQRF platform. The simplified block diagram of the system (figure 1.1a) includes temperature

sensor connected to the transmitter (node). The transmitter communicates with the receiver (coordinator) using the IQMESH protocol (wireless link). The receiver is connected to the end device using RS232 link.

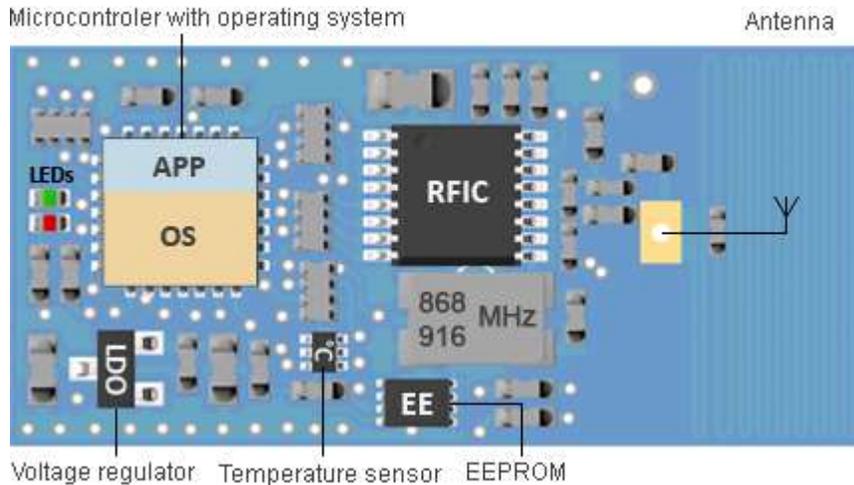


Figure 1.1b Structure of IQRF transceiver module [1].

The IQRF transceiver module is shown in Figure 1.1b. Radio frequencies can be selected among 868/916 MHz or 433 MHz bands. The temperature sensing is done using 1-Wire Digital Thermometer, Maxim DS18B20. The sensor has the supply voltage range from 3 to 5.5 V. The measurement range is $-55\text{ }^{\circ}\text{C}$ to $+125\text{ }^{\circ}\text{C}$, with accuracy of $\pm 0.5\text{ }^{\circ}\text{C}$, in the range $-10\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$. The resolution can be set to 9, 10, 11 or 12 bits by the user, where the 12-bit (maximum) resolution corresponds to $0.0625\text{ }^{\circ}\text{C}$.

Another application uses the same 1-Wire digital temperature sensor (Maxim DS18B20) but with Bluetooth wireless communication, for synchronous machine rotor temperature measurement [2]. In [2] a prototype of the measurement system along with 27 DS18B20 temperature sensors was mounted on a 400kVA, 1000 RPM synchronous generator. Since the rotor is in motion, the measurement system is battery powered. Reliable communication between the sensor and the Bluetooth link has been established.

In [3] an industrial application is reported for temperature monitoring and warning system. This application uses Zigbee, a wireless protocol popularly used in battery-powered low power applications (details in section 2.3). In this application, the vehicle consists of three parts: the wireless temperature detection module, inter-vehicle transmission networks module, and the remote monitoring server. Wireless temperature detection module consists of several wireless temperature sensors, with each wireless temperature sensor detecting a switchgear contact temperature [3]. The direct sequence spread spectrum (DSSS) is used for modulation. The temperature sensors use the low-temperature co-fired ceramic (LTCC) built-in antenna - a high performance patch antenna, for transmission of measured values.

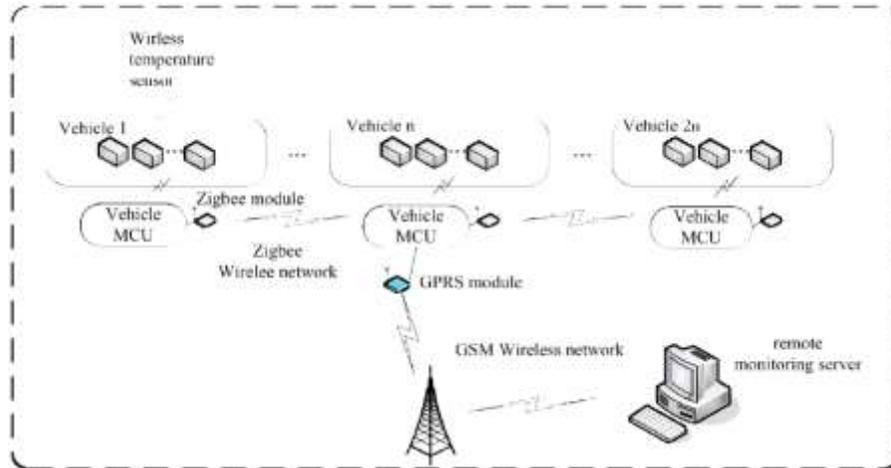


Figure 1.2: The complete system for the temperature measurement reported in [3]

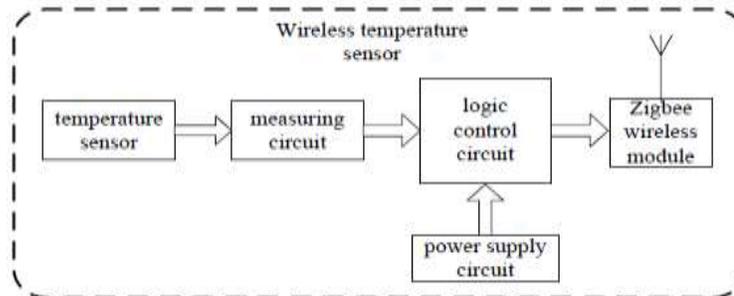


Figure 1.3: Wireless temperature sensor block diagram used in [3]

High operating temperatures can also be considered as harsh environment in industrial machines. Many of the commercially available sensors cannot withstand the higher temperature and hence are not suitable. In order to prevent system failure due to increasing temperatures, monitoring is essential. In [6], a completely passive LC resonant telemetry scheme is integrated with high-k temperature sensitive ceramic materials. The sensor is fabricated and successfully implemented for high temperatures of 235 °C. This work proved the concept of temperature sensing through passive wireless communication.

For similar high temperature applications, some Surface Acoustic Wave (SAW) sensors are also available in the market [21-22]. Their operating range is 200-300 °C and the accuracy is ~0.5 °C.

Apart from industrial applications, wireless temperature sensors have been used for many other applications like health monitoring, patient monitoring, ambient temperature monitoring, home automation networks and so on [7-13]. However, both SAW sensors and other wireless temperature sensing applications are out of scope for this thesis work as it is restricted to wireless temperature sensor application for advanced industrial machines.

2.2 Available products

2.2.1 Wireless temperature sensors

A comparison of commercially available wireless temperature sensors, using wireless protocols like Wifi, Zigbee [14-22], is given in table 1.1.

Table 2.1: Comparison of different wireless temperature sensors available in the market

Name	Range (m)	Dimensions (L x B x H) (mm)	Accuracy (° C)	Precision (° C)	Resln (° C)	Battery life (hrs)	Samples /s	LTD (mK/yr)
Connect sense	-	-	± 0.2	± 0.5	-	-	-	-
Monnit	91.44	44.9x26.4 x 19.9	± 1	-	-	8760	1/3600	-
STH-03ZB	100	92 x50 x 23	-	-	-	8760	1/60	-
THN132N	-	59 x36 x 21	-	-	0.2	-	-	-
Go wireless	-	-	± 0.5	-	0.07	2	1/20	-
SHT21	-	-	± 0.3	± 0.1	0.01	-	1/30	<200
Temp @lert	-	129.3x 67.3 x 40.6	± 0.5	-	-	43800	1/300	-
AcuRite	-	121.9x 40.6 x 22.9	± 2	-	-	-	-	-
TSAH10X	5	80 x 9	± 2	-	-	-	-	-
SENSoeR	-	5 x 5; 9.8 x 3;	± 0.5	-	0.1° C	-	-	-

Note: ‘-’denotes that this information is not available with the product specifications.

To summarize the characteristics of commercially available products:

- Wide operating ranges (-40°C to 125°C, 0°C to 60 °C), while many advanced industrial machines use much lower working temperature range, often within 20 – 25 °C.
- The commercially available products cover a wide temperature range, resulting in lower accuracy: from 0.2 to 2 °C, and low resolution: the best figure being 0.07 °C. Hence, for accurate and high-resolution temperature measurement, such sensors are not suitable.

- They are bulky with size starting from hundreds of millimeters and more. For advanced industrial applications, small foot-print sensors are preferred to save space.
- They also have slow rate of data transmission. Usually, this is traded-off with the power consumption of the sensor and the battery lifetime. They are not suitable for applications requiring constant temperature monitoring (for example one sample every second).
- The long-term drift information of the sensors is typically not specified for most of the products, and when specified, it is not impressive at all. For example, one of the products is reported to have a stability of 0.2 °C/year [19]. Long-term drift is an important piece of information which is missing.

2.2.2 High-performance temperature sensors

Apart from the wireless temperature sensors reported in literature and the commercially available ones in the market, we have investigated temperature sensors with high-performance in terms of resolution, power and size [23-43], which might be used in wireless configuration. They are made either using off-the-shelf components with various sensing elements (Platinum sensors, thermocouples, thermistors), or integrated in silicon (BJT and CMOS) with competitive performance [23-29]. In this section, few of such sensors are presented which could meet the requirements of advanced temperature sensing applications and could be potentially made wireless. A few temperature sensors having sub-mK resolution, low power consumption, small volume and small bandwidth are presented in Table 2. The CMOS and BJT sensors are clear winners in terms of size and power consumption, but they do not have sub-mK resolution, which can be achieved by thermistors. Also the extra circuitry employed for reaching lower power and resolution levels in integrated ICs are expected to introduce drift over time. The long-term drift parameter is not measured for most of the designs in table 2.

The BJT sensor in [44] is reported to have high resolution (3mK in a 2.2ms measurement time) and high energy efficiency, as expressed by a resolution FoM (figure of merit) of $3.6\text{pJ}\cdot\text{K}^2$, which is said to be a 3× improvement on the state of the art. By employing chopping, dynamic element matching and a single room temperature trim, the sensor also achieves a spread of less than $\pm 0.15^\circ\text{C}$ (3σ) from -45 to 130°C . In [44], the sensor's performance is compared with that of other energy-efficient precision temperature sensors. The design achieves the highest energy efficiency, as well as the highest reported resolution for a BJT-based temperature sensor. However, the long-term drift performance has not been reported. In chapter 4, the experiment conducted to measure the drift of this sensor is presented. In chapter 5, a comparative study of the results of long-term drift of this record-holding BJT-based temperature sensor [44] and the wireless prototype designed in this work is presented in detail.

Table 2.2: A few sub-mK resolution, low power and small foot print temperature sensors reported in the literature.

Authors	Dissipation (mW)	Volume (mm)	Resolution (μ K)	Sensor type	BW (Hz)	year
Wudy, Franz E et al [30]	0.04 (thermistor) 12.5 (Electronics)	1000x 1000x600	± 100 (-30 to +30 $^{\circ}$ C)	30 K Beta Therm NTC (epoxy coated bead type)	10	2011
Ambrosetti. R, E.Matteoli, D. Ricci [31]	¹ 5.125 (Electronics)	Not mentioned	380	PT100; but the Readout can be used for thermistor and thermocouple as well	0.1-10	2012
Sanjuan, J, et al [32]	² 0.010 (thermistor)	Not mentioned	10	Beta therm NTC 10K G10K4D372	1-30 (10^{-3})	2007
Libbrecht, K. G., et al [33]	¹ 14.5 (Electronics)	Not mentioned	100	Thermistors	0.1-4	2004
Wu. Rong, et al [34]	1.35 ³ (Electronics)	3x3	1	Bridge sensors	5	2011
Guijie wang, et al [35]	0.374 (Sensor + Electronics)	0.8x0.8	200	BJT sensors	Not mentioned	2014

¹Power dissipation of the sensor is not mentioned

²Power dissipation of the Electronics is not mentioned

³ This design is only a read out IC which can be used on different types of temperature sensors used in bridge

2.3 Introduction to low-power, wireless communication

In this section, the main features of the four short-range wireless communication protocols, namely: Bluetooth, Zigbee, WiFi and UWB are presented. Further, a comparison of Bluetooth and Bluetooth Low Energy is made. Finally, the two most suitable protocols for industrial applications: Zigbee and BLE are compared. The information presented here is widely available in literature [45-51]. An attempt is made to summarize the characteristics of each protocol, thereby deducing which among them is the most suitable for the intended application.

Bluetooth: For replacing cables between mobile phones, small peripheral devices like keyboard, mouse and headsets, a radio frequency communication was established known as Bluetooth. It usually supports short-range data communication: 1-10 m. However, it can also support higher distances with increased power consumption. When two devices are connected using Bluetooth, they form a 'pico net'. Many pico nets together form a 'scatter net'. Bluetooth is an example for "Single-hop-point to multi-point" data transfer, which means all the data is transferred through a central point [ref]. It is used in device-to-device, low power (maximum 1 W) and small bandwidth applications.

ZigBee: It is an open wireless standard designed for low power, low data-rate (250 kbps) and low cost applications. It is most suitable for applications requiring longer battery life, as the wireless device is in sleep mode for most of the time. For industrial applications, where machine-to-machine interaction is required using battery powered sensors, ZigBee is found to be useful. It has a star network topology in its most basic form (Single-point to multi-point data transfer), but also allows constructing mesh network applications (multi-point data transfer).

UWB (Ultra-Wide Band): As the name suggests, this wireless protocol has a large data rate of 110 Mbps. It uses a large band of frequency to transmit data in parallel, with each packet transmitted in a short pulse of time. It is suitable for applications requiring high speed and wide bandwidth for example: audio and video applications. However, a disadvantage of UWB is the high power consumption. Hence, under the condition to restrict its power consumption, it was made commercial in 2002. On the plus side, it has higher distance coverage (100 m) as compared to Bluetooth and ZigBee, but at the expense of higher power consumption.

WiFi (Wireless Fidelity): is a communication standard operating both at 2.4 GHz and ultra-high frequency like 5 GHz. Usually WiFi is used in computer to computer communication that can replace high-speed Ethernet cables. It is sometimes called wireless personal area network (WPAN). It can be used to provide internet access to different devices within certain distance (maximum 100 m). Wi-Fi has high data rate, high bandwidth and high power consumption.

A comparison of the different protocols mentioned above is presented in table 3. Maximum power consumptions for the commercially available chipsets (mentioned) using each of the protocols is calculated. For industrial applications where battery powered, stand-alone wireless sensors are deployed for short range data acquisition, requiring low power consumption and low data rate, Bluetooth and ZigBee are found to be the most optimum wireless protocols.

Table 2.3: Comparison of the four short-range wireless protocols

Parameters	Bluetooth	ZigBee	UWB	WiFi
IEEE standard	802.15.1	802.15.4	802.15.4a	802.11 a/b/g
Operating Freq	2.4 GHz	868/915 MHz; 2.4 GHz	3.1 – 10.6 GHz	3.4 GHz; 5 GHz
Range (meters)	10	10-100	10*	100
Data rate	1 Mbps	250 Kbps	110 Mbps	54 Mbps
Tx Power (dBm)	0 – 10 dBm	(-25) – 0 dBm	-41.3 dBm/MHz	15 – 20 dBm
Modulation	GFSK	BPSK(+ASK), O-QPSK	BPSK, QPSK	BPSK, QPSK, COFDM, CCK, M-QAM
Error mechanism	16-bit CRC	16-bit CRC	32-bit CRC	32-bit
Max power consumption	Bluecore 2 0.1 W	CC2430 0.081 W	XS110 0.749 W	CX53111 0.722 W

*UWB restricted for low power applications has limited range of 10 m. For higher range, the power consumption is higher

In Dec 2010, Bluetooth Low Energy (BLE), also called Bluetooth Smart was introduced specifically for low power applications in the area of health care, fitness, handheld devices and other battery-powered applications. It was believed to be a low cost and low power wireless communication standard, while still providing the same range as classic Bluetooth. The power consumption could be as low as 10 mW. A detailed comparison between classic Bluetooth and Bluetooth Low Energy is given in table 4.

Table 2.4 Comparison between classic Bluetooth and Bluetooth Low Energy

Technical Specification	Classic Bluetooth technology	Bluetooth Low Energy
Distance/Range (theoretical max.)	100 m (330 ft)	100 m (330 ft)
Over the air data rate	1–3 Mbit/s	1 Mbit/s
Application throughput	0.7–2.1 Mbit/s	0.27 Mbit/s
Active slaves	7	Not defined; implementation dependent
Security	56/128-bit and application layer user defined	128-bit AES with Counter Mode CBC-MAC and

		application layer user defined
Robustness	Adaptive fast frequency hopping, FEC, fast ACK	Adaptive frequency hopping, Lazy Acknowledgement, 24-bit CRC, 32-bit Message Integrity Check
Latency (from a non-connected state)	Typically 100 ms	6 ms
Minimum total time to send data (det.battery life)	100 ms	3 ms
Power consumption (max according to the standard)	1W	0.01 to 0.5 W (depending on use case)
Peak current consumption	<30 mA	<15 mA

After, analyzing the different wireless protocols, for our application we can narrow down our choices to either ZigBee or Bluetooth low energy. A comparison between the power consumptions of ZigBee and BLE in cyclic sleep scenario has been made [44]. Two commercially available chipsets, TI's CC2450 and XBee S2, which use BLE and ZigBee, respectively, were used. One package of 8-byte length was transmitted between the nodes. The current consumption of BLE during the sleep and awake mode was found to be 0.78 μA and 4.5 mA, respectively, while for ZigBee, the sleep and awake mode currents were found to be 4.18 μA and 9.3 mA, respectively [table 5]. The power supply used was 3.3 V. It is to be noted that, determining the sleep and awake currents for the protocols is mentioned to be challenging, as the total power consumption depends on a number of interacting factors, and is not just the combination of the two currents.

Table 2.5: Comparison of current and power consumptions of ZigBee and BLE

Parameters	ZigBee (XBee S2)	BLE (TI's CC2450)
Sleep current	4.18 μA	0.78 μA
Awake current	9.3 mA	4.5 mA
Power consumption (total)	30.07 mW	14.85 mW

2.4 Conclusions

In this chapter, a survey of the state of the temperature sensors for industrial applications was presented in detail. The wireless temperature sensors available in the market are bulky, power hungry and have larger footprint along with low resolution and accuracy. Other high-performance temperature sensors including sensors made with off-the-shelf components and ICs, which can be

potentially made wireless were also surveyed. Although the integrated temperature sensors win over the sensors with off-the-shelf components in power and area, they are not able to reach the required resolution. Also, the information of the long-term stability of the sensors is not available. In chapter 4 and 5, the long-term drift of a best available BJT-based temperature sensor is measured and compared with the wireless temperature sensor designed in this work.

Next, the different available wireless protocols were surveyed to choose the most suitable among them for the present application. Bluetooth Low Energy (BLE) was found to be the most suitable wireless protocol that can be used for the for short range industrial applications requiring low power and low data rate.

Finally, it could be concluded that for industrial applications, a wireless temperature sensor using off-the-shelf components would be suitable to reach the resolution, accuracy, power and long-term stability parameters, with BLE as the protocol. In chapter 3, the specifications and design of the sensor are presented, followed by the qualifying experiments in chapter 4 and corresponding results in chapter 5.

References

- [1] Bazydło, Piotr, Szymon Dąbrowski, and Roman Szewczyk. "Wireless temperature measurement system based on the IQRF platform." *Mechatronics-Ideas for Industrial Application*. Springer International Publishing, 2015. 281-288.
- [2] Kovacic, Marinko, Mario Vražić, and Ivan Gašparac. "Bluetooth wireless communication and 1-wire digital temperature sensors in synchronous machine rotor temperature measurement." *Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010*. 2010.
- [3] Wei, Lexiang, Xiaoqing Zeng, and Tuo Shen. "A Wireless Solution for Train Switchgear Contact Temperature Monitoring and Alarming System Based on Wireless Communication Technology." *International Journal of Communications, Network and System Sciences* 8.04 (2015): 79.
- [4] Bell, Alexis, et al. "Wireless patient monitoring system." *Healthcare Innovation Conference (HIC), 2014 IEEE*. IEEE, 2014.
- [5] Folea, Silviu C., and George Mois. "A Low-Power Wireless Sensor for Online Ambient Monitoring." *Sensors Journal, IEEE* 15.2 (2015): 742-749.
- [6] Wang, Ya, et al. "A passive wireless temperature sensor for harsh environment applications." *Sensors* 8.12 (2008): 7982-7995.
- [7] Milos, Frank S., and Joan B. Pallix. *Wireless Temperature Sensors for Health Monitoring of Aerospace Thermal Protection Systems*. NDE for Health Monitoring and Diagnostics, San Diego, 2002, 4702-58.

- [8] Caldara, M. I. C. H. E. L. E., et al. "A wearable sweat pH and body temperature sensor platform for health, fitness, and wellness applications." *Sensors and Microsystems*. Springer International Publishing, 2014. 431-434.
- [9] Bell, Alexis, et al. "Wireless patient monitoring system." *Healthcare Innovation Conference (HIC)*, 2014 IEEE. IEEE, 2014.
- [10] Jeong, Seokhyeon, et al. "A fully-integrated 71 nW CMOS temperature sensor for low power wireless sensor nodes." *Solid-State Circuits, IEEE Journal of* 49.8 (2014): 1682-1693.
- [11] Pengfei, Li, Li Jiakun, and Jing Junfeng. "Wireless temperature monitoring system Based on the ZigBee technology." *Computer Engineering and Technology (ICCET)*, 2010 2nd International Conference on. Vol. 1. IEEE, 2010.
- [12] Gomez, Carles, and Josep Paradells. "Wireless home automation networks: A survey of architectures and technologies." *IEEE Communications Magazine* 48.6 (2010): 92-101.
- [13] Zhang, Yu, ZhiShan Wang, and JinHua Li. "Design a wireless temperature measurement system based on NRF9E5 and DS18B20." *Measuring Technology and Mechatronics Automation (ICMTMA)*, 2010 International Conference on. Vol. 1. IEEE, 2010.
- [14] Connect sense, wireless temperature sensor, 2015 <https://www.connectsense.com/wireless-temperature-sensor> (Accessed: July 2015)
- [15] Monnit, "Wireless Temperature Sensor", Datasheet
- [16] Nietzsche Enterprise Co., Ltd., "STH-03ZB User's Manual (ZigBee HA Profile)".
- [17] Oregon Scientific, Wireless temperature sensor, <http://store.oregonscientific.com/us/wireless-temperature-sensor-thn132n-p.html> (Accessed: July 2015)
- [18] Vernier, Go wireless Temp, <http://www.vernier.com/products/sensors/temperature-sensors/gw-temp/> (Accessed: July 2015)
- [19] Sensirion, "SHT21 Humidity and Temperature Sensor IC", Datasheet.
- [20] Acurite, Temperature and Humidity sensor, 00592tx <http://www.acurite.com/temperature-and-humidity-sensor-06002rm-592txr.html> (Accessed: July 2015)
- [21] Sensoer, "TSA H100 to TSA H105 Thermowell packaged Wireless Passive SAW Temperature Sensors with high-performance Antenna", Datasheet
- [22] Sensoer, "TSA H160 to TSA H165 (-V), Fixture mounted Wireless Passive SAW Temperature Sensors with high-performance Antenna Standard & Voltage-compatible versions", Datasheet.
- [23] Li, Jiang, Xu Weisheng, and Yu Youlin. "Accurate operation of a CMOS integrated temperature sensor." *Microelectronics Journal* 41.12 (2010): 897-905.
- [24] Souri, Kamran, Youngcheol Chae, and Kofi AA Makinwa. "A CMOS Temperature Sensor With a Voltage-Calibrated Inaccuracy of 0.15 C (3) From 55 C to 125 C." *Solid-State Circuits, IEEE Journal of* 48.1 (2013): 292-301.

- [25] Kamran Souri, Youngcheol Chae, Frank Thus, Kofi Makinwa. "A 0.85V 600nW All-CMOS Temperature Sensor with an Inaccuracy of $\pm 0.4^{\circ}\text{C}$ (3σ) from -40 to 125°C ." *Solid-State Circuits, IEEE Journal*, 2014
- [26] Makinwa, K. A. A. "Smart temperature sensors in standard CMOS." *Procedia Engineering* 5 (2010): 930-939.
- [27] Gaurav Singh, "A Gm-C Continuous-Time Sigma-Delta Modulator with improved linearity," MSc Thesis, Department of Microelectronics, Faculty of Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands, 2011
- [28] De Marcellis, Andrea, and Giuseppe Ferri. *Analog circuits and systems for voltage-mode and current-mode sensor interfacing applications*. Springer, 2011.
- [29] Rong Wu, "Precision Instrumentation Amplifiers and a Read-Out IC for Sensor Interfacing," Ph.D. dissertation, Department of Microelectronics, Faculty of Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands, 2011
- [30] Wudy, Franz E., et al. "Fast micro-Kelvin resolution thermometer based on NTC thermistors." *Journal of Chemical & Engineering Data* 56.12 (2011): 4823-4828.
- [31] Sanjuán, J., et al. "Thermal diagnostics front-end electronics for LISA Pathfinder." *Review of Scientific Instruments* 78.10 (2007): 104904.
- [32] Wu, Rong, et al. "A 20-b 40-mV Range Read-Out IC With 50-nV Offset and 0.04% Gain Error for Bridge Transducers." *Solid-State Circuits, IEEE Journal of* 47.9 (2012): 2152-2163.
- [33] Ambrosetti, R., E. Matteoli, and D. Ricci. "Note: A versatile, stable, high-resolution readout system for RTD and thermistor sensors." *Review of Scientific Instruments* 83.9 (2012): 096101.
- [34] Unni, PK Madhavan, M. K. Gunasekaran, and A. Kumar. " ± 30 μK temperature controller from 25 to 103°C : Study and analysis." *Review of scientific instruments* 74.1 (2003): 231-242.
- [35] Perrott, Michael H., et al. "A Temperature-to-Digital Converter for a MEMS-Based Programmable Oscillator With Frequency Stability and Integrated Jitter." *Solid-State Circuits, IEEE Journal of* 48.1 (2013): 276-291.
- [36] Libbrecht, K. G., and A. W. Libbrecht. "A versatile thermoelectric temperature controller with 10 mK reproducibility and 100 mK absolute accuracy." *Review of Scientific Instruments* 80.12 (2009): 126107.
- [37] Libbrecht, K. G., and A. W. Libbrecht. "Response to "Comment on 'A versatile thermoelectric temperature controller with 10 mK reproducibility and 100 mK absolute accuracy'" [Rev. Sci. Instrum. 80, 126107 (2009)]." *Review of Scientific Instruments* 82.2 (2011): 027102.
- [38] Cuadras, Angel, and Oscar Casas. "Determination of heart rate using a high-resolution temperature measurement." *Sensors Journal, IEEE* 6.3 (2006): 836-843.
- [39] Wu, Rong, Johan H. Huijsing, and Kofi AA Makinwa. "A 21b \pm 40mv range read-out ic for bridge transducers." *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International. IEEE*, 2011.
- [40] Higuchi, Sei, et al. "THERMISTOR BRIDGE PERFORMANCE IN THE 10– 4 TO 1 HZ RANGE AND ITS APPLICATIONS IN GRAVITATIONAL WAVE DETECTION PROJECTS."

- [41] Guillet, Bruno, Didier Robbes, and Laurence Méchin. "Low noise temperature control: application to an active cavity radiometer." *Review of scientific instruments* 74.1 (2003): 243-249.
- [42] Bell, E. C., and L. N. Hulley. "Precision temperature control." *Proceedings of the Institution of Electrical Engineers*. Vol. 113. No. 10. IEE, 1966.
- [43] Robert David, "A Liquid-in-Glass Thermometer with Sub-MicroKelvin Resolution, and its Application for Calorimetry" Ph.D. dissertation, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2006
- [44] Heidary, Ali, et al. "12.8 A BJT-based CMOS temperature sensor with a 3.6 pJ· K²-resolution FoM." *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014 IEEE International. IEEE, 2014.
- [45] Dementyev, Artem, et al. "Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario." *Wireless Symposium (IWS)*, 2013 IEEE International. IEEE, 2013.
- [46] Lee, Jin-Shyan, Yu-Wei Su, and Chung-Chou Shen. "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi." *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*. IEEE, 2007.
- [47] Lin, Jiun-Ren, Timothy Talty, and Ozan Tonguz. "On the potential of bluetooth low energy technology for vehicular applications." *Communications Magazine*, IEEE 53.1 (2015): 267-275.
- [48] Gharghan, Sadik Kamel, Rosdiadee Nordin, and Mahamod Ismail. "A Survey on Energy Efficient Wireless Sensor Networks for Bicycle Performance Monitoring Application." *Journal of Sensors* 2014 (2014).
- [49] Gungor, Vehbi C., and Gerhard P. Hancke. "Industrial wireless sensor networks: Challenges, design principles, and technical approaches." *Industrial Electronics, IEEE Transactions on* 56.10 (2009): 4258-4265.
- [50] Baker, Nick. "ZigBee and Bluetooth: Strengths and weaknesses for industrial applications." *Computing and Control Engineering* 16.2 (2005): 20-25.
- [51] Chakkor, Saad, et al. "Comparative Performance Analysis of Wireless Communication Protocols for Intelligent Sensors and Their Applications." *arXiv preprint arXiv:1409.6884* (2014).

Chapter 3: Wireless Sensor Design

This chapter includes the specifications, system design, choice of components, PCB layout and the Bluetooth communication program of the wireless temperature sensor. Section 3.1 includes the focus points of the design, indicating the most important parameters to be considered while designing the prototype. Next, section 3.2 focusses on the system design including the principle, the choice of suitable components and block & circuit diagrams. Further, section 3.3 covers the PCB and layout. Finally, the Bluetooth communication program for the sensor is explained in detail (section 3.4).

3.1 Specifications

The wireless temperature sensor prototype designed in this work has to meet the specifications given in Table I. These specifications were provided by ASML B.V. The temperature sensor is intended for diagnostic purposes in lithography machines at ASML.

Table 3.1: List of specifications for the design of the wireless temperature sensor prototype.

Description	Requirements
Measurement range (including)	15-30 °C
Bandwidth	1 Hz
Area	30 mm x 20 mm x 20 mm?
Accuracy	0.1 °C
Repeatability	=<1 mK/year?
Resolution	=<200 µK
Response time	=<1 s
Data point storage	65536 measurement points
Power (maximum)	=<1 mW
Sample interval range	0.2-3600 s
Calibration interval	>= 1 year
Stability in environment =<25 °C	=<5 mK
Sensor measures temperature of the material (phase)	Solid
Minimum stand-alone operation period (battery operation)	Three months

Precision integrated temperature sensors (CMOS or bipolar) cater to a different set of applications which require ultra-low power (nW) and very small area (mm²). This performance is usually traded-off with lower resolution – often restricted to a few mKs, higher long-term drift – due to the analog frontend (amplifier) used for achieving low noise, low power and small area. Apart from this, the sensor being integrated with the read-out electronics, can sometimes be unsuitable for applications which need the sensor to be placed separately. Hence, a system-on-chip using off-the-shelf components and involving PCB design is preferred for this wireless prototype design (explained in detail in later sections).

To summarize, the primary focus points of the design are: *low long-term drift, low power, small area and high resolution.*

3.2 System design

3.2.1 Principle of operation

The wireless temperature sensor prototype is a system-on-chip, as mentioned, and is made using - off-the-shelf components. For this prototype, the Bluetooth Low Energy (BLE) radio is not integrated in the PCB and is a separate component.

To begin with, the power budget of 1 mW was divided equally among the front-end circuit and the wireless communication module. Thus, the maximum power consumption in each module was restricted to 0.5 mW.

Secondly, for sub-mK resolution (0.2 mK), an NTC thermistor was chosen as the sensing element, as it provides maximum sensitivity for limited power consumption. For a relatively linear response and immunity to supply voltage variations and external interferences, it is best to use the NTC in a bridge configuration, followed by a high CMRR¹ (common-mode rejection ratio) input stage. For this prototype, performances of both half and full bridge configurations have been tested.

To meet the long-term stability of as low as 1mK/year, use of any components which severely drift over time (ex: pre-amplifier) was avoided. The principle of “direct digitization” was applied, by connecting the bridge to an Analog-to-digital convertor with high CMRR.

The measurement values from the sensor which are digitized by the ADC need to be stored in a memory, before they can be accessed by the BLE module for transmission to avoid loss of data points. A laptop would be used as a receiver.

The power supply for the sensor should be stable over time to avoid errors due to supply voltage variation, along with facilitating autonomy to the design. For simplicity, a battery is the first choice.

¹ CMRR is defined as the ratio of differential gain to common mode gain [1]. It is the measure of how effectively the noise signals common to both input terminals are rejected relative to the measured signals.

Few other possible sources like super capacitors, energy harvesters increase the complexity of the design. Hence they are out of scope for this prototype implementation.

3.2.2 Choice of components:

The first step in choosing the components for the front-end was the selection of a suitable battery. Table III shows a survey of the available battery technologies. The ½AA industry standard battery with Lithium thionyl chloride chemistry (3.6 V power supply) was chosen[5]. This type of battery has excellent stability of supply voltage over time. Also it meets the capacity- and area requirements set for this project.

Table 3.2: Survey of best suitable battery technology.

Type	Voltage	Size	Shelf life
Alkaline cells	1.5 V	X	X
Lead acid	2 V	X	X
Ni-H, NiMH	3 V	X	X
Li-Po	3.7 – 4.2 V	X	O
Li-Mn	3.0 V	O	X
Li-SOCl ₂	3.6 V	O	O

X: Does not meet the requirements; O: Meets the requirement

Next, the sensing element – the NTC thermistor, was chosen. Other temperature sensors like BJT, Resistance temperature detector (RTD) and many more, do not meet the resolution requirements. Thermocouples are also highly sensitive like thermistors, but they are suitable for applications requiring larger temperature range [ref]. Thermistors provide the sub-mK resolution along with accuracy, for the smaller temperature range in this design (around room temperature, 20 – 25 °C). The value of the NTC is chosen to be 10 kΩ at 25 °C. A disadvantage of the thermistors is their drift from the initial resistance value over time. A recent study [2] indicates that the best performing NTC for applications requiring long-term stability is NCP15XH103D03RC from Murata Manufacturing with drift rates as low as 0.54 mK/year (provided minimum mounting stress). Apart from that, it has small footprint being a surface mount device (SMD). This type of NTC is used in our design.

To achieve the required stability and resolution, the resistors used in the bridge are as important as the sensor itself. They must have low 1/f noise to be stable over time. In the resistance range of 10 kΩ, excellent performance is demonstrated by small SMD type resistors from both metal foil and top quality thin film technology. The current noise mentioned in the datasheets, is low enough for a noise target below 0.2 mK pp (pick-to-pick), for both technologies. The performance trade-off of precision is always between accuracy versus long term stability. Metal foil resistors are more

sensitive to humidity due to large active area and sandwich pattern of the trimming links. Some “heat dissipation” could be used to keep out humidity, in principle. But that is not possible here due to the low power requirement. However, newer manufacturing technologies focus on metal foil resistors that have better humidity performance. Also, the unique ‘trimming’ procedure used for foil resistors avoids creation of hot spots leading to better long-term stability [3]. Keeping all these constraints in mind, the FRSM type from Vishay Precision group with Load Life Stability of 0.0025% (25 ppm) and TCR of ± 0.05 ppm/°C has been chosen for our application [4]. The thermal noise of the full bridge was found to be 14 nV/sqrt Hz (corresponding to 10 kΩ equivalent bridge resistance), which is much less than the dominant noise source: the input voltage noise of the ADC.

The output of the bridge is directly connected to an analog-to-digital converter (ADC) with differential input and high CMRR. A survey of different available ADCs was undertaken. A comparison of the three most suitable ADCs is presented in Table II. The best match in terms of noise, power and CMRR was found to be AD7789 [5]. The input voltage white noise of the ADC is specified as 1.5 μV rms, corresponding to 4.5 μV 3σ-noise for a bandwidth (BW) of 16.6 Hz. For our application, the average of 16.6 samples was taken per second, which reduced the noise of the ADC to 1.02 μV pp. The output voltage of the bridge at 25 °C for 0.2 mK change in temperature, should be greater than the noise of the ADC to meet the resolution specification. If a voltage of 0.8 V is applied over the bridge, corresponding to ~34 μA current through the thermistor, the output voltage of the bridge corresponding to 0.2 mK temperature change at 25 °C is ~1.6 μV, which is well above the 1.02 μV pp noise floor.

Table 3.3: Comparison of ADCs regarding noise, power and CMRR.

Part Number	Power (mW)	Rms Noise (μV)	CMRR (db)
LT2413	1	0.8	120
LT2484/85	0.8	0.6	120
AD7789	0.2	1.5	100

In order to store 64k of 16-bit data 1MB memory storage is required. There are 2 options for the data storage – volatile or non-volatile memory. The volatile memory that could be suitable for this application is serial Static Random Access Memory (SRAM) in 8-pin socket. Typical representatives are 23LC1024 and 23LCV1024 from Microchip. The non-volatile memory can be Electrically Erasable Programmable Read Only Memory (EEPROM) or Ferro-electric RAM (FRAM). But taking into account the higher consumption and much longer write time, EEPROM is excluded. The lowest

consumption FRAM currently available is FM24V10-G/ FM25V10-G from Ramtron (Cypress). A comparison between the parts is given on table 4:

Table 3.4: Comparison of most suitable SRAM and FRAM memory chips for this design, available in the market

Part No.	Density (Mbit)	Supply Current (TYP)		Supply Current (MAX)	
		Sleep Mode (μA)	Read/Write (mA)	Sleep Mode (μA)	Read/Write (mA)
23LC1024	1	4	3	10	10
23LCV1024	1	4	3	10	10
FM24V10-G	1	5	-	8	0.4
FM25V10-G	1	5	-	8	0.3

FRAM has the advantage of having lower power consumption and the ability to retain the data even if the battery is completely discharged (all the data acquired during the operation of the device can be read after the battery is replaced). On the other hand FRAM is ~ 4 times more expensive than serial SRAM. So, if the price is a concern, then the cheaper option among the two can be used. Both devices are pin-to-pin compatible and can be replaced without a PCB change (a small change in the firmware will be required). In this design, FRAM (FM25V10-G) is used, as it has the lowest power consumption.

Finally, the other half of the design- the BLE wireless communication module was chosen. There are different BLE chipsets on the market. As the space and the power consumptions are limited in this application, the main criteria is to have a single chip design with available user flash memory for the specific application and minimum power consumption. A comparison of some of the most wide-spread solutions is given on table 5. Based on the most common criteria, the component that best suits this application is CY8C4xxx-BL from Cypress. It has also integrated balun² that simplifies the antenna matching network and additionally reduces the required board space (if there is no balun at least 9 external components are required, CY8C4xxx-BL requires only 2 parts).

² Balun definition: Balun is used to "balance" unbalanced systems - i.e. those where power flows from an unbalanced line to a balanced line (hence, balun derives from balance to unbalanced) [6].

Table 3.5: Comparison of BLE chipsets available in the market suitable for this design

Features	CY8C4xx7-BL (CYPRESS)	nRF51822 (NORDIC)	CC2541 (TEXAS INSTRUMENTS)	CSR1011 (CSR)
CPU, Speed	ARM CM0, 48 MHz	ARM CM0, 16 MHz	8051, 32 MHz	16-bit RISC, NA
Flash, BLE Stack (KB)	128/256, 64	256, 80	256,130	64(ROM),64
Supply Voltage	1.9-5.5	1.8-3.6	2.0-3.6	1.8-3.6
Serial Interface	4 SPI, 2 I2C, 3 UART	2 SPI, 2 I2C, UART	2 SPI, I2C, 2 UART	SPI, I2C, UART
ADC	12-bit, 1-Msps	10-bit, 14.7-kbps	12-bit, 7.5-kbps	10-bit, 0.7-kbps
Packages/GPIOs	QFN, CSP / 36	QFN, CSP / 32	QFN / 23	QFN / 32
Rx Sensitivity (dBm)	-92	-93	-88	-92.5
Avg. Current (μ A) (1-sec, 4-sec connection)	18.9, 5.7	15.5, 5.6	24, 6.75	28, 10.75
Deep-Sleep Current (μ A)	1.3	2.3	1	5
Hibernate Current (nA)	150	2,400	500	Not supported
Stop Mode Current (nA)	60	No	No	600
Integrated Balun	Yes	No	No	Yes

3.2.3 Block diagram:

The block diagram of the prototype is given in Figure 1. This provides an overview and the flow of information of the whole temperature measurement system.

The temperature measured by the sensor goes to the ADC. The converted values from the ADC are stored in the memory, which is then accessed by the BLE module for transmission. A laptop is deployed as a receiver. A functional block diagram of the sensor (transmitter side), indicating the components in each stage is given in Figure 3.1.

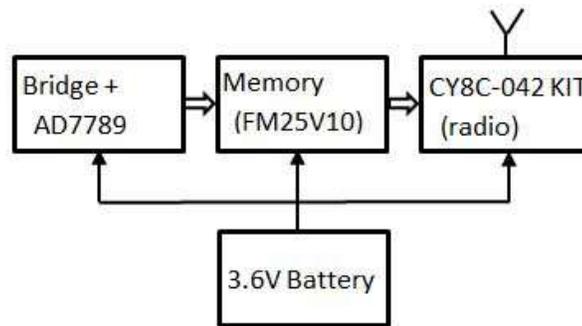


Figure 3.1: Functional block diagram of the wireless temperature sensor prototype

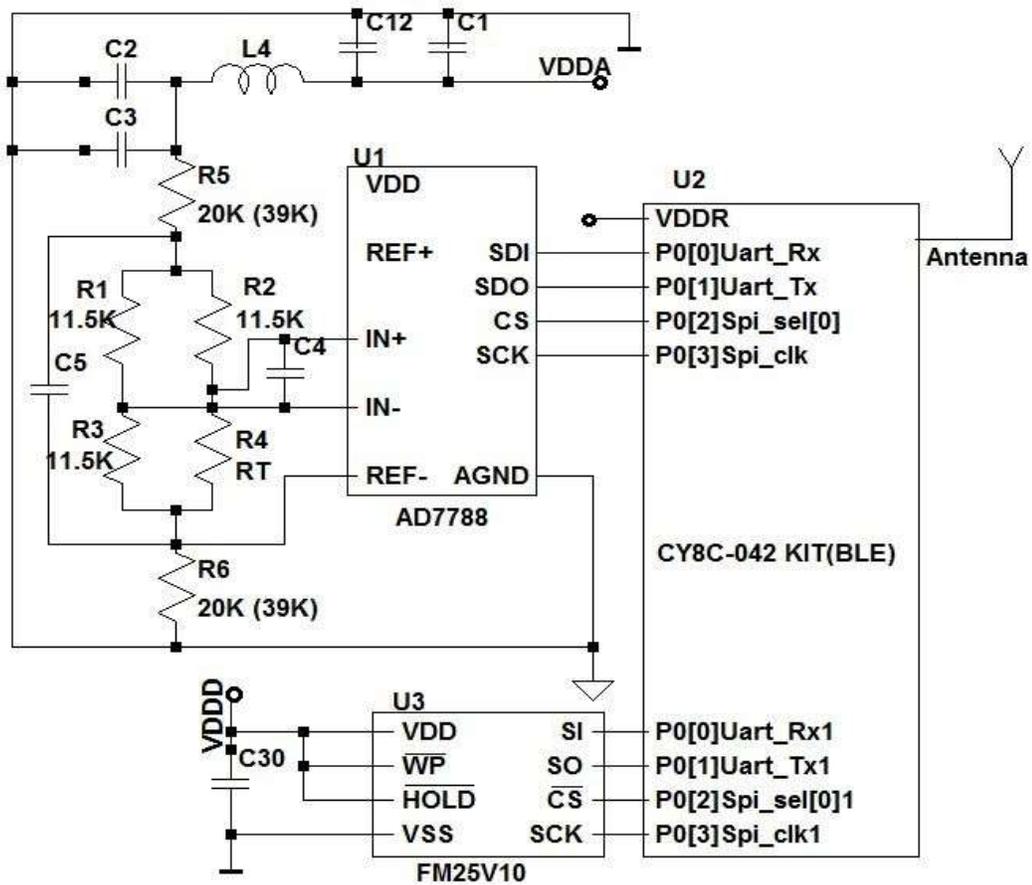


Figure 3.2: circuit diagram of full (half) bridge configuration of the wireless sensor

iii. Circuit Diagram: Figure 3.2 depicts the front-end circuit diagram half/full bridge configuration. The design is kept simple, with minimum possible components, in order to meet the long-term stability specification of ≤ 1 mK/year. The half bridge configuration comprises one arm of the full bridge, consisting of the NTC and resistor R2. In this case resistors R5 and R6 have the value mentioned in parenthesis (39 K Ω). In this way, the current through the NTC in both configurations is ~ 34 μ A.

The resolution of the sensor can be increased to 0.1 mK by reducing the values of resistors R5 and R6 to 10 k Ω and 20 k Ω respectively for full and half bridge configurations, thereby forcing ~ 55 μ A current through the NTC in both the cases. However the power dissipated in the NTC increases, from 11.56 μ W to 30.25 μ W.

3.3 PCB and Layout

Two prototypes of the wireless temperature sensor were made: one with full and one with half bridge configuration³. Figure 3.3a and 3.3b show the layout of the half bridge and 3.3c depicts the arrangement of components on the PCB for the same. For the full bridge, everything remains the same except the addition of an extra connector adjacent to connector C2. Some snapshots of the final prototype are shown in Figure 3.4.

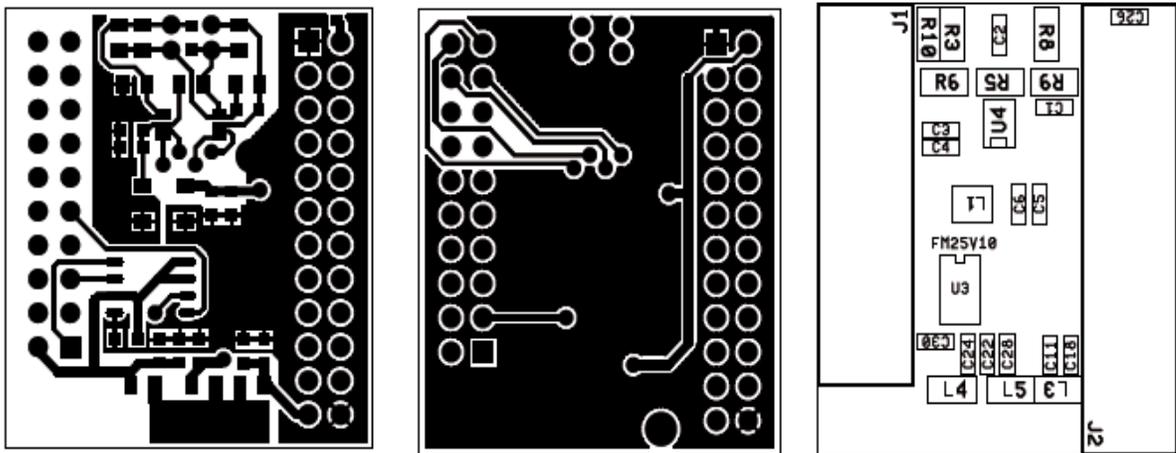


Figure 3.3 (a, b, c): Layout and component arrangement of the half bridge configuration.

³ While suggesting the front-end design, simulating it and choosing components was entirely my work, I had the assistance of Mr. Alexander Kerezov from Sofia University, Bulgaria, for making the PCB, the Layout and the Bluetooth communication program (section 3.4).

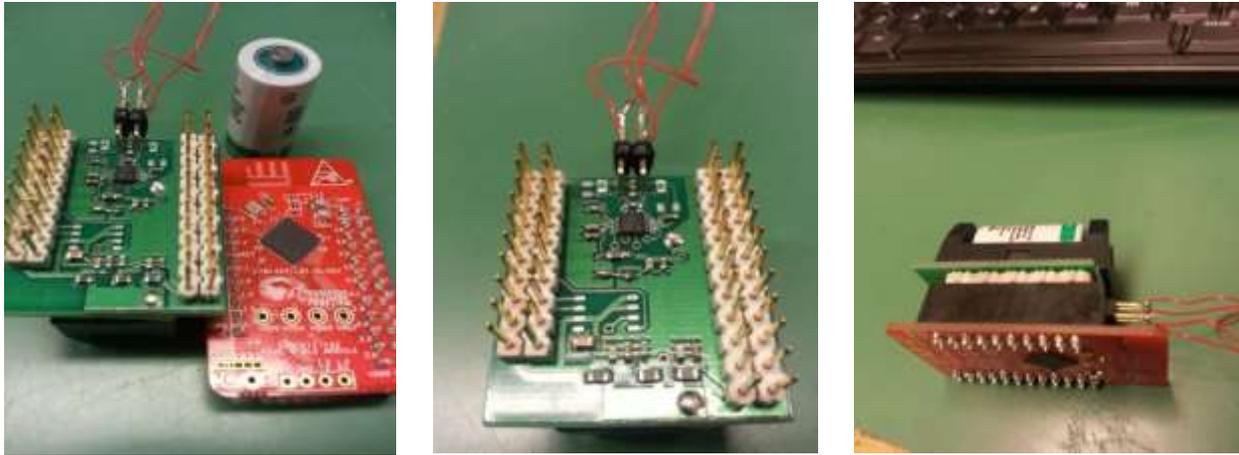


Figure 3.4: Snapshots of the wireless sensor prototype along with the BLE module.

3.4 Bluetooth Communication Program

In this section, an overview of the programming part of the wireless communication is presented. Detailed steps with figures can be found in Appendix I.

3.4.1 CY8Ckit-042 BLE

The BLE module from Cypress, CY8C042-Kit is used for the wireless transmission of the temperature measured. CY8Ckit-042 BLE is one among the families of different embedded programmable system-on-chip (PSoC) devices, based on PSoC 4 architecture from Cypress [ref]. Programmable analog and digital peripheral functions, memory and a microcontroller are integrated on a single chip. It also includes two BLE modules especially for wireless communication. It is a user-friendly, cheap development platform (costs \$25), enabling the user to realize different designs using the flexibility of PSoC 4. The designs can be created using PSoC creator 3.1. It is an integrated development environment (IDE) for hardware and firmware designs of different families of PSoC devices. It has different analog and digital components (Example BLE, interrupt, watch-dog timer, ADC) with which users can “drag-and-drop” to create desired applications. The main program, written in C language, includes the way these components will be used and connected with each other to realize the design. For temperature measurement, there is an example project provided with the software, which is used as the basis in creating the communication of the wireless prototype. The program is edited to suit the requirements of the present application. A flow chart of the main program is presented next.

3.4.2 Program (flow chart)

The flow chart of the main function is presented in Figure 3.5. The program measures the temperature using AD7789 and transmits it over BLE Health Monitor Service. The complete code can be found in Appendix I.

1. BLE Events: In this step, various Application Program Interfaces (APIs) are called to clear the BLE stack and send/receive the necessary events to put the device to discoverable mode.
2. GAP Events: Generic Access Profile (GAP) is responsible for advertising the BLE device to the receivers. Basically it determines the way devices (BLE receiver and transmitter) can interact with each other (handshaking). In this step, information like address of the BLE sensor, pass key and other advertisement data are exchanged.
3. If the Authorization information received in the GAP events is correct, then the connection between the two devices is established.
4. GATT Events: The acronym GATT stands for Generic Attribute Profile. It defines the way the two devices can exchange data using parameters like **services** and **characteristics**. GATT is responsible for all the information exchange after the two devices are connected and ready to exchange data. After this step, the BLE is ready to send the temperature information.
5. Start WDT: This function will trigger an interrupt every second.
6. SPI: Serial Peripheral Interface is required to read data from the ADC and store it in the memory, from where it can be accessed and sent via BLE. This function is called to perform these steps.
7. Read the data: First the MSbyte is read, followed by the next byte and lastly the LSByte is read.
8. ISR: As the WDT is set to send trigger an interrupt every second, this ISR is executed every second.
9. Send data: As soon as the interrupt is received, data is sent using the antenna.

The flow chart represents flow of events to transmit one value of the measured temperature. This process continues, sending one value per second, unless the device gets disconnected from the receiver due to various possible reasons (packets lost, connection lost, device put to hibernate mode, power supply discharge leading to loss of connection and so on). Once the code is ready it is programmed into the BLE module using the PSoC programmer.

3.4.3 CySmart

At the receiver end, CySmart 1.0 GUI is used in windows laptop for receiving the values. The measured temperature values are received in 5-byte hexadecimal format out of which the first byte is used by the BLE stack to indicate the kind of service given by BLE (ex: temperature monitoring, Battery level and so on). The next three bytes are the data in the order of MSbyte followed by the next byte and LSbyte. The last byte is empty as the ADC is only 24-bit. The screenshots of the whole process of creating the program using PSoC creator and receiving the information using CySmart is included in Appendix II.

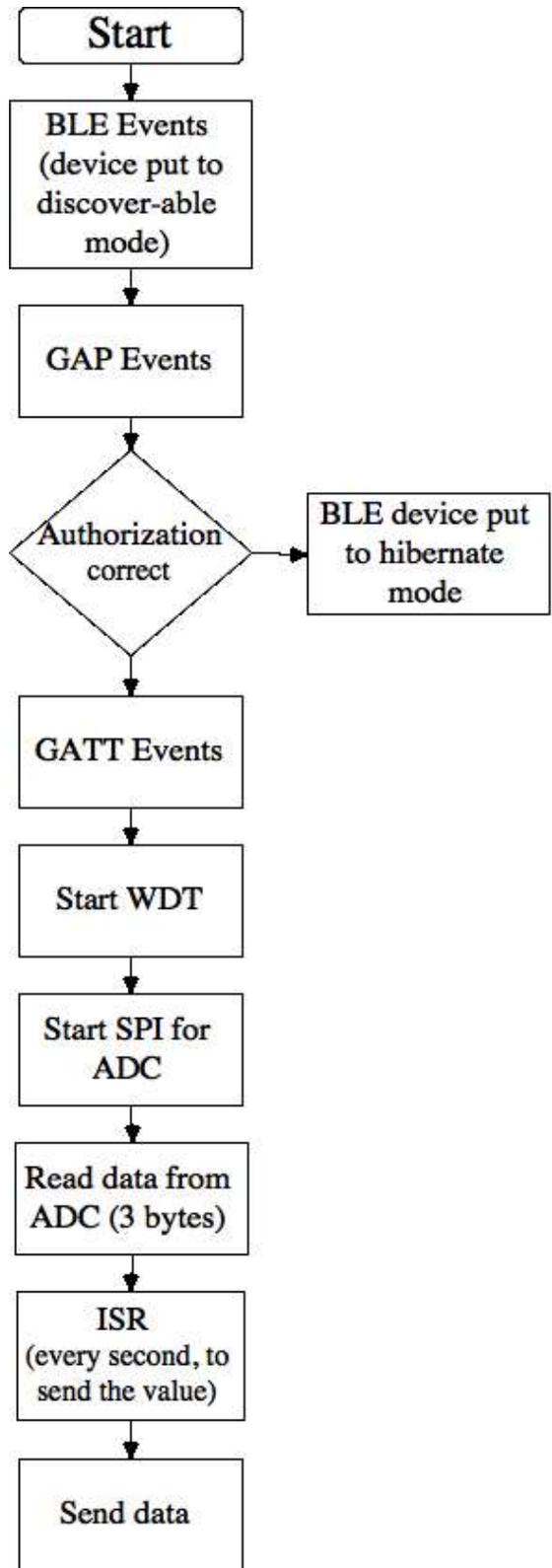


Figure 3.5: Flow chart describing the various events taking place to transmit one data of temperature measure by the sensor

References

- [1] MIT OpenCourseWare, "RES.6-010 Electronic Feedback Systems", Spring 2013, <http://ocw.mit.edu>.
- [2] Anupama Kulkarni, Mihai Patrascu, Yuri van de Vijver, Jaap van Wensveen, Robert Pijnenburg, Stoyan Nihtianov, "Investigation of Long-Term Drift of NTC Temperature Sensors with less than 1 mK Uncertainty", presented at the 24th IEEE International Symposium on Industrial Electronics, 3-5 June 2015, Búzios, Rio de Janeiro, Brazil
- [3] Vishay Precision Group, "How to select resistors and avoid unforeseen stress factors", January 2011.
- [4] Vishay Precision Group, "Z-1 Foil Ultra High Precision Wrap-around Chip Resistor for improved Load Life Stability of 0.0025% (25 ppm) with TCR of ± 0.05 ppm/ $^{\circ}$ C and withstands ESD of 25 KV min", FRSM resistor Datasheet, November 2012.
- [5] Analog Devices, "Low Power, 16-/24-Bit, Sigma-Delta ADCs AD7788/AD7789", 2006.
- [6] Antenna-theory.com, "Baluns", 2009-2011 <http://www.antenna-theory.com/definitions/balun.php>

Chapter 4: Experiments/tests

The chapter encompasses all the experiments conducted during the thesis. Section 4.1 includes the long-term drift measurement of NTC temperature sensors, which is one of the very extensive tests carried out during this project. Next, the qualifying tests for the wireless temperature sensor prototype, which cover the other solid half of the experiments, are presented (section 4.2). Finally, in section 4.3, experiment for long-term drift measurement of a state-of-the-art BJT temperature sensor is presented.

4.1 Long term drift of NTC temperature sensors

4.1.1 Motivation

Long-term drift of temperature sensors is critical to applications requiring high reliability. However, documentation and knowledge regarding long-term stability is limited. Usually manufacturers promulgate drift margins down to 10-20 mK/year, while the performance of the sensors might be much better. For some advanced industrial applications like lithography, optical radiometers, and space applications (the target applications of the wireless temperature sensor designed in this project), which demand drift rates down to a few mK/year, this information is inadequate. These applications also require sub-mK resolution along with long-term stability.

Temperature sensors like semiconductor diodes, Bipolar Junction Transistors (BJT), Resistance temperature detectors (RTDs) and so on, offer resolution in the range of a few mKs. Their resolution depends on a number of factors, for example, if the measurement time is long enough, the resolution can be very high, provided the drift is small. For temperature sensing applications requiring sub-mK resolution, semiconductor oxide (ceramics) based temperature sensors with high sensitivities are preferred. However, with the semiconductor oxide based sensors (often called thermistors), there is a gradual change in resistance from the initial value over time. This is known as “drift” and is more evident in high sensitivity sensors with Negative Temperature Coefficient (NTC). Hence, for applications requiring higher resolution, long-term stability usually has to be compromised. The reasons for drift are ambiguous and include migration of cat-ions in the material due to oxidation, process conditions (e.g: sintering temperatures), thermal cycling, humidity, stress, self-heating, material compositions and manufacturing techniques [1]-[8]. Few studies on sensor drift performance of temperature sensors have been conducted in the past [9]-[14]. Some have been able to reach measurement setup stability in the mK range [11], [12]. Other than the performance reported in the literature, manufacturers also measure drift rates and report them in datasheets [14]. However, their measurement setups feature drift margins >10 mK/year, and therefore they cannot measure drift in the mK/year range accurately.

In this project, a few state-of-the-art NTC temperature sensors available in the market are tested for long-term drift performance⁴. The results are obtained using a measurement setup with less than 1 mK uncertainty. The tested sensors are of the following types: glass bead, probe and SMD (Table 1).

Table 4.1: Details of the type, manufacturers and excitation voltages applied to the sensors being tested for drift performance

	Amt	Supplier	Type	Name	Operating conditions
1	10	Murata	SMD	NCP15XH103D03RC	0.6V AC
2	5	Meas Spec	SMD	SMD410KF38H	0.6V AC
3	5	Meas Spec	Probe	10K3MCD1	0.6V AC
4	5	Meas Spec	Bead	46036	0.6V AC
5	5	Meas Spec	Bead	55036	0.6V AC
6	5	Meas Spec	Bead	46036	0.6V DC
7	5	Murata	SMD	NCP15XH103D03RC	1.2V AC
8	5	Panasonic	SMD	ERTJ0EG103FA	0.6V AC
9	2	Vishay	Foil	Ref resistors[8]	
				Total = 45 Samples	

4.1.2 Experimental setup

The NTCs are inserted in a water bath and maintained at 22 °C (room temperature) by 16 Peltier elements. This is the temperature point of interest for this project and hence the measurements for drift are done at this temperature. Primarily, the drift in the resistance of the NTCs is measured and recorded. Later, this will be translated into equivalent drift in temperature. IPRTs (Industrial Platinum Resistance Thermometers) are used as reference sensors to measure the change in resistance of the NTCs (Fig. 1a). They are calibrated using a Triple Point of Water (TPW)⁵ cell (Fig. 1b). In addition to measuring drift, IPRTs are also used to regulate the temperature of the bath. The resistance of the IPRTs is measured using an F17 automatic resistance (AC) Ratio Bridge [15]. The AC Bridge is calibrated by two fixed reference resistors [16]. Additionally, a Digital Multimeter (DMM) from Keithley [17] is used to measure the absolute values of impedance of the NTCs (Fig. 1c).

⁴ The measurements were done at TempControl B.V, the Netherlands, under the expert guidance of Mr. Jaap van Wensveen.

⁵ The surface where all three phases of water co-exist in thermal equilibrium is called the triple point of water (TPW). As per definition, this occurs exactly at 0.010°C.



Fig. 4.1. a) IPRT sensors; b) Triple point of water cell; c) Readout set-up used to measure drift of NTCs, which includes AC resistance bridge and Keithley DMM.

4.1.3 Methodology

4.1.3.1. Calibration procedure

To maintain a setup with 1 mK/year stability, the blocks being calibrated during every measurement are: the waterbath, IPRTs and the AC resistance ratiometric bridge. Firstly, the short-term temperature fluctuations within the bath are kept below 50 μ K. The water in the bath is circulated at the rate of 0.6L/second and the temperature of the water bath is recorded continuously. Next, IPRTs are calibrated each time they are used, using a TPW cell (Fig. 1b). This cell is used to define the ITS 90 temperature scale. Its uncertainty contribution is considered to be 20 μ K/year, although drift rates of a triple point of water cells are reported to be much less [18] [19]. Each of the IPRTs is inserted into TPW cell and their resistances are logged for 15 minutes at the TPW. The values are translated into room temperature (22°C) using the Callendar-Van Dusen equation [20]:

$$R_t = R_0 (1 + A * t_n + B * t_n^2) \quad (1)$$

where R_t is the resistance measured at 22°C; Nominal Values of Coefficients according to IEC 751 (1995): $A = 3.9080 * 10^{-3} \text{ } ^\circ\text{C}^{-1}$; $B = -5.775 * 10^{-7} \text{ } ^\circ\text{C}^{-2}$; $R_0 = 100\Omega$ at 0 °C; and t_n is the measured temperature. The coefficients A and B are multiplied by t_0 and t_0^2 , respectively, ($t_0 = 273.15 \text{ K}$) to make them dimensionless to ease calculation and inverse use. Coefficients A and B were calculated for each thermometer before the experiment was initiated, in order to match the thermometers at 0 °C and 22 °C. If the slope of the curve of a thermometer needs adjustment, only the linear component A will be recalculated, leaving B unchanged.

Any deviation from the last value measured larger than 100 μ K at the TPW, leads to recalculation of the value of R_0 . Next, all sensors (Table 1) are placed in the water bath at 22°C. After complete stabilization, the IPRTs are inter-compared. If the total spread is larger than 200 μ K, then the sensors are checked again for correctness at 22°C. The linear coefficient A of the sensor which does not match the other will be recalculated.

The accuracy of the AC ratiometric bridge is ensured by using a set of fixed resistors. The resistors are placed in a bridge configuration. Next, they are swapped in their positions to obtain exactly the reciprocal of the previous value. This procedure is repeated with several resistor pairs. The bridge should be able to reproduce to at least 1 LSB (least significant bit), which corresponds to 255 μK for a Pt 100 sensor.

4.1.3.2 Drift measurement of NTCs

Initially, the stability of the water bath is measured. Later, the same procedure is repeated for drift measurement of the NTCs. To measure the temperature of the water bath, a fixed resistor and a Pt100 sensor (IPRT) are inserted into the bath. It is assumed that both register the same temperature in the water bath. The resistance value of the IPRTs is measured using the AC resistance ratiometric bridge. Since the measurement range of the AC Bridge is only 0-400 Ω for 100 Ω resistors, a Keithley Multimeter is used to measure the absolute values of impedances which are of the order of 11.5 k Ω . After complete stabilization of the water bath, the NTCs being tested are measured for drift. Three Pt100 sensors are kept in the water bath together with different sets of NTCs enclosed in tubes made of stainless steel (Fig. 2).

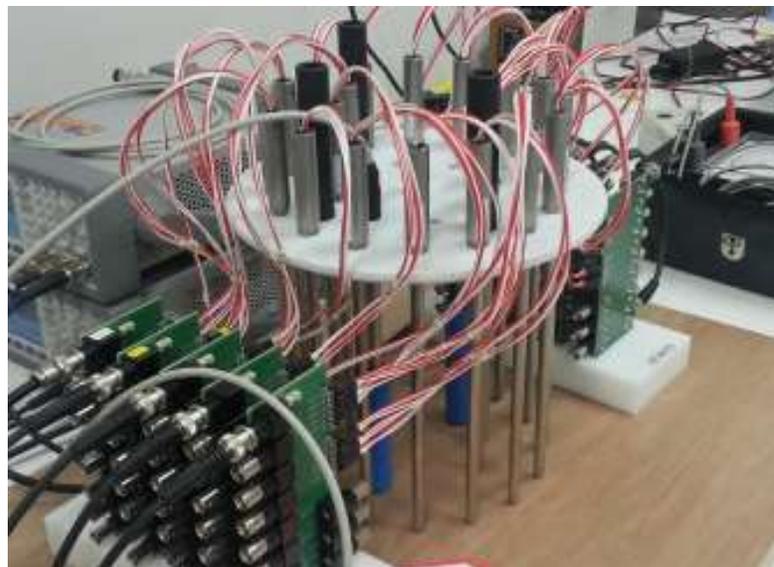


Fig. 4.2. NTCs enclosed in tubes along with three Pt100 sensors.

The tubes are placed in the same position each time the measurement is done. A software program ensures that the measured resistance of each NTC is recorded in the same order every time so that any offset or uncertainty is translated into a systematic error and can be disregarded later. Each measurement consists of the temperature of the water bath measured with the calibrated Pt300 sensor and 20 measurements of the NTC resistance, spread over 20 seconds. The mean and standard deviation of 2σ of the 20 measurements are calculated along with the slope, which represents the probable drift during a series of 20 measurements. All raw data as well as the calculated data are recorded.

The next step is to convert the raw data of resistance drift into equivalent temperature drift. The Steinhart-Hart method, which characterizes non-linear NTCs, is widely used for this step. By using a fitting equation with only three coefficients, temperature can be derived from the measured NTC resistance. However, nowadays accurate NTCs which reproduce within millikelvins are available. A curve-fit over larger temperature ranges may be less accurate than the accuracy of the measurements gathered by calibration. Once characterized, the total uncertainty in the temperature, determined with a calibrated NTC using the Steinhart-Hart method, can be unacceptably large. This point was noticed, thus an alternative characterization was developed⁶. It is based on a polynomial using the natural logarithm of the measured resistance divided by the resistance at 0 °C (or any other temperature within the calibrated range versus the temperature in °C) (2).

$$t_{bn} = \sum_{m=0}^i b_m \ln\left(\frac{R_{tn}}{R_0}\right)^{m+1} \quad i = 3; m = 0 \dots i; \quad (2)$$

A restriction of the Steinhart-Hart method is that only three coefficients can be used. With enough data available, more coefficients can be added to the alternate fitting method. Experimentally obtained calibration data was used to test this alternative characterization. For example, a comparison of the two methods for YSI 55033 NTC yielded the following results. The uncertainty in the curve using Steinhart-Hart equation with three coefficients was found to be 0.0122 °C. With (2), (four coefficients) the differences and uncertainties in °C were reduced to 9×10^{-4} , an order of magnitude of more than 10 times compared to the Steinhart-Hart equation. For each NTC type, a set of parameters was calculated from the resistance versus temperature tables, as published by the respective manufacturers.

Now, we can summarize all the uncertainties mentioned. The temperature fluctuations allowed within the bath are kept below 50 µK. The TPW cell has a defined reproducibility of 20 µK. The margin of error during calibration of Pt100 sensors at the TPW is 100 µK. When the IPRTs are inter-compared for deviation, 200 µK of margin is allowed. Finally, for the AC resistance ratioetric bridge, a margin of error of 255 µK is allowed. This corresponds to a total margin of error for the measurement setup of less than 1 mK. Note that this is the worst case estimation. In reality, the uncertainty value of the setup is expected to be well below 1 mK, as will be confirmed by the measurement results. As an extra check for the stability claim, two fixed resistors are measured for drift along with the NTCs.

4.1.4 Error minimization

The measurement results (presented in detail in chapter 5, section 5.1) exhibit that all the sets of NTCs follow a certain pattern in drift, indicating the presence of deterministic measurement error in addition to actual temperature drift. The sources of this kind of error are probably related to the

⁶ I had the assistance of Mr. Jaap van Wensveen for the development of this alternative characterization.

test set up and the measurement conditions which need further investigation. Also, the reason for drift of all the sensors in negative direction is unknown and needs further inspection.

To segregate the deterministic error from the actual temperature drift, the following error minimization techniques were employed:

1. All data from the same type of NTC were averaged and plotted on one graph (Fig. 5.2). Each line in the graph represents the average temperature drift including the deterministic error, in each set.
2. The initial phase where the NTCs show a larger variation (Fig 5.3) is considered to be a “pre-aging phase” (out of scope). Hence, it is excluded from the final result and no error minimization techniques are applied to this part of the data. No quantitative threshold has been used for setting this phase.
3. The drift behavior of the NTCs over time is equivocal. Some studies report about a few types of NTCs that have exponential behavior over time [11], indicating that the drift is higher initially. Gradually, the sensor stabilizes.

To predict the sensor behavior over time, either a linear or logarithmic fit is used. Both approximations were applied to the graphs and compared. Since no significant difference was found for the present data sets, the graphs were approximated to a linear function of offset (deterministic error) and slope (actual drift).

$$\text{Drift, } Y = mx + C \quad (3)$$

m = actual drift of the sensors; C = deterministic error

The segregated deterministic error is shown in Fig 5.4. Every line in the graph represents the error in each set of NTCs. Most of the NTCs exhibit an error of about 0.6 mK. In principle this value is the uncertainty of the measurement setup, which is less than 1mK.

4. In the final step, the error values of all the sensors are averaged and subtracted from the actual signal. The outliers present in the graph are not included while averaging the errors. The reasons for the behavior of the outlier-sensors are unknown and need further investigation. Including them while averaging the errors, increases the overall error. By performing this step, we segregate the deterministic error from the results. What remains is the actual drift value (Figure 5.5).

To summarize, through measurements we obtained results with an uncertainty of better than 1mK, while the uncertainty is further reduced by applying error minimization techniques.

4.1.5 Conclusion

The experimental setup and methodology for testing the long-term drift performance of a few state-of-the-art NTC temperature sensors was presented in this section. A measurement setup with stability better than 1 mK was introduced. After applying techniques to minimize the deterministic

error in the measurements, most of the sensors appear to have drift rates of less than 1 mK/year pp.

4.2 Wireless sensor qualification tests

After the design and implementation of the wireless temperature sensor prototype, the next step was to check if it meets all the specifications described in chapter 3, section 3.1. All the tests conducted to qualify the performance of the prototype are described in this section in detail.

4.2.1 Noise measurements

To measure the noise, 11.5 k Ω resistors were connected to the bridge instead of the NTC, so that the output voltage of the bridge gives the noise in the circuit. 420 samples for each full and half bridge (1 sample per second from the BLE module) were recorded and plotted. Figure 5.5 and Figure 5.6 show the noise performance of full and half bridge configurations, respectively. For full bridge, the noise measured is in μV , since the bridge is balanced. For the half bridge, we see an offset of about 0.4V due to the ratio metric measurement. If the first few samples are discarded, the noise in the full bridge is about 2 μV pp. The calculated standard deviation of the noise is 0.659 μV . For the half bridge, we observe some spikes in the noise measurements. If we ignore these spikes, then the noise is about 5 μV peak to peak and the standard deviation is 2.68 μV . For the full bridge, these spikes are less noticeable. The exact cause for the spikes is not known at present and shall be further investigated.

4.2.2 Transfer Characteristics

In order to plot the transfer characteristics of the sensor, a temperature chamber shown in Figure 4.3a was used, which has a temperature fluctuation of < 2 $^{\circ}\text{C}$. Temperature was increased in steps of 2 $^{\circ}\text{C}$ from 10 to 30 $^{\circ}\text{C}$ and then decreased similarly from 30 to 10 $^{\circ}\text{C}$ to observe if there are any hysteresis effects. A calibrated PT100 reference sensor and the designed wireless sensor both were inserted into holes drilled in an Aluminum block, and measured its temperature (Figure 4.3b). In this experiment, since the fluctuation inside the oven is large (~ 2 $^{\circ}\text{C}$), an aluminum block was used to filter the fluctuations (Figure 4.4). The Pt100 sensor was read out using a Keithley multimeter, connected to the laptop using a LabView program. Every time the temperature of the chamber was increased/ decreased, 45 to 60 minutes were needed for the temperature fluctuations in the block to settle within an inaccuracy of 3 mK. At this instant, the temperature of the reference sensor and the wireless sensor were taken. This experiment was done for both full and half bridge configurations.



Fig 4.3 (a) Temperature chamber used to measure the transfer characteristics of the sensor (b) PT100 sensor

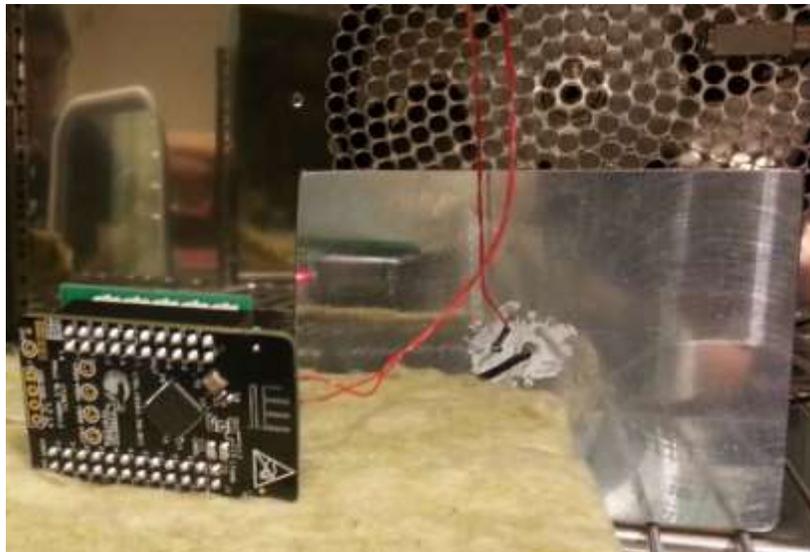


Fig 4.4. Aluminum block inside the chamber, connected to reference and wireless sensor.

4.2.3 Power consumption

The following are the power consumption calculation for both full and half bridge configurations for the prototype board. Since, the half bridge has two resistors less; it is expected to consume less current than the full bridge, hence being more power efficient. However, as mentioned in the section 4.2.2, the half bridge has higher noise than the full bridge. This is the trade-off that has to be noted.

The ADC consumes 65 μA current. The consumption of the BLE prototype is split into three categories: Average current (1-4 seconds connection time), Deep sleep mode current and current consumption during transmission. The current during the transmission is measured and found to

be 12.5 mA. The transmission time of the BLE is less than 100 μ s. The deep sleep current was measured and found to be 11 μ A.

For a half bridge, the current in the bridge is \sim 36 μ A while for a full bridge the current in the bridge is twice that of the half bridge. The calculation results will be presented in chapter 5.

It is to be noted that the BLE current consumption mentioned is for the prototype module from CY8C-Kit 042 pioneer kit. From the datasheet of CY8C4xx7-BL, the current is expected to be much lower.

4.2.4 Power supply rejection ratio (PSRR)

To check the immunity of both full and half bridge configurations to the variations in the supply voltage, the supply voltage is varied from 3 to 3.6 V in steps of 50 mV. For each value of supply voltage, 10 samples are taken and averaged. This is done to reduce the random variation in the graph so that the dependence on the supply voltage can be more explicit.

4.2.5 Reliability of communication

In advanced industrial applications like ASML machines, few vacuum tight, closed parts are used in some sections of the machine. The wireless temperature sensor designed in this project should be able to transmit and receive data in such an environment. In order to check the quality of communication of BLE protocol, tests involving signal strength vs distance were performed using a closed vessel (which can mimic the behavior of the closed parts of the machine) and compared to the performance of a sensor in an open environment.

As a first step, TI's CC2540 BLE kit was used for the tests. This kit comes with a "packet sniffer" profile which allows the user to capture the number of packets in error.

The BLE module was kept inside the vessel pumped down to vacuum and sealed at all the ends using copper tapes. The only way for the signal to come out was the feedthroughs (small screws coming out of the vessel).

4.2.6 Life-time tests of batteries in vacuum

After testing the signal strength of the BLE protocol from the closed vessel, the next experiment was conducted to check the behavior the batteries selected for this design, in vacuum. A vacuum vessel (figure 4.5) was used for the experiments. Two samples of LiSOCl₂ battery used in our application were mounted on a PCB as shown in figure 4.6, so that the power dissipation for each battery is restricted to 1 mW. They were connected through the feedthrough of the vessel to a data acquisition board (DAQ). Their operating voltages were recorded for duration of three months.



Figure 4.5: vacuum vessel used for testing the effects of vacuum on life-time of batteries

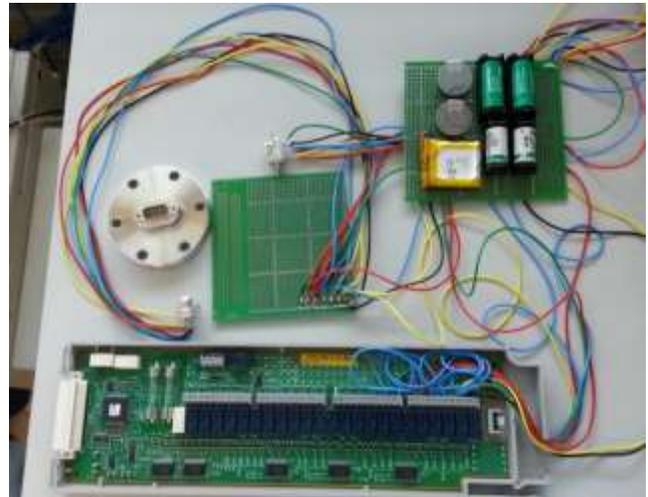
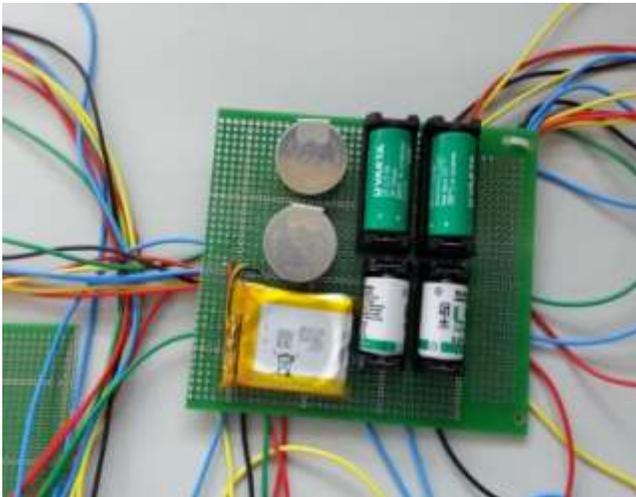


Figure 4.6: a) batteries listed in table 1 soldered to a PCB for vacuum tests b) The batteries connected to DAQ and feedthrough of the vacuum vessel

4.2.7 . Long-term drift of electronics

The last set of qualifying tests for the wireless temperature sensor prototype is the long-term drift tests of the rest of the electronics apart from the NTC.

The test was conducted room temperature. A reference temperature sensor was used to record the variation in the room temperature, so that any correlation of the results with room temperature variation can be disregarded later. To filter the variation in temperature, an aluminum block was used. Both the sensors are inserted in the holes inside the block, which are very close to each other, so that they measure the same temperature. The test setup is shown in Figure 4.7.

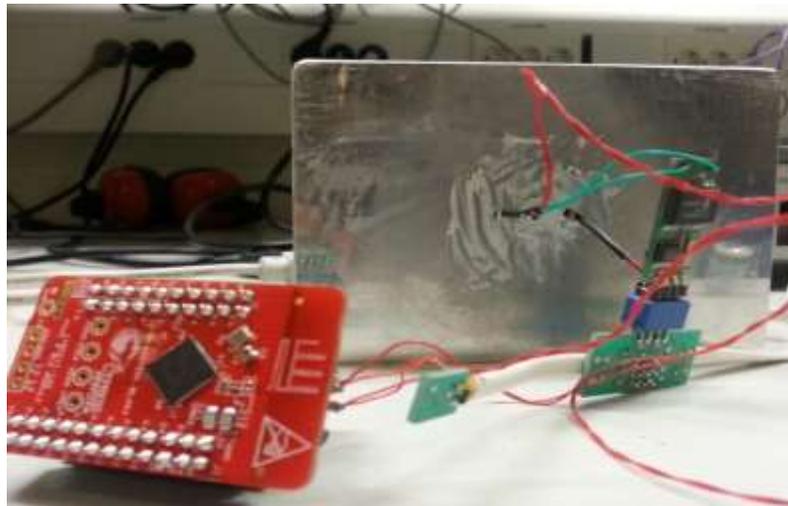


Figure 4.7: long-term drift test setup (wireless prototype, reference sensor and Al block)

4.2.8 Conclusion

In this section, a series of experiments conducted for the qualification of the wireless temperature sensor prototype were presented. Numbers of tests, including noise measurement, transfer characteristics, power consumption, power supply rejection ratio (PSRR), reliability of communication, life-time tests of batteries in vacuum and long-term drift of electronics were conducted. The results of these tests are listed in chapter 5.

4.3 BJT-based temperature sensor stability tests

MOS and BJT temperature sensors have catered to temperature sensing applications requiring ultra-low power consumption (nW range), very small areas (mm) and wide temperature ranges [22-25]. This is traded off with their resolutions being of the order of a few mKs. The long-term drift rates of such sensors are seldom measured. In [26], the drift of an npn transistor is reported to be 0.1K. An integrated temperature sensor from Smartec B.V, SMT 160-30 [27], which uses a bipolar npn transistor as the temperature sensing element is also reported to have a similar drift rate. Later, Fruett and Meijer [28] found that the main reason for the drift is package stress which will be 2-7 times less for CMOS devices using pnp-transistors. To produce an accurate band-gap

reference, dynamic element (DEM) matching was used. The accuracy was further improved by using other circuit techniques like chopping, auto-zeroing and auto calibration.

In this thesis, a new version of SMT 160-30 from Smartec technologies [29], which has improved performance parameters than its previous version, is measured for long-term drift performance. A comparison of the drift rates of a CMOS based sensor (Smartec) and the wireless sensor designed in this project is presented in chapter 5, section 5.3.

4.3.1 Drift measurement⁷

The setup used for the drift measurements presented in this section also uses a water bath as in section 4.1. The bjt based sensor uses eight current references, the average value of which is recorded as temperature data. The eight sensors are connected to the PCB using twisted pair wires (figure 4.8a). The output of the PCB is eight duty cycles, which is then fed to the MCU (microcontroller unit) by a flat cable (figure 4.8b). The averaging of eight cycles is done in the MCU to give digital bit-stream as the output. To begin with, only one of the eight sensors is inserted in the bath for long-term drift measurement.

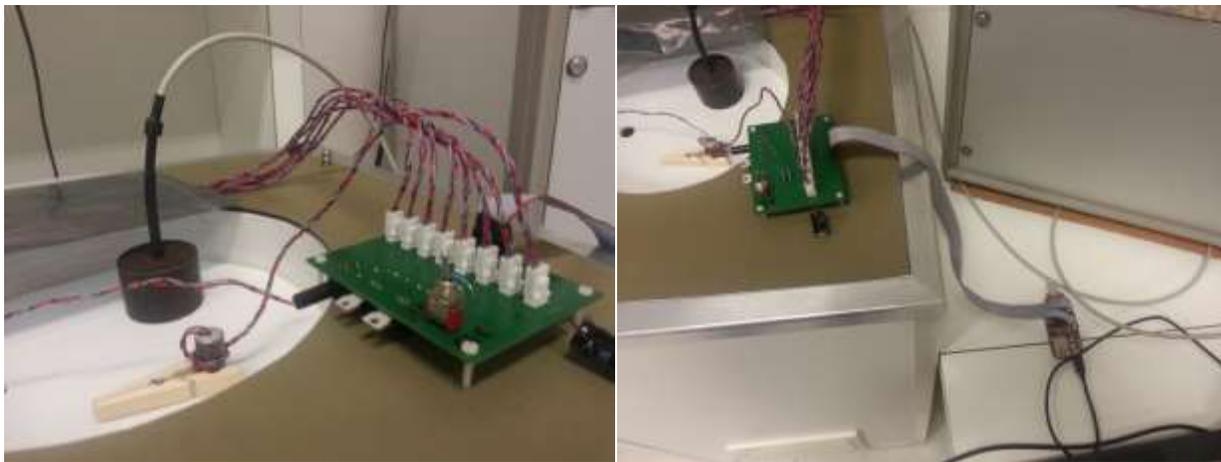


Figure 4.8 : a) The integrated temperature sensor (8 BJT sensors), with one of them inserted in the bath b) The flat wire connection between the pcb and MCU.

In the beginning of the experiment, a couple of problems were encountered, which were solved.

Problem 1: The measured raw data seemed too noisy

The original frequency of measurement is, $f = 27$ Hz. However, due to large amount of data, the data was filtered at 1Hz bandwidth and a moving average of 27 data points was taken. The noise measured was 0.007 K ($2\text{-}\sigma$), while the expected noise was ~ 2 mK ($1\text{-}\sigma$).

⁷ The measurements are conducted and data is recorded at TempControl B.V.

Solution 1: It was identified that the extra noise was from the power supply. When a decoupling capacitor was used on the 5V supply line, the noise reduced to ~ 4 mK ($2\text{-}\sigma$) = ~ 2 mK ($1\text{-}\sigma$), which is expected.

Problem 2: "Hand-effect"

The 8 sensors are connected to the MCU using flat cables. All the lines in the flat cable have constant capacitances between them. Since the capacitances are uniform, they do not appear as interferences during the course of measurement. However, when the wire of only one of the sensors (which is transmitting data) is touched, the capacitance of that particular line changes. This appears as a dip in the measurements. Apparently, it is a high-frequency effect called as "hand-effect".

Solution 2: Replace the flat cable with twisted cable and see if the effect still persists

After solving a few more trivial problems (reducing data recording interval), the experiment was started and is on-going. In chapter 5, section 5.3, results of the drift measurement (until present) is included.

4.4 Conclusion

In this chapter, all the experiments involved in qualifying the wireless temperature sensor design were presented. Next, an introduction to long-term drift experiments of a BJT-based sensor from Smartec was presented. Chapter 5 includes the results of all the tests explained in this chapter, along with a comparison of the long term drift performance of the BJT and the wireless temperature sensor.

References

- [1] Feteira, A. "Negative temperature coefficient resistance (NTCR) ceramic thermistors: an industrial perspective." *Journal of the American Ceramic Society* 92.5 (2009): 967-983.
- [2] Feteira, A, and K. Reichmann. "NTC ceramics: past, present and future." *Advances in Science and Technology* 67 (2011): 124-133.
- [3] Fritsch, S., et al. "Correlation between the structure, the microstructure and the electrical properties of nickel manganite negative temperature coefficient (NTC) thermistors." *Solid State Ionics* 109.3 (1998): 229-237.
- [4] Fang, D.L, et al. "Aging of nickel manganite NTC ceramics." *Journal of Electroceramics* 22.4 (2009): 421-427.
- [5] Hosseini, M. "The effect of cation composition on the electrical properties and aging of Mn-Co-Ni thermistors." *Ceramics International* 26.3 (2000): 245-249.
- [6] Castelan, P., et al. "Aging study of nickel-copper-manganite negative temperature coefficient thermistors by thermopower measurements." *Journal of Applied Physics* 72.10 (1992): 4705-4709.

- [7] Varghese, J. M., Seema A., and K. R. Dayas. "Ni-Mn-Fe-Cr-O negative temperature coefficient thermistor compositions: Correlation between processing conditions and electrical characteristics." *Journal of Electroceramics* 22.4 (2009): 436-441.
- [8] Groen, W. A., et al. "Aging of NTC ceramics in the system Mn-Ni-Fe-O." *Journal of Electroceramics* 7.2 (2001): 77-87.
- [9] Lawton, K. M., and S. R. Patterson. "Long-term relative stability of thermistors." *Precision Engineering* 25.1 (2001): 24-28.
- [10] Lawton, K. M., and S. R. Patterson. "Long-term relative stability of thermistors: Part 2." *Precision Engineering* 26.3 (2002): 340-345.
- [11] Wood, S. D., et al. "An investigation of the stability of thermistors." *Journal of Research of the National Bureau of Standards* 83.3 (1978): 247-263..
- [12] Edwards, T. J. "Observations on the stability of thermistors." *Review of Scientific Instruments* 54.5 (1983): 613-617
- [13] Siwek, W. R., et al. "Stability of NTC thermistors." *Temperature, Its Measurement and Control in Science and Industry* 6.Part 1 (1992): 497-502.
- [14] Measurement Specialties, Application Note TD011.
- [15] Automatic Systems Laboratories, " F700 Thermometry Bridge ", F700 Datasheet.
- [16] Vishay Precision Group, " Ultra high precision Z-foil resistor with TCR of ± 0.05 ppm/ $^{\circ}$ C, PCR of 5 ppm at rated power, tolerance of ± 0.005 % and load life stability of ± 0.005 % ", Z201 Datasheet, March 2010.
- [17] Keithley, " Low noise 7½ digit autoranging multimeter ", 2010 Datasheet
- [18] Hill, K. D. "Is there a long-term drift in triple point of water cells?." *Metrologia* 38.1 (2001): 79.
- [19] Yan, X. K., et al. "The long-term drift of triple-point-of-water cells." *International Journal of Thermophysics* 29.3 (2008): 815-824.
- [20] Preston-Thomas, H. "The international temperature scale of 1990(ITS-90)." *Metrologia* 27.1 (1990): 3-10.
- [21] Lavenuta, G. "Negative temperature coefficient thermistors." *Sensors-the Journal of Applied Sensing Technology* 14.5 (1997): 46-55.
- [22] R. A. Blauschild, P. A. Tucci, R. S. Muller, and R. G. Meyer, "A new NMOS temperature-stable voltage reference," *IEEE J. Solid-State Circuits*, vol. SC-13, pp. 767-774, Dec. 1978.
- [23] H. J. Song and C. K. Kim, "A temperature-stabilized SOI voltage reference based on threshold voltage difference between enhancement and depletion NMOSFET's," *IEEE J. Solid-State Circuits*, vol. SC-28, pp. 671-677, June 1993.
- [24] Y. P. Tsvids and R. W. Ulmer, "A CMOS voltage reference," *IEEE J. Solid-State Circuits*, vol. SC-13, pp. 774-778, Dec. 1978.
- [25] M. Tuthill, "A switched-current, switched-capacitor temperature sensor in 0.6- μ m CMOS," *IEEE J. Solid-State Circuits*, vol. SC-33, pp. 1117-1122, July 1998.
- [26] G. C. M. Meijer, "Thermal sensors based on transistors," *Sens. Actuators A*, vol. 10, pp. 103-125, 1986.

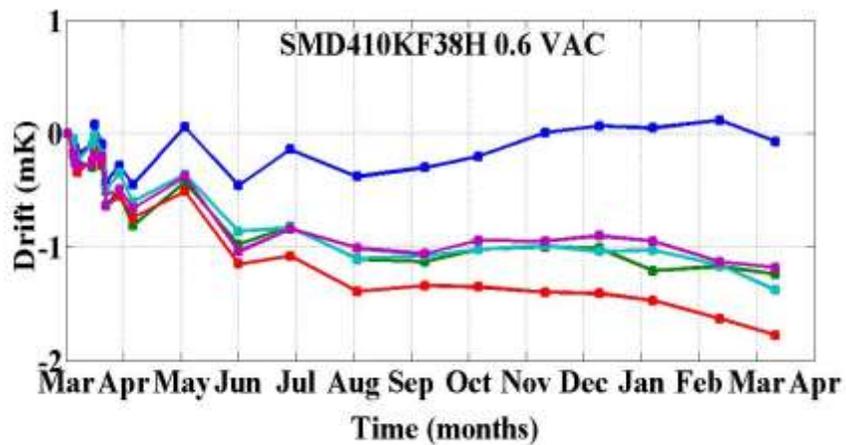
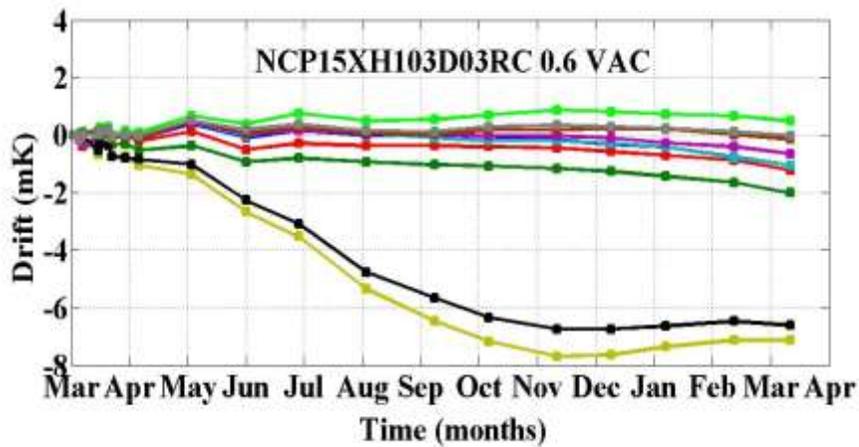
- [27] J. F. Creemer, F. Fruett, G. C. M. Meijer, and P. J. French, "Piezjunction effect in silicon sensors and circuits and its relation to piezoresistance," *IEEE Sensors J.*, vol. 1, pp. 98–108, Aug. 2001.
- [28] F. Fruett and G. C. M. Meijer, "The influence of the piezo-junction effect at different temperatures on the accuracy of silicon temperature sensors," in *Proc. ICMP/2000*, Manaus, Brazil, Sept. 2000.
- [29] Heidary, Ali, et al. "12.8 A BJT-based CMOS temperature sensor with a 3.6 pJ·K²-resolution FoM." *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014 IEEE International. IEEE, 2014.

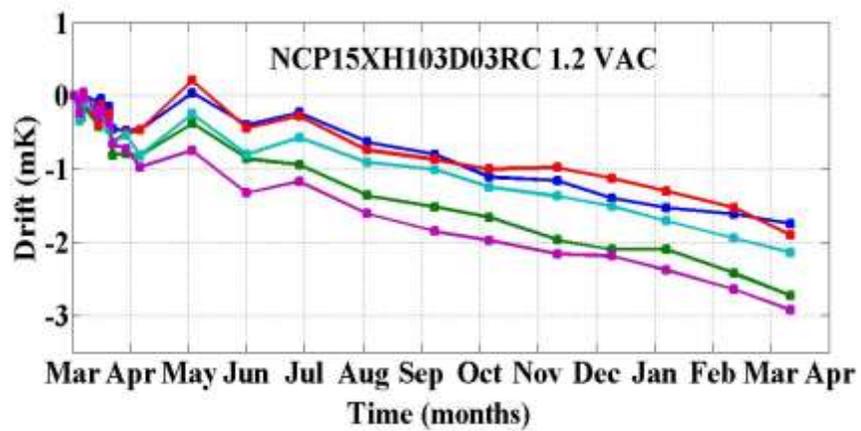
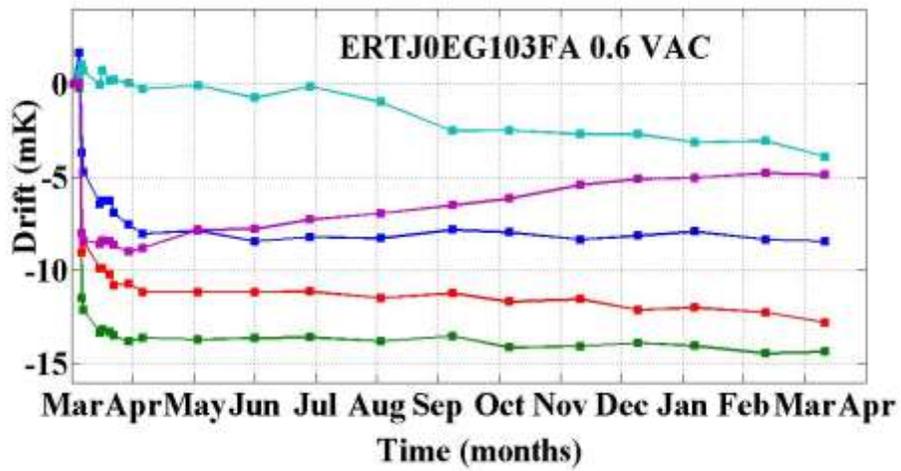
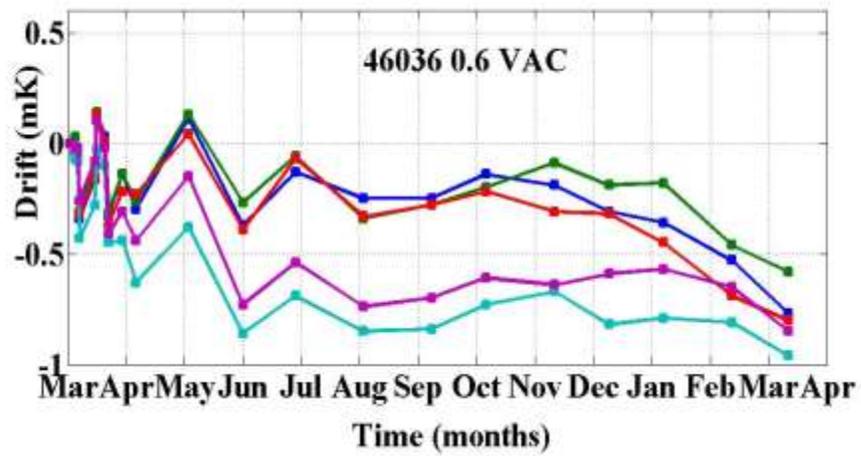
Chapter 5: Experimental results

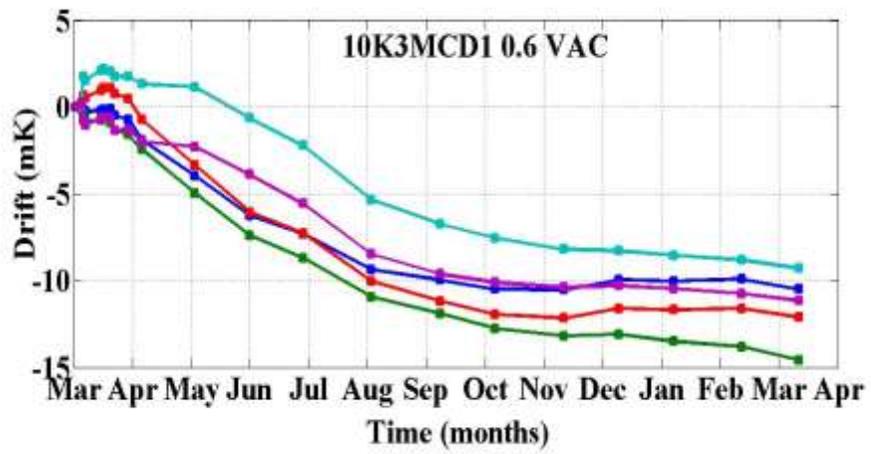
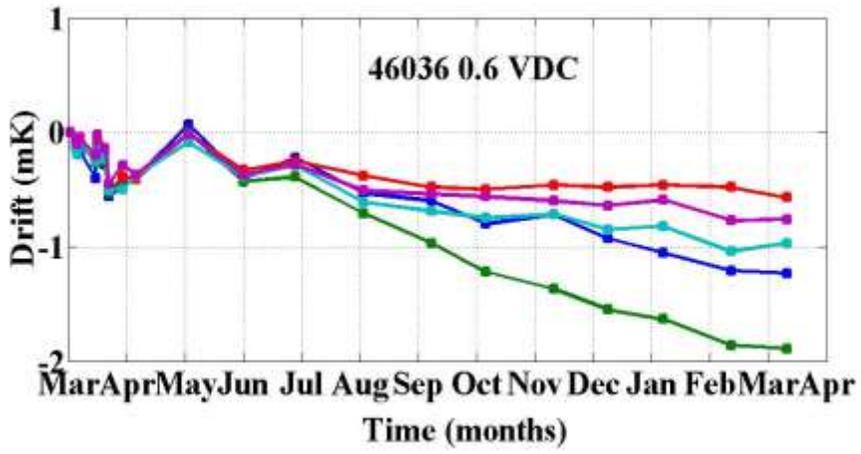
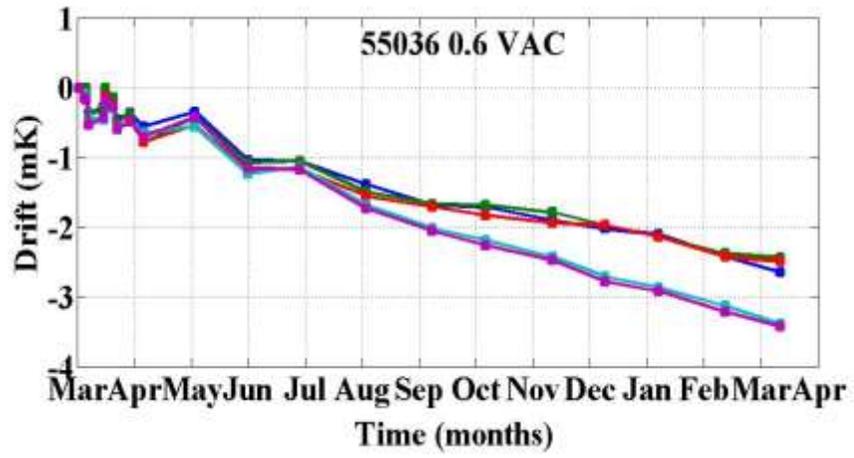
This chapter includes the results of experiments explained in chapter 4. Firstly, in section 5.1, the measurement results of the long-term drift of NTC temperature sensors are presented, along with results after error minimization. Section 5.2 includes results of the wireless temperature sensor qualifying tests. Finally, the long-term drift results of BJT-based Smartec sensor are presented in section 5.3

5.1 Long-term drift of NTC temperature sensors

Figure 5.1 (a-i) shows the measurement results of the long-term drift tests of NTC temperature sensors. Many sensors have the drift performance of the order of 1 mK or more, which is same as the uncertainty of the measurement setup.







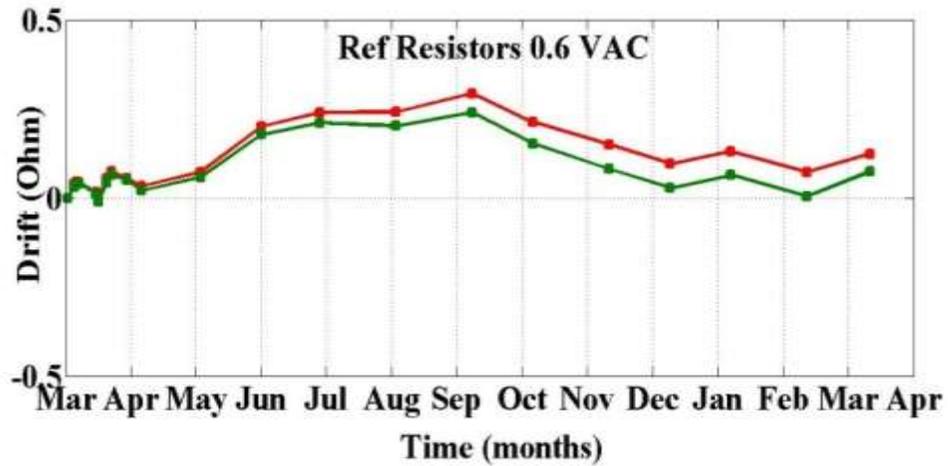


Figure 5.1 (a) – (i) Drift measurement results of various NTCs and fixed resistors, mentioned in Table 1.

As mentioned in chapter 4, the presence of deterministic error was observed and error minimization techniques were applied (figures 5.2 – 5.5).

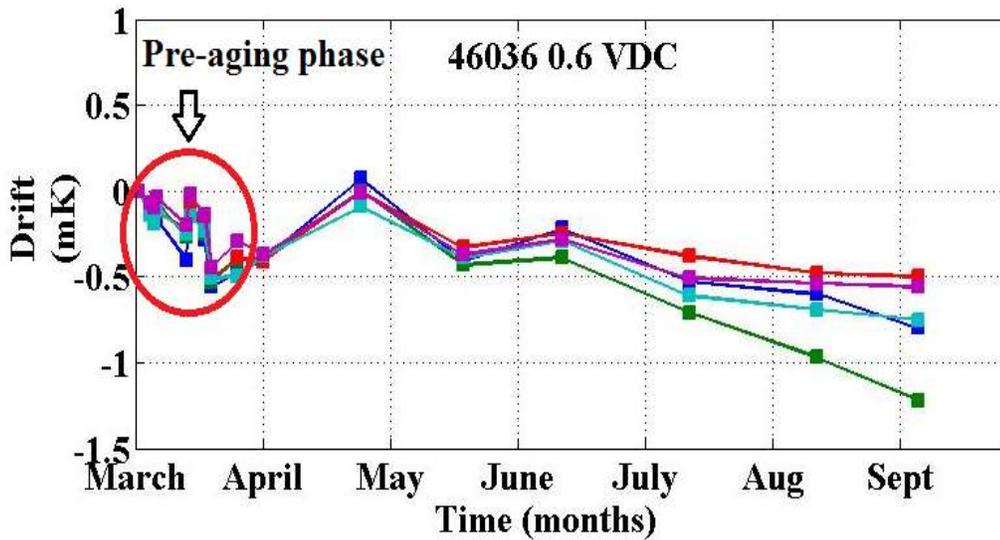


Fig. 5.2. Example case showing the pre-aging phase

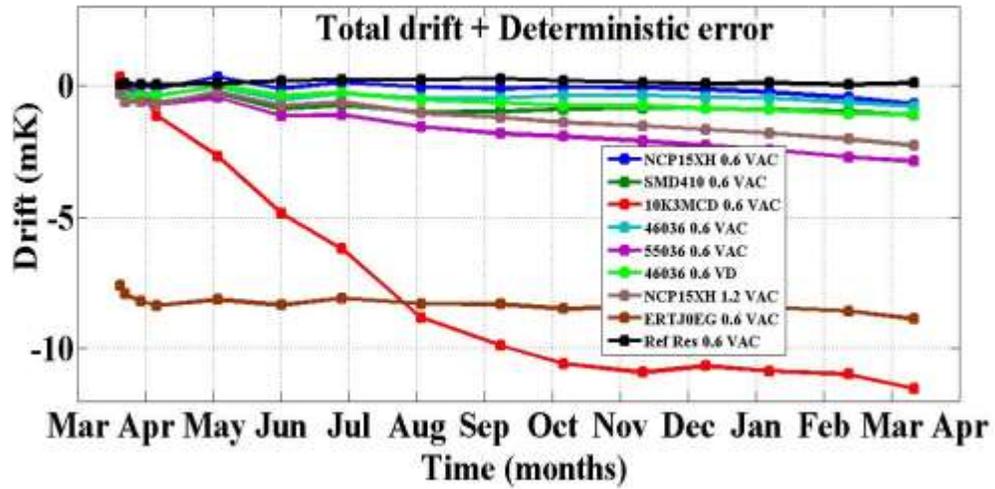


Fig 5.3. Graph representing the average of each set of NTC

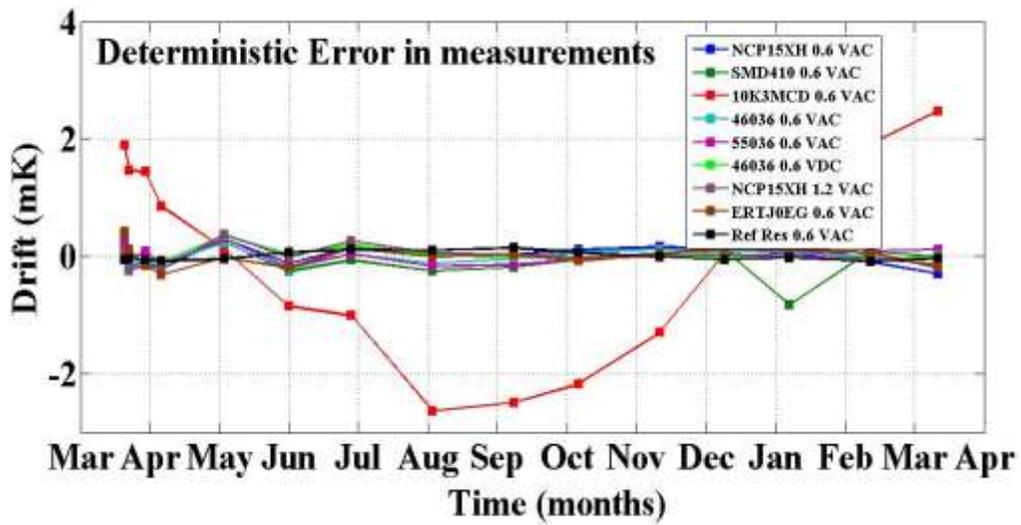


Fig 5.4. Deterministic error in each set of NTCs introduced during measurements

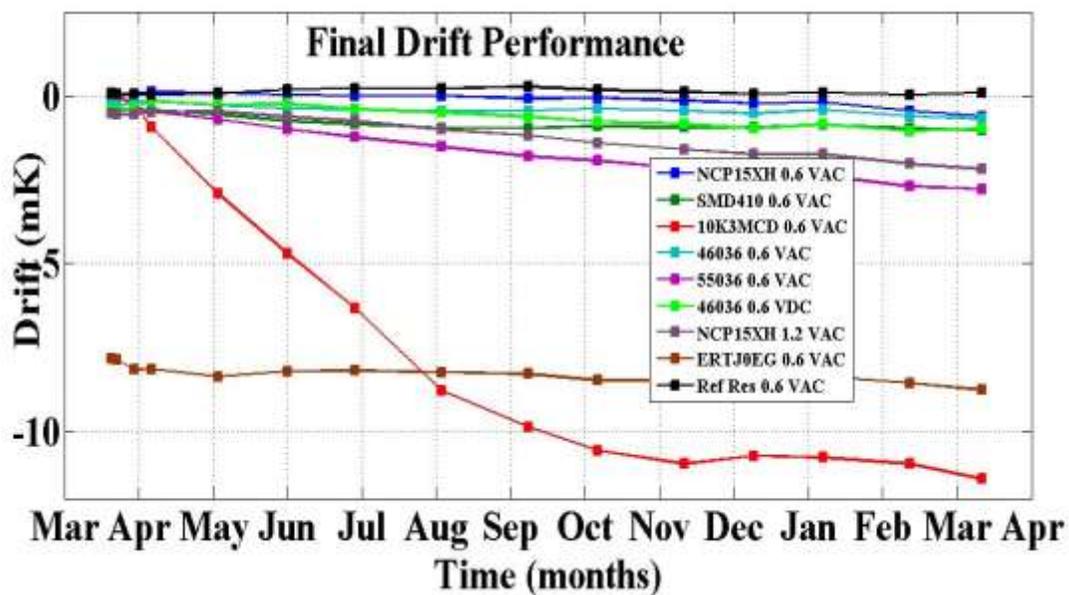


Fig. 5.5. Final drift results after the removal of deterministic error

5.2 Wireless sensor prototype qualifying tests

In this section, the results of the qualification tests for wireless temperature sensor explained in chapter 4, section 4.2 are presented.

5.2.1 Noise measurements

Figure 5.6 and Figure 5.7 show the noise performance of full and half bridge configurations, respectively. For full bridge, the noise measured is in μV , since the bridge is balanced. For the half bridge, we see an offset of about 0.4V due to the ratio metric measurement.

If the first few samples are discarded, the noise in the full bridge is about 2 μV pp. The calculated standard deviation of the noise is 0.659 μV . For the half bridge, we observe some spikes in the noise measurements. If we ignore these spikes, then the noise is about 5 μV peak to peak and the standard deviation is 2.68 μV . For the full bridge, these spikes are less noticeable. The exact cause for the spikes is not known at present and needs further investigation.

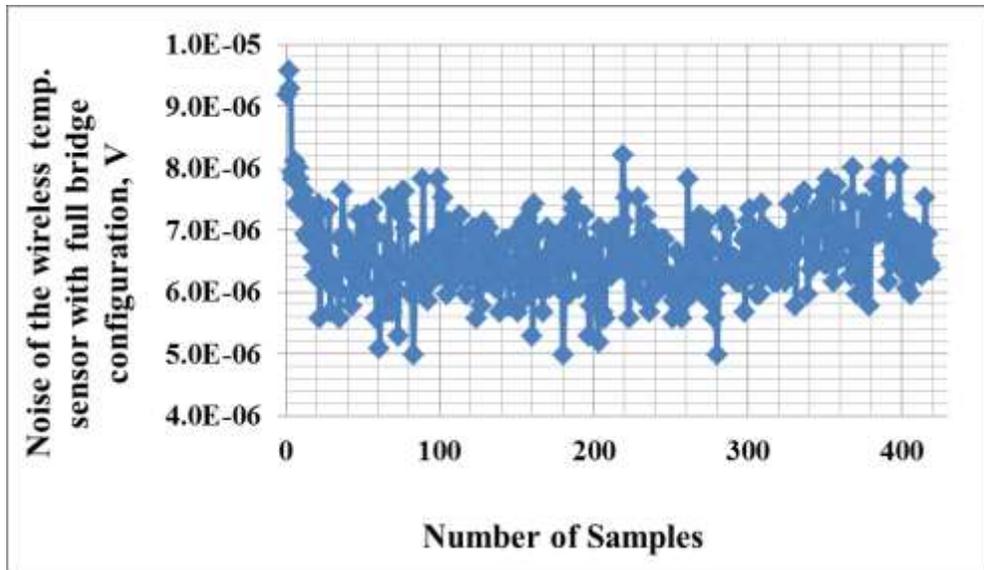


Fig 5.6. Measured noise performance of full bridge configuration

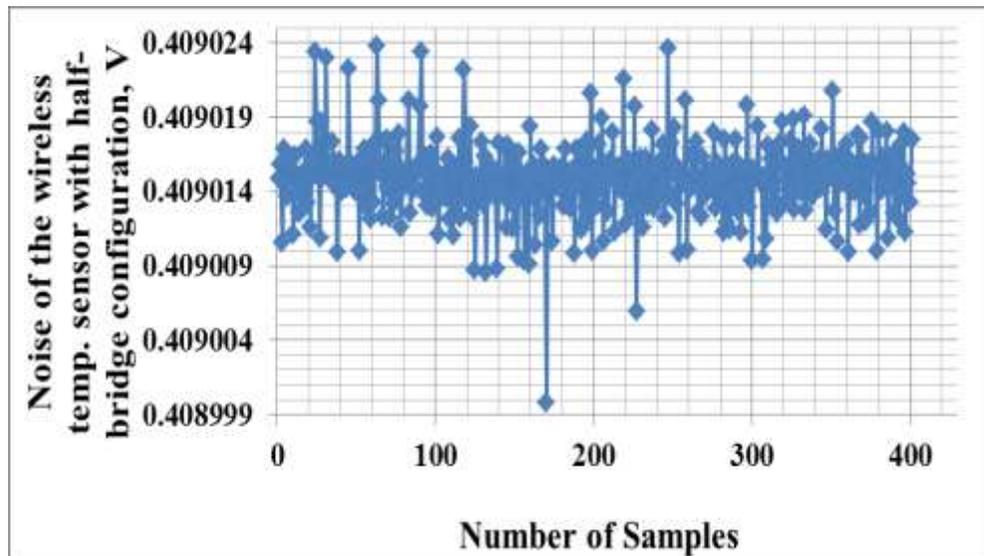


Fig 5.7. Noise performance of half bridge configuration

5.2.2 Transfer characteristics

This experiment was done for both full and half bridge configurations and the results are plotted in Figure 5.8 and Figure 5.9, respectively. Since the ADC is not considered to exhibit non-linear characteristics, the transfer characteristics of the sensors depend on the NTC sensor. For a full bridge configuration, the output of the NTC is linearized and hence we see a linear characteristic. The bridge is perfectly balanced at 22 °C, at which the output voltage is zero. For the half bridge the transfer characteristic is found to be much more non-linear due to the non-linearity of the NTC. No hysteresis is observed for both half and full bridge.

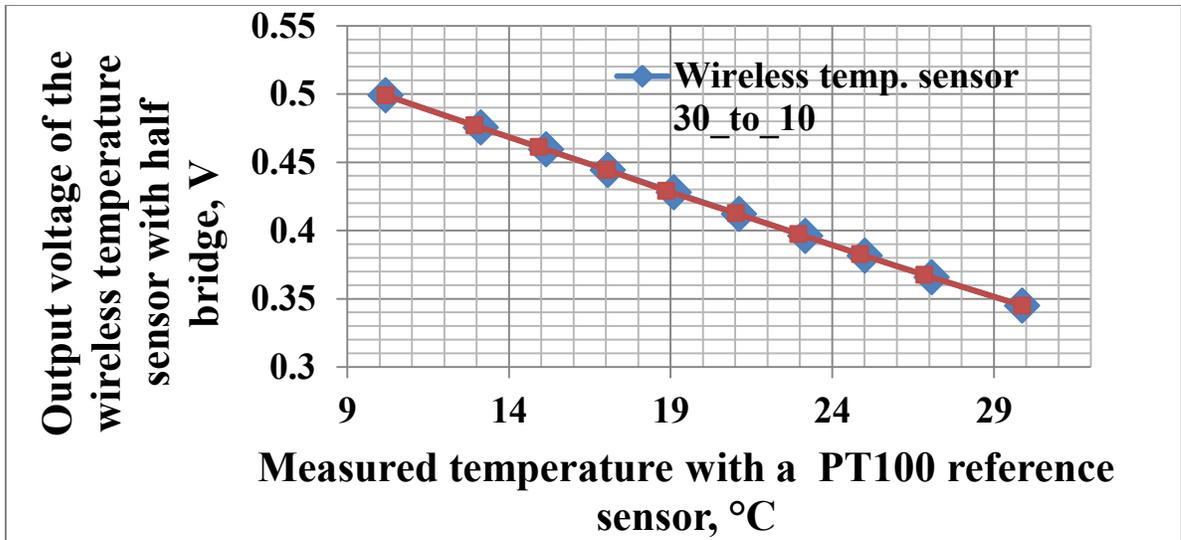


Figure 5.8 Transfer characteristics of wireless sensor with half bridge

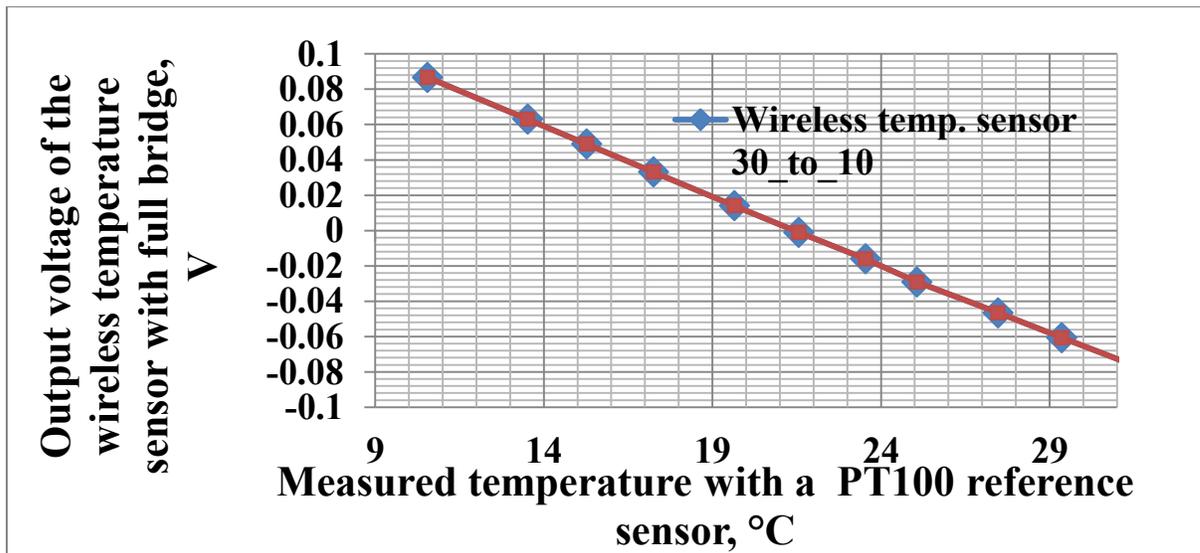


Figure 5.9 Transfer characteristics of wireless sensor with full bridge

The transfer characteristics for full and half bridge are apparently linearly. To calculate the hidden non-linearity, the ideal transfer characteristic was calculated for the temperature range of 10 – 30 °C and subtracted from the measured transfer characteristics. Figure 5.10 and 5.11 represent non-linearity performance of full and half bridge configurations respectively. The spikes are considered as outliers due to measurement errors. The maximum inaccuracy observed in both the graphs is ~ ±1 °C.

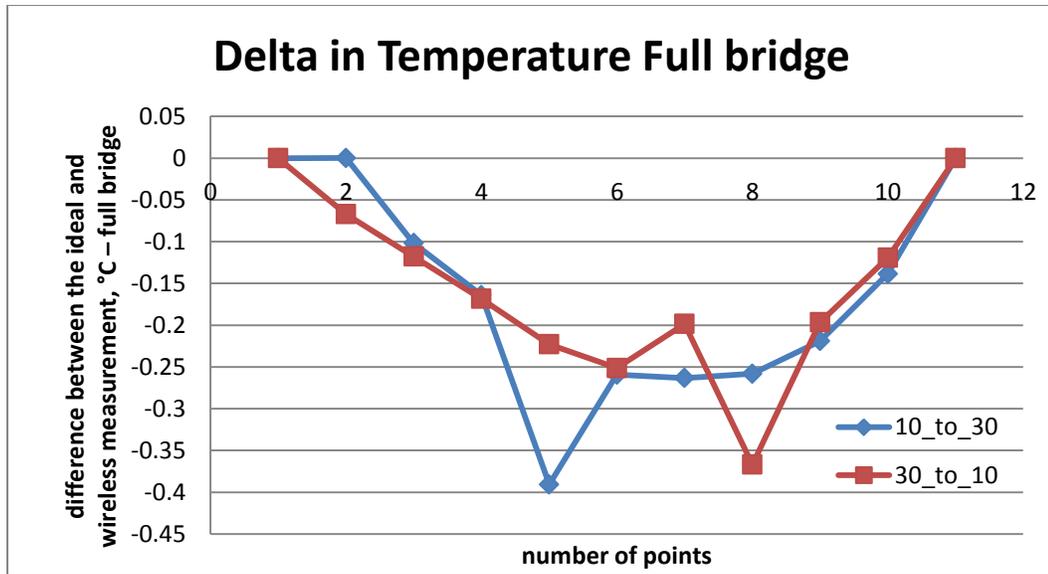


Figure 5.10: Difference between the ideal and measured transfer characteristics for full bridge

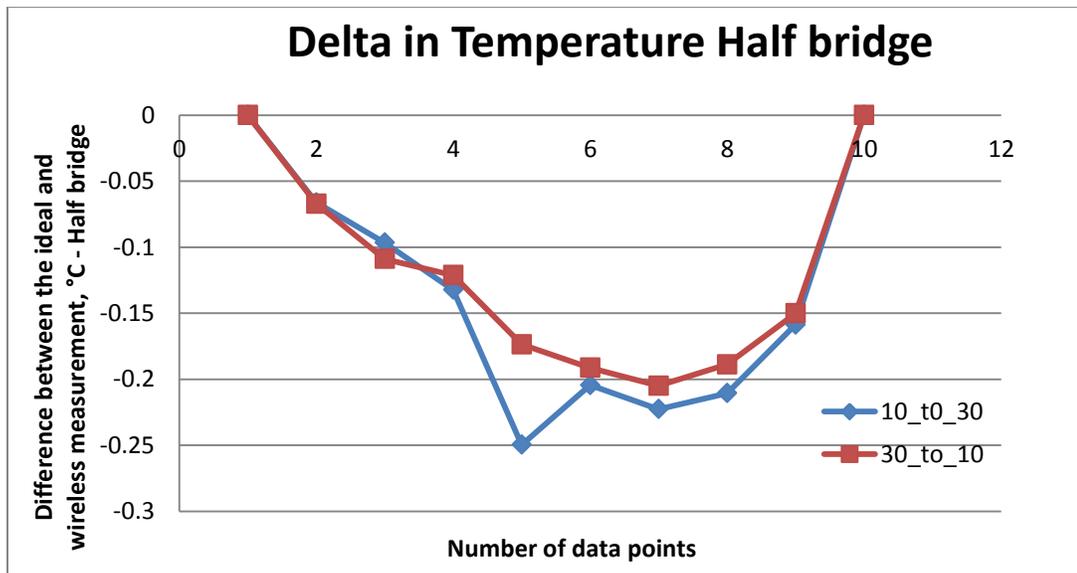


Figure 5.11: Difference between the ideal and measured transfer characteristics for half bridge

5.2.3 Power consumption

The power consumption of the Bridge, ADC and the BLE module together is ~0.850 mW for a half bridge. For a full bridge, the current in the bridge is twice that of the half bridge. Hence, the power consumption is higher, which equals ~0.980 mW. However, for both the configurations, the power consumption is well below the specification of 1 mW.

5.2.4 Power supply rejection ratio (PSRR)

Figures 5.12 and 5.13 show the PSRR performance of full and half bridge, respectively, for supply voltage variation from 3 to 3.6 V. Since the full bridge measures the difference of the two arms, the effect of variation of supply voltage is almost completely canceled out. What is seen in Fig. 5.12 is mostly the equivalent input noise. However, the half bridge shows certain supply voltage dependence equivalent to ~ 0.5 mK. It is worth mentioning that the expected battery voltage variation will be less than 0.6V.

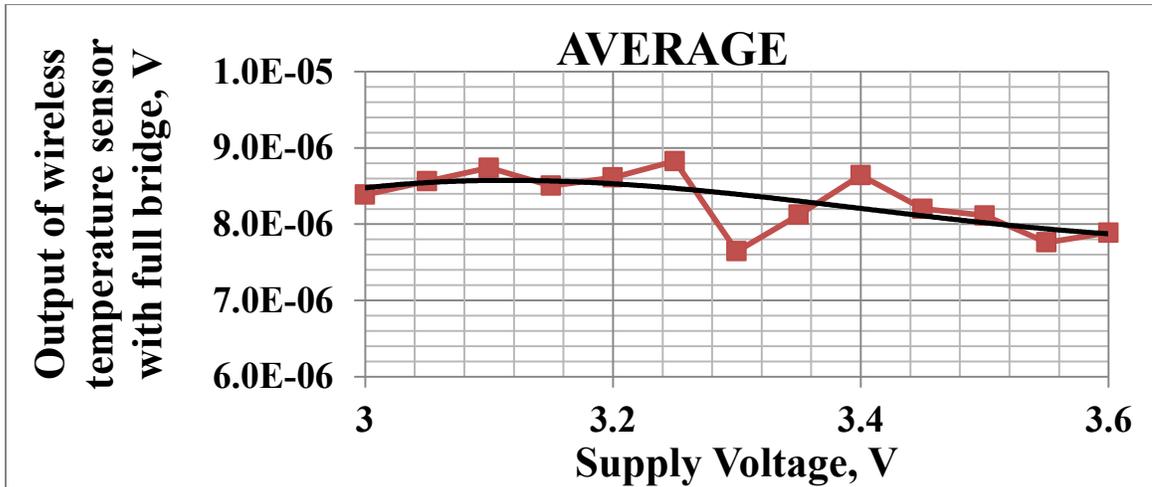


Figure 5.12. Behavior of full bridge configuration with respect to variations in the power supply

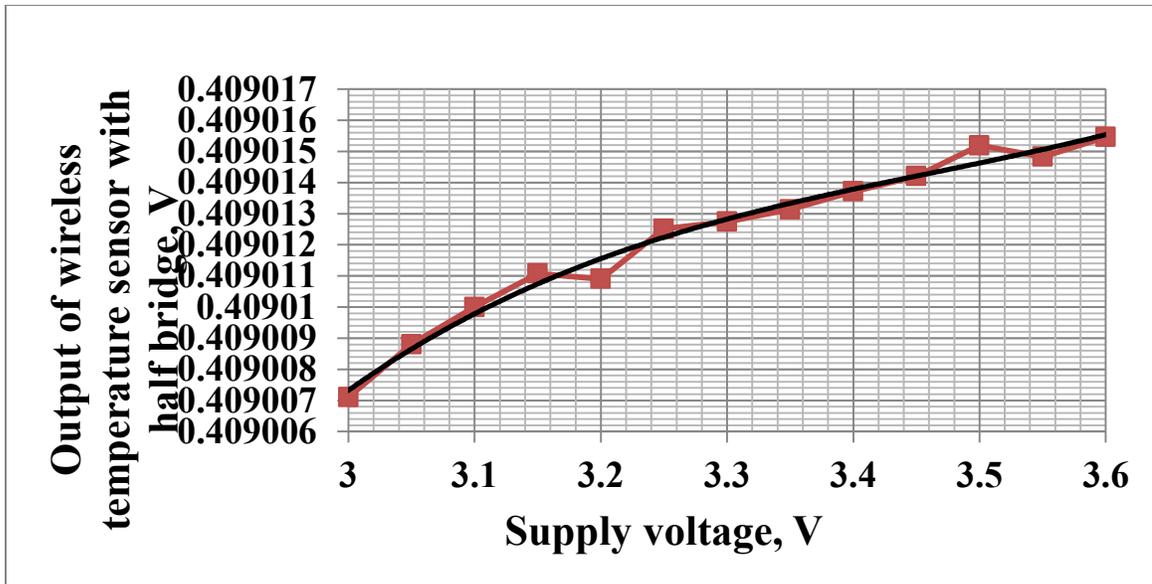


Figure 5.13. Behavior of half bridge configuration with respect to variations in the power supply

5.2.5 Reliability of communication

The results obtained are shown in Figure 5.13. Packets lost during communication from the closed vessel and in open environment are plotted. The packet loss in closed vessel is found to be much higher than for open vessel after a distance of 1m.

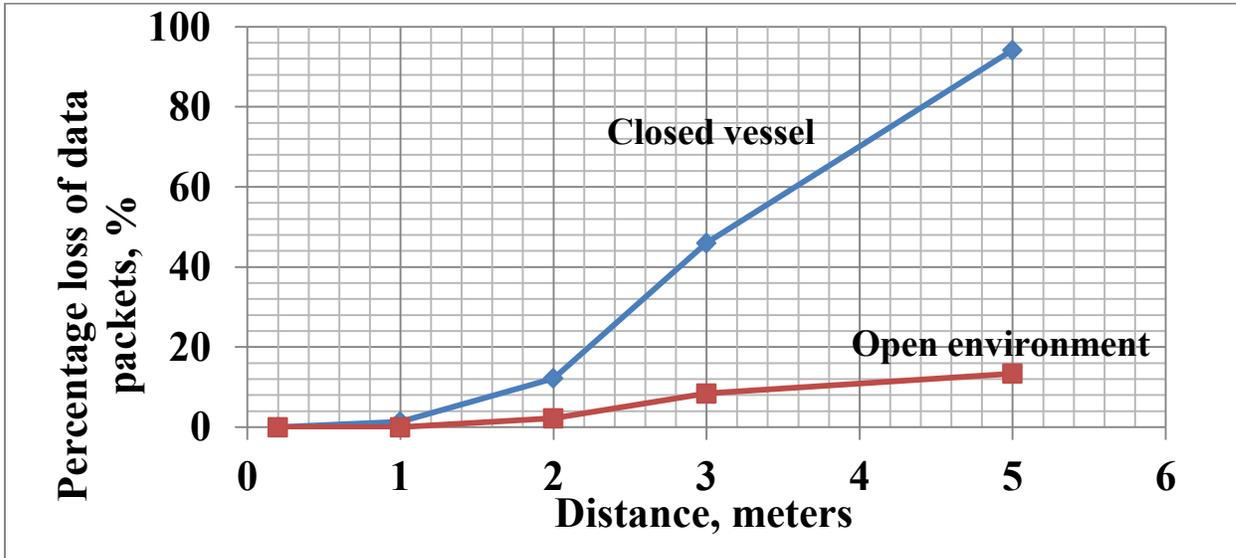


Figure 5.14 Distance vs. percentage loss of packets

5.2.6 Life-time tests of batteries in vacuum

The operating voltages of two LiSOCl_2 batteries were recorded for the duration of three months (Figure 5.14). Curves (1) and (2) represent the two samples.

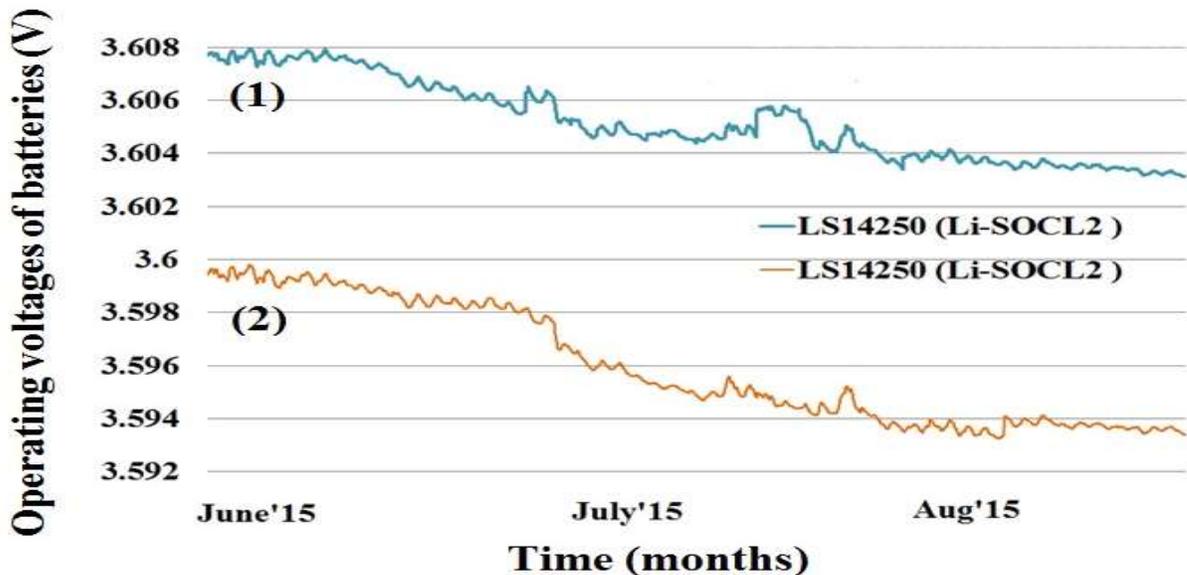


Figure 5.15: Life time of batteries in vacuum test results for the list of batteries in Table 4.2

Note that LiSOC12 battery was chosen for our application mainly due to its excellent supply voltage stability over time. For a period of three months, the discharge of the operating voltage in vacuum for each sample is found to be $\sim 5 - 6$ mV.

5.2.7 Long-term drift of electronics

The drift of the electronics of the wireless temperature sensor was measured for duration of around 1 month. The results show that the drift was found to be below the resolution of the sensor (Figure 5.16). The flat line is the drift of the wireless sensor, which is basically noise. For each value plotted, an average of 10 samples was taken and the drift was found to be <0.1 mK ($3\text{-}\sigma$) / 1 month. The red line in the graph is the variation of the reference sensor, PT100. Note, to measure the drift performance of the electronics, four resistors were used in the full bridge of the wireless sensor prototype. No correlation is observed between the room temperature and the drift of the electronics. The resistors are extremely stable with low TCR.

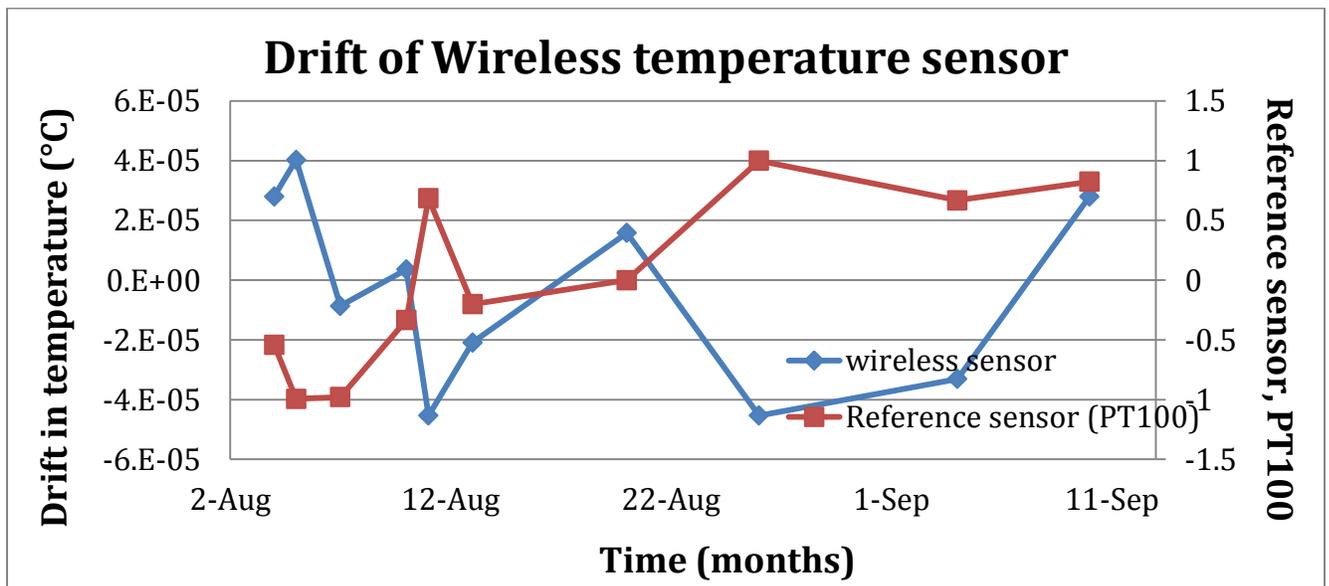


Figure 5.16: Drift results of electronics part of the wireless temperature sensor prototype

Hence the total drift of the wireless sensor can be calculated as the sum of drift of the sensor (0.492 mK/yr) and the drift of electronics (~ 0.1 mK), which corresponds to $\sim <0.6$ mK.

5.3 BJT-based temperature sensor stability tests

Figure 5.17 shows the time drift results of the BJT temperature sensor. The average drift value is found to be about ~ 2 mK p-p/100 days for each of the sensors, which is higher than the drift of the wireless temperature sensor designed in this project. For <1 mk long-term drift specifications defined for this prototype design, wireless temperature sensor made from off the shelf components is found to perform better than the BJT sensor.

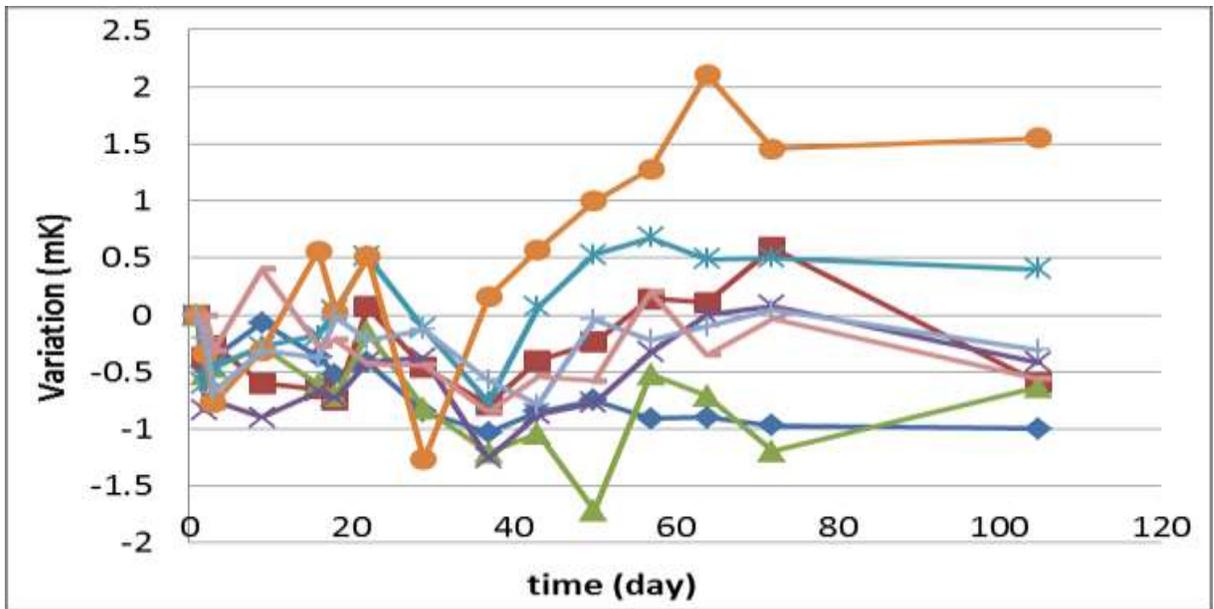


Figure 5.17: Drift results of BJT-based temperature sensor

Chapter 6: Conclusion

Section 6.1 clearly describes my contributions to the project. In the section 6.2, a summary of the thesis is presented along with emphasis on the most important results. Finally, section 6.3 includes some recommendations for the future work of the wireless temperature sensor.

6.1 Contributions

The work done in this thesis primarily included two major parts: Long-term drift of NTC temperature sensors and design and qualification of the wireless temperature sensor prototype. The sensor was developed to suffice the requirements from ASML for a low power, high resolution, small foot print and stable wireless temperature sensor for diagnostic purposes. My supervisors directed me to the idea. My major contributions were:

1. Long-term drift of NTC temperature sensors: Since long-term drift was an important parameter in the design of the sensor, experiments were conducted to deduce the drift of a few state-of-the-art NTC temperature sensors. The best performing sensor among them was used in the prototype.
 - a. Understanding the long-term tests results of NTC temperature sensors and relating their causes to physical phenomena (migration of cat-ions in NTCs, sintering temperatures, oxidation, stress, etc.).
 - b. Proposed and implemented error minimization techniques to reduce the deterministic error in the drift measurements, which improved the results.
2. Design and qualification of wireless temperature sensor
 - a. Proposed the design of the analog front-end circuitry. Conducted a few initial simulations for the design using LTSpice, followed by selection of components for the design.
 - b. Conducted all the experimental tests listed in chapter 4, for the qualification of the wireless sensor. This involved setting up the experiments, carrying them out and finally data-processing of results. I used MatLab and Excel for the post processing of the results.

On each of these topics I have scripted one IEEE conference paper each and both the papers have been accepted for publication [1-2].

6.2 Summary

- A wireless temperature sensor that can be used for diagnostic purposes in advanced industrial applications like ASML machines was developed and qualified.
- Both half and full bridge configurations were tested for performance.
- The half bridge was found to be more power efficient than the full bridge, but it was traded with worse noise performance and low PSRR.
- The achieved performance for both the configurations is:
 - Power < 1 mW; Resolution 150 uK (3-sigma).
 - Accuracy: 0.1 °C
 - Area = 30 mm x 20 mm x 20 mm
- Long-term drift of NTC temperature sensor + electronics: <0.6 mK
- For the long-term drift of NTC temperature sensors, the error minimization techniques were employed as mentioned earlier. Table 6.1 shows the drift rates of various sensors along with the drift rates before and after error minimization⁸.

Table 6.1: Long-term drift results of NTC temperature sensors, with drift rates before and after error minimization

	Amt	Name	Operating conditions	Drift rates (mK/year pk-to-pk)	After error minimization
1	10	NCP15XH103D03RC	0.6V AC	1.0425	0.492
2	5	SMD410KF38H	0.6V AC	0.93	0.638
3	5	46036	0.6V AC	0.776	0.546
4	5	ERTJ0EG103FA	0.6V AC	1.258	0.899
5	5	46036	0.6V DC	1.076	0.9
6	5	NCP15XH103D03RC	1.2V AC	2.064	1.704
7	5	55036	0.6V AC	2.658	2.388
8	5	10K3MCD1	0.6V AC	11.836	11.307
9	2	Ref resistors	0.6V AC	0.24	0.24 Ω/Ω/year

⁸ The best and worst drift rates are highlighted in red color.

6.3 Future work

- Integrate the wireless part in the PCB
- Design of a suitable mechanical housing so that the sensor is easily mountable (on-going)
- Dwell deep into the BLE protocol to understand/solve few issues regarding the connection between the receiver and transmitter.
- Create more number of sensor nodes, so that a network of sensor can be created to monitor different parts.

Appendix I

This section includes the main program used for the BLE communication of the wireless temperature sensor prototype

```
/*
 * File Name: main.c
 *
 * Version: 1.0
 *
 * Description:
 * Temperature measurement example project that measures using AD7789
 * and sends it over BLE Health Thermometer service.
 *
 * Note:
 *
 * Hardware Dependency:
 * CY8CKIT-042 BLE
 *
 * Copyright 2015, Cypress Semiconductor Corporation. All rights reserved.
 * You may use this file only in accordance with the license, terms, conditions,
 * disclaimers, and limitations in the end user license agreement accompanying
 * the software package with which this file was provided.
 */

#include "common.h"
#include "hts.h"
#include "bas.h"

uint8 busStatus = CYBLE_STACK_STATE_FREE;    /* Status of stack queue */

volatile uint32 mainTimer = 0;
volatile uint8 test[4];
volatile uint32 TemperatureBuffer[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
volatile int TemperatureBufferPointer = 0;

extern uint32 temperatureValue;

/*
 * Function Name: AppCallBack()
 *
 * Summary:
 * This is an event callback function to receive events from the BLE Component.
 *
 * Parameters:
 * event - the event code
 * *eventParam - the event parameters
 */

void AppCallBack(uint32 event, void* eventParam)
{
```

```

// CYBLE_API_RESULT_T apiResult;
CYBLE_GAP_BD_ADDR_T localAddr;

// uint8 i;

switch (event)
{
/*****
*       General Events
*****/
    case CYBLE_EVT_STACK_ON: /* This event received when component is Started */
        /* Enter in to discoverable mode so that remote can search it. */
//        apiResult =
        CyBle_GappStartAdvertisement(CYBLE_ADVERTISING_FAST);

/*        if(apiResult != CYBLE_ERROR_OK)
        {
            printf("StartAdvertisement API Error: %d \r\n", apiResult);
        }
        printf("Bluetooth On, StartAdvertisement with addr: ");

*/
        CyBle_GetDeviceAddress(&localAddr);
/*        for(i = CYBLE_GAP_BD_ADDR_SIZE; i > 0u; i--)
        {
            printf("%02.2x", localAddr.bdAddr[i-1]);
        }
        printf("\r\n");

*/
        break;
    case CYBLE_EVT_TIMEOUT:
        /* CYBLE_GAP_ADV_MODE_TO - Advertisement time set by application is expired */
        /* CYBLE_GAP_AUTH_TO - Authentication procedure timeout */
        /* CYBLE_GAP_SCAN_TO - Scan time set by application is expired */
        /* CYBLE_GATT_RSP_TO - GATT procedure timeout */
//        printf("CYBLE_EVT_TIMEOUT: %x \r\n", *(uint8 *)eventParam);
        break;
    case CYBLE_EVT_HARDWARE_ERROR: /* This event indicates that some internal HW error has
occurred. */
//        printf("Hardware Error \r\n");
        break;
    case CYBLE_EVT_HCI_STATUS:
//        printf("EVT_HCI_STATUS: %x \r\n", *(uint8 *)eventParam);
        break;

/*****
*       GAP Events
*****/
    case CYBLE_EVT_GAP_AUTH_REQ:
/*        printf("EVT_AUTH_REQ: security=%x, bonding=%x, ekeySize=%x, err=%x \r\n",
        (*(CYBLE_GAP_AUTH_INFO_T *)eventParam).security,
        (*(CYBLE_GAP_AUTH_INFO_T *)eventParam).bonding,
        (*(CYBLE_GAP_AUTH_INFO_T *)eventParam).ekeySize,
        (*(CYBLE_GAP_AUTH_INFO_T *)eventParam).authErr);

*/
        break;
}

```

```

    case CYBLE_EVT_GAP_PASSKEY_ENTRY_REQUEST:
//     printf("EVT_PASSKEY_ENTRY_REQUEST \r\n");
        break;
    case CYBLE_EVT_GAP_PASSKEY_DISPLAY_REQUEST:
//     printf("EVT_PASSKEY_DISPLAY_REQUEST %6.6ld \r\n", *(uint32 *)eventParam);
        break;
    case CYBLE_EVT_GAP_AUTH_COMPLETE:
/*     authInfo = (CYBLE_GAP_AUTH_INFO_T *)eventParam;
        printf("AUTH_COMPLETE: security:%x, bonding:%x, ekeySize:%x, authErr %x \r\n",
            authInfo->security, authInfo->bonding, authInfo->ekeySize, authInfo->authErr);
*/
        break;
    case CYBLE_EVT_GAP_AUTH_FAILED:
//     printf("EVT_AUTH_FAILED: %x \r\n", *(uint8 *)eventParam);
        break;
    case CYBLE_EVT_GAPP_ADVERTISEMENT_START_STOP:
//     printf("CYBLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: %x\r\n", CyBle_GetState());
    if(CYBLE_STATE_DISCONNECTED == CyBle_GetState())
    {
        /* Fast and slow advertising period complete, go to low power
        * mode (Hibernate mode) and wait for an external
        * user event to wake up the device again */
/*     printf("Hibernate \r\n");
        Advertising_LED_Write(LED_OFF);
        Disconnect_LED_Write(LED_ON);
        LowPower_LED_Write(LED_OFF);
        while((UART_DEB_SpiUartGetTxBufferSize() + UART_DEB_GET_TX_FIFO_SR_VALID) != 0);
        SW2_ClearInterrupt();
        Wakeup_Interrupt_ClearPending();
        Wakeup_Interrupt_Start();
        */
        CySysPmHibernate();
    }
        break;
    case CYBLE_EVT_GAP_DEVICE_CONNECTED:
//     printf("CYBLE_EVT_GAP_DEVICE_CONNECTED: %x \r\n", *(uint8 *)eventParam);
//     Advertising_LED_Write(LED_OFF);
        break;
    case CYBLE_EVT_GAP_DEVICE_DISCONNECTED:
//     printf("CYBLE_EVT_GAP_DEVICE_DISCONNECTED\r\n");
        /* Put the device to discoverable mode so that remote can search it. */
//     apiResult =
        CyBle_GappStartAdvertisement(CYBLE_ADVERTISING_FAST);
/*     if(apiResult != CYBLE_ERROR_OK)
        {
            printf("StartAdvertisement API Error: %x \r\n", apiResult);
        }
        */
        break;
    case CYBLE_EVT_GAP_ENCRYPT_CHANGE:
//     printf("ENCRYPT_CHANGE: %x \r\n", *(uint8 *)eventParam);
        break;
    case CYBLE_EVT_GAPC_CONNECTION_UPDATE_COMPLETE:
//     printf("EVT_CONNECTION_UPDATE_COMPLETE: %x \r\n", *(uint8 *)eventParam);
        break;
    case CYBLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT:

```

```

//     printf("CYBLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT \r\n");
break;

/*****
*         GATT Events
*****/
case CYBLE_EVT_GATT_CONNECT_IND:
//     printf("EVT_GATT_CONNECT_IND: %x, %x \r\n", cyBle_connHandle.attId, cyBle_connHandle.bdHandle);
break;
case CYBLE_EVT_GATT_DISCONNECT_IND:
//     printf("EVT_GATT_DISCONNECT_IND \r\n");
break;
case CYBLE_EVT_GATTS_WRITE_REQ:
//     printf("EVT_GATT_WRITE_REQ: %x = ",((CYBLE_GATTS_WRITE_REQ_PARAM_T *)eventParam)-
>handleValPair.attrHandle);
//     ShowValue(&((CYBLE_GATTS_WRITE_REQ_PARAM_T *)eventParam)->handleValPair.value);
(void)CyBle_GattsWriteRsp(((CYBLE_GATTS_WRITE_REQ_PARAM_T *)eventParam)->connHandle);
break;
case CYBLE_EVT_GATTS_INDICATION_ENABLED:
//     printf("CYBLE_EVT_GATTS_INDICATION_ENABLED \r\n");
break;
case CYBLE_EVT_GATTS_INDICATION_DISABLED:
//     printf("CYBLE_EVT_GATTS_INDICATION_DISABLED \r\n");
break;

        default:
//     printf("OTHER event: %lx \r\n", event);
break;
    }
}

```

```

/*****
* Function Name: Timer_Interrupt
*****/
*
* Summary:
* Handles the Interrupt Service Routine for the WDT timer.
*
*****/
CY_ISR(Timer_Interrupt)
{
    if(CySysWdtGetInterruptSource() & WDT_INTERRUPT_SOURCE)
    {
        /* Indicate that timer is raised to the main loop */
        mainTimer++;

        /* Clears interrupt request */
        CySysWdtClearInterrupt(WDT_INTERRUPT_SOURCE);
    }
}

```

```

/*****
* Function Name: WDT_Start
*****/

```

```

*
* Summary:
* Configures WDT to trigger an interrupt every second.
*
***** /

void WDT_Start(void)
{
    /* Unlock the WDT registers for modification */
    CySysWdtUnlock();
    /* Setup ISR */
    WDT_Interrupt_StartEx(&Timer_Interrupt);
    /* Write the mode to generate interrupt on match */
    CySysWdtWriteMode(WDT_COUNTER, CY_SYS_WDT_MODE_INT);
    /* Configure the WDT counter clear on a match setting */
    CySysWdtWriteClearOnMatch(WDT_COUNTER, WDT_COUNTER_ENABLE);
    /* Configure the WDT counter match comparison value */
    CySysWdtWriteMatch(WDT_COUNTER, WDT_1SEC);
    /* Reset WDT counter */
    CySysWdtResetCounters(WDT_COUNTER);
    /* Enable the specified WDT counter */
    CySysWdtEnable(WDT_COUNTER_MASK);
    /* Lock out configuration changes to the Watchdog timer registers */
    CySysWdtLock();
}

/*****
* Function Name: WDT_Stop
*****
*
* Summary:
* This API stops the WDT timer.
*
***** /

void WDT_Stop(void)
{
    /* Unlock the WDT registers for modification */
    CySysWdtUnlock();
    /* Disable the specified WDT counter */
    CySysWdtDisable(WDT_COUNTER_MASK);
    /* Locks out configuration changes to the Watchdog timer registers */
    CySysWdtLock();
}

/*****
* Function Name: CY_ISR
*****
*
* Summary:
* This API reads AD7789 when DUOT/RDY goes low.
*
***** /

CY_ISR(AD7789_Interrupt)
{

```

```

SPIM_SpiUartWriteTxData(0x38); // Read once the data in order to accept the command
while(0u == (SPIM_GetMasterInterruptSource() & SPIM_INTR_MASTER_SPI_DONE));
SPIM_ClearMasterInterruptSource(SPIM_INTR_MASTER_SPI_DONE);

SPIM_SpiUartClearRxBuffer();
SPIM_SpiUartWriteTxData(0xFF); // Dummy write to read 1 byte - write 0 to avoid ADC RESET
SPIM_SpiUartWriteTxData(0xFF); // Dummy write to read 1 byte - write 0 to avoid ADC RESET
SPIM_SpiUartWriteTxData(0xFF); // Dummy write to read 1 byte - write 0 to avoid ADC RESET
while(0u == (SPIM_GetMasterInterruptSource() & SPIM_INTR_MASTER_SPI_DONE));
SPIM_ClearMasterInterruptSource(SPIM_INTR_MASTER_SPI_DONE);
test[0] = SPIM_SpiUartReadRxData(); // Read MSByte
test[1] = SPIM_SpiUartReadRxData(); // Read next Byte
test[2] = SPIM_SpiUartReadRxData(); // Read LSByte

TemperatureBufferPointer ++;
if (TemperatureBufferPointer > 15) TemperatureBufferPointer = 0;
TemperatureBuffer[TemperatureBufferPointer] = (test[0]<<16)+(test[1]<<8)+test[2];

InputPin_ClearInterrupt();
}

/*****
* Function Name: main
*****/
*
* Summary:
* This is the main() C function.
*
*****/
int main()
{
    SS_AD7789_Write(1); // ADC Chip Select unactive
    CYBLE_LP_MODE_T lpMode;
    CYBLE_BLESS_STATE_T blessState;

    /* Start CYBLE component and register generic event handler */
    CyBle_Start(AppCallBack);
    /* Register service specific callback functions */
    CyBle_HtsRegisterAttrCallback(HtsCallBack);

    WDT_Start(); // Start WDT
    SPIM_Start(); // Start SPI for ADC communication
    SS_AD7789_Write(0); // ADC Chip Select active

    SPIM_SpiUartWriteTxData(0x38); // Read once the data in order to accept the command
    while(0u == (SPIM_GetMasterInterruptSource() & SPIM_INTR_MASTER_SPI_DONE));
    SPIM_ClearMasterInterruptSource(SPIM_INTR_MASTER_SPI_DONE);
    SPIM_SpiUartClearRxBuffer();
    SPIM_SpiUartWriteTxData(0xFF); // Dummy write to read 1 byte
    SPIM_SpiUartWriteTxData(0xFF); // Dummy write to read 1 byte
    SPIM_SpiUartWriteTxData(0xFF); // Dummy write to read 1 byte
    while(0u == (SPIM_GetMasterInterruptSource() & SPIM_INTR_MASTER_SPI_DONE));
    SPIM_ClearMasterInterruptSource(SPIM_INTR_MASTER_SPI_DONE);
    test[0] = SPIM_SpiUartReadRxData(); // Read MSByte
    test[1] = SPIM_SpiUartReadRxData(); // Read next Byte

```

```

test[2] = SPIM_SpiUartReadRxData();    // Read LSByte

SPIM_SpiUartWriteTxData(0x10);        // Write Mode register command
SPIM_SpiUartWriteTxData(0x02);        // Continuous conversion mode, Bipolar
while(0u == (SPIM_GetMasterInterruptSource() & SPIM_INTR_MASTER_SPI_DONE));
SPIM_ClearMasterInterruptSource(SPIM_INTR_MASTER_SPI_DONE);

ExternalADC_Interrupt_StartEx(AD7789_Interrupt);

CyGlobalIntEnable;

/*****
* Main polling loop
*****/
while(1)
{
    if(CyBle_GetState() != CYBLE_STATE_INITIALIZING)
    {
        /* Enter DeepSleep mode between connection intervals */
        lpMode = CyBle_EnterLPM(CYBLE_BLESS_DEEPSLEEP);

        blessState = CyBle_GetBleSsState();

        if(lpMode == CYBLE_BLESS_DEEPSLEEP)
        {
            if(blessState == CYBLE_BLESS_STATE_ECO_ON || blessState == CYBLE_BLESS_STATE_DEEPSLEEP)
            {
                CySysPmDeepSleep();
            }
        }
        else
        {
            if(blessState != CYBLE_BLESS_STATE_EVENT_CLOSE)
            {
                CySysPmSleep();
            }
        }
    }
}

/*****
* Wait for connection established with Central device
*****/
if(CyBle_GetState() == CYBLE_STATE_CONNECTED)
{
    /*****
    * Periodically measure a battery level and temperature and send
    * results to the Client
    *****/
    if(mainTimer != 0u)
    {
        mainTimer = 0u;
        if(temperatureMeasure == ENABLED)

```

```

    {
        MeasureTemperature();
        CyBle_ProcessEvents();
    }
}

/*****
* Processes all pending BLE events in the stack
*****/
CyBle_ProcessEvents();

    } // End of infinite loop
} // End of main

/* [] END OF FILE */

```

Appendix II

This section includes the explanation of creating the BLE communication program and receiving the data using CySmart.

1. CY8C042-Kit BLE: Figure A.1 shows the details of the kit and the items provided with it.



Figure A.1: Different components of the CY8C042-kit BLE from Cypress

2. PSoC Creator

In the PSoC creator software, there are several analog, digital and communication components present (Figure A.2). For creating the communication program for the sensor, the following components are used: Serial Peripheral Interface (SPI) for values from ADC, BLE for transmission, Watch dog Timer (WDT), and external interrupt (Figure A.2). The source and header files for each of the components that are placed in the work space are generated internally and need not be separately written (figure A.4). The main program, written in C language, includes the way these components will be used and connected with each other to realize the design. For temperature measurement, there is an example project which is used as the basis in creating the communication

of the wireless prototype. The program is edited to suit the requirements of the present application, a flow chart of which was discussed in section 3.4.

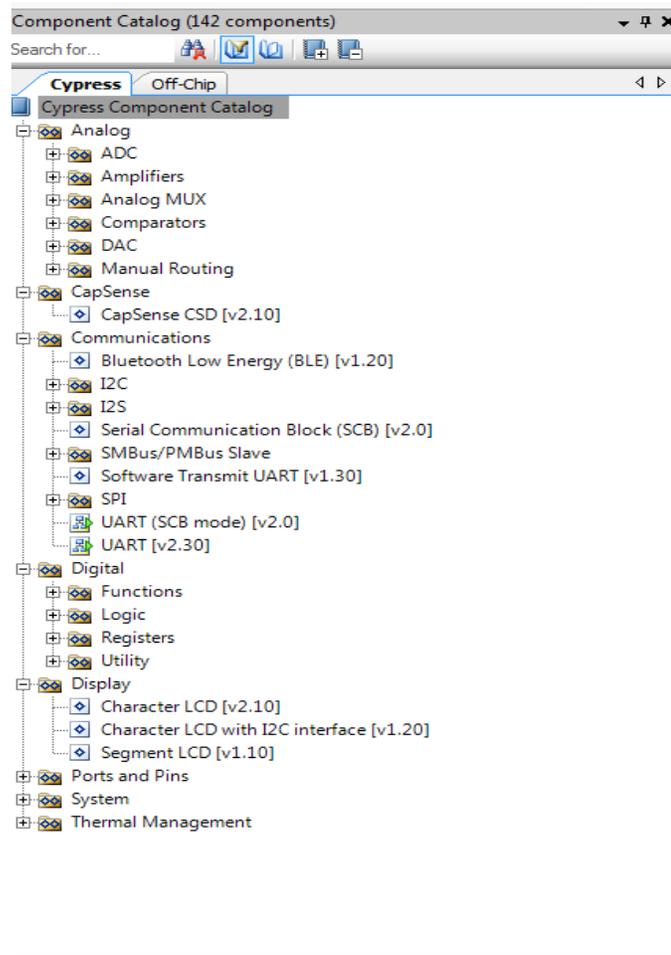


Figure A.2: Displays the component catalogue from PSoC creator through which different components can be selected for the design

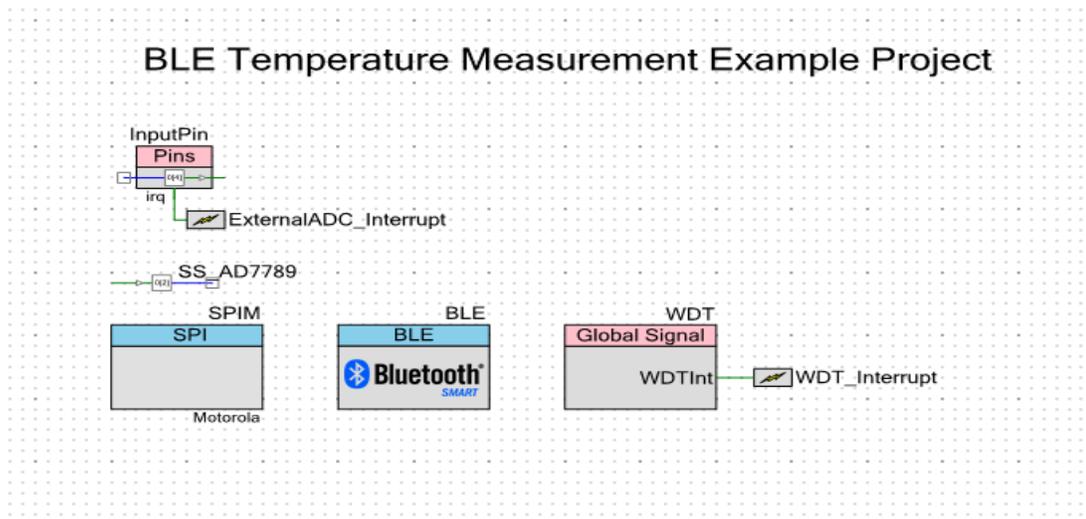


Figure A.3: The PSoC creator workspace where in the components can be dragged and dropped.

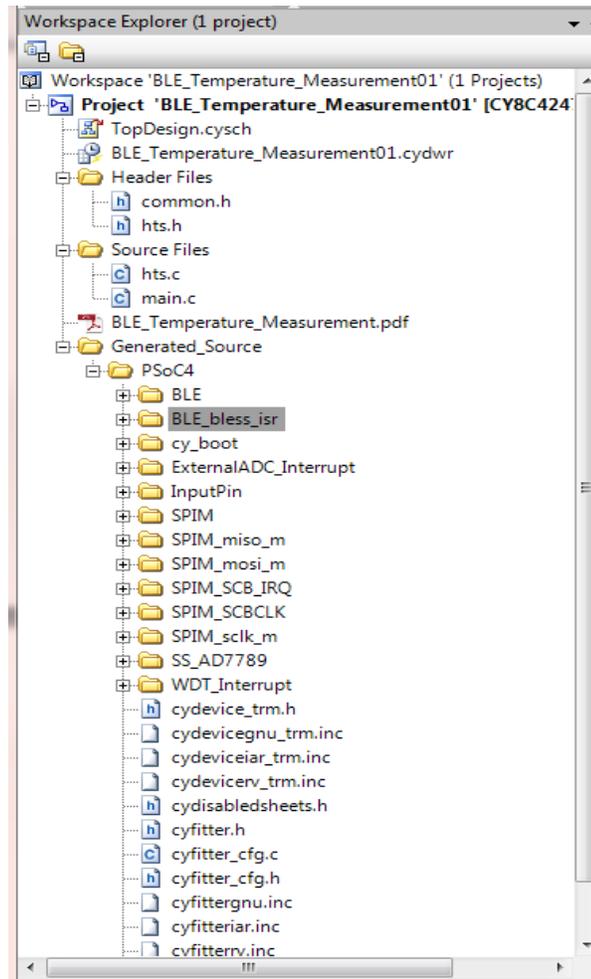


Figure A.4: The source and header files of the desired components created by the PSoC creator

Once the code is ready it needs to be checked for syntax/semantics or other errors. For this, the first step is to select the device to which the programmed has to be written. In the Project drop down menu, the device selector option can be used for this. For example, the device CY8C4247LQI-BL493 is selected for the full bridge prototype (Figure A.5). Next, the program is checked for errors using “build” option from the build menu. After this step a hex file is created which can be dumped into the inbuilt microcontroller in the kit. This can be done either directly using the “Program” option from the debug drop down menu. Or, the hex file can be loaded into the board using PSoC programmer 3.2.2.

4. Receiver: CySmart 1.0 GUI is used in windows laptop for receiving the temperature values. It is easy to use and user friendly. Here are the steps that need to be followed in receiving the information.

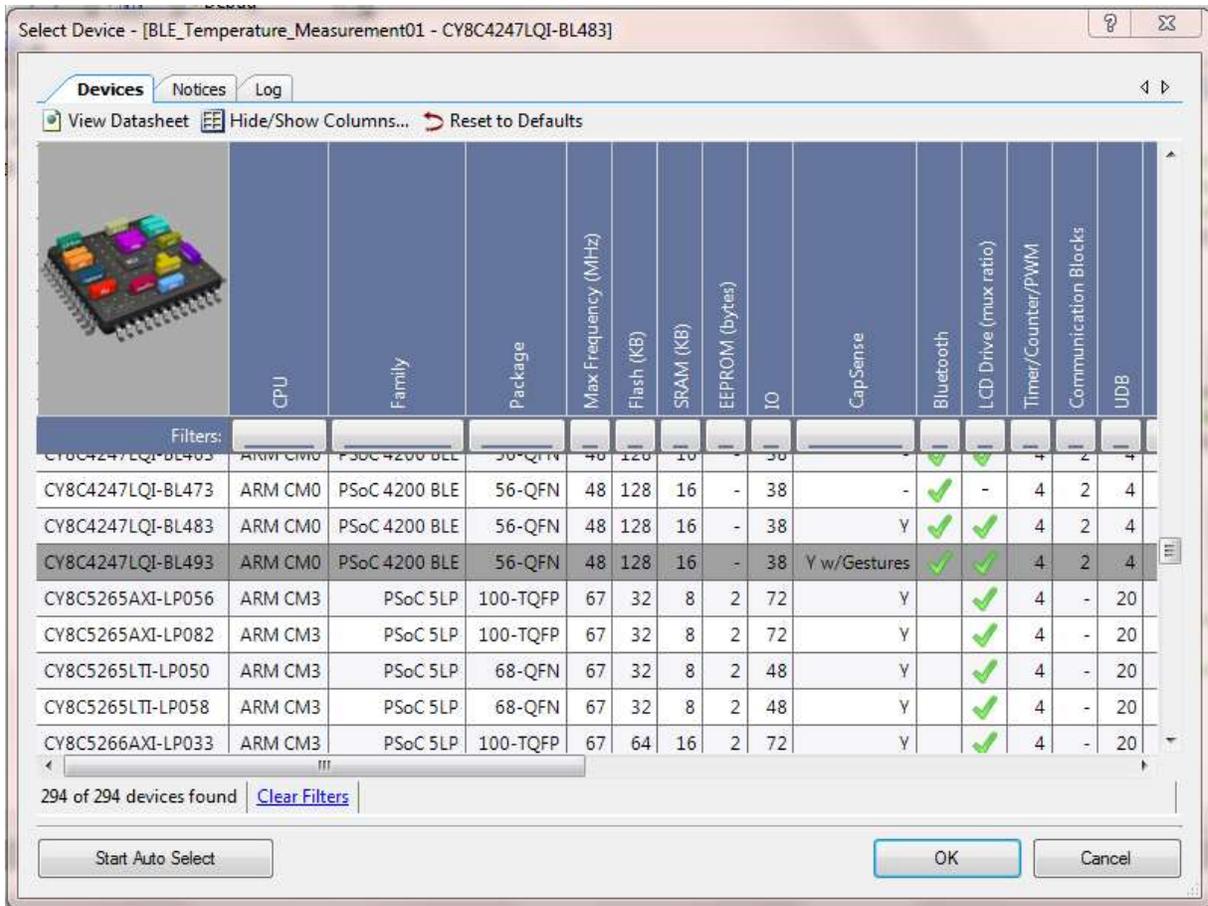


Figure A.5: Screen shot of the device selector dialogue box from PSoC creator

- i. The startup window shows the available devices, and the connection has to be made by double clicking the device name (Figure A.6).
- ii. Once the connection is established, the devices have to be “paired”.
- iii. This is followed by clicking “discover all notifications” tab, which enables the receiver to see all the services and characteristics offered by the BLE device.
- iv. Finally, “enable all notifications”, will alert the device to start sending the data. A screen shot of CySmart highlighting steps ii, iii, iv is shown in Figure A.7.
- v. The measured temperature values are displayed in 5-byte hexadecimal format in the “log window”, out of which the first byte is used by the BLE stack to indicate the kind of service given by BLE (ex: temperature monitoring, Battery level and so on). The next three bytes are the data in the order of MSbyte followed by the next byte and LSbyte. The last byte is empty as the ADC is only 24-bit (Figure A.8). The steps for extracting the temperature value from the 5-byte hex format are explained in Appendix III.

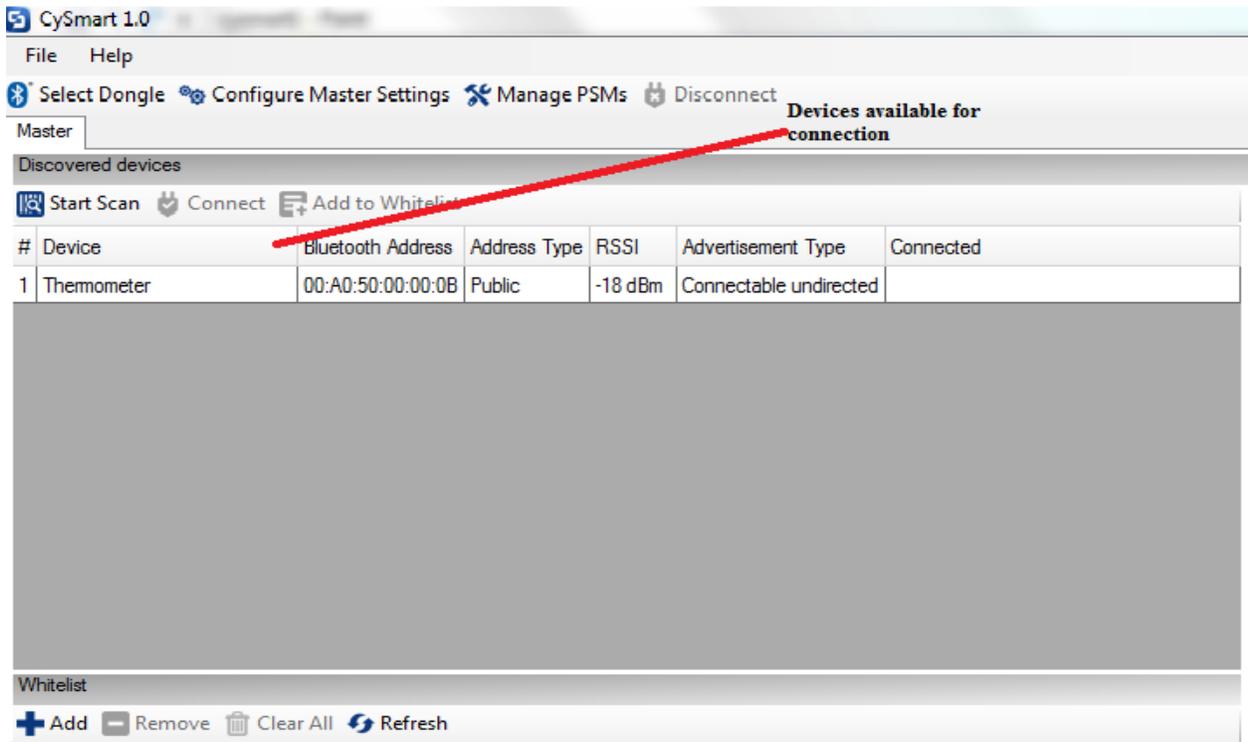


Figure A.6: Screen shot of CySmart GUI showing the available devices for connection

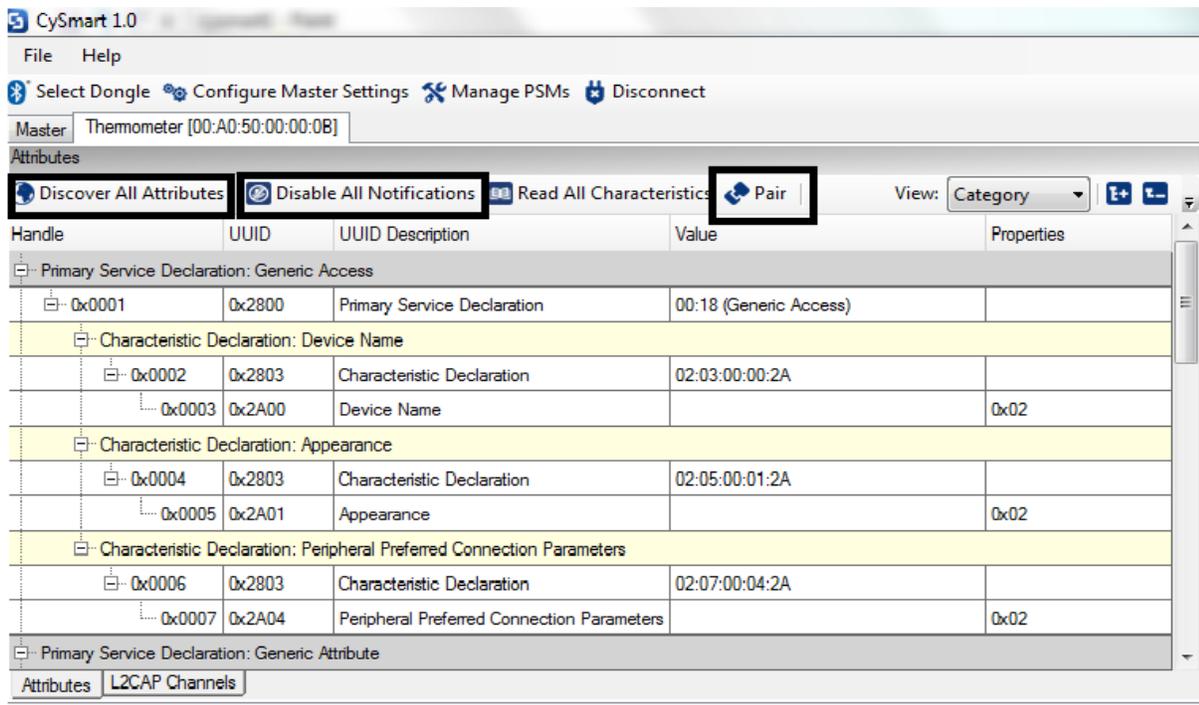


Figure A.7: The screen shot of Cysmart GUI highlighting the steps: pair, discover all attributes, and enable notifications.

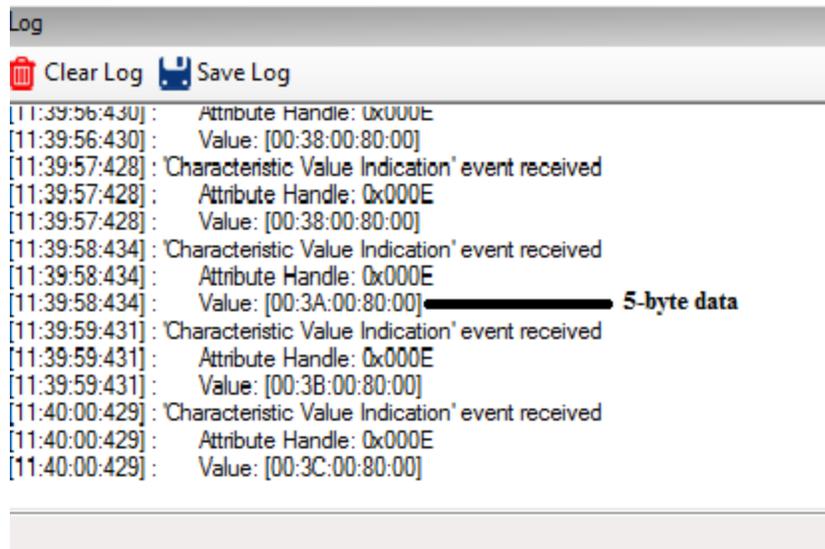


Figure A.8: Screen shot of the “log window” on Cysmart where the temperature data is displayed

Appendix III

For all the qualifying tests conducted on the wireless temperature sensor (described in section 4.2), the data is received using CySmart as mentioned earlier. In this section, steps followed for extracting the temperature data from the 5-byte hexadecimal format, displayed in CySmart is explained.

1. Among the 5-bytes received, only three bytes are of interest to us as mentioned earlier. Let us consider an example value received in CySmart: 00:78:88:8D:00
2. In the data, LSB of the value sent by ADC is received first, followed by the next byte and then the MSB. Hence, the actual data in hexadecimal would then be: 8D8878.
3. This value is converted to decimal value using HEX2DEC function in excel, which amounts to 9275512. This is the code sent by the ADC.
4. Next step is to convert the ADC code to measured output voltage of the bridge.
 - a. For the full bridge, the ADC operates in bipolar mode. Hence the formula used is:

$$Code = 2^{N-1} \times [(A_{IN}/V_{REF}) + 1]$$

Where, N=24 (number of bits of ADC), Vref = 0.82 V (reference voltage of the bridge), A_{in} = the measured output voltage of the bridge

- b. For the half bridge, the ADC operates in unipolar mode. Hence the formula used is:

$$Code = 2^N \times (A_{IN}/V_{REF})$$

5. Continuing the example, the value of 9275512 from the ADC translates to 0.086696301V for a full bridge configuration. The value read by the ADC is the ratio of V_{out} (output voltage of the bridge) and V_{ref}. Hence, this value is divided by reference to get V_{out} = 0.1057 V.
6. Next steps are converting the voltage to corresponding resistance value. This can be calculated using the output equation for the bridge. For a full bridge, the output is expressed as:

$$V_{out} = \frac{R - R_t}{2(R + R_t)} * V_{dd}$$

where R = 11.5 kΩ, V_{dd} = 3.6V, V_{out} = 0.1057 (calculated in the previous step), R_t = resistance of the thermistor

7. The value of R_t obtained after this step is = 17.667 kΩ.
8. The final step is to convert the resistance data to equivalent temperature data. For this step, the Steinhart – Hart equation is used.

$$\frac{1}{T} = A + \frac{B}{\ln(Rt)} + \frac{C}{[\ln(Rt)]^3}$$

where $A = 0.000631079$, $B = 0.000295679$, $C = -4.64221E-10$, Rt = resistance value calculated in step 7.

9. The temperature calculated using this equation corresponds to 283.911 K, which is 10. 911 °C.

These steps are followed to calculate values of temperature for various tests conducted on the sensor.

Appendix IV

In this section, the steps followed for error minimization of NTC temperature sensors is explained in detail. To process the data, excel was used. The raw data was received from TempControl B.V., which included the actual drift of the sensors and the deterministic error. To separate the drift from the deterministic error, the following steps were taken.

1. Multiple samples of each sensor were measured for drift. All the samples were taken from the same batch. As the first step, the initial phase, where the variation of the resistances was very high, was considered as pre-aging and disregarded during error minimization. Next, the average of each of these set of sensors was taken. The two outliers in NCP15XH103D03RC were disregarded during the averaging. All the other samples of each sensor were included while averaging.
2. Next, a linear approximation of the average of each sensor was made by plotting the “linear trend line” in excel (Figure A.9). Both logarithmic and linear approximations were applied. As there was not large change in the results, linear approximation was finalized.

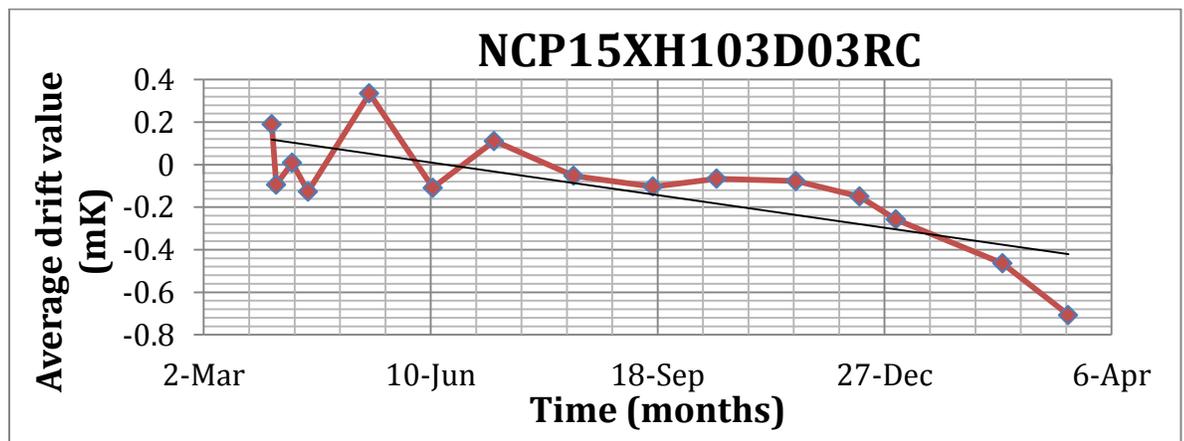
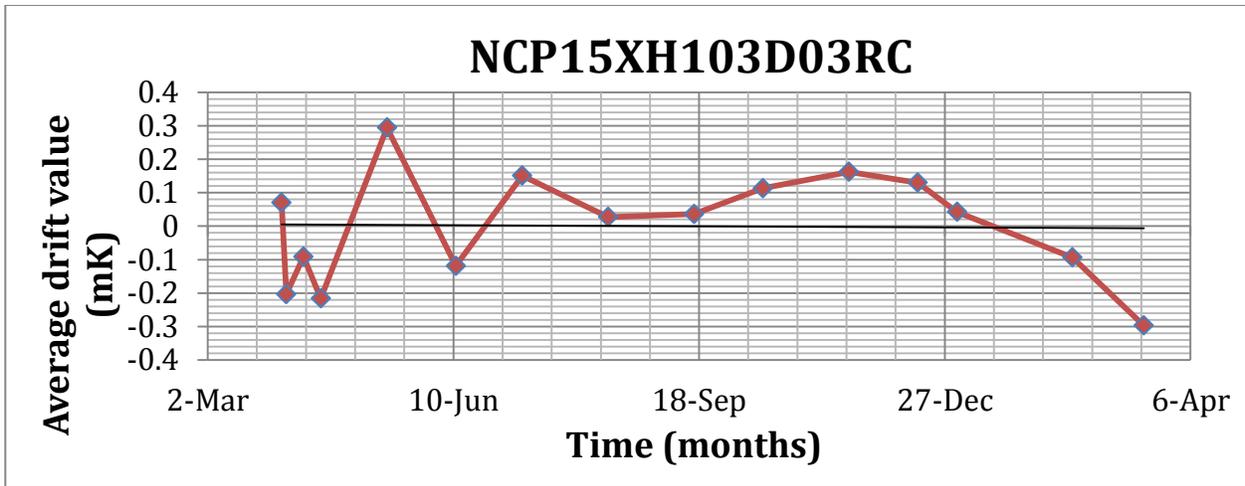


Figure A.9: Plot of average value of 8 samples of NCP15XH103D03RC, along with the linear approximation

3. To calculate the error value, the values of the linear approximation was noted. This value was subtracted from the average value of each of the sensors. That is, the actual drift is subtracted from the average value. What remains is the deterministic error (Figure A.10). A screenshot of the excel file is shown (figure A.11).



Figure

A.10: Example plot showing the difference of average value of the sensors and the linear trend line

Date	AVG	linear_line_Value	Error
1-Apr	0.19	0.12	0.07
3-Apr	-0.09375	0.11	-0.20375
10-Apr	0.00875	0.1	-0.09125
17-Apr	-0.12625	0.09	-0.21625
14-May	0.335	0.04	0.295
11-Jun	-0.10875	0.01	-0.11875
8-Jul	0.11125	-0.04	0.15125
12-Aug	-0.0525	-0.08	0.0275
16-Sep	-0.10375	-0.14	0.03625
14-Oct	-0.06625	-0.18	0.11375
18-Nov	-0.0775	-0.24	0.1625
16-Dec	-0.15	-0.28	0.13
1-Jan	-0.2575	-0.3	0.0425
17-Feb	-0.46375	-0.37	-0.09375
18-Mar	-0.7075	-0.41	-0.2975

Figure A.11: Screen shot of the excel fine for the error calculation of NCP15XH103D03RC

- Similarly, for each other the sensor sets, steps 1-3 are applied. A plot of deterministic errors of all the sensors was presented in Figure 5.4. The average of the deterministic error was taken. The outliers (error values of 10K3MCD1) were not included while averaging. This average deterministic error was subtracted from each of the original (average) value of the sensors, which had both deterministic error and drift (figure A.9). The difference between them was considered as the final drift of the sensors.

Appendix V

Figure 5.8 and 5.9 show the transfer characteristics of the wireless temperature sensor. Apparently, they seem linear. However, when we zoom into the curve, an inaccuracy of ± 1 mK (shown in Figure 5.10 and 5.11) is observed. In this section, the calculation of this inaccuracy of the transfer characteristics is explained in detail. An ideal transfer characteristic was calculated. The difference between the measured and the ideal transfer is the inaccuracy hidden in Figure 5.8 and Figure 5.9.

1. The ideal, linear transfer curve would have same slope for two measured data points. By using the geometrical formula,

$$\frac{Y_2 - Y_1}{X_2 - X_1} = \text{slope}$$

2. From the measured graph, $X_2 - X_1$ (difference between two consecutive points of the reference temperature), slope and Y_1 (from the temperature measured by the wireless sensor) are obtained.
3. Y_2 is calculated for each temperature point and the ideal line is plotted.
4. The difference between the two curves is the inaccuracy of the transfer characteristics. It is found to be ± 1 mK.

