Hybrid RFID Sensors: Design, Implementation and Application

Dissertation presented for the degree of
Master of science

University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

SPES BONA

ISAT laboratory
Department of Computer science
University of Cape Town

Jarred Martin

August 2014

Supervisor: Dr A. Bagula
Head of Department: Dr S. Berman
Date of submission: August 2014

Declaration of Authorship

I, Jarred Martin, declare that this thesis titled, ”Hybrid RFID Sensors: Design, Implementation and Application” and the work presented in it is my own. I confirm that:

- This work was done wholly for the degree of Master of science at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed: ______________________________

Date: ______________________________

Dedicated to my mother.

Abstract

The fields of Wireless sensor networks and RFID technology are two examples of the current move to ubiquitous computing. Wireless sensor networks has emerged as a tool for long term remote monitoring for applications ranging from agriculture to military. While in RFID we have already seen it being used in everyday life from access control to asset tracking. The integration of these two fields allows for a whole range of new applications, the focus of this dissertation is to present a wireless sensor network platform which incorporates a hybrid RFID sensor mote for the detection of environmental conditions and the locating of objects or personnel within an environment.

The solution that is proposed comprises of both hardware and software but focuses on the design of the platforms' prototype wireless sensor mote which provides object detection through the use of an RFID reader and environmental conditions by using low cost slave sensors.

The solution was then applied to solving the problem of locating mining personnel and detecting hazardous levels of methane gas for use in underground mines.

Acknowledgements

First and foremost I would like to thank my supervisor Dr. Antoine Bagula for introducing me to the field of wireless sensor networks during the Internet of Things course in my honours year and his constant guidance during the development of this Msc project. When this journey started I was so lost and unsure of where and how to start, the meetings and group presentations quickly guided me in the right direction. The opportunities and knowledge you have provided me with in my academic career are also much appreciated.

I would also like to thank the group members of the ISAT laboratory for their assistance and input during the many days spent in the lab. The fun conversations we had in the lab about things happening in our lives kept me smiling throughout.

To the Computer science department, it has been many years since I started in the GEPS program in 2008. Thank you all for the solid foundation laid during my undergraduate years.

To my family; my dad, brother and aunt Stephanie I doubt I would have been able to complete my studies and achieve all that I have if it wasn't for your immense support.

And finally to my mom I wish you could have been here to see the man I have become, this is all for you...

Contents

Abstract **v**

Acknowledgements **vi**

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Problem statement . . . 2

1.2.1 Mining Industry . . . 2

1.2.2 Supply chain management . . . 2

1.2.3 Smart buildings . . . 2

1.2.4 Healthcare . . . 3

1.3 Aims . . . 3

1.4 Dissertation outline . . . 4

2 Background **5**

2.1 Wireless Sensor Networks . . . 5

2.1.1 Sensor mote . . . 7

2.1.2 Gateway device . . . 7

2.1.3 Sensor network applications . . . 8

2.1.4 Discussion . . . 11

2.2 Radio Frequency Identification . . . 12

2.2.1 Types of RFID tags . . . 12

2.2.2 Types of RFID readers . . . 12

2.2.3 RFID monitoring applications . . . 13

2.2.4 Discussion . . . 14

2.3 RFID and WSN integration . . . 15

2.3.1 Integrating RFID tags with sensors . . . 15

2.3.2 Integrating Wireless sensor motes with RFID readers . 15

3 System design 16

3.1 Design requirements 16

3.1.1 Functional requirements 16

3.1.2 Performance requirements 17

3.1.3 Performance evaluation 21

3.2 Design overview 22

3.2.1 Slave sensor 22

3.2.2 Hybrid mote 23

3.2.3 Gateway device 24

3.3 Component selection 24

3.3.1 Gateway selection 25

3.3.2 Microcontroller selection 34

3.3.3 Transceiver selection 43

4 System implementation 48

4.1 Overview 48

4.2 Hybrid RFID mote components 49

4.2.1 Arduino Uno 49

4.2.2 OpenBeacon RFID reader 54

4.3 Gateway components 55

4.3.1 Serial reader 56

4.3.2 MySQL database 56

4.4 Slave sensor components 57

4.4.1 I2C manager 58

4.4.2 Gas sensor: NGM-2611 58

5 Design Evaluation 60

5.1 Overview 60

5.2 Laboratory tests 60

5.2.1 Power consumption 62

5.2.2 RFID tag detection range 68

5.2.3 Slave sensor response time 70

6 Conclusion and future work 72

6.1 Conclusion 72

6.1.1 Summary 72

6.2 Future work 73
6.2.1 Large scale network testing 73
6.2.2 Interference testing 74
6.2.3 Multiple sensors 74
6.2.4 Environmental testing 74
6.2.5 Power saving 74

Appendix A System preparation 75
A.1 MySQL setup 75
A.2 Database creation 76
A.3 Apache2 installation 76
A.4 MySQL INSERT 77
A.5 Measuring execution time 77
A.6 Measuring system resources 77

Appendix B Httperf 78
B.1 Sample output 78

Bibliography 79

List of Figures

1.1 The proposed result of integration between RFID and WSN's 4

2.1 Generic WSN architecture. 6
2.2 ZebraNet sensor mote [1] 9
2.3 GDI sensor mote [2] 10

3.1 Communication between components. 22
3.2 Hardware components needed by each tier 24
3.3 CPU usage of Raspberry pi and Alix board 28
3.4 I/O wait time of Raspberry pi and Alix board 28
3.5 CPU usage of Raspberry pi and Alix board 29
3.6 Upper bound of connection rate 33
3.7 Web server reply time 33
3.8 Test code structure. 36
3.9 Output trace from oscilloscope. 37
3.10 Comparison of run times. 38
3.11 MPS430G2553 test circuit with shunt resistor. 40
3.12 ATMEGA328P test circuit with shunt resistor. 41
3.13 Comparison of the current draw for the microcontrollers. . . . 42
3.14 RSS vs. Distance. 46

4.1 Hardware configuration for the Hybrid RFID platform 48
4.2 The hardware and software components which comprise the Hybrid mote. 49
4.3 I2C bus circuit. 51
4.4 Hybrid mote - Slave sensor communication protocol. 51
4.5 Scheduler operations. 52
4.6 RFID tag detection algorithm. 53

4.7 Interface between Arduino UNO and openBeacon RFID reader. 54
4.8 Gateway middleware 55
4.9 Serial reader algorirthm 56
4.10 Database tables 57
4.11 Slave sensor circuit diagram 58
4.12 Methane gas sensor module [3] 59

5.1 Test setup 61
5.2 Evaluation time span 62
5.3 Solar power supply 63
5.4 Voltage vs. time 64
5.5 The Openbeacon RFID tag that was used [4] 68
5.6 RFID tag power level 69
5.7 Sensor test chamber 71
5.8 Detection time 71

6.1 Specifications of Hybrid mote 73
6.2 Specifications of Slave sensor 73

List of Tables

3.1 Comparison between suitable gateway devices. 25
3.2 Device configuration for experimentation. 26
3.3 Comparison between suitable microcontrollers. 34
3.4 Comparision between suitable transceivers. 44

Chapter 1

Introduction

1.1 Motivation

The two fields of wireless technologies, RFID and wireless sensor networks are useful technologies that have a wide range of applications. In RFID applications include supply chain management and manufacturing, in wireless sensor networks the technology has been applied in deploying sensor motes to monitor air pollution [5] and battlefield surveillance [6] . The development of these technologies have largely been done in parallel, thus there has not been much research in integrating these two technologies [7].

An RFID system would consist of an RFID tag reader and RFID tags [7], the function of the reader would be to use its radio transceiver to transmit an encoded signal and listen on a certain frequency for signals from an RFID tag [8]. Depending on the type of RFID system being implemented the RFID tag may periodically emit a radio frequency signal which may contain the tags ID or information about the object the tag is attached to. The RFID system is able to identify individual objects within an environment however it is unable to give an indication of the physical condition of that object or whether the environment is suitable for the object, this is where wireless sensing technology can be used.

A typical wireless sensor network is comprised of sensor motes equipped with; wireless communications hardware [5], sensors to detect changes in its environment and a gateway device [9] which functions as a link between the ad-hoc network of sensor motes and traditional computing devices such as desktop computers on a network [9]. Users are then able to query or monitor the status of the environment in which the sensor motes are in.

1.2 Problem statement

Current wireless sensor paradigms do not allow for the detection and locating of an object within an environment [10] as well as giving an indication as to whether that environment is suitable for the object to be in. A system which is able to provide such information can be highly useful to the following industries;

1.2.1 Mining Industry

In the application of the coal mining industry where miner safety in underground tunnels is paramount in running an efficient operation and the threat of explosions due to methane gas [11], [12] seeping from the surrounding strata provides a very real problem. Current systems do exist to detect the concentrations of harmful gases such as methane, carbon monoxide and carbon dioxide [13] within the tunnels, however they do not provide information as to whether there are miners within the vicinity of the dangerous concentrations of gases [14].

1.2.2 Supply chain management

For companies which specialize in the logistics of transporting fresh produce from the farm to the supermarket, it would be helpful to know whether the vehicle or warehouse in which the produce is stored is at the correct temperature in order to preserve the quality of the produce. The produce could be tagged with smart RFID tags equipped with temperature sensors which would sense the surrounding environments temperature and alert them if the produce is being exposed to too low or high temperatures.

1.2.3 Smart buildings

In a smart building application, the monitoring of the power consumption of a building is critical in order to save on the costs of providing that power. A building power management system which controls heating, ventilation, air conditioning and lighting could be equipped with a system to be able to detect where people are within the building and where most of the power is being consumed and thereby save energy by diverting power away from unoccupied rooms.

1.2.4 Healthcare

In a healthcare application a system which monitors the location of patients in a hospital can be used to keep track of patients who have mental disabilities. Such patients would often leave their hospital rooms and get lost within the building, a system has been implemented at a Belgian hospital which achieves patient and staff tracking [15].

1.3 Aims

The focus of this research is to develop a platform which supports wireless sensor motes that are able to identify and locate objects within an environment and to provide information on whether the environmental conditions are suitable for the object to be in.

A system which incorporates both RFID, wireless sensing technology and an associated interfacing platform, such as the Hybrid RFID sensing mote proposed in this document would allow for a large number of new applications. With that said this dissertation aims to answer the following questions;

1. Which technique in the integration of WSN and RFID would be able to provide a platform with the environmental awareness envisioned?

2. How can the platform be designed to be easily reconfigured for a multitude of different sensing application?

3. How can the cost of deploying a sensing network be reduced while increasing the sensing resolution of an environment.

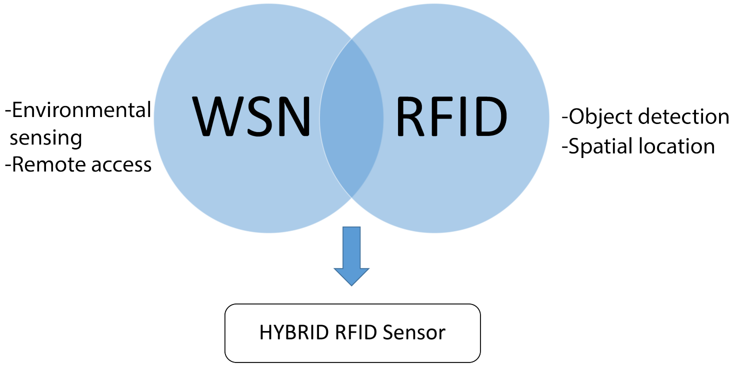


Figure 1.1: The proposed result of integration between RFID and WSN's

1.4 Dissertation outline

Chapter 2 presents an overview of the current trends in the fields of WSN and RFID, examples of major projects in the fields are mentioned. Techniques of integrating RFID and WSN are also discussed.

Chapter 3 presents the high level design of the proposed system and evaluates by experimentation the suitability of each component of the design.

Chapter 4 The implementation of the system is presented, with each components functional systems being described.

Chapter 5 laboratory evaluation of the platform as a mine monitoring system is presented with results.

Chapter 6 provides a conclusion of the results of evaluation and future work to improve the platform.

Chapter 2

Background

2.1 Wireless Sensor Networks

The field of Wireless sensor networks has been around for many years, it was originally developed by the military for use in battlefield communication and surveillance [6]. But now academia and industry are leading developments [16]. Within the last decade there has been tremendous progress in terms of the number of research papers being published and applications of the technology, this is mostly due to;

- **Increased processing capabilities of low powered computing devices** [17]
  Progress in manufacturing of semiconductors and integrated circuits have allowed much more components to be placed on a silicon wafer, these components allow devices to have more transistors ,RAM , Flash memory and input/output peripherals to name a few.

- **Efficiency of wireless communications protocols for low powered devices** [18]
  Wireless communications protocols such as IEEE 802.15.4, 6LoWPAN, ZigBee, etc. have been specifically designed for use by systems which are typically battery powered [19] and as a result need to be as efficient as possible when it comes to power consumption and processor resource usage. Adoption of these protocols as standards have been key in advancing the technologies wide spread use.

- **Cost of hardware**
  Current low powered microcontrollers which are typically 8 bit [20], 16-MHz microcontrollers could cost as low as $2. With these devices very few external components are required for them to function.

- **Need for data**
  The remote monitoring of an envrionment which may be hard to gain access to or the process of monitoring requires minimal interference, wireless sensor networks are able to provide these monitoring services with minimal interference as the sensing devices are accessed remotely through the wireless sensor network.

A wireless sensor network consists of many components, they can be either physical hardware or Software which operates the hardware. The diagram in figure 2.1 shows how the physical hardware of a typical wireless sensor network are connected.

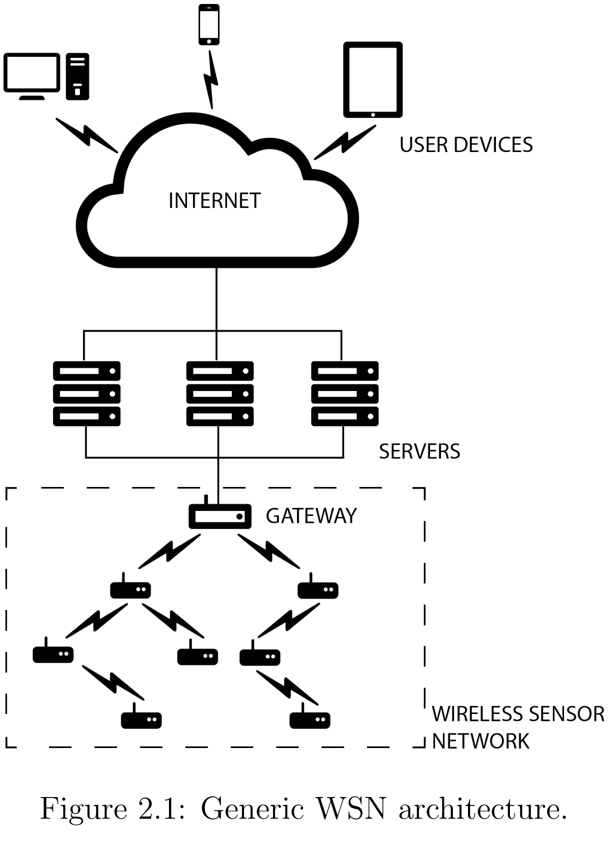


Figure 2.1: Generic WSN architecture.

The three main components of a wireless sensor network include:

- Sensor mote
- Gateway
- Middleware

2.1.1 Sensor mote

The sensor mote lies at the core of the wireless sensor network. Its primary function is to read various parameters of the environment in which it is in [9], depending on the type of sensors it is equipped with. That sensor data is then stored on the sensor mote in its internal memory to be communicated to the networks gateway device where the data would be disseminated across various devices which require the data. In most situations a sensor mote does not have a direct communications link between itself and the gateway device, this may be because the gateway may not be within communications range or the wireless link may be of poor quality [21]. The sensor mote would have to communicate the data to a nearby mote [19] based on the type of routing protocol used with the goal of the data being communicated between sensor motes until it reaches the gateway device.

2.1.2 Gateway device

The gateway device in a wireless sensor network is the connection between the sensor motes and traditional computer networks [22]. This is needed due to the differences in communications protocols used by the sensor network and computer networks [23], in the case of sensor networks the protocols used could be;

- IEEE 802.15.4
- 6LoWPAN
- ZigBee

In computer networks protocols used could be;

- UDP

- TCP/IPv4
- TCP/IPv6

The gateway device would typically receive sensor data from the sensor motes in the network wirelessly then software running on the gateway called the middleware would store the sensor data in a database and perform processing on it and then disseminate the data to devices requesting it.

2.1.2.1 Middleware

The middleware functions as a portal to the sensor network and its data [24], the sensor data becomes formatted so that it becomes easier to understand [24]. Underlying patterns within the sensor data can also be realized by using data mining techniques. The user requesting the sensor data would access the portal via a web browser or by using a native software application on devices such as PCs and mobile phones.

2.1.3 Sensor network applications

In recent years the scientific community has benefited greatly from the use of sensor networks in the remote monitoring of hard to access locations. The advantages allowed by sensor networks outweighs those of traditional invasive monitoring techniques [25], especially in monitoring of wildlife habitats. What follows are two scientific research projects where sensor networks are used, from these two projects we will identify the processes used and lessons learnt in designing the sensor networks. Then apply it in the design of the Hybrid RFID sensing platform.

2.1.3.1 ZebraNet

An example of the power of wireless sensor networks is presented below in the ZebraNet system. It was designed by researchers at Princeton University to support the research of biologists stationed at the Mpala Research Center in Kenya, by using the latest wireless sensor technology to track herds of zebra's. As the project name suggests the network is made up of zebra's.
The ZebraNet system consists of wireless sensing devices that have been embedded into collars that have been placed around the necks of the zebra's. The sensing devices' functions are to take regular readings of the zebras'

location using a GPS unit and then transmit the data from zebra collar to zebra collar until infrequent communications percolate the GPS data to the base stations [26] located around watering holes. The researchers system has been designed to fulfill three goals;

1. Generate detailed, accurate logs of each zebras' position,
2. To recover those logs for analysis
3. To operate autonomously for months.

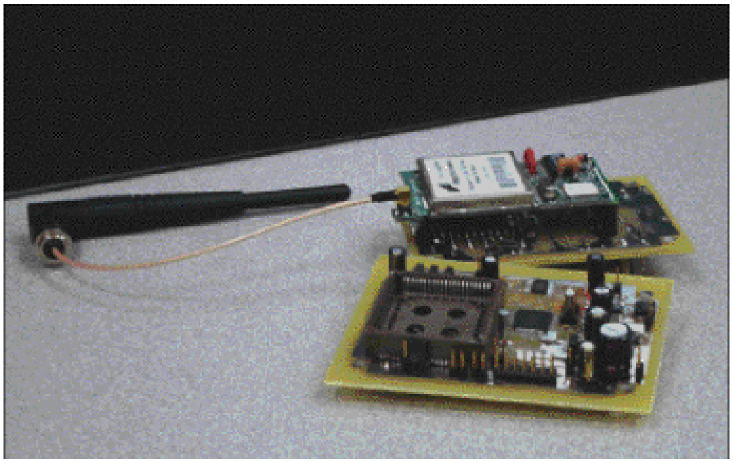


Figure 2.2: ZebraNet sensor mote [1]

To move data through the network of collared devices, the collars communicate via pairwise connections so that when two zebra's come close enough together the collared devices are within communications range. What the autonomous operation means in this system is that the collared devices need to be able to operate for long periods of time without any maintenance, the maintenance that would be involved in wireless sensor networks include; replacing batteries that have been depleted of charge and freeing up storage space on the flash memory device.

Without this type of technology remote monitoring of such situations would not be feasible due to the alternatives being man power intensive and the data not being as fine grained as that provided by ZebraNet.

2.1.3.2 The Great duck island project

The Great duck island (GDI) project undertaken by the College of the Atlantic (COA) involved the monitoring of the habitat of the ”Leachs’ storm Petrel” sea bird, for the purpose of understanding the breeding and nesting characteristics.

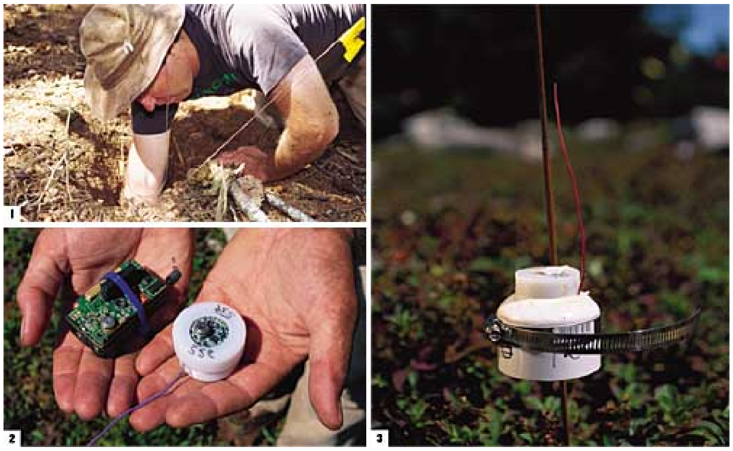


Figure 2.3: GDI sensor mote [2]

The use of a non-invasive form of monitoring was essential due to the sea birds being sensitive to outside disturbances such as researchers, therefore a sensor network was set-up on the island. The researchers system design had to fulfil three design goals [25];

1. Sensor motes needed to operate for 9 months, which is the length of the field season of the sea birds.

2. Small size, to be able to fit inside the sea birds burrows.

3. To operate autonomously.

In trying to achieve the design goals researchers decided on using a tiered network approach consisting of three levels; the lowest level being the sensor motes, which has the function of performing the sensing as well as communicating with other motes in the network.

The next level up is the gateway, all sensor data from the sensor network is routed to the gateway which then transmits the data to a base station which pushes the data to servers on the internet via a satellite link.

The sensor mote of which we are focused was the Mica2Dot with weather sensor board, the battery life of the sensor mote needed to be atleast 270 days; for the small size 2, 2200 mAh AA batteries were used as a power source, this meant that on a daily basis the sensor mote could only consume

$$8.148 = \frac{2200}{270} \tag{2.1}$$

8.148 mAh of battery per day, with this tight energy constraint close attention was paid to the use of energy by each component.

This projects tight power constraints highlights the importance of component selection and employing energy saving features in a sensor network.

2.1.4 Discussion

From what we have seen in the two examples of WSN applications we learnt about the processes guiding the design of a sensor network and sensor mote. The nature of the applications highlight the difference in each ones design requirements, while both being examples of a wildlife monitoring system. In the case of ZebraNet the researchers chose to design a mobile sensor mote instead of opting to use a commercially available product, whereas in the GDI project they opted for the commercially available Mica2Dot.

The designers of these two research projects have each respectively chosen different approaches in their sensor network design. The two types of requirements that guided each projects design are summarised as follows;

System performance

Battery life span, price and size.

System functionality

Sensor types and method of data dissemination.

It is from these design guidelines that we will attempt to realise the Hybrid RFID sensor platform.

2.2 Radio Frequency Identification

The acronym RFID stands for Radio Frequency Identification, the technology has been around for the last seventy years [27], but only recently has there been large scale adoption. RFID technology allows for the identification of objects by means of using electronic circuits within an RFID tag which may be attached to an object which transmit a unique code to an RFID reader. This technology is similar to that of using barcodes and a barcode scanner in its function [6], there are however some advantages over using traditional barcodes to identify an object; line of sight is not required to identify an object and tag detection range is up to 12 meters depending on the tag-reader system.

2.2.1 Types of RFID tags

In RFID the technology can be divided into two categories; Active and Passive. The active type of RFID tag is that which uses an external power source such as a battery to power its radio to communicate with an RFID reader. The passive RFID tag does not contain its own power source, it makes use of energy provided by the RFID reader when it is interrogated by means of electromagnetic induction [28]. The frequency ranges at which RFID tags and readers operate range from 128 kHz to 5.8 GHz, the active tags would typically operate at the 2.4 GHz frequency band while passive tags operate at under 100 MHz [27].

2.2.2 Types of RFID readers

RFID reader technology can be divided into three categories separated by the operating frequency band, with each being suited to different applications as radio waves perform differently based on the frequency. The frequency bands used by RFID readers are;

- Low frequency RFID
  The Low frequency RFID operates in the 30 kHz to 300 kHz with tag detection ranges of up to 10 cm [29].

- High frequency RFID
  The High frequency RFID is most commonly used in payment applica-

tions where a tag detection range of more than 10 cm is needed. The frequency band used is 3 MHz to 30 MHz [29].

- Ultra high frequency RFID
  The Ultra-high frequency RFID readers are currently the fastest growing of the three RFID frequency types, this is due to the lower cost of manufacturing of passive UHF RFID tags. The UHF type is the one most susceptible to interference, this is due to the high frequency band of 300 MHz to 3 GHz. Tag detection range is up to 12 metres [29].

2.2.3 RFID monitoring applications

In this section we will look at how RFID is being used in different industry applications to solve problems and design factors relating to each systems' RFID component.

2.2.3.1 Asset Management in a Hospital Setting

With the retail and industrial manufacturing sectors currently implementing RFID based systems for the purpose of tracking objects, the healthcare sector as such is next in receiving attention from researchers wishing to exploit RFID technology capabilities in healthcare. A pilot asset tracking system which used RFID was installed in a large 120 bed hospital. The purpose of the pilot project was to understand the limitations and benefits of RFID technology in order to improve patient safety by being able to track the location of mobile medical equipment such as infusion pumps, etc. The researchers selection requirements for the RFID component of the system were as follows;

- RFID tag size
  The size of the tag will determine what type of devices could be equipped with an RFID tag. The researchers were interested in tagging all mobile medical devices.

- Battery life
  Hospital equipment is replaced every seven years, the researchers wanted the battery of the tag to last at least seven years.

- Location resolution
  The resolution of the location data provided by the reader needed to

be able to accurately locate small items such as blood products as well as larger items such as infusion pumps.

- Tag density
  The system needed to be able to detect at least one hundred items in a small 5 metres by 5 metres room.

As mentioned there exists two options for an RFID system, either active or passive. With a passive reader configuration there would have to be a high number of readers deployed in order to maintain a high location resolution, this would be physically intrusive and expensive. The alternative active RFID system was selected as it would provide greater performance with fewer RFID readers.

2.2.3.2 Toyota SA manufacturing

In the automotive manufacturing industry the technology has become a welcomed addition to improving the efficiency in which vehicles are manufactured by being able to effectively track each vehicle during the post-production process. The need to effectively track vehicles is due to the high volume of vehicles produced (220 000 per year) at the Toyota facility in Durban, South Africa. This is accomplished with reusable highly robust RFID tags mounted to each vehicle during assembly, the RFID tag would then communicate with a reader at each station along the assembly line, updating the facilities vehicle tracking server with the vehicle position and the time spent at each station.

2.2.4 Discussion

The design requirements facing researchers in tracking assets using RFID were that they needed;

1. To be able to simultaneously detect approximately 100 items with the RFID reader
2. Track the locations of items
3. High spatial resolution

Of the two types of RFID technology which exist; active and passive, only active RFID tags would be able to meet these requirements, however active RFID tags require its own power source. This would affect the amount of maintenance and the financial cost in implementing.

2.3 RFID and WSN integration

The idea of integrating RFID technology with wireless sensor networks is brought on by the need to maximise the usefulness of the technologies as combined they allow for a new perspective on possible applications in the real world as opposed to academic research projects. The research done on integration is a relatively new field when compared to WSN and RFID, however techniques of integration can already be split into categories.

2.3.1 Integrating RFID tags with sensors

Within this category of integration techniques there are two possible methods in which the sensor tags communicate. The first is by having the sensor tags communicate with each other by forming an ad-hoc wireless network and relaying the sensed data and Identification code via the ad-hoc network to an RFID reader. Then there is the simpler configuration, which is the RFID tag equipped with a sensor and having the data only being communicated while it is being interrogated by an RFID reader.

2.3.2 Integrating Wireless sensor motes with RFID readers

An alternate approach to sensor tags is to incorporate RFID readers onto wireless sensor motes, by doing this we overcome the biggest problems of RFID technology; the passive nature of RFID, what this means is that one cannot actively search for the location of an RFID tag. A person would typically have to move the RFID reader to scan objects in order to find a specific one.

In the configuration of integrating an RFID reader to a wireless sensor mote the RFID tag detection range is effectively extended as the tags can be read from a remote location via the wireless sensor network.

Chapter 3

System design

3.1 Design requirements

The system that is to be designed would allow the users to be able to remotely detect objects that have been equipped with RFID tags and give an indication as to whether the environment the object is in, is suitable for its well-being. This will be achieved by developing an architecture using a combination of Wireless sensor network and RFID paradigms. As a proof of the concept the system will be applied to solving the problem of locating personnel in underground mines and detecting whether the underground atmosphere is safe for humans.

3.1.1 Functional requirements

The functional requirements refer to the functions that the system needs to be able to perform, they are listed below.

3.1.1.1 Sensing capabilities

Sensor capabilities required for the hybrid RFID platform are that it should be easily configurable so that the platform appeals to many different sensing applications, however for testing purposes the system would require the ability to detect the locations of mining personnel and the level of methane gas in the environment.

3.1.1.2 Sensor data dissemination

The sensor data which is collected by the various sensors is required by the operators of the system in order to make informed decisions on whether the conditions within the mine are safe, in order for the sensor data to reach those operators the data needs to be disseminated. This would be achieved through the use of a wireless sensor network and a local area network.

3.1.1.3 Scalability

The scalability of the system refers to the sensor networks ability to expand with an increase in the number of Hybrid RFID motes without having to reconfigure the network.

3.1.2 Performance requirements

The performance requirements are those quantitative requirements that the system has to meet in order to be considered a success.

3.1.2.1 Low power consumption

The power consumption for the hybrid motes and slave sensors need to be kept at a minimum, this is because in some situations access to the power grid will not always be available and the system would have to operate on batteries or be powered from renewable energy sources such as wind or solar. The hybrid mote would have to be able to operate for long periods of time using its battery, therefore it needs to be efficient in its energy use. As for design requirements the following parameters have been decided upon for the power consumption of the components of the system.

Hybrid RFID mote The hybrid RFID mote which incorporates the following;

- RFID reader
- Wireless transceiver
- Microcontroller

Should consume no more than 100 mA during its peak processing activity i.e. when wireless communication is in progress and RFID reader is communicating with nearby RFID tags. At 100 mA constant current consumption a standard Lithium-ion battery pack of 6000 mAh would only last

$$time = \frac{capacity}{current} \tag{3.1}$$

therefore

$$time = \frac{6000mAh}{100mA} \tag{3.2}$$

$$time = 60hours \tag{3.3}$$

This battery life time would not be suitable for the hybrid mote. A means to be more efficient in the hybrid motes' power consumption would need to be implemented. The simplest method to achieve low power consumption is to employ sleep functions which turn off components when they are not used.

Gateway device The gateway device would be installed in a location where grid power is not necessarily available, therefore a low power computing device would have to be selected. One that is capable of operating off a battery and solar power where it is available. The battery size would be limited to 200 mm by 100 mm by 100 mm as it is light enough to be easily carried, battery sizes of these dimensions range in capacity from 7.6 Ah to 15.2 Ah, using equation 3.4 with a typical current used by low powered computing platforms of 1.5 Amperes the battery life would be

$$time = \frac{15.2Ah}{1.5A} \tag{3.4}$$

$$time = 10.1hours \tag{3.5}$$

Which is sufficient to last through the night and then be recharged during the day using a solar panel.

As for solar power, the size of the solar panel is also limited due to the space available when installing the system. A solar panel of dimensions 750 mm by 500 mm is set as the maximum size. A panel of approximately this size can produce a maximum of 35 Watts using a Multicrystalline type solar panel.

The gateway device should be able to support the use of an operating system, preferably a distribution of UNIX. With support of modern networking protocols such as TCP/IP. The software requirements of the gateway device are the following:

- Database application.
  The database software would store the sensor data being generated from the sensor motes in the wireless sensor network.

- Webserver
  A Webserver would distribute the sensor data to clients remotely connecting to the gateway.

In terms of the hardware that is needed on the gateway device it would be:

- USB ports
  The USB ports for wireless communication peripherals (802.11b/g/n or 802.15.4)

- Ethernet port
  Ethernet port for local network and internet access to the gateway device.

- Non-volatile storage
  The sensor data stored within the database would require permanent storage.

3.1.2.2 Long range wireless communication

The distance between hybrid motes would be made as far as possible to increase the sensor networks coverage area. The limiting factor on the distance between motes is that of the wireless communications range. When selecting wireless communications hardware there is a trade-off between

- Power consumption
  Longer range needs more power to transmit due to more energy needed to over-come the losses caused by moisture in the air.

- Range
  Range depends on the frequency of the radio signal used and high frequencies are more susceptible to wireless interference.

- Data rate
  A high data rate requires a high transmit frequency, as more bits need to be transmitted per frequency cycle.

A low data rate of between 9600 and 115200 bits per second would be sufficient as the data that will be communicated between the hybrid motes will be short text data.

3.1.2.3 Object detection range

The RFID system that will be used to detect RFID tags needs to be able detect tags at a range of at least 2 meters, for this requirement an active RFID tag/reader system will need to be used.

3.1.2.4 Low cost

The cost requirement for the system should be that it is cheaper than existing wireless sensor network platforms which could potentially provide a similar service. The following is a list of the cost of existing wireless sensor network platforms;

- Waspmote
  The Waspmote wireless sensor mote is a product of the Libelium Corporation, it is a low powered sensor mote which can be equipped with many different sensors including RFID readers. The cost per Waspmote which includes the sensor mote and wireless transceiver is €163.00 [30].

- Mica2
  The Mica2 mote is a wireless sensor mote developed by Crossbow technologies, it is running the TinyOs real-time operating system which was developed by Berkeley University. The Mica2 also allows for sensors to be attached via the sensor board, but there are far fewer sensors which are supported compared to the Waspmote. The Mica2 mote costs $150.00 [31].

- Z1 Mote
  The Z1 mote from Zolertia has support for TinyOs as well as the Contiki operating system, the networking stack has support for 6LowPAN protocol which allow for the mote to communicate IPv6 packets. The cost for the Z1 mote is €95.00 [32].

3.1.2.5 Real-time data

The system should be able to provide information on the locations of the RFID tags as well as data on what environmental conditions the object finds itself in as close to real-time as possible. There are however limitations on how quickly a networked system can deliver data to a storage location, so as to appear as if the data is being received in real-time. There is an upper bound in the response time of the gateway in delivery content to a client requesting it of 1.0 seconds, which is the limit for a persons flow of thought to be uninterrupted without noticing [33].

3.1.3 Performance evaluation

Now that the system's requirements have been decided upon, the question of how the evaluation will be performed remains. What performance evaulation means for this system is, will the hybrid RFID mote meet the set requirements?
There exist many methods in order to evaluate the performance of a system, in this document we will apply two;

3.1.3.1 Prototype building

Developing a type of sensor mote, like in the case of the hybrid RFID mote requires a prototype to be built first. The prototype being the first of its kind is used to identify weaknesses in the design before the final design stage.

3.1.3.2 System Benchmarking

One of the most common approaches in system evaluation is system benchmarking, in benchmarking the requirements of a system are set beforehand and competing systems run tests in order to determine which one is the most suitable based on results of the tests. The tests which are usually run involve simulating the expected workload a system may encounter if it was to be deployed for the application. The problem with benchmarking is that the simulated workload can only be approximated as the actual workload a system may endure is unknown.

3.2 Design overview

The platform which provides object identification and environmental sensing would consist of both Hardware and software. The following figure 3.1 shows the interconnection of the hardware components which comprise the system.

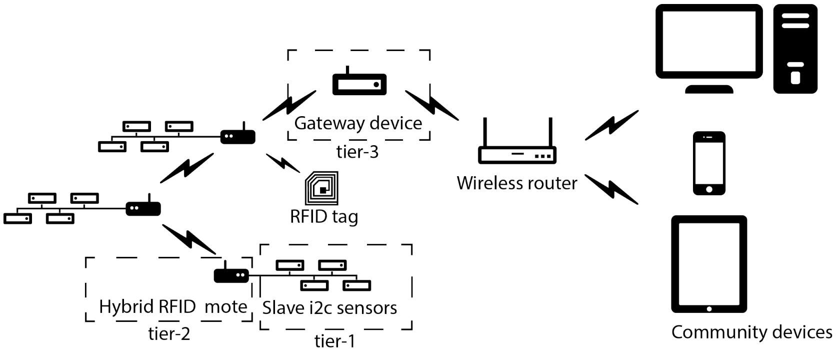


Figure 3.1: Communication between components.

We have selected the sensor mote/RFID reader technique of integrating the RFID functionality with the wireless communications capabilities of Wireless sensor networks. Using a tiered approach where;

tier-1 devices provide the environmental sensor data

tier-2 provides the network communications capabilities and object identification through RFID

tier-3 gateway devices disseminate the environmental and object locations to clients

3.2.1 Slave sensor

The lowest level of the system is the slave I2c sensors. The slave I2c sensor is designed so that it can be fitted with a large variety of electronic sensing devices. This is possible due to the large variety of inputs a microcontroller has i.e.

- Analogue input
  Input pins which are connected to an Analogue to Digital converter.

- Serial UART
  A serial data communication protocol.

- SPI
  The Serial Peripheral Interface is a bus interface used to connect many peripheral devices such as; sensors to a microcontroller.

Types of sensors;

- Gas

- Proximity

- Temperature

- Electrical current

The light weight and small size allows it to be deployed in hard to reach areas of an environment. The slave sensor communicates with the hybrid sensor mote by means of a cable using the I2c bus protocol, the use of a wired communications link between the hybrid RFID mote and the slave sensor frees up the radio spectrum for the hybrid motes to communicate with each other without much interference.

3.2.2 Hybrid mote

The next component of the system is the hybrid RFID mote, its function is to;

- Act as a controller of the many slave I2c sensors

- Detect the presence of objects equipped with RFID tags

- Relay the sensor data from slave sensors and the ID number of any RFID tag detected by its RFID reader

With the I2c bus configuration the hybrid mote is able to control up to 127 slave I2c sensors at a maximum cable length of approximately 100 meters [34].

3.2.3 Gateway device

The gateway is used to store the data being produced by the hybrid sensor mote and slave sensors in a database. The data is made available to other computing devices which are authorised to view it by means of a web service.

What follows is a detailed description of the design of the system as a whole and how individual components contribute to a working system.

3.3 Component selection

From the design we move to the selection of components which would make-up the Hybrid RFID sensing platform. Here we benchmark the selection of suitable components then select the best based on the design requirements. In figure 3.2 we present the types of individual components needed by each tier of the system.

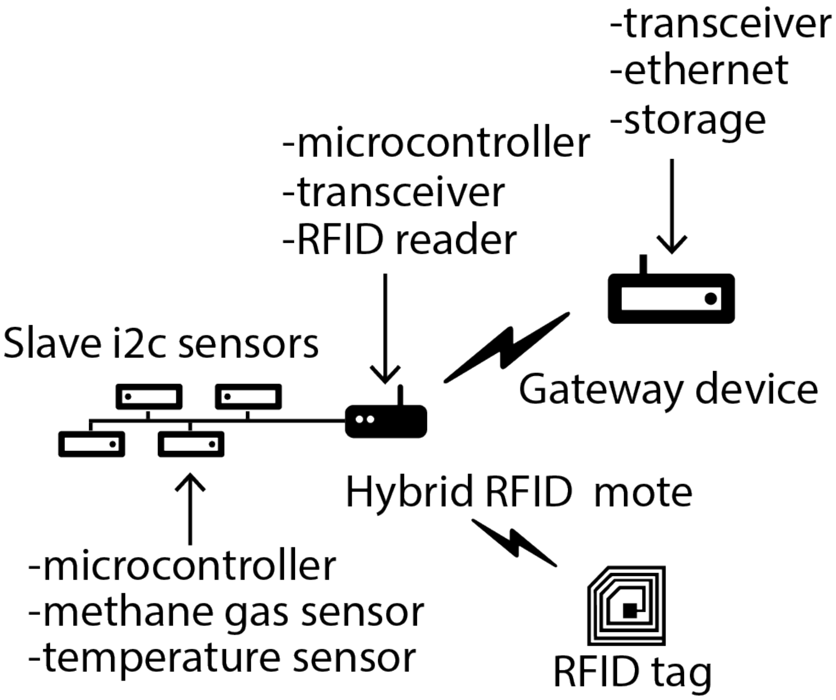


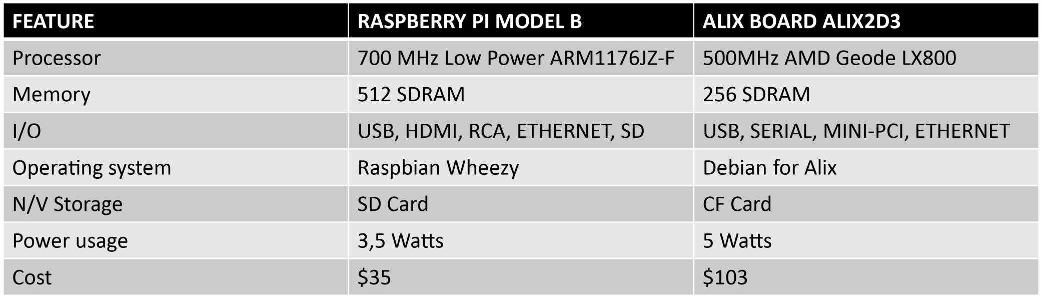
Figure 3.2: Hardware components needed by each tier

3.3.1 Gateway selection

3.3.1.1 Hardware options

The choice of a gateway device for the wireless sensor network was based on experimentation done on the Raspberry pi model B and the Pc Engines Alix board. Both of the devices are low powered computing boards, the Raspberry pi was developed by the Raspberry pi foundation for use as a teaching tool for school children to learn how to program [35] but has become a popular choice for researchers and hobbyists when it comes to a low powered embedded computing platform. The Alix board developed by Pc Engines (GmbH) is a small low powered embedded computer which was designed to be used as wireless routers, industrial user interfaces and firewalls [36]. Both of these devices run Linux operating systems [37], [38] which give them full IPv4/IPv6 networking capabilities. In table 3.1 the hardware features of each device are compared.

Table 3.1: Comparison between suitable gateway devices.



FEATURE	RASPBERRY PI MODEL B	ALIX BOARD ALIX2D3
Processor	700 MHz Low Power ARM1176JZ-F	500MHz AMD Geode LX800
Memory	512 SDRAM	256 SDRAM
I/O	USB, HDMI, RCA, ETHERNET, SD	USB, SERIAL, MINI-PCI, ETHERNET
Operating system	Raspbian Wheezy	Debian for Alix
N/V Storage	SD Card	CF Card
Power usage	3,5 Watts	5 Watts
Cost	$35	$103

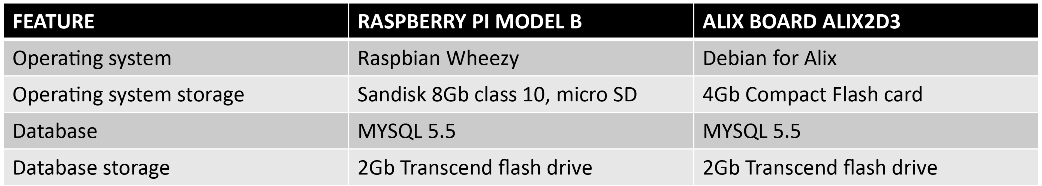
From the table it can be seen why the Raspberry pi is the most attractive low powered computing platform; its cost, processor, memory and power consumption are superior than the Alix board.

3.3.1.2 Experiment 1: Real-time data storage

The first experiment aims to determine which of the two devices are able to store sensor data in real time. This is necessary if the device is used as a gateway in a large wireless sensor network, each sensor mote within the network will be transmitting sensor data in to the gateway device. The gateway should be able to store all incoming sensor data within a database as it is received, so that the relevant I/O buffers do not become overflowed and sensor data gets lost.

Method of experimentation The experiment was conducted using a Raspberry pi model B and Alix board Alix2d3. Each device was running a variant of Linux operating system, below is the software configuration of each device.

Table 3.2: Device configuration for experimentation.



FEATURE	RASPBERRY PI MODEL B	ALIX BOARD ALIX2D3
Operating system	Raspbian Wheezy	Debian for Alix
Operating system storage	Sandisk 8Gb class 10, micro SD	4Gb Compact Flash card
Database	MYSQL 5.5	MYSQL 5.5
Database storage	2Gb Transcend flash drive	2Gb Transcend flash drive

Installation of the operating systems were achieved by following instructions from the manufacturers documentation. The database that was chosen to be used was the MySQL 5.5 this was due to its high performance when writing data to the database. Database installation was achieved by following instructions from the MySQL documentation [39].

The proposed experiment would comprise of each device writing large amounts of static data to the database while performance metrics such as CPU, memory and I/O were monitored. This however presented a challenge, due to the limitations of flash based memory devices, the limited number of write cycles available [40]. Writing the large amounts of data to the same device which stored the operating system would have significantly decreased its life span. An alternate to writing the data to the same device that stored the operating system had to be found.

It was decided that using flash drives as external storage was a good alternative as they are cheap and low powered compared to hard disk drives. The MySQL database needed to be configured to store the database on the external storage devices, this was a simple procedure that needed few commands. This is detailed below;

The storage device needed to be formatted and mounted to be usable by the operating system, this process is detailed in A.1 The next procedure was to create the database and tables where the simulated test data would be written to. This was achieved by using SQL statements, the commands are detailed in Appendix A.2

The next step was to write the code which would write to the database and measure the time and system resources which were used during the writing process. It was decided that Linux bash scripts would be used to perform the inserts into the database, this was chosen over using a compiled language such as C or JAVA due to the need for a database driver to communicate between the MySQL database and the compiled code. The use of a database driver would add unnecessary overhead and affect performance.

The bash script in A.4 consisted of MySQL INSERT statements which wrote data into the database, to get an accurate measure of system performance, 10 000 MySQL INSERTs where performed.

The time command on Linux was used to measure the time the script took to complete, the listing A.5 shows how the time command was used.

While the time command was executing the systems resources were being monitored using vmstat, in order to run vmstat at least one parameter needs to be specified. For this experiment vmstat was set to measure system resources every second and output the information to a text file using the command in listing A.6.

Results

CPU usage The experiment was run on both the Raspberry pi and the Alix board, the time it took for each device to perform the 10 000 SQL INSERT's statements were compared, graphs were then generated from the system performance.

From the graph in figure 3.3 which shows the CPU usage while performing the SQL INSERT statements, it can be seen that the ALIX board maintains a comparatively high CPU usage compared to the Raspberry pi while taking nearly half as much time as the Raspberry pi. The erratic CPU usage of the Raspberry pi was caused by the CPU having to wait for I/O operations to complete i.e writing of the data to the flash drive.

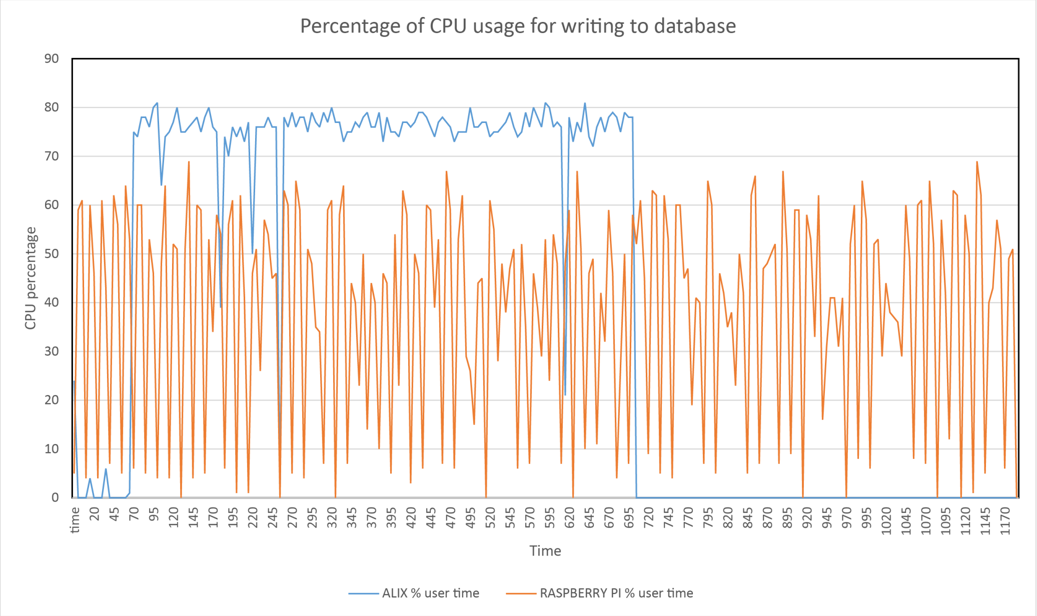


Figure 3.3: CPU usage of Raspberry pi and Alix board

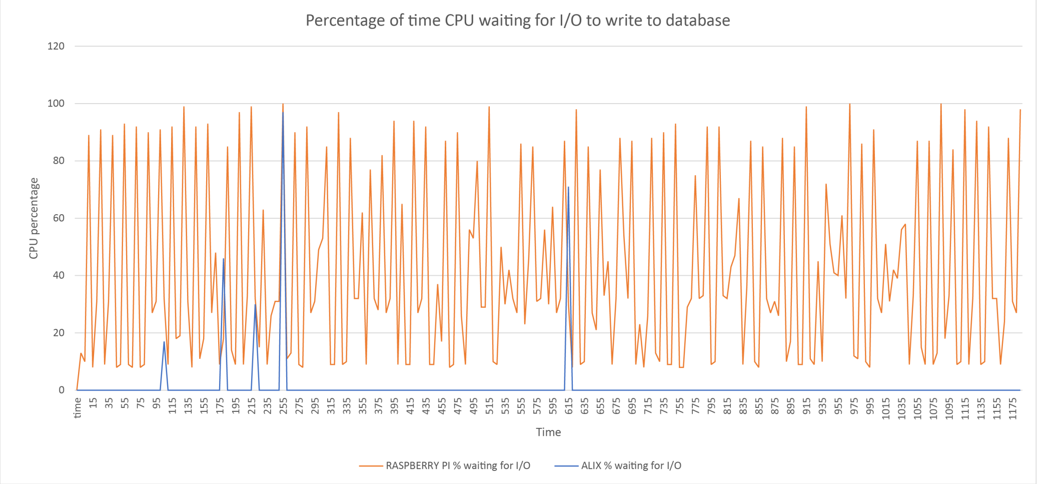


Figure 3.4: I/O wait time of Raspberry pi and Alix board

The assumption that I/O operations was the cause of the slow completion time when the Raspberry pi was performing the experiment was proven true when the vmstat output was analysed. The graph in figure 3.4 which shows the percentage of time which the CPU was waiting for I/O operations to complete indicate that the CPU was not being fully utilized for much of the time during the experiment.

This was a problem as the purpose of the experiment was to determine which platform is able to store data as close to real time with minimum delay. So a solution to the I/O bottleneck problem would be to use a storage device which has a higher Read/Write capability while maintaining the following;

- Low power consumption
- Small size
- Low cost

It was decided that a portable 2.5 inch USB hard disk drive would be the most suitable replacement for flash memory with the Raspberry Pi. The experiment was performed again, this time using a portable hard disk for database storage.

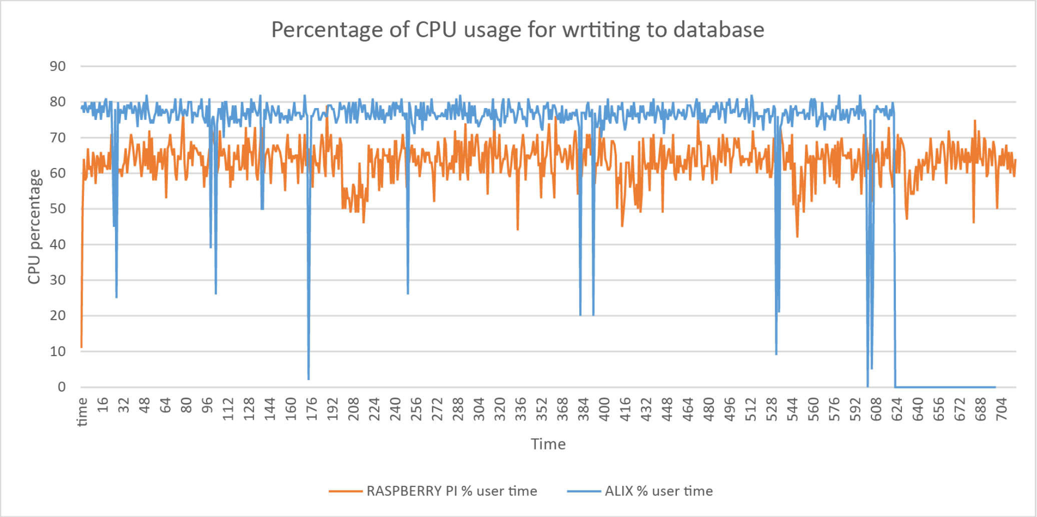


Figure 3.5: CPU usage of Raspberry pi and Alix board

From the graph in figure 3.5 it can be seen that the Raspberry pi's CPU maintained a high usage percentage while taking nearly half the time as compared to when flash memory was used as a storage device, therefore our assumption that the I/O operations were the cause of the poor performance in writing to the database was correct. Using the portable hard disk as the storage device for the Raspberry pi the I/O problem was mitigated.

Writes per second What is important in the results is determining the number of database writes per second the two platforms are capable of achieving. The calculation to determine the number is as follows;

$$writespersecond = \frac{writes}{time} \tag{3.6}$$

Raspberry pi

$$writespersecond = \frac{10000}{11 \times 60 + 58.137} = 13.92 \tag{3.7}$$

Alix board

$$writespersecond = \frac{10000}{11 \times 60 + 37.17} = 15.69 \tag{3.8}$$

The number of writes per second should not be exceeded by the hybrid motes in the sensor network as it is the maximum number the two platforms are capable of performing. The number will be used as metric for setting up the sampling frequency of the hybrid motes in the sensor network.

3.3.1.3 Experiment 2: Real-time data delivery

The last aspect in selecting a suitable gateway device is its ability to communicate the sensor data from the sensor network to clients wanting to access it. The simplest method in which clients can access the data is by using a web server, this is due to the fact that almost all computing devices today; personal computers, mobile phones and tablet computers have a web browser application.

Method of experimentation The aim of this experiment is to determine how the Raspberry pi and Alix board perform in delivering requested sensor data to clients under different web server load conditions. The performance measurement that will be made is;

- Number of web server requests per second vs. response time
  The maximum number of requests per second before the web server

crashes will give an indication of how many clients are able to be served with sensor data before the gateway is no longer able to provide the service. The average response time is the time it takes for a clients request to the server to be fulfilled. The maximum number of requests per second will be determined at which point the response time is unacceptable.

The raspbery pi was configured by installing apache2 webserver and PHP5. This was achieved by using the following commands in A.3.

The Alix board was configured with NGINX webserver instead of Apache2, due to the lack of support of certain dependencies required for the installation of Apache2.

To perform the experiment an application called httperf was used, it is a testing tool used to test performance of web servers which was developed by hewlett packard research labs [41]. The application httperf was installed on the two platforms as it would be performing its performance measurements on the web server as localhost, this was done so as to eliminate any effects caused by local networks router in preventing a high number of network connections to a host computer.

The experiment will focus on the connection rate/requests per second As the variable parameter in httperf. Connection rates of 20, 40, 80, 160, 320, 400 and additionally for the Alix board 650, 800, 1000 per second and as fast as possible will be measured in httperf. A test page was written in html which would be used as sample data which the client connections would access during the experiment. The page consisted of simple text which was 4kB and 16kB in size.

Httperf functions as follows;

Listing 3.1: Httperf usage

```
#!/bin/bash
httperf --server localhost --port 80 --num-conns 1000 --rate
    connectionRate
```

The application returns various diagnostic information about the target web server, a sample of the output from one of the runs is presented in B.1.

Of the information provided by httperf, what is relevant to the investigation is;

- connection rate
  The connection rate is the amount of connections per second attempting to retrieve data from the web server

- reply rate
  This is the rate at which the web server can respond to web requests

- reply time
  The reply time is the time in which the web server takes to serve the web page

The maximum connection rate per second that the web server can accept was determined by using httperf and specifying it to attempt to make connections to the web server at connection rates from 20 to 1000 connections per second. The upper bound at which the web server reached its maximum was noted when the connection rate specified in the httperf test and the connection rate reported in the httperf diagnostics report differed.

When the connection rate was too high for the web server the httperf diagnostics report indicated a lower value than what was specified in the test. Once the maximum number of connections per second were determined, the reply time was measured to determine the level of service ie, whether the web server responded with the requested page within an acceptable time.

Results The experiment was run on both the Raspberry pi and Alix board, the results were collected and are presented on graphs. The first part of the experiment aimed to determine the maximum number of connections per second that the web server can accept using the httperf diagnostics application. From the graph in figure 3.6 the connection rates per second at which the experiment was run is plotted to identify the upper bound. At about 730 connections per second the Alix board reaches its maximum and the Raspberry pi at about 190 connections per second.

While the upper bound has been determined the reply time was needed, the graph of how the reply time varies depending on connection rate is shown in figure 3.7. The reply time for both the Raspberry pi and Alix board is steady at 1 ms for connection rates up to the upper bound which was identified in the graph above. When approaching the upper bound the reply time increases sharply as the web server becomes overloaded with web page connection requests

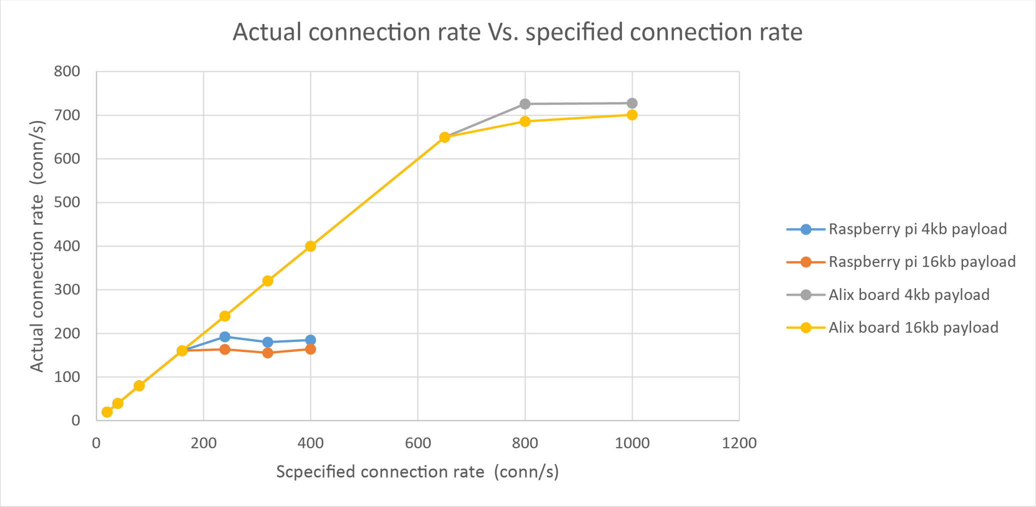


Figure 3.6: Upper bound of connection rate

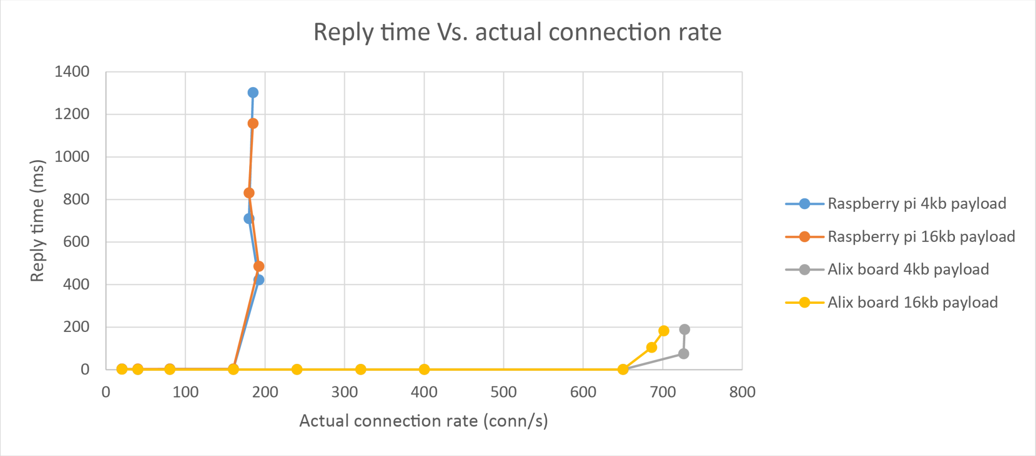


Figure 3.7: Web server reply time

3.3.1.4 Experimental conclusion

In conclusion we have seen the storage, processing and ability to deliver content of each of the two possible gateway devices. For data storage the two devices performed similarly with each having the ability to write 13 and 15 data strings to the data base per second, which is a sufficient amount for the system.

As for delivering the web pages to clients requesting the data it is expected that the amount of clients would not exceed more than 100. This is due to the gateway being on a local area network and closed off from external networks such as the internet.

Therefore taking into consideration the financial cost, power consumption, data storage/delivery aspects of each platform the Raspberry pi is selected as the most suitable choice of gateway device for the hybrid RFID platform.

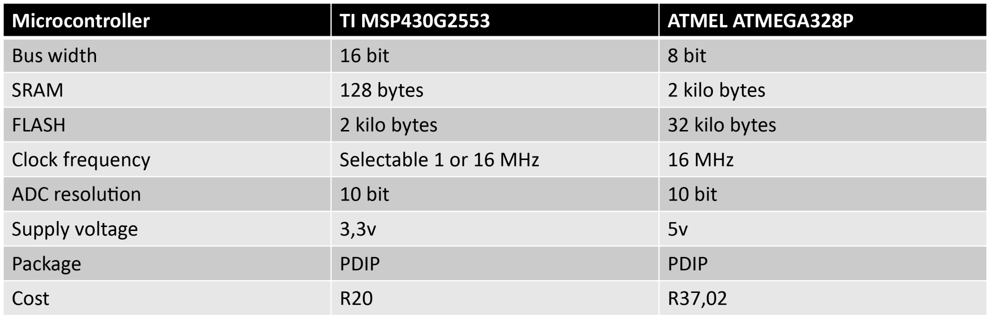
3.3.2 Microcontroller selection

3.3.2.1 Hardware options

In this section the possible hardware components for the design of the slave sensor are listed, each one is then quantitatively evaluated to determine the most suitable combination of hardware which meets the design requirements as stated in the previous section. The following microcontrollers were identified as possible options for use in the slave sensor, the initial choice of microcontrollers were based on;

- Availability within South Africa
- Low cost
- Developer support of software libraries

Table 3.3: Comparison between suitable microcontrollers.



Microcontroller	TI MSP430G2553	ATMEL ATMEGA328P
Bus width	16 bit	8 bit
SRAM	128 bytes	2 kilo bytes
FLASH	2 kilo bytes	32 kilo bytes
Clock frequency	Selectable 1 or 16 MHz	16 MHz
ADC resolution	10 bit	10 bit
Supply voltage	3,3v	5v
Package	PDIP	PDIP
Cost	R20	R37,02

From the data sheets of each of the two microcontrollers, it was then determined whether the microcontrollers supported the following Input-Output ports needed for the slave sensor;

I2C is a bus protocol used to interface many devices with a single bus connection.

Analog-to-digital converter is a device used to convert analog voltage readings from a sensor to a digital form which can be understood by a microcontroller.

Universal asynchronous receiver/transmitter is a serial communication protocol used to serially receive/transmit data on a single wire.

Both microcontrollers come in the PDIP package which allow them to be easily integrated onto a PCB and reprogrammed by using development boards, in the case of the ATMEGA328P the Arduino UNO board is used and for the MSP430G2553 the Texas Instruments Launch Pad can be used. The development environment for both microcontrollers are open source when using;

- Arduino editor for Arduino UNO
- Energia for the Texas Instruments Launch pad

The fact that the tools are open source helps keep the cost of the development of the system down as some other microcontroller design companies would charge a license fee for use of their proprietary compilers.

3.3.2.2 Experimental outline

Two experiments were conducted to determine the suitability of each of the microcontrollers as a slave sensor. The experiments aimed to determine the following about each microcontroller;

- The processing capability of each microcontroller
  By measuring the length of time it takes for the microcontroller to execute code which encrypts data with the AES-256 algorithm
- The power consumed by the microcontroller
  While the microcontroller is actively executing code

3.3.2.3 Experiment 3: Processing capability

The time taken for ATMEGA328P and MSP430G2553 to encrypt 16 bytes of sensor data

Method of experimentation The experiment was conducted with the two microcontrollers and their associated programming boards, the software implementation of the AES-256 encryption algorithm was written in C programming language by [42]. The implementation was portable and functioned correctly on both microcontrollers when using avr-gcc compiler for ATMEGA328P and mspgcc for MSP430G2553.

The challenge in design of the experiment was to measure the time the encryption block of code took to execute while not adding additional overhead, options included;

- Outputting messages from the microcontroller via a serial connection to be read by a PC
- Using digital output pins

The simplest option was to use digital clock pulses to measure the time between the rising and falling edge of a digital pulse, before the encryption code executes a digital pin is set to HIGH and when it is complete the pin is set to LOW. To be able to measure the rising and falling pulse edge an oscilloscope is needed as an accurate execution time is needed. The code was structured as follows in figure 3.8;

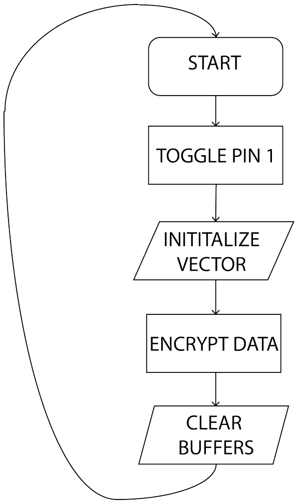


Figure 3.8: Test code structure.

The time was measured by using a digital oscilloscope which was connected to a digital output pin on the microcontroller, when the output of pin 1 switched from;

- 5 V HIGH in the case of the ATMEGA328P
- 3.3 V HIGH for the MSP430G2553

to 0 V LOW the difference between the two pulse edges is the time the encryption block of code takes to execute.

Results The edges where the output switches HIGH/LOW were clearly visible on the oscilloscopes LCD. For analysis of the waveform produced by digital output pin 1, the waveform data was transferred from the digital oscilloscope to a PC. The format of the data from the oscilloscope was in a comma-separated file, the file contained the graph of the output of pin 1.

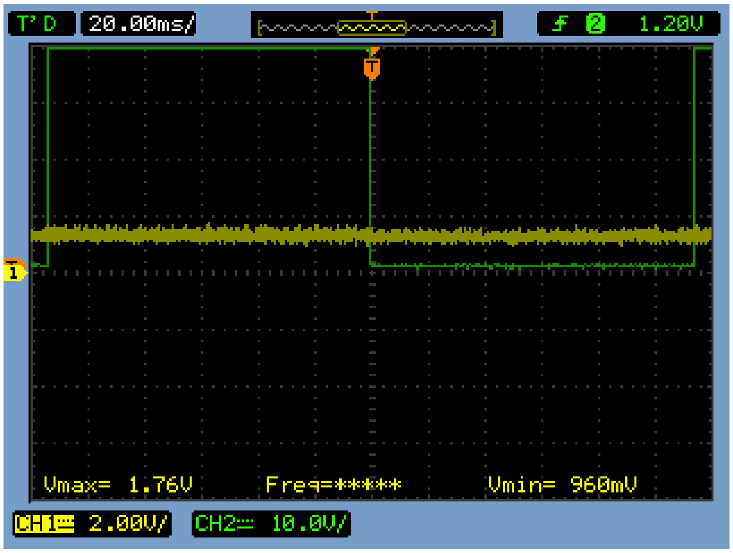


Figure 3.9: Output trace from oscilloscope.

The Channel 2 trace (in green) is the output of the pin which signals when the encryption code begins executing and completes.

The time results from the graph in figure 3.10 were gathered from runs of the experiment. From the results it's seen that the ATMEGA328P at 16 MHz has similar performance to MSP430G2553 when its clock frequency is set to 16 MHz.

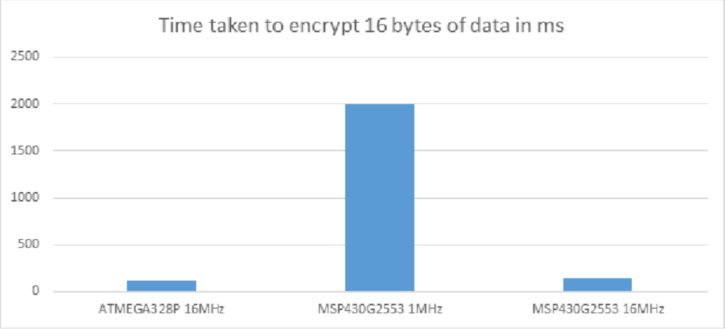


Figure 3.10: Comparison of run times.

When considering code execution time for AES-256 bit encryption a higher clock frequency is more important compared to memory capacity and bus width, this was noticed due to the large execution time difference when the MSP430G2553 operates at 1 MHz versus 16 MHz. The time taken at 16 MHz clock frequency is within the design requirements for the ATMEGA328P and MSP430G2553.

3.3.2.4 Experiment 4: Power consumption

Method of experimentation The power consumption of a system can be measured in many ways using different hardware, depending on the type of hardware being analysed and the power consumed by the hardware. There are three possible ways to measure the electrical current being consumed by a system, then calculate the power. From the simplest to implement they are;

- Digital multimeter with Amperemeter and Voltmeter function
- Shunt resistor
- Clamp current probe

The method that was used to measure current consumption in the following experiments was the shunt resistor method, it was chosen because of its simplicity to implement, low cost and availability of equipment needed to use this method. A brief explanation of each method follows;

Digital multimeter A digital multimeter is a common piece of equipment found in electronics laboratories, in the Universities electronics departments laboratory access to this equipment was not an issue. The multimeter has the function to measure current flowing in a circuit by placing the leads of the multimeter in series with the system under test and voltage by placing the leads in parallel, the current and voltage is then simply read off the display of the multimeter. The multimeter is however not suitable for the measuring of the current used by the system in the experiments, this is because of the low sample rate and resolution of a multimeter in measuring voltage and current. The measurement of current or voltage provided by the multimeter is not reliable when testing systems where the current or voltage fluctuates many times per second. Therefore multimeter only provides reliable measurements where a circuits current and voltage is steady.

Clamp probe A current clamp meter is an electrical testing equipment used to measure the current flowing through a wire. The clamp probe works by connect the instruments jaws to a current carrying wire, the current is coupled into the instruments measurement circuitry which consists of a primary and secondary wire coil. The use of a primary and secondary coil in a clamp probe allows it to measure very large currents, in the order of 1000 Amperes. However the sensitivity of the instrument in measuring low DC current is poor, this is due to the input-output ratio of the measurement coils. The cost of a clamp probe was also prohibitively expensive and therefore was not available in the laboratory.

Shunt resistor The shunt resistor technique is the most common approach engineers use when measuring the current draw characteristics of a circuit. A shunt resistor is a high precision resistor typically less than 10 ohms which is placed in series with the circuit being tested, the shunt resistor linearly converts the current flowing through it into a voltage which can be measured by an oscilloscope. The downside of using this technique is that the resistor can interfere with the measured circuit as it consumes some power as it converts electrical energy into heat. There would be an error in the measured voltage as a result of the energy conversion.

For this experiment to measure the current consumption of the circuit the shunt resistor technique was used. This was due to its simplicity in implementing and the necessary measuring equipment was available for use in the laboratory.

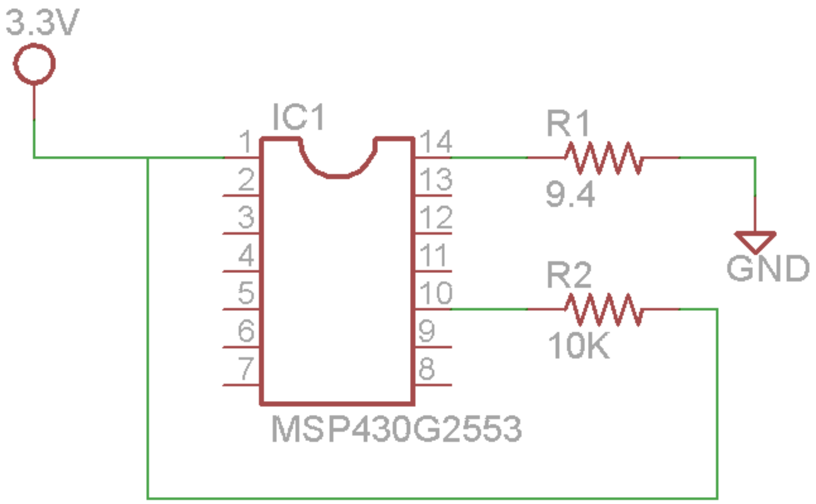


Figure 3.11: MPS430G2553 test circuit with shunt resistor.

The MSP430G2553 used a shunt resistor (R1) connected between pin 14 and GROUND, the resistance was 9.4 ohm. By using this configuration only an oscilloscope is needed to be able to measure the voltage then in turn calculate the current and power consumption.

The voltage across resistor R1 is measured using the oscilloscope. The current passing R1 is calculated using ohms law

The formula

$$v = i \times r \tag{3.9}$$

Making i (current, the subject of the formula) with the resistance R1

$$i = \frac{v}{9.4} \tag{3.10}$$

Then the power consumed by the microcontroller is

$$p = i \times 3.3 \tag{3.11}$$

The same procedure is followed when calculating power consumption for the ATMEGA328P. The circuit used to measure the current follows in figure 3.12.

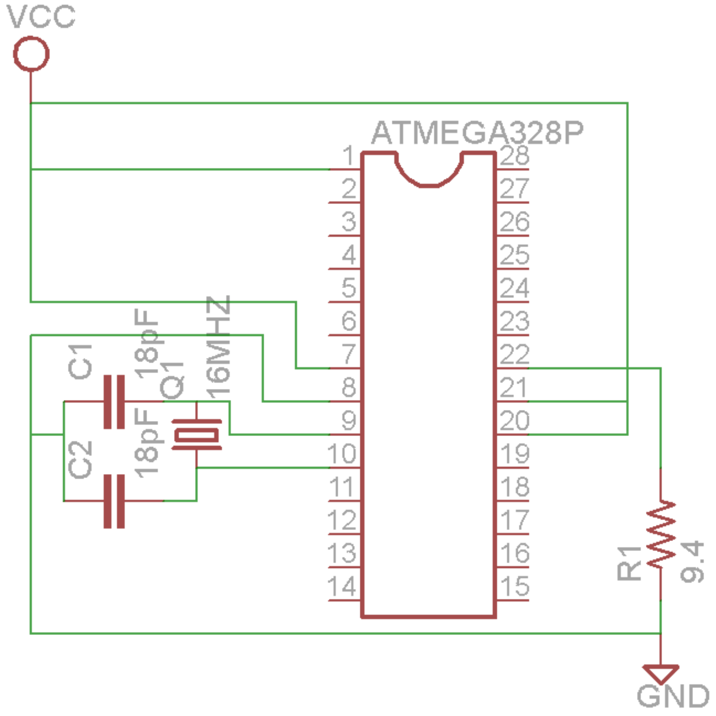


Figure 3.12: ATMEGA328P test circuit with shunt resistor.

Results The results indicate that the ATMEGA328P consumes much more current than the MSP430G2553.

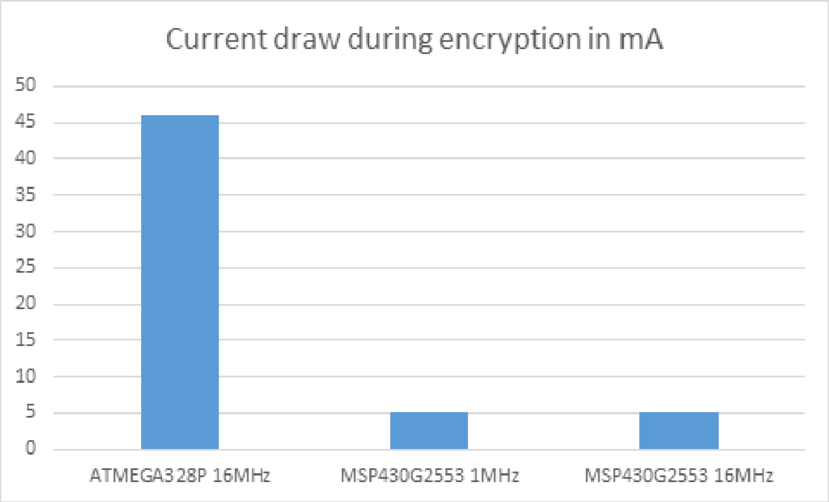


Figure 3.13: Comparison of the current draw for the microcontrollers.

The resulting power consumption of each microcontroller is calculated as;

- MSP430G2553
  16.5mW

- ATMEGA328P
  235mW

3.3.2.5 Experimental conclusion

The aim of the experiment was to determine the most suitable microcontrollers for use in the slave I2C sensors. The short-listed microcontrollers were chosen based on availability within South Africa, price, documentation and low power consumption.

From the results presented in determining the power consumption of the microcontrollers it was found that the MSP430G2553 consumed about fourteen times less power than that of the ATMEGA328P while performing the AES-256 encryption operation which would be the most resource intensive task that the I2C slave sensor would perform.

The result of the time taken to encrypt 16 bytes of data using the AES-256 encryption algorithm indicated that either of the two microcontrollers had sufficient computational power to complete the encryption task within the required time.

Therefore the decision on which microcontroller to use for the slave I2C sensor was chosen based on the low power consumption and financial cost. The hybrid RFID motes' microcontroller was then chosen to be the ATMEGA328P due to the power consumption being within acceptable limits, the large amount of peripherals which can be easily interfaced and the simplicity of writing the firmware for the platform.

3.3.3 Transceiver selection

The choice of radio for the hybrid mote is one of the most important design decisions as the radio has a large effect on the systems usability due to factors such as communication range, power consumption and wireless interference tolerance. Fortunately there exist many different types of radios supporting different wireless communication protocols, some of them include Bluetooth, 802.11, 802.15.4, Zigbee and 6LowPan.

3.3.3.1 Hardware options

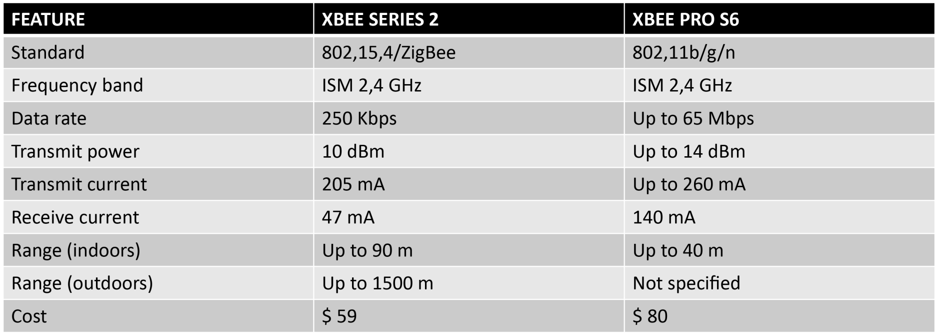
In determining which radio and communication protocol is most suitable for the hybrid mote experiments on each possible option will be performed. As per the design requirements mentioned previously for the wireless communication of the hybrid mote, the following options for the radio were selected.

- Digi XBee series 2
- Digi XBee PRO 6B

In table 3.4 is a list of the possible options. Of the possible communications protocols options that were initially suggested, Bluetooth was not selected due to its lack of routing capabilities within the protocol and the short range compared with other protocols. 6LowPan allows for the use of IPv6 addressing which in turn would give the hybrid motes the ability to communicate directly with traditional computer networks which support

IPv6, however for the purpose of the mine monitoring system there would be no need for each hybrid mote to be able to communicate directly with a computer on an IPv6 network. The two options above are both products of the Digi Corporation, they have the same I/O footprint which allows them to be used interchangeably with other hardware. This is ideal for the experiment which will be performed as no modifications will have to be made when swapping between the series 2 and PRO 6B.

Table 3.4: Comparision between suitable transceivers.



FEATURE	XBEE SERIES 2	XBEE PRO S6
Standard	802,15,4/ZigBee	802,11b/g/n
Frequency band	ISM 2,4 GHz	ISM 2,4 GHz
Data rate	250 Kbps	Up to 65 Mbps
Transmit power	10 dBm	Up to 14 dBm
Transmit current	205 mA	Up to 260 mA
Receive current	47 mA	140 mA
Range (indoors)	Up to 90 m	Up to 40 m
Range (outdoors)	Up to 1500 m	Not specified
Cost	$ 59	$ 80

3.3.3.2 Experiment 5: Range and Packet loss

The experiment aims to determine the range and packet loss of the XBee 802.15.4 and 802.11g radios for indoor and outdoor applications, this information is highly important as it range affects the type of network topology, location of sensor mote within the network and power consumption of each sensor mote.

Method of experimentation The experiment comprised of two performance measurements, the two being.

- Range vs. the Received Signal Strength (dBm)
  The Received Signal Strength is the measurement of the power in a radio signal, the combination of Received signal strength and the range between the transmitter and receiver will give an indication of the maximum range.

- Packet loss vs. Range
  With the maximum range it is also useful to know how stable the wireless link is at different distances between the transmitter and receiver.

The packet loss, which is the percentage of data packets sent by the transmitter, which are not received by the receiving radio. Will give an indication of the reliability of wireless link at the different distances.

Each of these measurements would be taken within an indoor and outdoor environment, the outdoor environment where the experiment would be conducted would be an open field with clear fresnel zone between the transmitting radio and the receiving radio with no interference from 2.4 GHz wireless sources such as Wifi. The fresnel zone and height at which the transmitter and receiver needed to be placed was calculated to be 2 meters using an online fresnel zone calculator.

The indoor section of the experiment would be conducted in a building with a large corridor.For both the indoor and outdoor experiments data packets of 64 bytes were used as test data where the transmitter was sending packets at a rate of 1 packet per second for 100 packets. Three trials were performed at distances of 100m, 50m, 20m and 5m for outdoors and at 20m and 5m for indoors. An application was written in JAVA using the Xbee api version 0.9 which listened for packets on the receiving Xbee, the application recorded the following information from each packet;

- Received signal strength indicator
  The RSSI of the received packet

- Remote address
  The network address of the sending Xbee

- Sent data
  The data contained within the packet

On the laptop that was sending the test data packets an application called X-CTU was used, it is also the application which is able to reprogram the firmware of the Xbee's. For the purpose of the experiment the two firmwares which were used by the Xbee's were;

- XB24-ZB:Zigbee coordinator API version 21A7
  The receiving Xbee was loaded with the coordinator firmware.

- XB24-ZB:Zigbee router API version 23A7
  The transmitting Xbee was loaded with the router firmware

The two different firmwares are necessary because of the way in which Zigbee networks operate. A Zigbee network needs to have a coordinator device, its responsibility is to create the network and assign addresses to nodes. The router is a device which can send, receive information as well as route traffic to specific nodes. Together the two devices form a simple Zigbee network.

The API mode was used by the Xbee's, it is a frame based method in which the computer can communicate with the Xbee via its serial UART port. The API mode allows access to the following functions;

- Change parameters of the Xbee without having to enter command mode
- View the RSSI and source address of a packet

Results The experiment were carried out with the following results being obtained. In measuring the RSS vs. transmission range for 802.11g and 802.15.4 the following graph was produced;

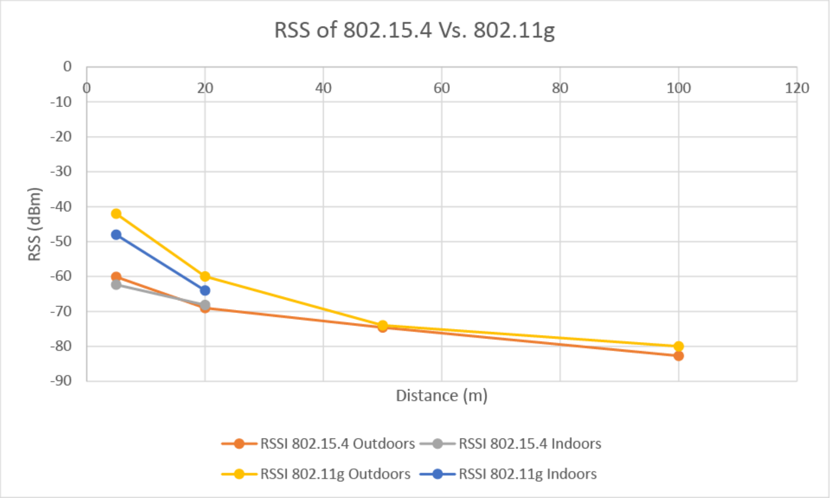


Figure 3.14: RSS vs. Distance.

3.3.3.3 Experimental conclusion

In conclusion it was seen from the results that in the indoor and outdoor scenarios the 802.11g XBee outperformed the 802.15.4 XBee in terms of the signal strength. This is due in part to the higher transmit power of the 802.11g transceiver, which as a result provides a further point to point range. The packet error rate between the two XBee's were negligible at the ranges that were tested, as a result it was not a factor in selecting the appropriate transceiver.

The 802.15.4 XBee radios would be simpler than the 802.11 XBee to implement due to them not needing additional infrastructure such as 802.11 wireless routers to support the network, as the 802.11 XBees do not support the ability to route traffic.

In terms of the financial cost and power consumption the 802.15.4 radios are superior, therefore the transceiver which will be used by the Hybrid mote and gateway will be the 802.15.4 XBee series 2.

Chapter 4

System implementation

4.1 Overview

This chapter introduces the hardware configuration of each of the components that form the hybrid RFID platform. The software components which provide the functionality are also presented.
Based on the benchmarking evaluation done in chapter 3 we have the hardware selection for each component shown below.

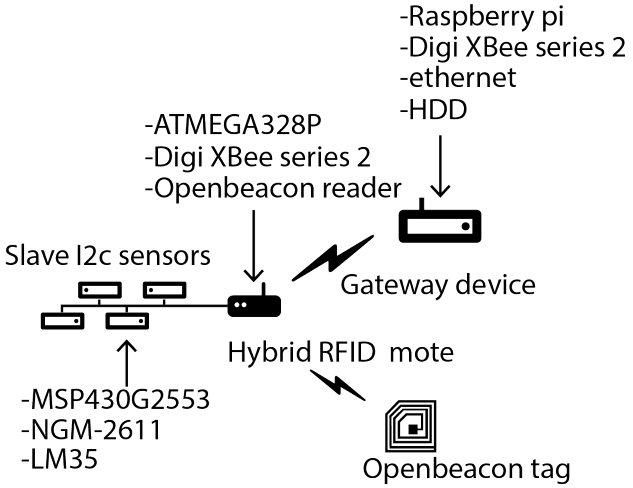


Figure 4.1: Hardware configuration for the Hybrid RFID platform

4.2 Hybrid RFID mote components

The Hybrid RFID mote is based on the open source Arduino Uno platform, the Arduino Uno uses the Atmel ATMEGA328P which is an 8 bit microcontroller. The following diagram shows the various hardware and software components which make up the hybrid RFID mote.

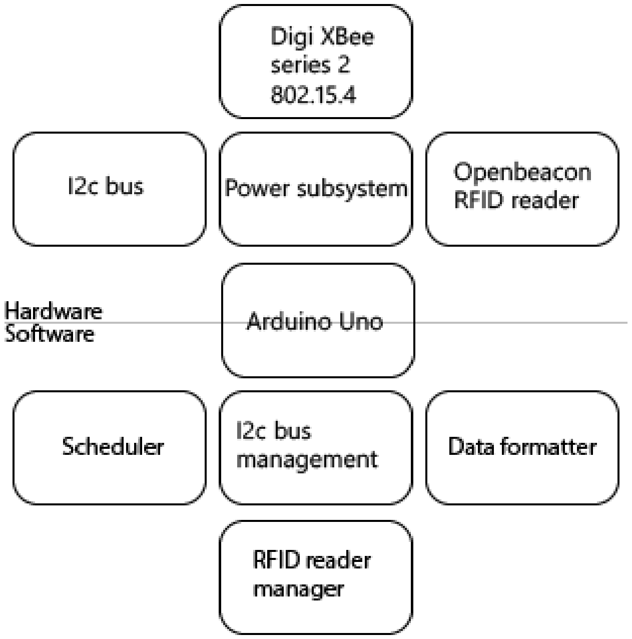


Figure 4.2: The hardware and software components which comprise the Hybrid mote.

4.2.1 Arduino Uno

The Arduino is an open-source prototyping platform for developing low powered and low cost embedded electronics projects. The creators of the platform have made the hardware designs of the Arduino available under the open-source license for anyone to build their own or modify the designs for their own purpose.

The ease of use in rapidly prototyping an embedded computing application, the amount of support giving by the large community of developers/designers in the form of software libraries and technical help was also a contributing factor. In terms of technical specifications; the latest revision of the Arduino Uno, the R3

The Arduino Uno allows peripherals to be connect via

- SPI
- I2C
- UART
- Logic HIGH/LOW

The wide variety of interfaces allows many different types of hardware to interface with the Hybrid mote, so as to not be constrained by a particular device manufacturers choice of hardware interface.

4.2.1.1 I2C bus interface

The I2c bus interface is the data bus on which the many slave sensors connect to the Hybrid RFID mote. The configuration used in the system is a master-slave setup where the Hybrid mote is the master and the slave sensors are in slave mode. When a device is setup in master mode it controls which of the slaves can communicate on the bus, only one device may communicate on the bus at any one time. The master device achieves this by requesting data from a particular slave using its address, the address space used by the slaves are 8 bits. The slaves would listen on the bus for their address followed by the requested number of bits, the slave would then respond with the data. The circuit used for the I2c bus is comprised of two 10 kOhm pull-up resistors which pull the SDA (data) and SCL (clock) lines up to the 3 V supply. All the slave devices would be using a supply voltage of 3 volts due to the MSP430G2553 using 3 volt logic. The SDA line is used to transfer the data serially while the SCL line is used to synchronize the slave and master by means of clock pulses. Figure 4.3 shows the I2C bus circuit.

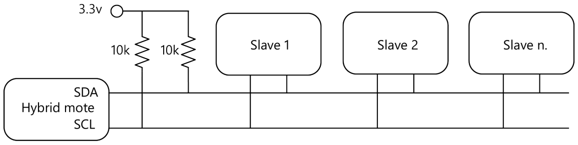


Figure 4.3: I2C bus circuit.

4.2.1.2 I2C bus management

The I2C bus manager is responsible for controlling which devices on the I2C bus are allowed to communicate with the master at a time, a communication protocol was developed in order to achieve the following functions when communicating with the slave I2C sensors.

- Request the number of sensors a slave is equipped with
- Request the encrypted sensor reading

The protocol in which the hybrid mote communicates with the I2C slave occurs is laid out below

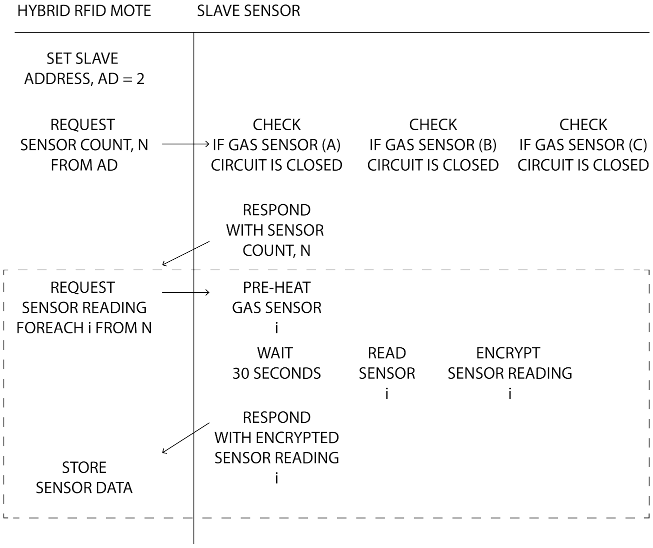

HYBRID RFID MOTE
SLAVE SENSOR
SET SLAVE ADDRESS, AD = 2
REQUEST SENSOR COUNT, N FROM AD
CHECK IF GAS SENSOR (A) CIRCUIT IS CLOSED
CHECK IF GAS SENSOR (B) CIRCUIT IS CLOSED
CHECK IF GAS SENSOR (C) CIRCUIT IS CLOSED
RESPOND WITH SENSOR COUNT, N
REQUEST SENSOR READING FOREACH i FROM N
PRE-HEAT GAS SENSOR i
WAIT 30 SECONDS
READ SENSOR i
ENCRYPT SENSOR READING i
RESPOND WITH ENCRYPTED SENSOR READING i
STORE SENSOR DATA

Figure 4.4: Hybrid mote - Slave sensor communication protocol.

4.2.1.3 Scheduler

The scheduler managers all the functions of the hybrid RFID mote and when each function is meant to occur and its frequency.

Slave sensor interrogation is the process of communicating with each of the connected slave I2C devices and requesting sensor data from it.

RFID tag detection is the function which manages the serial data stream that is being received from the OpenBeacon RFID reader.

Wireless communication function controls the XBee transceiver, it formats the data to be sent within the packet as a UNICODE string.

Operating mode is when the hybrid RFID mote switches between sleeping and active modes.

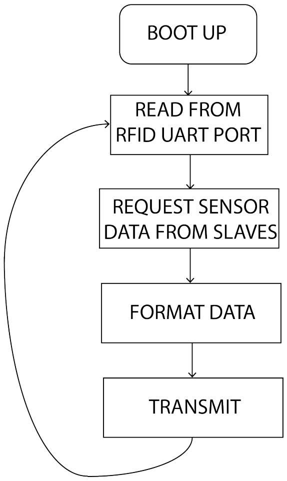


Figure 4.5: Scheduler operations.

4.2.1.4 Data formatter

The data formatter function is responsible for the preparation of sensor data from the slave devices and the list of nearby RFID tags that have been detected by the RFID reader for transmission over the network to the gateway.

The formatter is limited in the size of data that can be communicated at a time by the MTU of the Xbee radio, which is set to 90 bytes. So special attention has to be made to keep the transmission of data under 90 bytes,the type of data that is transmitted is simply text. The format is as follows;

> *moteID detectedTagCount tagIDs slaveID sensorCount encryptedSensorData*

This string of characters is itself encrypted during transmission as the Xbee radio includes circuitry which allows for the data packets to be encrypted using AES-256, key management of the encryption algorithm is handled by the firmware of the Xbee radio.

4.2.1.5 RFID reader manager

The RFID reader managers function is to control the flow of data from the Openbeacon RFID reader, the Openbeacon RFID reader is setup to continuously output the Identification number of any tag that has been detected in the format:

> *RF: tagID*

The UART port on which the Openbeacon is connected to is continuously polled to check if there is any data within the UART buffer, the state chart for the RFID manager is below.

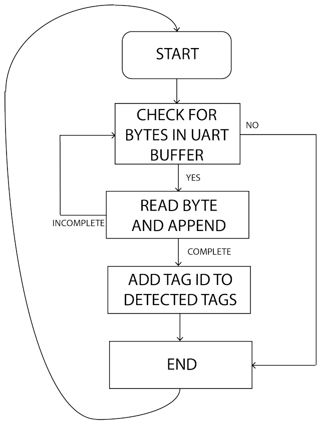


Figure 4.6: RFID tag detection algorithm.

4.2.2 OpenBeacon RFID reader

The OpenBeacon RFID reader is a small USB device, for the purpose in which it will be used in the hybrid RFID mote a method of interfacing the Arduino and OpenBeacon needs to be made as the Arduino Uno does not have a USB host controller for communicating via USB. On the OpenBeacon there is however an auxiliary serial port which can be repurposed to output the Identification number of any detected RFID tag. To use the auxiliary serial port a pin header was soldered to expose the following outputs;

- Serial DRXD
  Serial receive pin
- Serial DTXD
  Serial transmit pin
- GND
  Circuit ground

For the purpose of interfacing the OpenBeacon RFID reader with the Arduino Uno, the UART port was used. The UART is a two wire serial interface used for low rate data transfer. The two serial pins Tx (transmit) and Rx(receive) on the Arduino can be assigned to any two digital pins on the digital port header, in this case the first two pins; pin0 and pin1 were used. The figure below indicates how the Arduino Uno is connected to the OpenBeacon RFID reader.

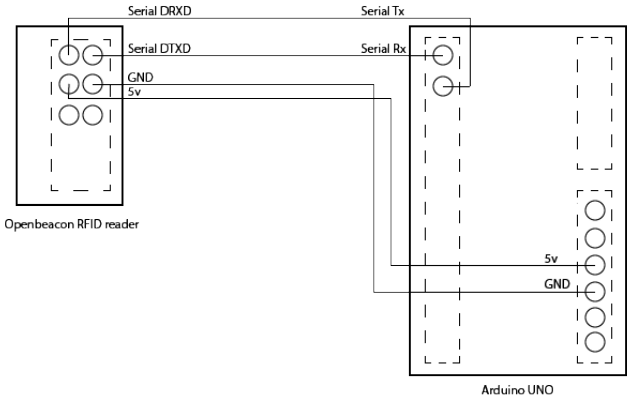


Figure 4.7: Interface between Arduino UNO and openBeacon RFID reader.

4.3 Gateway components

The gateway device that is used for the system is based on the Raspberry pi model B, the raspberry pi is connected to the system as shown in figure**??** on page**??**. The gateways hardware components include the following;

- **Digi XBee 802.15.4**
  This is the wireless transceiver which is connected via USB
- **USB hard drive**
  The hard drive is the device on which the systems sensor database is stored
- **Router**
  The router provides the gateway with access to the local area network

The Raspberry pi is running software which manages all the aspects of storing and disseminating the sensor information, such software is typically called the middleware in wireless sensor networks. The components which comprise the systems middleware is shown in the figure below.

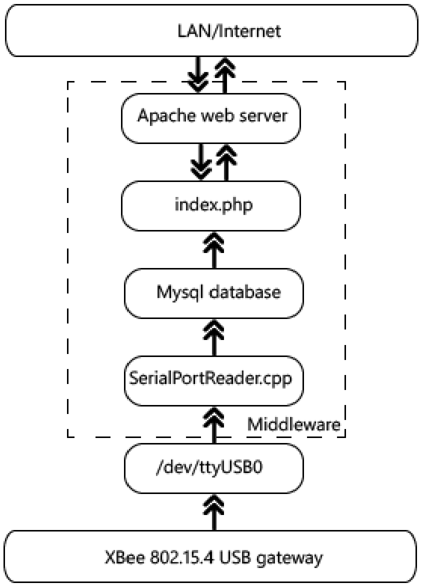


Figure 4.8: Gateway middleware

4.3.1 Serial reader

The serial reader applications function is to read data from the XBee via the USB-serial adapter, parse the data and then write that sensor data to a local database. The algorithm continuously polls the USB serial port to check if there is data waiting in the buffer to be read.

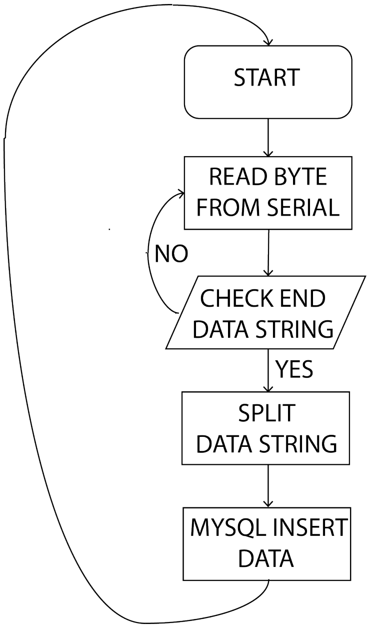


Figure 4.9: Serial reader algorirthm

After splitting and parsing the data string, the data objects are inserted into the tables of the MySQL database.

4.3.2 MySQL database

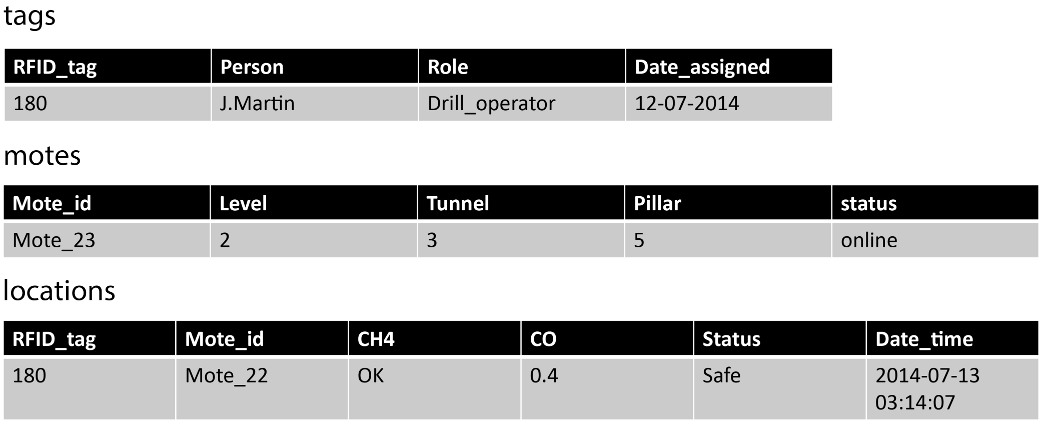
As mentioned previously the database that was chosen to be used for the system is the MySQL 5.5 database. It comprises 3 tables;

- tags
  The purpose of this table is to manage which RFID tags are assigned what miners. The table is only updated when a miner is assigned or is unassigned a new RFID tag.

- motes
  This table managers the location of the hybrid mote within the mine tunnels, as the tunnel system is expanded new hybrid motes will be added to the network. That is when this table will become updated.

- locations
  The locations table is where the methane gas concentration level and location of detected RFID tags are stored, each time a hybrid mote detects a tag it will send the tag ID along with the methane gas concentration level to the gateway. Where the locations table will be updated.

The tables' columns are shown in table 4.10;



tags

RFID_tag	Person	Role	Date_assigned
180	J.Martin	Drill_operator	12-07-2014

motes

Mote_id	Level	Tunnel	Pillar	status
Mote_23	2	3	5	online

locations

RFID_tag	Mote_id	CH4	CO	Status	Date_time
180	Mote_22	OK	0.4	Safe	2014-07-13 03:14:07

Figure 4.10: Database tables

4.4 Slave sensor components

The slave device has the advantage of being small, cheap and configurable. Due to its size the slave devices can be placed along the tunnel walls in large numbers and provide a high resolution data of the condition of the atmosphere within the mine. The interconnection between the hardware components of the slave sensor is shown in figure 4.11, with the header port JP1 where the NGM-2611 gas sensor is inserted.

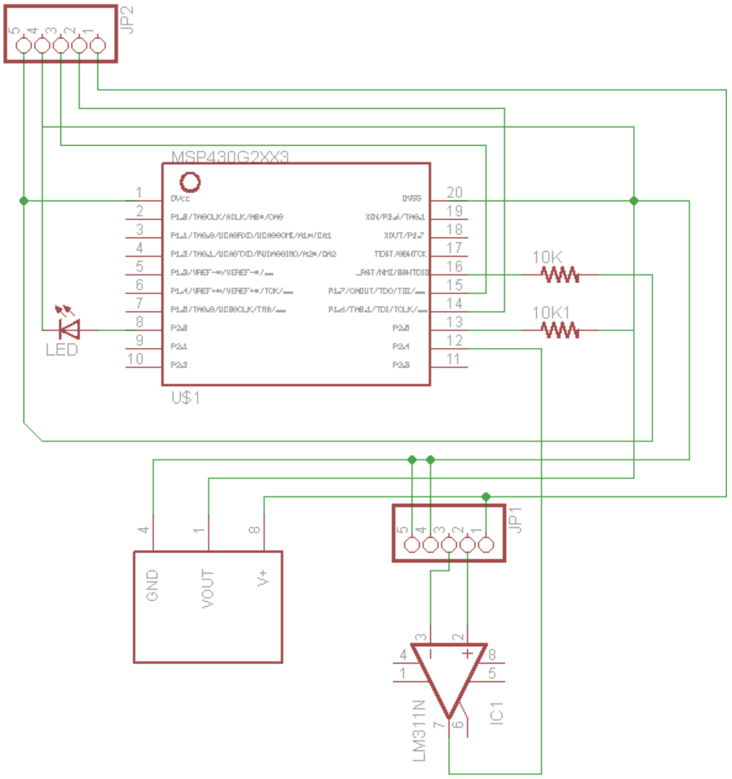


Figure 4.11: Slave sensor circuit diagram

4.4.1 I2C manager

The I2C manager controls the communication between the hybrid mote and the connected slave device. The I2C manager works by responding to communication requests from the master device, which is the hybrid mote. The protocol used in communication between hybrid mote and slave device is detailed in figure 4.4 on page 51.

4.4.2 Gas sensor: NGM-2611

The gas sensor which is used for testing of the system is a methane sensor which was developed by Figaro corporation [43]. The sensor is pre-calibrated and allows for its sensitivity to be manually adjusted by means of a potentiometer. The sensor is a simple device to communicate with, it has 5 pins. The pins 3 and 2 are connected to a comparator which when Vout exceeds the Vref voltage the NGM-2611 will signal its output HIGH and cause the MSP430G2553 microcontroller to set the alarm flag, signalling that the gas concentration level is above the safe limit.

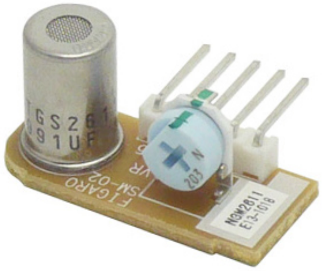


Figure 4.12: Methane gas sensor module [3]

Chapter 5

Design Evaluation

5.1 Overview

This chapter aims to provide a description of the quantitative evaluation of the system as a whole which was designed and developed in this document. Testing that was perform consisted of answering the following questions;

1. What are the power consumption characteristics of the system?
2. How effective is the RFID in detecting tags?
3. Under what conditions will the slave sensor send an alert when methane gas concentration is at a dangerous level?

5.2 Laboratory tests

The test set-up consisted of one Hybrid RFID mote, a gateway Raspberry pi and an Android mobile phone. The Hybrid RFID mote was placed against the wall at a height of 2 meters with a slave sensor 1 meter away. This positioning of the mote and slave sensor is due to the fact that methane gas would rise in the presence of air because of its lower density.

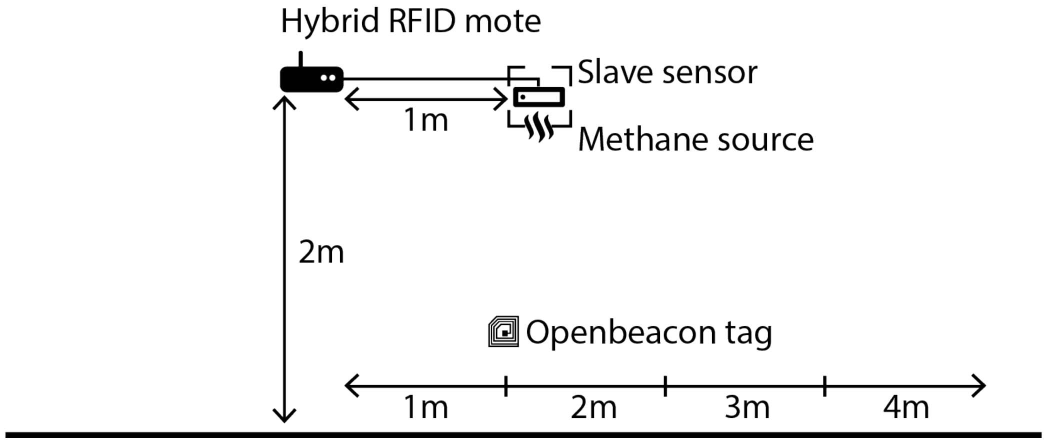


Figure 5.1: Test setup

The evaluation occurred over a period of 72 hours so that different aspects of the system could be evaluated, during the time RFID tags were carried by 2 assistants so as to simulate workers moving from place to place, while the slave sensor was periodically exposed to low volumes of methane gas.

The tag detection reliability phase looked at the reliability of the Openbeacon reader in successfully detecting a tags unique ID and what effect multiple tags within the reader range has.

Gateway solar power supply reliability looked at the possibility of using a solar power and a lead acid battery charger to provide electrical energy to the Raspberry pi over the 72 hour evaluation period.

The Hybrid mote power usage evaluation part looked at how effective the energy saving modes of operation were in conserving the motes battery charge.

The Slave sensor response time evaluation looked at the reliability of the slave sensor to accurately and timeously respond with an alert to a gas level over the alarm point. And finally the range of the RFID tags were determined by experimentation.

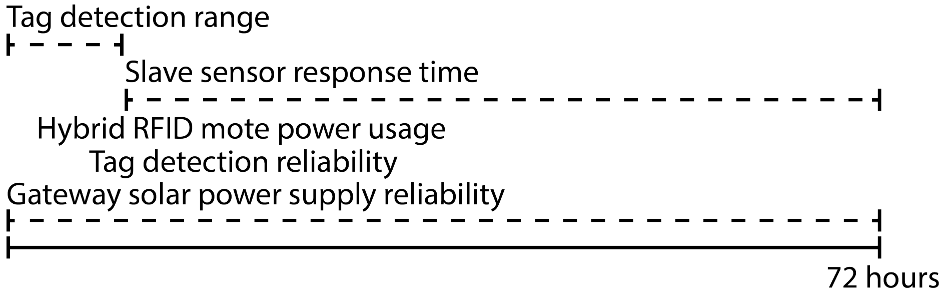

Tag detection range
Slave sensor response time
Hybrid RFID mote power usage
Tag detection reliability
Gateway solar power supply reliability
72 hours

Figure 5.2: Evaluation time span

Each one is described fully in the following sections:

5.2.1 Power consumption

The power consumption will determine the power characteristics of the hybrid motes and attached slave sensor devices in the event that the external power supply is interrupted, in the mining example this could occur in the event of a tunnel collapse which severs power cables. The battery supply would be able to power each hybrid mote for a time while still performing its function as mine personnel locator and mine methane gas detector.

5.2.1.1 Gateway

The power consumption evaluation for the gateway device explored the possibility of using renewable solar energy to supplement the charge provided by a battery. This was done in order to explore alternate applications of the Hybrid RFID system where the devices will be deployed above-ground over long periods for the sensor of the environmental conditions and the detection of nearby objects.

Experiment 6: Gateway power The gateway power system consisted of four components;

1. **Solar panel**
   The solar panel that was used was 35 Watt Polycrystalline silicon type.

2. **Battery charger**
   Battery charger was based around the LM317 integrated circuit.

3. **Lead acid battery**
   The battery brand was Rita.

4. **Power supply**
   The power supply that was used was of a switch mode type, outputting the required 5 Volts with a maximum output current of 3 Amperes.

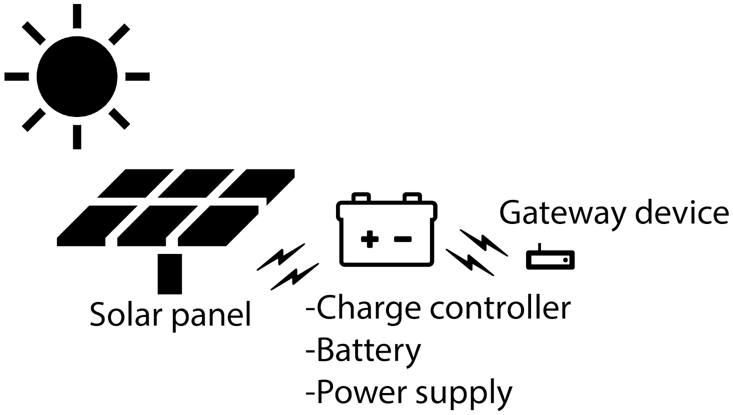


Figure 5.3: Solar power supply

Method of experimentation As mentioned above the evaluation period was 72 hours, it consisted of powering the Raspberry pi gateway using a solar panel during the day and off a lead acid battery during the night. What we are interested in was whether the Raspberry pi gateway could perform its function while being powered exclusively by solar energy. The process of evaluating the use of solar energy involved periodically measuring the voltage of the lead acid battery in order to approximate the level of charge remaining in the battery and then determine the battery life of the gateways solar power supply.

Results What we found during this test was that the battery only lasted for 61 hours, this was due in part to the cloudy weather that was experienced during the 72 hours. The cloud cover caused the solar panel to not receive sufficient solar energy in order to provide an electrical current to the battery charger. This was unfortunate but it's the reality of using renewable energy such as solar, the use of one battery also affects the performance

due to the battery being charged by the solar panel and at the same time discharged by the Raspberry pi during the day cycle.
A possible solution would be to simply use two batteries, when one battery is being charged during the day, the other is being discharged. At any one time during the day a battery is being charged.

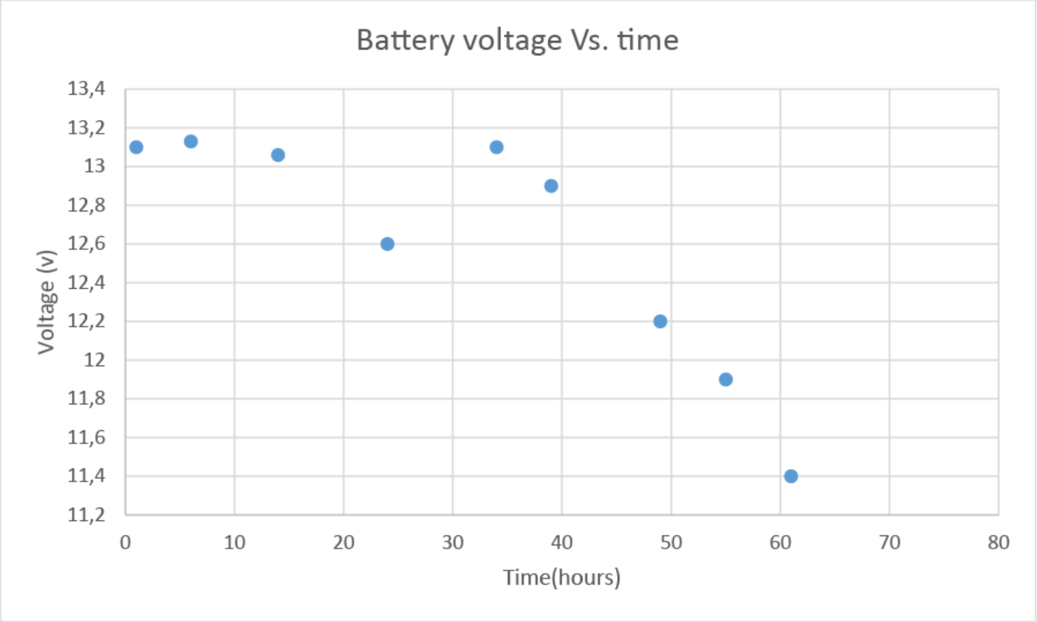


Figure 5.4: Voltage vs. time

To continue the other tests the Raspberry pi was connected to mains power supply for the remaining time.

5.2.1.2 Hybrid mote

A hybrid RFID mote prototype includes;

- Arduino UNO
- XBee Series 2 PRO
- Wireless Shield
- OpenBeacon RFID reader

With all these hardware components the battery power consumption and battery life needs to be known. The Hybrid mote has two modes, Active and sleep. In active mode the microcontroller, XBee and RFID reader are all operating at highest performance without any power saving features enabled. Sleep mode has the microcontroller put into low power idle mode which reduces its current draw. The XBee has its own built-in power saving features which reduces current draw while not transmitting or receiving. The RFID reader is not in any power saving mode as it is constantly sending out signals trying to interrogate any nearby detected RFID tag. The reader wakes the microcontroller out of sleep mode by means of communicating data on the UART port.

Experiment 7: Hybrid mote energy saving feature In order for the hybrid motes life span to be increased, energy saving features were employed in its firmware. This section looks at the improvements in reducing the current consumed of the hybrid mote.

Modes of operation In measuring the advantage acquired in terms of the battery life span when using the energy saving features that were developed. The target of the alternating modes evaluation was to beat the 39 hours battery life span which is theoretically how long the battery will last if the Hybrid mote functions only in its active mode. The active mode consumed a substantial amount of current, the maximum constant current was 153 mA.

- Digi Xbee: 42 mA
- ATMEGA328P: 45 mA
- Openbeacon RFID reader: 26 mA
- Arduino UNO board: 40 mA

The sleep mode of the Hybrid mote reduced current draw by turning off the Xbee transceiver and putting the ATMEGA328P into low power sleep mode, which would in effect reduce power usage.

Method of experimentation In measuring the difference in power consumption between the active and sleep modes an ammeter was placed in series to the power input of the hybrid mote in order to measure the constant current and subsequently calculate the power consumption. The experiment consisted of forcing the hybrid mote to switch between the two modes of operation, this was achieved by introducing an RFID tag to the openBeacon reader. The tag would be detected by the reader and its ID would be transmitted on the serial input of the ATMEGA328P, which would wake it and power the Xbee for transmitting of the tag ID and gas level.

Results The current consumed by the Hybrid mote in active mode was measured to be 153 mA, which would allow for 39 hours of battery operation using the standard 6000 mAh Lithium battery. When in sleep mode the current was measured to be 72 mA. Each time the mote switches to active mode, it only stays there for 5 seconds at a time, which is how long it takes to read from the openBeacon , slave sensors and transmit.

During the 72 hours evaluation the hybrid mote interrogated tags a total of 32 times, which means the mote was in active mode for approximately:

$$activeTime = (tagInterrogationCount) + (slaveSensorPollFrequncy) \tag{5.1}$$

In this case the total active time with one slave sensor polling every minute was:

$$4476seconds = 156 + (\frac{1}{60} \times 259200) \tag{5.2}$$

that is 4476 seconds of the 72 hours of evaluation. In terms of the energy used in detecting the environmental conditions around the mote and transmitting that data the mote consumed:

$$energyConsumed(Joules) = p(Watts) \times t(time) \tag{5.3}$$

Therefore active mode consumed:

$$energyConsumed(Joules) = (5 \times 0.153) \times 4476 = 3424.14 \tag{5.4}$$

and sleep mode:

$$energyConsumed(Joules) = (5 \times 0.072) \times 254724 = 18340.12 \tag{5.5}$$

Combined the energy consumed as a percentage of the LiPo battery is:

$$percentageOfBattery = \frac{3424.14J + 18340.12J}{3.7V \times (6Ah \times 3600)} = 27 \tag{5.6}$$

5.2.1.3 Slave sensor

The slave sensors power consumption is also required as it is important to know how many slave sensors can be supported by the hybrid RFID motes' power system so as to not overload it by drawing too much current.
The slave sensors' constant current draw was measured using an ampere meter in its active sensing and idle waiting states.

Active sensing state In active sensing state the methane gas sensor is turned on, it works by using a heating element to heat up the gas which is passing over the sensing resistor. The heating element consumes 50 mA while the measurement circuit consumes 7 mA. Therefore for each slave device in active sensing state the total current would be a maximum of 65 mA.

Idle waiting state The idle waiting state has the methane gas sensor turned off, this state is only used when the Hybrid RFID mote detects that the external power has been shut off. The current draw is however extremely low at a measured amount of 7 mA as it is only the microcontroller which is operating.

From measurements performed each slave sensor will consume a maximum 65 mA. Due to the power supply of each hybrid mote only being able to supply 1 A of current, each hybrid mote is only able to support up to 10 slave sensors while having sufficient head room for safety purposes.

5.2.1.4 RFID tag

Each RFID tag is powered by a lithium 3.3 V 620 mAh coin cell, using lab equipment the constant current draw was measured to be 6.7 mA. At 6.7 mA current draw the battery life is only 92 hours, this is extremely poor for an RFID tag. The reason behind the poor battery life was due to the tags firmware configuration, the firmware used transmits the tags unique ID every second which for our purpose is much too high. Optimisations in the firmware of the RFID tag would have to be made in order to reduce the current draw from the battery.

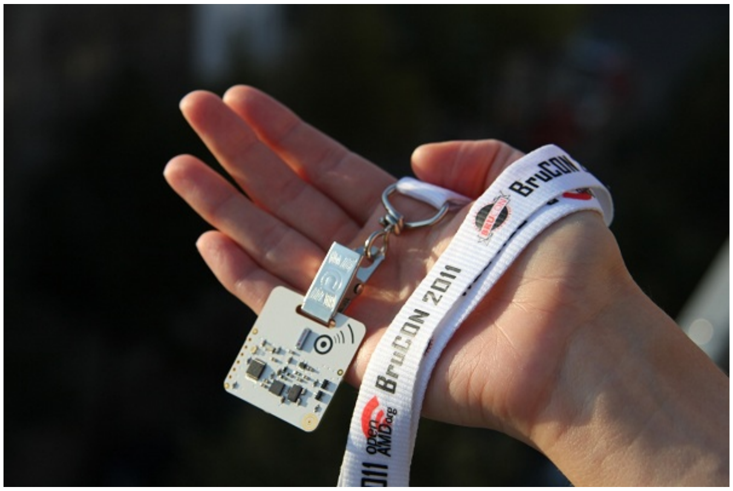


Figure 5.5: The Openbeacon RFID tag that was used [4]

5.2.2 RFID tag detection range

The read range of the OpenBeacon RFID reader is also an important measurement, as it would affect the placement of the Hybrid mote so as to be in the most likely position in which it is able to detect any nearby RFID tags. Tag detection range testing consisted of one experiment.

5.2.2.1 Method of experimentation

The RFID reader was reconfigured to be able to output the RSSI of any RFID tag that it interrogates. This was accomplished by reprogramming the firmware of the OpenBeacon RFID reader. The way in which the OpenBeacon responds with the RSSI is very different to how a typical active RFID system works, usually an RFID tag would transmit its' signal at full power while the RFID reader compares the RSS with an expected full signal strength. The technique employed by OpenBeacon is for the RFID tag to step its transmit power level down from full strength to almost zero in 25% steps and then sending the power level in the data packet, therefore RSSI resolution is only given in 4 steps.

- 0 = Very close (Low transmit power)
- 85 = Close

- 170 = Near
- 255 = Far (Highest transmit power)

The average of 10 of the received power levels is then used to determine the actual power level and then infer how far away the tag is.

The RSS was measured at a distance starting at 1 meter and incrementing by 1 meter until the openBeacon RFID reader was unable to respond with a detected tags RSSI.

5.2.2.2 Results

In the results the tag was only detectable for 3.5 meters away from the hybrid mote, this range is better than what was expected for a small low powered RFID tag. The graph below shows the transmit power level (RSSI for OpenBeacon) against the distances that were experimented with.

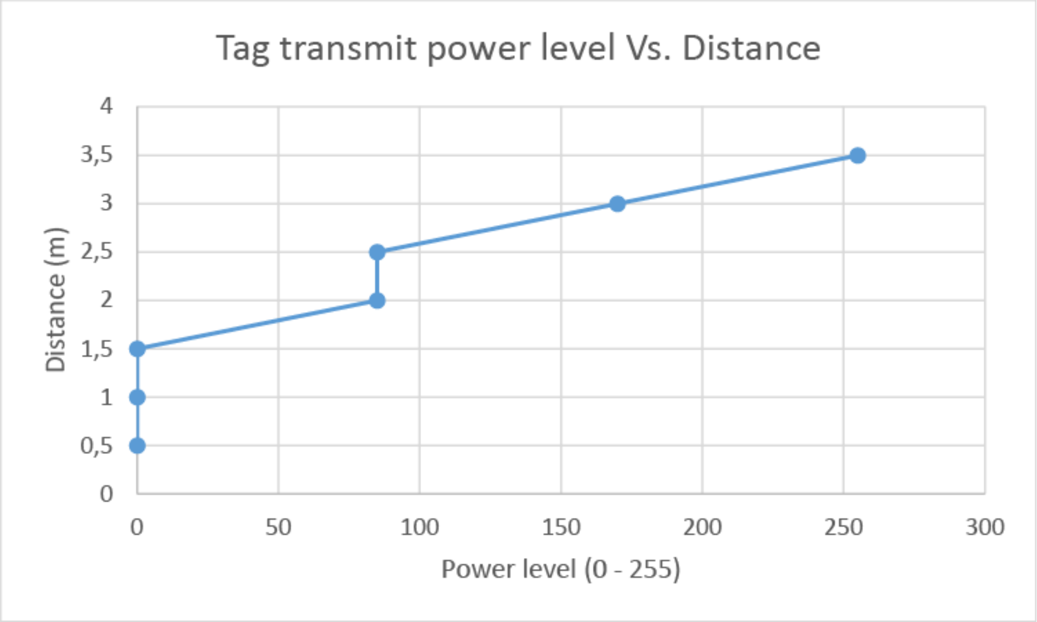


Figure 5.6: RFID tag power level

5.2.3 Slave sensor response time

The effectiveness of the slave sensor sensor in detecting the methane needed to be tested in order to know under what conditions it will respond with a warning.

5.2.3.1 Method of experimentation

The slave sensor was tested by applying a sample of the gas delivered to the sensor inside of a small perspex chamber by means of a syringe. A volume of methane gas was delivered to the sensor in order to measure the time it takes for the sensor to respond with a warning to the increase in methane gas concentration in its proximity. The experiment consisted of taking measurements of the time with concentrations of methane at the Lower explosive limit (LEL) and the Upper explosive limit (UEL) which is the concentration range at which methane is flammable;

- 2.5%
- 5%, the LEL of methane [44]
- 10%
- 15%, the UEL of methane [44]

by volume in air. The chamber used had a volume of 64000 mm^3 which is equivalent to 64 ml. Therefore in order to change the methane gas concentration of the air in the chamber to the required amounts, the volume of gas to be inserted to the chamber needed to be calculated.

$$volumeOfMethane = percentage * chamberVolume \tag{5.7}$$

The volume is calculated as;

- 2.5%, 1.6ml
- 5%, 3.2ml
- 10%, 6.4ml
- 15%, 9.6ml

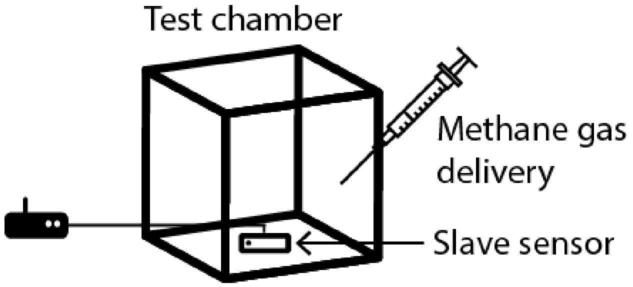


Figure 5.7: Sensor test chamber

Each run at the various concentrations was done at a cold system start i.e the Hybrid mote and slave sensor needed to boot-up, as well as the slave sensor needing to prepare the NGM-2611 to take measurements of the gas concentration.

5.2.3.2 Results

The results of the experiment yielded the following graph of the time it took for the system to detect a gas concentration over the set limit of 5%.

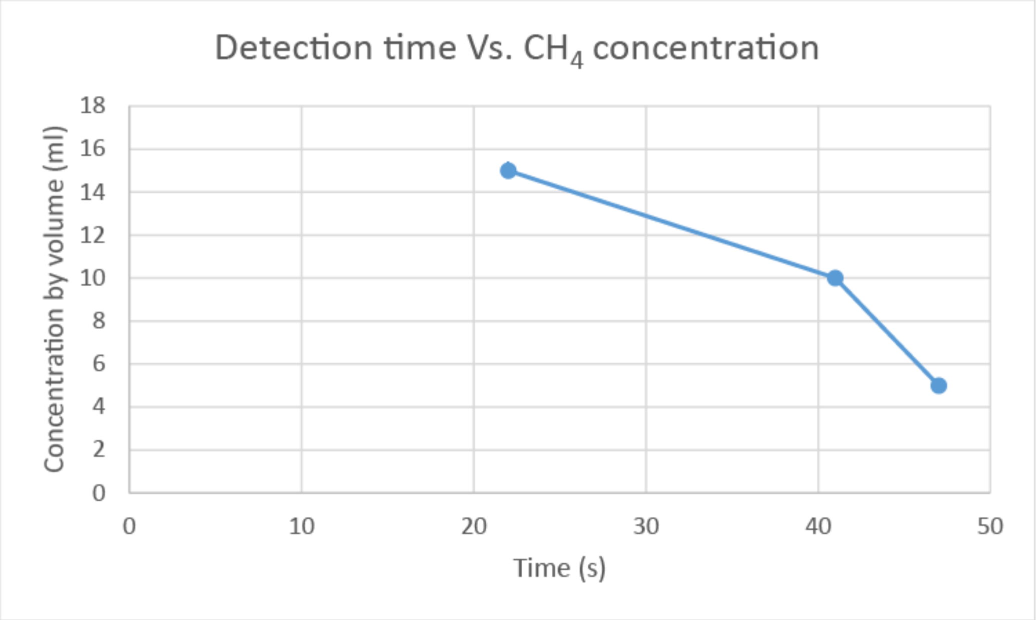


Figure 5.8: Detection time

Chapter 6

Conclusion and future work

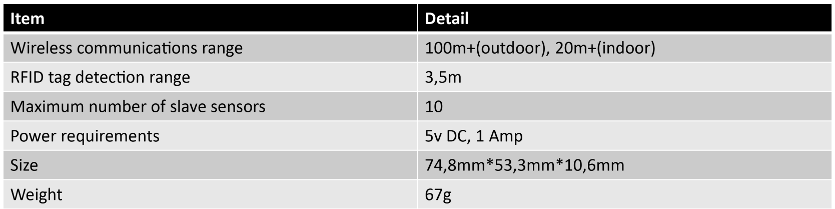
6.1 Conclusion

The range of possible applications afforded by the integration of WSN and RFID allows for many problems to be solved. As was demonstrated the threat of explosions posed by leaking methane gas from coal seams in underground mines could be mitigated. And in the event of a disaster underground the location of miners within the tunnel system would be known through the use of the Hybrid RFID sensor.

The use of cheap slave sensors showed that it is possible to increase the resolution of the sensor data provided by a sensor mote without adding more motes to the sensor network, which could be prohibitively expensive. The design of the Hybrid RFID sensor allows up to 10 of these slave sensors to be providing environmental data by daisy-chaining them on the I2c bus.

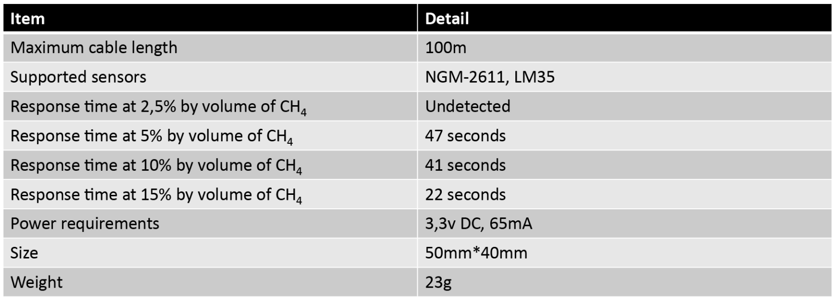
6.1.1 Summary

The systems specifications from experimentation and testing are detailed as follows in tables 6.1 and 6.2.



Item	Detail
Wireless communications range	100m+(outdoor), 20m+(indoor)
RFID tag detection range	3,5m
Maximum number of slave sensors	10
Power requirements	5v DC, 1 Amp
Size	74,8mm*53,3mm*10,6mm
Weight	67g

Figure 6.1: Specifications of Hybrid mote



Item	Detail
Maximum cable length	100m
Supported sensors	NGM-2611, LM35
Response time at 2,5% by volume of CH_4	Undetected
Response time at 5% by volume of CH_4	47 seconds
Response time at 10% by volume of CH_4	41 seconds
Response time at 15% by volume of CH_4	22 seconds
Power requirements	3,3v DC, 65mA
Size	50mm*40mm
Weight	23g

Figure 6.2: Specifications of Slave sensor

The evaluation identified where future improvements could be made, they are explained in the following section.

6.2 Future work

With this system there lies some areas in its design and testing which could be improved on, due to time and financial constraints placed on the project.

6.2.1 Large scale network testing

The wireless hardware which provides the sensor network is built on the proven Zigbee technology, there have been many research papers on the reliability and scalability in simulators such as NS [45]. Therefore it was beyond the scope of this work for in depth scalability testing of the mine monitoring system, however a real world implementation of this system would require this information to provide the network reliability and quality of service needed.

6.2.2 Interference testing

Due to the fact that both the RFID tags and the Hybrid RFID mote wireless transceivers operate on the same 2.4GHz frequency there is the possibility for communication to be interrupted. The CSMA technique employed by the transceivers do however limit the probability of interference occurring. However testing this notion would be required before implementation of this system.

6.2.3 Multiple sensors

Each Slave sensor was designed with the ability to attach additional analogue or digital sensors such as a vibration sensor. The addition of another sensor to the slave device has however not been tested.

6.2.4 Environmental testing

The Hybrid motes and slave sensor have not been tested for their ability to operate over long periods in a harsh environment where humidity and temperature may be harmful to electronic devices. For instance the time variation that the NGM-2611 takes to detect an over the limit methane concentration when operating in an environment with 20% humidity compared to 80% is vital information which can be used to improve the systems' effectiveness in different conditions.

6.2.5 Power saving

Some improvements could be made in improving the power consumption characteristics of the hybrid mote, starting with designing a custom PCB using a switching mode power supply instead of the linear regulator that was used on the Arduino UNO.

Appendix A

System preparation

This appendix lists MySQL code samples and BASH script commands that were referenced to during the preparation and testing of the gateway devices.

A.1 MySQL setup

Listing A.1: Formatting

```
#!/bin/bash
Sudo mkfs.ext4 /dev/sda1 L untitled
```

Listing A.2: Adding a mount point

```
#!/bin/bash
Sudo mkdir /mnt/usbdrive
Sudo mount /dev/sda1 /mnt/usbdrive
```

Listing A.3: Copying MySQL database files to the storage device

```
#!/bin/bash
Sudo cp  rv  /var/lib/mysql /mtn/usbdrive
```

Listing A.4: mysql configuration file with new location

```
[mysqld]
#
```

```
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /mnt/usbdrive/mysql
tmpdir          = /tmp
```

A.2 Database creation

Listing A.5: Create test database

```
CREATE DATABASE perftest;
Create tables
CREATE TABLE ‘perftest‘.‘test‘ (
  ‘id‘ INT NOT NULL AUTO_INCREMENT ,
  ‘col1‘ INT NULL ,
  ‘col2‘ INT NULL ,
  ‘col3‘ INT NULL ,
  ‘col4‘ INT NULL ,
  ‘col5‘ INT NULL ,
  ‘col6‘ INT NULL ,
  ‘col7‘ INT NULL ,
  ‘col8‘ INT NULL ,
  ‘col9‘ INT NULL ,
  PRIMARY KEY (‘id‘) );
```

A.3 Apache2 installation

Listing A.6: Formatting

```
#!/bin/bash
sudo apt-get install apache2 apache2-utils libapache2-mod-php5
    php5 php5-mysql
```

A.4 MySQL INSERT

Listing A.7: The bash script to perform INSERT's

```
#!/bin/bash
echo start
for i in {1..10000}
do
mysql --host=localhost --user=root --password=jarred perftest <<
   EOF
insert into perftest.test
   (col1,col2,col3,col4,col5,col6,col7,col8,col9) values
   ($i,$i,$i,$i,$i,$i,$i,$i,$i);
EOF
done
```

A.5 Measuring execution time

Listing A.8: Command to measure execution time of test script which performs the SQL INSERT's

```
#!/bin/bash
(time ./inserttest.sh) >> inserttestoutput.txt 2>&1
```

A.6 Measuring system resources

Listing A.9: Measuring the system resources usage while SQL INSERT's are being executed

```
#!/bin/bash
vmstat 1 >> inserttestoutput.txt
```

Appendix B

Httperf

This appendix lists the sample output for the Httperf application that was referenced to previously in this document.

B.1 Sample output

Listing B.1: Httperf sample output

```
#!/bin/bash

Connection rate: 20.0 conn/s (50.0 ms/conn, <=1 concurrent
   connections)
Connection time [ms]: min 2.7 avg 2.9 max 8.4 median 2.5 stddev 0.3
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 20.0 req/s (50.0 ms/req)
Request size [B]: 62.0

Reply rate [replies/s]: min 20.0 avg 20.0 max 20.0 stddev 0.0 (9
   samples)
Reply time [ms]: response 2.9 transfer 0.0
Reply size [B]: header 260.0 content 4287.0 footer 0.0 (total
   4547.0)
Reply status: 1xx=0 2xx=1000 3xx=0 4xx=0 5xx=0
```

Bibliography

[1] "The E-Quad News, Princeton University." [Online]. Available: http://www.princeton.edu/engineering/eqnews/spring03/feature4.html

[2] "Storm Petrel." [Online]. Available: http://spectrum.ieee.org/images/apr04/images/0404birdf4_123.jpg

[3] "NGM2611-E13 : Gas Sensors & Modules - Products - Figaro Engineering Inc." [Online]. Available: http://www.figaro.co.jp/en/product/entry/ngm2611.html

[4] "OpenBeacon RFID tag." [Online]. Available: http://www.openbeacon.org/File:BruCON2011-Tag.jpg

[5] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," *Computer Communications*, vol. 30, no. 7, pp. 1655–1695, May 2007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0140366406004749

[6] L. J. Celentano, "RFID-Assisted Wireless Sensor Networks for Cardiac Tele-healthcare," 2007.

[7] H. Liu and M. Bolic, "Integration of RFID and wireless sensor networks," *Proceedings of Sense ID . . .* , pp. 1–6, 2007. [Online]. Available: http://www.comp.hkbu.edu.hk/~hliu/publications/RFIDWSNs09.pdf

[8] L. Ho, M. Moh, Z. Walker, T. Hamada, and C. Su, "A prototype on RFID and sensor networks for elder healthcare: progress report," *. . . network design and analysis*, pp. 70–75, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1080164

[9] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and J. I. Robla, "A review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends." *Sensors (Basel, Switzerland)*, vol. 9, no. 6, pp. 4728–50, Jan. 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3291936&tool=pmcentrez&rendertype=abstract

[10] A. Mitrokotsa and C. Douligeris, "Integrated RFID and Sensor Networks : Architectures and," pp. 511–536, 2009.

[11] W. Eubanks and P. Smale, "CRS Report to Congress,," *Payment Card Interchange Fees: An Economic ...*, 2008. [Online]. Available: http://heinonlinebackup.com/hol-cgi-bin/get_pdf.cgi?handle=hein.tera/crser0010&section=1

[12] X. Niu, M. Ieee, X. Huang, Z. Zhao, and Y. Zhang, "The Design and Evaluation of a Wireless Sensor Network for Mine Safety Monitoring," vol. 12, pp. 1291–1295, 2007.

[13] "Cost-effective gas detection, vibration multisensor network suited to mining applications." [Online]. Available: http://www.miningweekly.com/article/cost-effective-gas-detection-vibration-multisensor-network-suited-to-mining-applications-2012

[14] Z. Pei and Z. Deng, "A distributed location algorithm for underground miners based on rescue robot and coal-mining wireless sensor networks," *Robotics, Automation and Mechatronics, 2008 ...*, pp. 884–888, 2008. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4690866

[15] "Hospital Using RTLS to Monitor Patients' Conditions - RFID Journal." [Online]. Available: http://www.rfidjournal.com/articles/view?6685

[16] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware Design Experiences in ZebraNet Categories and Subject Descriptors," pp. 227–238.

[17] A. Milenkovic and M. Milenkovic, "An environment for runtime power monitoring of wireless sensor network platforms," *System Theory, 2005 ...*, pp. 406–410, 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1460946

[18] E. Callaway, "Low Power Consumption Features of the IEEE 802.15.4/ZigBee LR-WPAN Standard," 2003.

[19] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, pp. 1–29, Mar. 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1498915.1498916

[20] R. Clauberg, "RFID and sensor networks," *Proc. RFID Workshop, St. Gallen, Switzerland*, pp. 1–6, 2004. [Online]. Available: http://intranet.daiict.ac.in/~ranjan/isn2007/papers/ibm_paper_rfid.pdf

[21] W. Heinzelman, "An application-specific protocol architecture for wireless microsensor networks," *Wireless . . .* , vol. 1, no. 4, pp. 660–670, 2002. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1045297

[22] P. Song, C. Chen, K. Li, and L. Sui, "The Design and Realization of Embedded Gateway Based on WSN," *2008 International Conference on Computer Science and Software Engineering*, pp. 32–36, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4722557

[23] D.-f. Ye, L.-l. Min, and W. Wang, "Design and Implementation of Wireless Sensor Network Gateway Based on Environmental Monitoring," *2009 International Conference on Environmental Science and Information Application Technology*, pp. 289–292, Jul. 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5199892

[24] I. Chatzigiannakis, G. Mylonas, and S. Nikoletseas, "50 ways to build your application: A survey of middleware and systems for Wireless Sensor Networks," *2007 IEEE Conference on Emerging Technologies & Factory Automation (EFTA 2007)*, pp. 466–473, Sep. 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4416805

[25] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," *Proceedings of the 1st ACM international workshop on Wireless sensor networks*

and applications - WSNA '02, p. 88, 2002. [Online]. Available: http://portal.acm.org/citation.cfm?doid=570738.570751

[26] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-s. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking : Design Tradeoffs and Early Experiences with ZebraNet," pp. 96–107, 2002.

[27] K. Bailey and B. Wiebe, "SensorFusion: Localization and Tracking of Sensors Using RFID," *Earthquake* ..., 2007. [Online]. Available: http://mceer-nt4.mceer.buffalo.edu/education/reu/2007/content/17bailey_uribe_wiebe.pdf

[28] R. Want, "The magic of RFID," *Queue*, no. October, 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1035619

[29] "The Different Types of RFID Systems — Impinj." [Online]. Available: http://www.impinj.com/resources/about-rfid/the-different-types-of-rfid-systems/

[30] "Waspmote 802.15.4 PRO SMA 2 DBI price." [Online]. Available: http://www.cooking-hacks.com/waspmote-802-15-4-pro-sma-2-dbi

[31] "Beginner's Guide to Crossbow Mica2 cost." [Online]. Available: http://www.pages.drexel.edu/~kws23/tutorials/motes/motes.html

[32] "Z1 Platform, Zolertia Shop." [Online]. Available: http://webshop.zolertia.com/product_info.php/products_id/32

[33] "Blogs — Forget application response time standards its all about the human reaction — Riverbed." [Online]. Available: http://www.riverbed.com/blogs/human-reaction-drives-application-response-time-standards.html

[34] N. X. P. Semiconductors, "AN10658 Sending I2C-bus signals via long communications cables," no. February, pp. 1–28, 2008.

[35] "About us — Raspberry Pi." [Online]. Available: http://www.raspberrypi.org/about/

[36] "PC Engines ALIX system boards." [Online]. Available: http://www.pcengines.ch/alix.htm

[37] "FrontPage - Raspbian." [Online]. Available: http://www.raspbian.org/

[38] "OpenWrt." [Online]. Available: https://openwrt.org/

[39] "MySQL :: MySQL 5.5 Reference Manual :: 2.2 Installing MySQL on Unix/Linux Using Generic Binaries." [Online]. Available: http://dev.mysql.com/doc/refman/5.5/en/binary-installation.html

[40] "Flash Memory Survives 100 Million Cycles - IEEE Spectrum." [Online]. Available: http://spectrum.ieee.org/semiconductors/memory/flash-memory-survives-100-million-cycles

[41] D. Mosberger and T. Jin, "httperf—a tool for measuring web server performance," *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, no. 3, pp. 31–37, Dec. 1998. [Online]. Available: http://dl.acm.org/citation.cfm?id=306225.306235

[42] "A byte-oriented AES-256 implementation — Literatecode." [Online]. Available: http://www.literatecode.com/aes256

[43] P. Information, "NGM-2611 - pre-calibrated module for Methane Features : * Factory calibrated Applications : * Residential natural gas alarm," pp. 1–3.

[44] L. Gas, "Lower and Upper Explosive Limits for Flammable Gases and Vapors (LEL/UEL)," *Matheson gas products*, p. 22, 2013. [Online]. Available: http://www.chrysalisscientific.com/pg443-Lower-LEL-Upper-UEL-Explosive-Limits.pdf

[45] "Simulation of IEEE 802.15.4/ZigBee with Network Simulator-2 (ns-2)." [Online]. Available: http://www.ifn.et.tu-dresden.de/~marandin/ZigBee/ZigBeeSimulationEnvironment.html