

# A Prototype Animal Borne Behaviour Monitoring System

by

Solomon Petrus le Roux



*Thesis presented in partial fulfilment of the requirements for  
the degree of Master of Engineering in the Department of  
Electrical and Electronic Engineering at the University of  
Stellenbosch*

Supervisor: Dr. R. Wolhuter

March 2016

# Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the authorship owner thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2016

# Abstract

## A Prototype Animal Borne Behaviour Monitoring System

S.P. le Roux

*Department of Electrical and Electronic Engineering,  
University of Stellenbosch,  
Private Bag X1, Matieland 7602, South Africa.*

Thesis: MEng (EE)

October 2015

Rhinos across the globe are suffering from an immense onslaught from rhino poachers. The latter show no regard for the remaining rhino species. New technologies must be implemented to provide scientists, biologists and nature conservationists with key information regarding the behaviour and well being of animals. This project entailed the design and development of a prototype Animal Borne Behaviour Monitoring System (ABBMS). The system was based on animal borne sensor driven devices known as WildMotes. The latter were used to collect and communicate sensor data to a base station, by means of a multi-hop Wireless Sensor Network (WSN) that exists between WildMotes, repeaters and the base station. The project considered the hardware design of the WildMotes, which included various components such as an ultra-low power microcontroller, GPS, accelerometer, nano-power tilt and vibration sensor, RF communication module, microSD card and FRAM modules. Apart from the hardware design, the project included all software required of the ABBMS. Initially the WildMotes collected data that could be used for the automatic behaviour classification of animals, by means of computer based techniques. The data was utilised with techniques such as linear- and quadratic discriminant analysis and decision trees, to classify the behaviour of rhinos and sheep with high accuracies. This behaviour included running, walking, standing, grazing and laying down. Furthermore, this project successfully implemented an On-animal Behaviour Classification System (OABCS). To the best of our knowledge, this is the first implementation where behaviour classification is performed in real time, on the animal. This technique provides live updates of animal behaviour, as opposed to post processing, computer based techniques. The OABCS was able to accurately distinguish between similar behavioural classes as above mentioned. In addition to the OABCS, a nano-power tilt and vibration sensor was applied, as an ultra-low power alternative, to classify the behaviour of animals. The latter could accurately distinguish between the same behaviour, while consuming very little energy. This technique was further utilised, in combination with the OABCS, to extend the battery life of the WildMotes from roughly 47 days to 270 days. Finally, GPS coordinates were obtained and utilised to reveal repetitive movement patterns of rhinos, by means of a heat map. In future work, the ABBMS can be combined with the OABCS, GPS locations and key stress level indicators, such as pulse rates, to learn more about endangered species and serve as tools in the fight against illegal poaching activities.

# Uittreksel

## Prototipe Dier Gedrags-moniteringstelsel

S.P. le Roux

*Departement Elektries en Elektroniese Ingenieurswese,  
Universiteit van Stellenbosch,  
Privaatsak X1, Matieland 7602, Swid Afrika.*

Tesis: MIng (EE)

Oktober 2015

Renosters regoor die wêreld ly onder 'n hewige aanslag van renosterstropers. Laasgenoemde toon geen agting vir die oorblywende renosterspesies nie. Nuwe tegnologie moet geïmplementeer word om wetenskaplikes, bioloë en natuurbewaarders met belangrike inligting oor die gedrag van diere te voorsien. Hierdie projek behels die ontwikkeling van 'n prototipe “*Animal Borne Behaviour Monitoring System*” (ABBMS). Die stelsel is gebaseer op mobile sensorgedrewe toestelle bekend as *WildMotes*. Laasgenoemde word gebruik om sensor data te kommunikeer na 'n basis stasie, deur middel van 'n “*Wireless Sensor Network*” (WSN) wat bestaan tussen *Wildmotes*, herhalers en die basis stasie. Die hardeware-ontwerp van die *WildMotes* sluit verskillende komponente in soos: 'n ultra-laekrag mikrobeheerder, GPS, versnellingsensor, nanodrywing kantel-en-vibrasie-sensor, RF kommunikasiemodule, microSD-kaart en FRAM-modules. Afgesien van die hardewareontwerp, is alle sagteware soos benodig vir die projek, ook ontwikkel. Aanvanklik is die *WildMotes* gebruik om data in te samel om d.m.v. rekenartegniese dieregedrag outomaties te klassifiseer. Die data is verwerk met tegniese soos lineêre-en-kwadratiese diskriminante-analise en besluitbome, om die gedrag van renosters en skape met 'n hoë akkuraatheid te klassifiseer. Die gedrag het hardloop, loop, staan, wei en lê ingesluit. Gedurende die projek is 'n “*On-animal Behaviour Classification System*” (OABCS) suksesvol ontwikkel. Na die beste van ons kennis, is dit die eerste implementering waar gedragsklassifikasie intyds op die dier uitgevoer word. Hierdie tegniek bied oombliklike opdaterings van die dier se gedrag, in teenstelling met die rekenaargebaseerde tegnieke, waar die inligting eers veel later beskikbaar is. Die OABCS was in staat om akkuraat te onderskei tussen soortgelyke gedragspatrone soos hierbo genoem. Bykomend tot die OABCS, is 'n nanodrywing kantel-en-vibrasie-sensor aangewend, as 'n laedrywing alternatief, om die gedrag van diere te klassifiseer. Dit kon akkuraat onderskei tussen die laasgenome gedragspatrone. Hierdie tegniek is verder benut, in kombinasie met die OABCS, om die batterylewe van die *WildMotes* te verleng vanaf ongeveer 47 dae tot 270 dae. Laastens was GPS-koördinate verkry en aangewend om herhalende bewegingspatrone van renosters te openbaar, deur middel van 'n hittekaart. In toekomstige werk kan die diere se gedrag in kombinasie met GPS posisies en sleutel stresvlak aanwysers, soos polsslak, gebruik word om meer te leer oor bedreigde spesies.



# Acknowledgements

I would like to express my sincere gratitude to the following people:

- Dr. Riaan Wolhuter, for believing in the project, for his guidance and support.
- Barend and Amanda le Roux for their wisdom and support.
- Fanie and Elmien Dippenaar for making the practical tests of the system possible.
- Luca Lategan for his help and support.

# Dedications

*This report is dedicated to my beloved wife **Salomi le Roux** and our extended family.*

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Uittreksel</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Dedications</b>	<b>v</b>
<b>Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>xi</b>
<b>Nomenclature</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background to the Project . . . . .	1
1.2 Project Objectives . . . . .	3
1.3 Project Summary . . . . .	4
1.4 Project Outcomes and Contributions . . . . .	5
1.5 Structure of this Thesis . . . . .	6
<b>2 Literature Study</b>	<b>7</b>
2.1 Previous Work Done in Related Areas . . . . .	7
<b>3 Design Requirements and System Formulation</b>	<b>12</b>
3.1 Design Requirements . . . . .	12
3.2 System Formulation . . . . .	14
<b>4 Hardware Design</b>	<b>17</b>
4.1 WildMotes . . . . .	17
4.2 Base Station and Repeaters . . . . .	34
4.3 Collar and Casing Design . . . . .	34
<b>5 Software Design</b>	<b>37</b>
5.1 Initialisation Software . . . . .	37
5.2 Functional Software . . . . .	42

<b>6</b>	<b>Network Implimentation</b>	<b>61</b>
6.1	Network Design Requirements . . . . .	61
6.2	Network Protocol . . . . .	62
6.3	Network Implementation . . . . .	66
<b>7</b>	<b>Energy Harvesting</b>	<b>72</b>
<b>8</b>	<b>Measurements and Results</b>	<b>76</b>
8.1	Initial Hardware Tests . . . . .	76
8.2	Experiment 1: 100 Hz Accelerometer Logger for Sheep Behaviour Classification . . . . .	77
8.3	Experiment 2: 40 Hz Accelerometer Logger for Rhino Behaviour Classification	83
8.4	Experiment 3: On-animal Behaviour Classification System . . . . .	88
8.5	Behaviour Classification using an Omni-directional Vibration Sensor . . . . .	93
8.6	Pulse Rate Detection . . . . .	103
<b>9</b>	<b>Conclusion</b>	<b>108</b>
9.1	Project Motivation . . . . .	108
9.2	Work Done Under the Project . . . . .	108
9.3	Project Outcomes and Contributions . . . . .	109
9.4	Future Work and Recommendations . . . . .	110
	<b>List of References</b>	<b>112</b>
	<b>Appendices</b>	<b>117</b>
<b>A</b>	<b>Measurements and Results Additional Info</b>	<b>118</b>
<b>B</b>	<b>PCB Footprints</b>	<b>128</b>
<b>C</b>	<b>PCB Layouts</b>	<b>136</b>
C.1	WildMote PCB Layout . . . . .	136
C.2	RF Shield PCB Layout . . . . .	137
C.3	GSP Expansion Card PCB Layout . . . . .	138
<b>D</b>	<b>Schematic Diagrams</b>	<b>139</b>
D.1	WildMote Schematic Diagram . . . . .	139
D.2	RF Shield Schematic Diagram . . . . .	139
<b>E</b>	<b>Casing Design Drawings</b>	<b>142</b>

# List of Figures

1.1	PREDNET project structure. . . . .	1
1.2	Kruger National Park within South Africa. . . . .	3
3.1	Proposed communication system. . . . .	16
4.1	WildMotes block diagram. . . . .	18
4.2	MSP430FR5739: Ultralow-power mixed signal microcontroller. . . . .	19
4.3	DM3CS: MicroSD card connector. . . . .	20
4.4	FM25V20: 2 Mbit Ferroelectric RAM modules. . . . .	21
4.5	GSN602: GPS receiver. . . . .	22
4.6	ADXL345: Digital accelerometer. . . . .	22
4.7	SQ-SEN-200: Nano-power tilt and vibration sensor. . . . .	23
4.8	MiniSense 100: Piezoelectric cantilever-type vibration sensor. . . . .	23
4.9	ADMP504: Ultralow noise analog microphone. . . . .	24
4.10	ASB1200V1: Analog barometric pressure sensor. . . . .	24
4.11	PCB contacts for external contact microphone. . . . .	25
4.12	CC1101: Low-power sub - 1 GHz RF transceiver. . . . .	27
4.13	L6932: Ultralow drop output linear regulator. . . . .	28
4.14	TPS61222: Low input voltage step-up converter. . . . .	29
4.15	LTC3105: Step-up DC/DC converter. . . . .	30
4.16	MSP430: 4 - Wire JTAG interface. . . . .	31
4.17	Tactile switch and LED's used for debugging. . . . .	32
4.18	PCB headers for external access 1. . . . .	32
4.19	PCB headers for external access 2. . . . .	33
4.20	Assembled WildMote. . . . .	33
4.21	WildMote closeup photos. . . . .	34
	(a) SQ-SEN-200. . . . .	34
	(b) CC1101. . . . .	34
	(c) ADXL345. . . . .	34
	(d) LTC3105. . . . .	34
	(e) GNS602. . . . .	34
	(f) FM25V20. . . . .	34
4.22	RF Shield and CC1101EMK433. . . . .	35
4.23	Raspberri Pi with RF Shield and CC1101EMK433. . . . .	35
5.1	GRACE summary of MSP430FR5739. . . . .	38
5.2	GRACE UART setup. . . . .	38
5.3	Routine 1: Interrupt handlers. . . . .	43
5.4	Routine 1: Initialisation. . . . .	44

5.5	Routine 1: Main flow A. . . . .	45
5.6	Routine 1: Main flow B. . . . .	46
5.7	Routine 2: Interrupt handlers. . . . .	47
5.8	Routine 2: Initialisation. . . . .	48
5.9	Routine 2: Main flow A. . . . .	49
5.10	Routine 2: Main flow B. . . . .	50
5.11	Routine 2: Main flow C. . . . .	51
5.12	Routine 3: Timer interrupt handler. . . . .	55
5.13	Routine 3: Switch interrupt handler A. . . . .	56
5.14	Routine 3: Switch interrupt handler B. . . . .	57
5.15	Routine 3: Initialisation. . . . .	58
5.16	Routine 3: Main flow A. . . . .	59
5.17	Routine 3: Main flow B. . . . .	60
6.1	Star network typology. . . . .	62
6.2	Multi-hop network typology. . . . .	63
6.3	RPL-UDP simulation 1: Nodes placement. . . . .	65
6.4	RPL-UDP simulation 1: Data flow. . . . .	65
6.5	RPL-UDP simulation 2: Node placement and data flow. . . . .	66
6.6	RIME implementation: Node placement. . . . .	67
6.7	RIME implementation 1. . . . .	67
6.8	RIME implementation 2. . . . .	68
6.9	RPL implementation: Node placement. . . . .	68
6.10	RPL implementation 1. . . . .	69
6.11	RPL implementation 2. . . . .	69
6.12	RPL implementation 3. . . . .	70
6.13	RPL implementation 4. . . . .	70
6.14	RPL implementation 5. . . . .	71
7.1	56 Cell solar array. . . . .	73
7.2	Sheep with collars. . . . .	73
7.3	Rhino wallowing. . . . .	74
7.4	Collar covered in mud. . . . .	75
7.5	Elephants swimming. . . . .	75
8.1	90 Degree GPS rotational error. . . . .	77
8.2	Sheep Collar placement. . . . .	78
8.3	Accelerometer orientation on sheep. . . . .	79
8.4	Acceleration data for the 4 <sup>th</sup> of July 2014. . . . .	80
8.5	100 Hz constant sampling frequency. . . . .	81
8.6	Acceleration of a running sheep. . . . .	81
8.7	Vibration sensor interrupt distribution for a sheep running. . . . .	82
8.8	Rhino Collar placement. . . . .	85
8.9	Confusion matrix of the LDA classifier: Rhino behaviour. . . . .	85
8.10	Confusion matrix of the decision tree classifier: Rhino behaviour. . . . .	86
8.11	Acceleration of a walking rhino. . . . .	87
8.12	Acceleration of a running rhino. . . . .	88
8.13	GPS coordinates obtained from free roaming rhinos. . . . .	89
8.14	Heatmap representation of the GPS coordinates. . . . .	90

8.15	Confusion matrix of the OABCS: Sheep behaviour. . . . .	91
8.16	Vibration sensor interrupt distribution for rhino behaviour. . . . .	94
8.17	Moving average of vibration sensor interrupts: Sheep running. . . . .	95
8.18	Moving average of vibration sensor interrupts: Sheep walking. . . . .	95
8.19	Moving average of vibration sensor interrupts: Sheep laying down. . . . .	96
8.20	Classification accuracy vs walking threshold for sheep behaviour. . . . .	96
8.21	Confusion matrix of the vibration sensor classifier: Sheep behaviour. . . . .	97
8.22	Moving average of vibration sensor interrupts: Rhino running. . . . .	98
8.23	Moving average of vibration sensor interrupts: Rhino walking. . . . .	99
8.24	Moving average of vibration sensor interrupts: Rhino standing. . . . .	99
8.25	Moving average of vibration sensor interrupts: Rhino laying down. . . . .	100
8.26	Classification accuracy vs running threshold for rhino behaviour. . . . .	100
8.27	Confusion matrix of the vibration sensor classifier: Rhino behaviour. . . . .	101
8.28	Confusion matrix of the vibration sensor classifier, using a 6000 point moving average: Rhino behaviour. . . . .	102
8.29	Finger placement for pulse oximeter measurements. . . . .	103
8.30	Pulse oximeter on the ear of a lamb. . . . .	104
8.31	Pulse oximeter output as measured on the ear of a lamb. . . . .	104
8.32	Single heart beat of a rhino. . . . .	106
8.33	26 Heart beats of a rhino. . . . .	106
8.34	Spectrogram of a rhino's pulse rate at 156 beats/min. . . . .	107
B.1	MSP430FR5739 PCB footprint. . . . .	128
B.2	MicroSD card holder PCB footprint. . . . .	129
B.3	FRAM PCB footprint. . . . .	129
B.4	GNS602 PCB footprint. . . . .	130
B.5	ADXL345 PCB footprint. . . . .	130
B.6	SQ-SEN-200 PCB footprint. . . . .	131
B.7	MiniSense 100 PCB footprint. . . . .	131
B.8	ADMP504 PCB footprint. . . . .	131
B.9	ABS 1200 PCB footprint. . . . .	132
B.10	Contact microphone PCB footprint. . . . .	132
B.11	Samtech connector PCB footprint. . . . .	133
B.12	L6932H1.2 PCB footprint. . . . .	133
B.13	TPS61222 PCB footprint. . . . .	134
B.14	LTC3105 PCB footprint. . . . .	135
B.15	MSP-FET430UIF female connector PCB footprint. . . . .	135
C.1	WildMote PCB layout. . . . .	136
C.2	RF Shield PCB layout. . . . .	137
C.3	GSP Expansion Card PCB layout. . . . .	138
D.1	WildMote schematic diagram. . . . .	140
D.2	RF Shield schematic diagram. . . . .	141
E.1	Casing Base drawing. . . . .	143
E.2	Casing Lid 1 drawing. . . . .	144
E.3	Casing Lid 2 drawing. . . . .	145
E.4	3D view of the casing. . . . .	146

# List of Tables

8.1	RAW data stored on the microSD card. . . . .	79
8.2	Running behaviour observation time stamps on the <i>4th</i> of July 2014. . . . .	80
8.3	Table of classification statistics: Sheep running. . . . .	82
8.4	Table of classification statistics: Sheep walking. . . . .	82
8.5	Table of classification statistics: Sheep grazing. . . . .	83
8.6	Table of classification statistics: Sheep standing. . . . .	83
8.7	Table of classification statistics: Sheep laying down. . . . .	83
8.8	Table of classification statistics: Rhino running. . . . .	84
8.9	Table of classification statistics: Rhinos walking. . . . .	86
8.10	Table of classification statistics: Rhino standing. . . . .	86
8.11	Table of classification statistics: Rhino laying down. . . . .	87
A.1	Observations made on 03_07_2014 . . . . .	119
A.2	Observations made on 04_07_2014 . . . . .	120
A.3	Observations made on 05_07_2014 and 06_07_2014 . . . . .	121
A.4	Observations made on 06_07_2014, 07_07_2014 and 08_07_2014 . . . . .	121
A.5	Observations made on 10_07_2014 . . . . .	122
A.6	Observations made on 12_07_2014 and 13_07_2014 . . . . .	123
A.7	Observations made on 14_07_2014 . . . . .	124
A.8	Observations made on 16_07_2014 . . . . .	125
A.9	Observations made on 17_07_2014 . . . . .	126
A.10	Observations made on 17_07_2014 and 18_07_2014 . . . . .	127



# Nomenclature

## Constants

$$c = 299792458 \text{ m/s}$$

$$g = 9.80665 \text{ m/s}^2$$

## Variables

$$\lambda \quad \text{Wavelength} \dots\dots\dots [\text{m}]$$

$$d \quad \text{Distance} \dots\dots\dots [\text{m}]$$

$$f \quad \text{Frequency} \dots\dots\dots [\text{Hz}]$$

## List of Abbreviation

ABBMS Animal Borne Behaviour Monitoring System

ABC Anonymous Best-effort Single-hop Broadcast

ADC Analog to Digital Converter

BS Base Station

CCA Clear Channel Assessment

CCS5 Code Composer Studio 5.5

DMA Direct Memory Access

DSP Digital Signal Processing

DT Decision Tree

EH Energy Harvesting

FIFO First In First Out

FIT Future Internet of Things

FRAM Ferroelectric Random Access Memory

GNS Global Navigation Systems

GPS Global Positioning System

IBC Identified Best-effort Single-hop Broadcast

IO Input-Output

IoT-LAB Internet of Things LAB

IPOLITE Identified Polite Single-hop Broadcast

kNN k-Nearest Neighbour

KNP Kruger National Park

LED Light Emitting Diode

LDA Linear Discriminant Analysis

LLN	Low power and Lossy Networks
LSB	Least Significant Bit
MH	Best-effort Multi-hop Unicast
MPPC	Maximum Power Point Control
NF	Best-effort Network Flooding
OABCS	On-animal Behaviour Classification System
PC	Personal Computer
PCB	Printed Circuit Board
POLITE	Polite Single-hop Broadcast
PREDNET	Predator Network
RAM	Random Access Memory
RF	Radio Frequency
RMH	Hop-by-hop Reliable Multi-hop Unicast
RP	Repeater
RTC	Real Time Clock
RUC	Reliable Single-hop Unicast
SPI	Serial Peripheral Interface
SSCK	Synchronous Serial Clock
SSI	Synchronous Serial Input
SSO	Synchronous Serial Output
SSU	Synchronous Serial Communication Unit
STUC	Stubbon Single-hop Unicast
TFF	Time to Firt Fix
TI	Texas Instruments
TG	Tag
UART	Universal Asynchronous Receiver/Transmitter
UC	Best-effort Single-hop Unicast
UDP	User Datagram Protocol
ULDO	Ultra Low Drop Out
ULL	Ultra-Low-Leakage
VHF	Very High Frequency
WSN	Wireless Sensor Network
QDA	Quadratic Discriminant Analysis

# Chapter 1

## Introduction

### 1.1 Background to the Project

This thesis forms part of a project called Predator Network (PREDNET) which is a research collaboration between the University of Stellenbosch and the research institution Inria, based in Lille, France. PREDNET has been established to research the most suitable topology and subsequent deployment of a wireless sensor network for sparsely populated outlying rural and wilderness areas, for effective monitoring and protection of resources and ecosystems [1].

PREDNET has various research focus areas and associated researchers. Figure 1.1 shows the four main research areas of the project. The Hardware Design team is responsible for designing all the hardware platforms required by the project. This includes designing sensors, wireless sensor motes, wireless sensor networks (WSNs), all software components needed to configure the motes, collect data and manage the WSN. The Energy Harvesting team investigates effective ways to harvest energy that will assist the Hardware Design team to extend the lifetime of the motes. The Digital Signal Processing (DSP) team analyses and processes the data collected by the Hardware Design team by means of various machine learning techniques. The Network Protocols team develops new routing protocols tailored for the WSNs used within the PREDNET project. The work described in this thesis relates to several aspects of the greater project. A significant portion entailed hardware and software design, but a first network protocol was also implemented. The same applies to some DSP techniques imbedded in the firmware.

This thesis is a natural result and response to the urgent and real crisis regarding the world wide rhino poaching dilemma. Rhinos across the globe are suffering from an

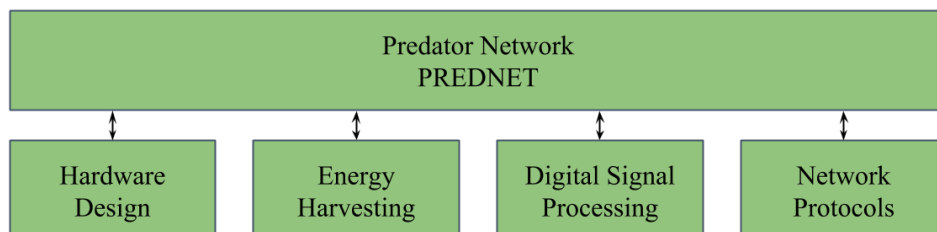


Figure 1.1: PREDNET project structure.

immense onslaught from rhino poachers. The latter show no regard for the remaining rhino species. Rhinos are being killed for their horns which are smuggled across borders in illegal trade. Nature conservationists, biologists, engineers and various other groups are going to extreme lengths to save the remaining rhino species, even risking their own lives. Despite all the efforts, rhino poachers still have the upper-hand and the rhino population is very rapidly decreasing.

Wildlife trafficking is one of the biggest forces driving many wildlife species towards extinction. In the last couple of years this has raised many international concerns and various solutions to the problem have been discussed among the world's most important conservation bodies. A report published by The White House [2] in early 2014 stated that, "wildlife trafficking undermines security across nations" [2]. The report further mentioned that among various other endangered wildlife species, elephant and rhino are seriously affected by poachers. The latter are two of Africa's most iconic animals and a major reason for tourism to the continent.

South Africa is home to most of the world's population of both Black Rhinoceros (*Diceros bicornis*) and White Rhinoceros (*Ceratotherium simum*). In the year 2012 approximately 75 % of the world's entire rhino population could be found in South Africa [3]. This might sound impressive, but the situation is not comforting at all. Historically, 85 % of the global rhino population was lost during the years 1970 and 1987. During this period approximately 11000 rhinos survived, while 59000 perished due to poaching [3]. Black rhino numbers declined from 60,000 in 1975 to 4,500 in 2010 [3]. The present situation regarding rhino poaching is certainly not better at all. From the year 2007 to 2015 South Africa has lost thousands of rhinos due to poaching [4]. At the current rate, very few or no rhinos will soon be left.

The Kruger National Park (KNP) is South Africa's largest national park and home to most of South Africa's rhinos. The location of the KNP in the North East of South Africa, is shown in Figure 1.2. Most, but not all, rhino poaching incidents occur in the KNP. Nature conservationists and rangers are actively running anti-poaching operations in order to fight the illegal activities. However, despite these efforts, the problems regarding the illegal wildlife trade are extremely complex and cannot be resolved by a single solution. Rhino poaching will only truly be limited when various different parties like politicians, nature conservationists, scientists, engineers and local communities join forces to combat the crime.

Nature conservationists are starting to use more technologically advanced counter-measures. The rhino poaching crisis has gained international exposure leading to more innovative ideas to combat the crime. One prominent technique utilised to assist nature conservationists, is tracking the animals. The current tracking equipment is expensive and depending on the particular technology, it can sometimes be very difficult to track the rhinos within the KNP and other similar areas in an effective manner. Although the localization of animals is a key metric for scientists and nature conservationists, we believe that it can be combined with animal behaviour in an effective manner to provide more insight in the well being of the animal. A system that can monitor the behaviour and various stress level indicators of animals may prove to provide very interesting insights, both as new data for scientists and as key information in running anti-poaching



Figure 1.2: Kruger National Park within South Africa.

operations. Collective, or group monitoring, coupled with pattern recognition techniques might also yield some very useful inputs into the management process.

It is clear that a technological platform needs to be developed to assist nature conservationists in their protective efforts. This project was initiated to develop a prototype Animal Borne Behaviour Monitoring System (ABBMS), as part of a greater technical biological information gathering platform. The ABBMS constitutes the design and development of a broader communication system comprising base stations, repeaters and animal borne sensing devices (known as WildMotes). The latter would all be connected through a multi-hop WSN, through which all WildMotes can send data to a base station (or server). The collected data, such as raw sensor measurements, animal behaviour and key stress level indicators of animals can then be used for further processing and interpretation, by biologists and conservationists.

## 1.2 Project Objectives

Further to the above, the objectives of this project, were defined as follows:

- To investigate previous work done in related areas
- To investigate a suitable system configuration for the ABBMS
- To design and develop the hardware for the ABBMS
- To design and develop the software for the ABBMS
- To implement a lightweight multi-hop WSN protocol for the system
- To provide design considerations for energy harvesting systems for animal borne applications

- To field test the system and evaluate the acquired measurements and results
- To recommend possible system improvements for future research efforts

### 1.3 Project Summary

This project entailed the design and development of a prototype ABBMS. The ABBMS consisted of a base station, repeaters and animal borne sensing devices known as WildMotes. The system enabled the WildMotes to communicate various sensor measurements to a base station, by means of a multi-hop WSN that exists between WildMotes, repeaters and the base station. The network coverage of the ABBMS was expanded by the implementation of two different multi-hop networking protocols, RIME and RPL. RIME proved to be a lightweight network protocol, that can easily be implemented on limited resource microcontrollers.

The hardware design of the WildMotes, included various components such as an ultra-low power microcontroller, Global Positioning System (GPS), accelerometer, nano-power tilt and vibration sensor, Radio Frequency (RF) communication module, microSD card and Ferro-electric RAM (FRAM) modules. Apart from the hardware design, the project also dealt with the software design of the ABBMS, including the WildMotes and their deployment framework. The initial goal of the WildMotes was to collect data to be used in the automatic behaviour classification of animals, by means of computer based techniques. Initial tests were done on sheep, due to their availability and tame nature. Further tests were carried out on rhinos. The WildMotes were successfully configured to log the accelerometer and nano-power vibration sensor values at sampling frequencies of 100 Hz and 40 Hz for sheep and rhinos, respectively. Tests on rhinos also included the successful collection of GPS coordinates every 3 min. The GPS coordinates were utilised to reveal their repetitive movement patterns, by means of a heat map. During the various experiments, the rhinos and sheep were recorded by means of video cameras. The recordings served as ground truth data for the behaviour classification process. The data and recordings were processed, by the DSP group, with techniques such as linear- and quadratic discriminant analysis and decision trees, to successfully classify the behaviour of rhinos and sheep with high accuracies. The behaviour of sheep could be classified as either running, walking, standing, laying down or grazing with respective precision of 99.5 %, 93.7 %, 95.2 %, 93.5 % and 66.4 %. For rhinos, the implementation of a linear discriminant analysis classifier resulted in an overall accuracy of 85.98 %, with an accuracy of 93.33 %, 98.97 % and 65.64 % for standing, walking and laying down respectively. Furthermore, decision tree classifiers resulted in an overall accuracy of 95.81 %, with an accuracy of 99.49 %, 97.18 % and 90.77 % for standing, walking and laying down respectively.

The definition of the abovementioned classifier algorithms was part of a research team effort. This project then implemented these algorithms on the Wildmotes to obtain an on-animal behaviour classification system (OABCS). The OABCS was able to accurately calculate and transmit the behaviour of animals, in real time. The live updates of animal behaviour is a great step forward from previous computer based techniques, which classified animal behaviour based on historical data. In addition to the OABCS, a nano-power tilt and vibration sensor was successfully applied as a low power alternative, to classify the behaviour of both rhinos and sheep. It was further utilised to extend the battery

life of the WildMotes. Further investigations regarding pulse rate measurement on rhinos were also carried out. The use of contact microphones proved to be more effective than pulse oximeters. The particular outcomes and contributions of this project are set out in the next section.

## 1.4 Project Outcomes and Contributions

The project realised the following outcomes and contributions:

- The design, implementation and testing on both sheep and rhinos, of on-animal Wildmote sensing devices. These devices present a state of the art hardware design, as could be satisfactorily confirmed during a recent visit to the network group at Trento University, in Italy. The units and prototype packaging have also proved suitably robust during field trials.
- The Wildmotes were actively tested in field trials, where acceleration- and vibration sensor data were sampled and continuously logged for both sheep and rhino. In the case of rhinos, the ABBMS' networking capabilities were further utilised to transmit GPS coordinates to a base station, every 3 min.
- Two different multi-hop networking protocols were implemented to prove that it can provide the WildMotes with an effective way to route data to the base station and at the same time increase the network coverage of the ABBMS.
- The collected data was successfully used in different implementations of animal movement/behavioural classifiers. For rhinos, the implementation of decision tree classifiers resulted in an overall classification accuracy of 95.81 %, with an accuracy of 99.49 %, 97.18 % and 90.77 % for standing, walking and laying down respectively. Video recordings were made to verify classifier performance.
- The definition of the classifier algorithms was part of a research team effort, but a challenge was presented in this project to implement these techniques in a resource constrained low power microcontroller environment. The success thereof depended on careful processing and memory management, data set reduction and classification system state management.
- The ABBMS successfully implemented an OABCS. The live transmission of on-animal classified behaviour has not been done before, to the best of our information and could make a significant contribution to research and real time abnormal behaviour recognition. The implementation of the OABCS on WildMotes, provided live updates of animal behaviour every 5.3s, as opposed to post processing, computer based techniques, on historical data. In the case of sheep it could accurately distinguish between running, walking and passive behaviour, with respective precisions of 76.92 %, 80.15 % and 90.03 %.
- An alternative and simplified route to effective classification and particularly battery life extension, was also explored. It was proved that data collected from a nano-power tilt and vibration sensor, could be utilised to this effect. A single decision tree was implemented using moving averages from the simple sensor and overall classification accuracy was acceptably close to the results from the full accelerometer

approach. For sheep, this technique could accurately distinguish between running-, walking- and passive behaviour, with a precision of 99.13 %, 91.77 % and 72.68 %, respectively. In the case of rhinos, it distinguished between running-, passive- and walking behaviour, with a precision of 100 %, 99.6 % and 30.29 % respectively. The battery life was extended from an estimated 47 to 270 d. This is important in an application type where battery life is a prime concern.

- An exploratory audio based recording of rhino pulse rate was made in order to correlate this with other behavioural information, as a step in the process to provide the necessary stress related information to biologists.

## 1.5 Structure of this Thesis

The rest of this thesis, is structured as follows: Chapter 2 considers previous work done in related areas and the associated methods. Chapter 3 considers the design requirements and system formulation for the ABBMS. Chapter 4 covers the hardware design of the WildMotes, whereafter the software design for the WildMotes is explained in Chapter 5. Chapter 6 considers two different management routing protocol implementations for the WSN. Chapter 7 provides design considerations for possible future energy harvesting systems in WildMotes. Chapter 8 analyses the measurements and results of the system whereafter the thesis concludes with some final remarks and recommendations in Chapter 9.



# Chapter 2

## Literature Study

### 2.1 Previous Work Done in Related Areas

A technological platform should assist scientists and nature conservationists in the acquisition of general behavioural and other biological information about rhinos. Such information should be useful for purposes of basic research as well as providing valuable input into the planning and execution of anti-poaching operations. Before commencing with the required systems design, it is necessary to review the availability and effectiveness of currently available anti-poaching techniques and technologies. A recent study done by Privilege Cheteni evaluated the current anti-poaching techniques being used in Africa and their effectiveness to protect rhinos [5]. According to the study, many African governments have tried various anti-poaching and conservation techniques, some of which were very expensive operations. However the study suggests that the battle have been lost to a certain extent and that rhino populations are moving towards extinction [5]. This is mainly because of the complexity of the organized crime network of the poachers. According to Rademeyer [6] and [5]: “poaching syndicates are multinational and are known to be involved in high risk criminal activities such as diamond smuggling, drugs, vehicle theft and armed robberies”. Furthermore, while poachers have a \$400 000 incentive to poach a rhino, the people protecting the rhinos do not have the same incentives.

#### 2.1.1 Current Anti-Poaching Techniques

A summary of the effectiveness of anti-poaching techniques are as follows. The dehorning and translocation of rhinos in targeted areas proved to be effective, if and only if, the security of the dehorned and translocated rhinos are increased. Protected and dehorned rhinos are 29.1% more likely to survive as opposed to horned- or dehorned unprotected rhinos [5]. Community based conservation is a technique that makes communities who are affected by rhino poaching part of the solution. At least one example of community based conservation was implemented successfully among Maasai people. They received financial support from the national park and as a result, they became protective towards the wildlife in the park [5]. Despite of this example, governments are afraid that the implementation of community based conservation will jeopardize the tourism industry. Furthermore they are afraid that diseases may spread to the wildlife as a result of human and livestock intrusion as were the case in previous community based conservation efforts [5]. Similar to community based conservation, wildlife management policies can

be an effective technique if governments promotes the locally managed commercial use of wildlife. However in 1993, this techniques failed miserably when a change of wildlife policy resulted in budget cuts affecting the national parks and caused poachers to slaughter rhinos to a point where only six white rhinos were left in a specific area of Zimbabwe [5].

The study mentions that law enforcement and patrols play a big part in protecting rhinos and the increase thereof can reduce poaching in the long run. However, the lack of law enforcement and patrols results in a doubling of poaching. Furthermore changes in law enforcement should be done regularly and can stop poaching in areas for many years as was the case in Nepal [5]. This is mainly because poachers get to know the law enforcement system and exploit loopholes to get away with illegal activities. Changes in law enforcement can prevent poachers from finding various flaws in the system. Penalties and the imprisonment of poachers forms part of law enforcement strategies, but has proved to have very little success, because poachers can easily pay the fines from previous illegal horn trading profits. It is however more effective to vary penalties based on the number of horns confiscated from a poacher, as opposed to fix penalties [5].

The study suggests the use of new forms of anti-poaching techniques. One such technique is based on a shoot to kill policy and has proved to be the only effective way to quickly reduce rhino poaching as poachers fear for their lives [5]. However this technique involves human rights issues and is, therefore, not wildly accepted and implemented. A new technological approach in South Africa aims to link the criminal to the crime by means of extensive DNA collection and profiling techniques. A Rhino DNA Index was developed by the Veterinary Genetics Laboratory of the Faculty of Veterinary Science at the University of Pretoria (Onderstepoort) [7]. In early 2015, I was part of a rhino conservation effort in Mpumalanga, South Africa. On the day, two white rhinos were dehorned and DNA samples collected and recorded from each rhino. The nature conservationists and veterinarians explained how various DNA samples are taken from each rhino including shaving of the horns and hair samples. The samples are carefully packaged and linked to unique microchips implanted into very specific parts of the rhinos. This samples are then taken to the Veterinary Genetics Laboratory where it is analysed and fed into their database. The database of DNA samples allows law enforcers to link specific criminals to specific rhino poaching incidents and increases the probability of successful prosecution. In general the nature conservationists and veterinarians said that they were positive about the outcomes of this technique, but new strategies need to be used alongside the DNA database to help them fight against poaching. The study [5] further mentions that the use of drones (or unmanned aerial vehicles) can significantly reduce the number of poaching incidents, because of their video recording abilities, which can later be used as evidence in law enforcement efforts, if poachers were caught on the crime scene. Groundbreaking work has been done by the Lindbergh foundation [8] in 2015. Although no formal research papers could be found, online articles and video documentaries of the Lindbergh foundation mentioned a project called Air Shepherd which combines drones with advanced predictive algorithms, geographical data and historical poaching incidents to predict where poachers might be [8]. The system then calculates flight routes for the drones while people in control rooms analyses the video footage collected by the drones. The latter seems like a very promising approach to fight against poaching. Drones in general are expensive and require some form of training before they can effectively be operated. This makes it hard to deploy drones in areas where they are required. Finally, the study also considers

the use of radio tags or collars as a means of assisting anti-poaching operations. Various types of collars exist for animal tracking. They range from the classic VHF collars where tracking is done manually by means of directional antennas, to the most advanced satellite based Global Positioning System (GPS) tracking collars. The study states that radio collars were used with very little success in protecting rhinos, mainly because of various problems with the collars [5]. Some of these problems include false transmissions and poor collar design resulting in injuries to the rhinos. Tranquillisation of the rhinos to attach the collars is also a very expensive and risky procedure and can lead to fertility problems and death of the animals.

### 2.1.2 Current Collars and Tracking Devices

An extensive survey was done by [9], to evaluate the performance of various collars and tracking devices used on animals. The study [9], considered the functional performance of 330 radio collars and tags. The study mentions that 49% of the collars malfunctioned before the end of their anticipated life. From the study it is clear that Very High Frequency (VHF) collars were used in the 1960s as the first form of animal tracking. Ultra-high frequency (UHF) became of use in the early 1970s and communicated to satellite constellations such as the Argos constellation. Calculating the exact location of animals based on the latter technique had its own difficulties and resulted in large errors, typically  $> 1000$  m [9]. The use of GPS collars came naturally in 1994 and provided precise localisation information. The data collected had to be retrieved by means of VHF or UHF transceivers [9]. This proved to be a difficult task, as animals first had to be located, whereafter the data could be retrieved, if you remain close enough to the animal. Later GPS collars allowed data to be uploaded to a computer by means of satellite constellations. More recent collars allowed data to be sent over local cellular networks [9].

Evaluating the performance of the different collars showed that the use of VHF collars were quite successful due to the simplicity of design. It provided very limited information on the animals exact location, as animals had to be tracked using directional antennas. Difficulties involved initially locating the animals, as you have to be within a specific range from the animal to detect the signal. The performance of Argos-only satellite collars or tags is quite successful, but only at high altitudes or elevations. It is therefore only recommended for studies on birds [9]. As mentioned earlier, large localisation errors occur when using this technique. GPS-Argos collars provide exact localisation information, but failed to upload even half of the collected data to the Argos constellation. This was a major problem as animals had to be located after the useful life of the collars to retrieve the collars and manually download the data [9]. Collars uploading data to the Iridium satellite network generally did not have a problem with the uplink of the data, but the collars never achieved their expected lifespan of 24 months. GPS collars which relied on VHF or UHF links to download the data, generally managed to do so successfully, but these collars also stopped before the expected lifespan [9]. Collars who used local cellular networks, communicated their data to a server with great success, despite the fact of early battery drainage [9]. The availability of cellular networks are, however, normally very limited in rural areas where wild animals naturally occur.

Home range estimation, migration and movement patterns and habitat preference are the primary results of telemetry studies; and are mainly concerned with the location of animals [9].

### 2.1.3 Animal Behaviour Classification

Animal behaviour classification is another developing research area. Animal behaviour classification by means of triaxial accelerometers are the research focus of a small number of researchers with little literature being available in the public domain. However, the popularity of the field is growing and some researchers have done great work in the animal behaviour classification field yielding good results.

In December 2012 an experiment done at the Tasmanian Institute of Agriculture Dairy Research Facility in Northwest Tasmania, focused on monitoring the behaviour of dairy cows [10]. According to the study, the behaviour of the animals can directly be related to the wellbeing of the animals and can also serve as an early indicator of specific diseases. It can further give an indication of pain experienced by the animal and the social interaction of the animals within the herds. The study “determined whether introducing supplements into the diet of cows would reduce their pasture intake” [10]. The study considered six behaviour classes namely resting, grazing, walking, searching, ruminating and head down. They were able to accurately classify three of the six classes by means of Linear Discriminant Analysis (LDA), binary Decision Tree (DT) and k-Nearest Neighbour (kNN) classifiers [10].

In September 2013 a study was performed to determine if adequate data can be collected to perform animal behaviour classification using a triaxial accelerometer placed on a dog [11]. According to the study, animal behaviour classification is important to study the effects of medicines on animals. The hardware described in [12] was placed on a 5-year-old male Staffordshire terrier and attached using a velcro strap. The accelerometer was sampled at 40 Hz with a  $\pm 2$  g resolution and the data was sent directly to a computer [11]. The study mentioned that the collected data could be used for behaviour classification in future studies, but the study did not implement any classification techniques. Finally the study proposed a segmenting algorithm that can be used prior to classification to separate periodic movements from non-periodic movements [11].

In December 2013 a study used accelerometers to investigate the behaviour of sheep to determine when sheep are grazing. Furthermore the study mentioned that future work will combine localization information with the behavioural information to exactly determine what the sheep are eating [13]. Hardware as described in [13] was placed in a food like container and attached to the sheep using velcro straps. The accelerometer data was directly sent to a computer by means of a WSN. From information gathered in the study, the data was manually classified by means of interpretation (using techniques as described in [14]) of the accelerometer data and no automatic behaviour classification techniques were used [13]. The manual classification proved to accurately distinguish between attaching the collar, standing, grazing, browsing and laying down [13].

A more recent study used a triaxial accelerometer to classify the behaviour of a tame Eurasian badger (*Meles meles*) [15]. A collar was placed around the neck of the badger and the accelerometer sampled at 25 Hz with a resolution of  $\pm 8$  g. The animal was filmed for a total of 45 min while the animal roamed freely. After downloading the collected data to a computer, the study used automatic behaviour classification techniques based on a supervised non-parametric k-NN machine learning algorithm and a decision tree to classify the data [15]. The accuracy ranged between 77.4% and 100% depending on the behaviour (walking, trotting, snuffing and resting) and classification method employed [15].

A few other studies also considered techniques for the automatic behaviour classification of animals. The first of which considered “Multiclass Semi-Supervised Learning for Animal Behaviour Recognition from Accelerometer Data” [16] and finally [17] considered the “Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system”.

The above mentioned studies are examples of good work that have been done in the field of the automatic behaviour classification of animals, however all the classification techniques required the data to be downloaded to a computer where advanced processing and machine learning techniques could be used to perform the classification. These techniques required large amounts of external storage where the accelerometer data could be stored for post processing. The information gathered from these classification techniques holds historical value. To the best of our knowledge, no system could provide live updates of animal behaviour, in real time. The next chapter sets out the design requirements for a system that can provide, among others, live updates of animal behaviours.

## Chapter 3

# Design Requirements and System Formulation

The investigations as set out in Chapter 2, provided a reasonable overview of the status regarding previous and current work in this area. It also provided insight in the commonly encountered engineering and related problems, as well as guidance in the definition of a set of systems design requirements towards a possible solution. This chapter will deal with the proposed methodology to achieve this objective, with its constituent subsystems.

### 3.1 Design Requirements

Based on the problems described in Chapter 1 and the research done in Chapter 2, it is evident that a technological platform is required that will assist engineers and scientists in better understanding rhinos and their behavioural patterns. This will provide nature conservationists with better information to run strategic anti-poaching operations. Such a technological platform (from now on referred to as the system or the Animal Borne Behaviour Monitoring System (ABBMS)) would be subject to various design limitations and design requirements.

From Chapter 2, it is apparent that no single solution without killing poachers, have had a great effect on rhino poaching and the rhino population is ever decreasing. It can therefore be of great interest to develop a system that can provide scientists and nature conservationists with more information to hopefully take a step in the right direction. This thesis proposes to investigate and design a system that will collect specific stress level indicators and behavioural patterns of rhinos and present this information to scientists in such a way that more effective decisions can be made to benefit the conservation of rhinos. It is important to note that we believe no single solution can solve the rhino poaching problem and that various techniques have to be combined to reach the same goal. We would hope that the outcomes of this project can be used in further research and in other anti-poaching systems to fight against poaching. We ideally would hope to at least develop a technological platform that can provide information regarding a rhino's exact location and the accelerations experienced by a rhino's back leg while roaming undisturbed in its natural environment. Furthermore we are interested to investigate ways of collecting the pulse rate of rhinos. Finally it is of great interest to investigate the possibility of measuring the behavioural state of the animal in its natural environment without the need of retrieving attached collars to download large amounts of data. By doing so scientist can

start to build a database of normal and abnormal behavioural patterns of rhinos. This information can be very valuable for scientists to study the behaviour of the animals. It can also be used with other anti-poaching techniques such as drones, where the behaviour of the animal with its exact location can be fed as inputs into advanced predictive algorithms to increase the probability of poacher detection. If abnormal behaviour is detected the drones can fly to the animal's location to investigate the situation. If poachers are detected the video footage will testify against them in court. This technique will also help parks with limited resources to better deploy resources and to focus on animals showing abnormal behaviour.

This thesis proposes to design a prototype ABBMS where the behaviour monitoring is done in real-time on the animal and do not require large amounts of external storage or large processors and advanced processing techniques, to perform the task.

The design limitations are defined by the physical dynamics and operation conditions of the system. With a bit of background knowledge on how rhinos go about everyday life, one can better derive and understand the various system constraints. White rhinos are the world's second largest land mammal after the African Elephant and have masses of up to 2700 kg. They have a lifespan of between 40 and 50 years. The males are more solitary and territorial while the females are more sociable. They are rather relaxed animals as they graze in the mornings and evenings and sleep or wallow (lay in a mud bath or water pool) in the afternoons. Furthermore, they sleep for a couple of hours at night [18]. Most of South Africa's rhinos are home in the Southern parts of the Kruger National Park where the vegetation is very lush and dense. These areas have lots of trees, bushes, grass, rocks and hills, making it the ideal habitat for these animals. Rhinos often rub against rocks and trees. Furthermore, they are very muscular, rough and tough. Rhinos use their horns for self defense and to defend their territory.

Taking the above mentioned facts into consideration, the design limitations of the system starts to unfold. The device needs to be physically attached to the rhino. This can be achieved by placing a collar around the leg of a rhino. In this case, the mass of the device is not so important, but the size must remain small enough not to harm or irritate the animal. The packaging must be waterproof and very durable to withstand the rhino's attempts to break it off against a rock or tree. The device must be battery powered with a lifetime as long as possible to limit the number of times a specific animal needs to be immobilised within its lifespan to replace the batteries. The device must be able to communicate all relevant information over a wireless medium in a very cluttered environment with limited or no cellular reception to a server where the data can be processed and analysed.

The design requirements of the system including the design limitations, are further derived from the system functionality and objectives. This project aims to design a system that can monitor and record behavioural patterns and stress level indicators of animals. The system must be able to track rhinos in their natural habitat and wirelessly communicate various system messages and GPS locations to an access point. The system must be able to detect useful information that can be used as key insights for nature conservationists to run strategic anti-poaching operations and leave room to gather information that can help connect a criminal to the crime scene at a later stage.

The design limitations and requirements can be summarized as follows:

- The system will consist of various small, durable and waterproof devices (known as WildMotes) attached to rhinos,
- The WildMotes shall be battery powered and optimized for low power consumption.
- The WildMotes must be able to determine their exact GPS locations and
- monitor the behaviour of the animals.
- The system shall be able to communicate all relevant information over a wireless network of WildMotes and repeaters to an access point (known as a base station).

## 3.2 System Formulation

This section provides an overview of the ABBMS. An idealistic model of the system can be described as one which consists of very small attachable or implantable sensor tags, able to monitor the behaviour and stress level indicators of rhinos and wirelessly communicate exact location and status messages, from within the habitat of the animals, to an access point. Furthermore, the battery lifetime of the tags should be longer than the expected lifetime of the animals. With current technology the idealistic model is still far from a practical solution, but must remain the goal for all other researchers to work towards, until other approaches prove to be more effective in achieving the same goal.

Before looking at the broader system, let's consider how we can gain information from rhinos to determine their stress levels and how it might benefit scientists and engineers. Let us consider the movement, heart rate and temperature as stress level indicators.

Animal movement or behaviour can be used as a stress level indicator. Animals move according to their needs or sensory inputs. For example if a rhino is hungry, it would slowly walk around and graze from one location to the other. If a rhino is tired or warm, it would lie and rest in the shade or wallow peacefully in a mud bath. Thus relating peaceful movement to a peaceful animal. On the other hand if a rhino is being attacked by predators or a rhino is wounded after being shot by poachers, the movement would be much more aggressive and the rhino would either run for cover or charge the intruder as they often do. This aggressive movement can be associated with a rhino showing high stress levels. The movement can be examined by using a passive vibration sensor which uses virtually no power but provides limited insight in the three dimensional movement of the animal. For further insights and useful information an accelerometer can be utilised. This information can be of great use to scientists. This data can furthermore be used in machine-learning techniques to classify the behaviour of rhinos, so that the tag can automatically compute whether the rhino is laying down, standing, walking or running. If no movement is detected for a long period of time it may indicate that the rhino has died.

The body temperature of animals can be used as a stress level indicator, as it varies depending on the movement of the animal. For example, if a rhino is grazing or wallowing peacefully in a pool of water, the body temperature would be much more regulated



and constant. However, if a rhino is running either to defend its life or territory, the assumption is that the body temperature would rise while the animal is under stress. If the rhino's temperature drops below the minimum allowed body temperature, it may indicate that the rhino has died. Based on research done by D.B. Allbrook, it was determined that the rectal temperature of a white rhinoceros (*Ceratotherium simum*) ranges from 33.6 °C to 37.5 °C throughout the day and night of a free roaming rhino [19]. The Concept Husbandry Guidelines for the White Rhinoceros reports the mean rectal temperature of white rhinos at 36.8 °C with a minimum of 36.6 °C and a maximum of 37.2 °C [20]. The latter only considered the unrestrained standing position of a white rhino. The EAZA Best Practice Guidelines for black rhinoceros reports the rectal temperature of black rhinos at a minimum of 34.5 °C and a maximum of 37.5 °C [21]. Temperature sensors are very small and some are integrated into micro controller units, thus making it easy to design a small temperature sensing device.

The heart rate or pulse rate of animals can also be used as a stress level indicator. The pulse rate of rhinos vary throughout the day, based on their movement and stress levels. A normally beating heart of a rhino can be related to low stress levels and vice versa. The pulse rate of an unrestrained standing white rhino has a mean value of 39 beats/min with a minimum of 32 beats/min and a maximum of 42 beats/min [20]. D.B. Allbrook reports the pulse rate of a free roaming rhino throughout the day and night to be between 30 beats/min and 40 beats/min [19]. The EAZA Best Practice Guidelines for black rhinoceros reports the pulse rate of black rhinos to be between 30 beats/min and 40 beats/min [21]. If no pulse is detected, it may indicate that the rhino has died.

The normal behavioural patterns, body temperature and pulse rates of these majestic animals are the basis from which one can work to develop a system to monitor these various stress level indicators and hopefully derive sensible information that can aid scientist in better decision making. Although the combination of the latter three stress level indicators can be used, it is firmly believed that one can derive the same information by using only one of the indicators. This would lead to much smaller sensor tags with a longer battery life.

We will now consider the broader system as proposed. From Figure 3.1 it is clear that the communication system is divided into three parts, namely base stations (BS), repeaters (RP) and tags (TG). A WSN exists between tags, repeaters and base station(s) allowing tags to send data either directly to a base station or from one tag to the other, or to a repeater which relays the data to a base station. Base stations either have on-board data storage capabilities, or it can be connected to the cloud to allow easy remote access of data. Tags are able to send data to a base station as long as it remains within the coverage area of the network. The ideal is to expand the network coverage as far as possible by using techniques such as multi-hop routing. In short, the purpose of a base station is to provide data access to the cloud or to store data locally. The purpose of a repeater is to relay data from tags to a base station, but the purpose of tags are slightly more complex.

The tags or WildMotes are the heart of the system. It would consist of a microelectronic printed circuit board (PCB), a durable casing and straps used to fasten the casing to the back leg of a rhino. The WildMotes would have various hardware components and sensors which are used to realize different system requirements. The WildMotes would

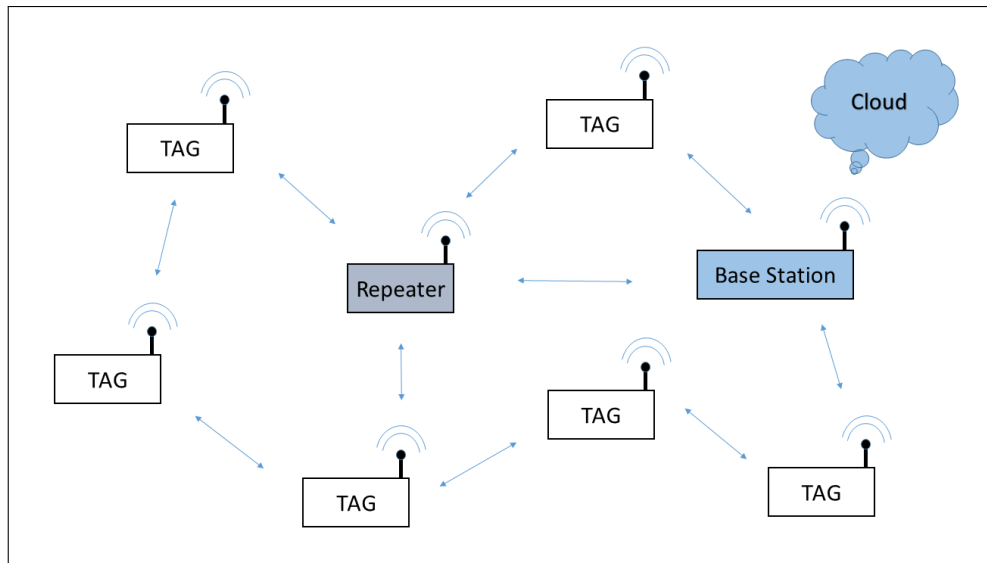


Figure 3.1: Proposed communication system.

ideally be able to monitor the behaviour of rhinos and to use this information to decide whether it needs to sound an alarm signal and transmit the exact GPS location of the rhino under stress.

This chapter considered the design requirements and system formulation of the ABBMS. It further gave a brief description of some ideal functionalities of the WildMotes, base stations and repeaters. The next chapter deals with the hardware design of the latter.

# Chapter 4

## Hardware Design

This section of the thesis deals with the hardware design of the ABBMS. The ABBMS is divided into base stations, repeaters and WildMotes. The base stations and repeaters have the same hardware design, but different software protocols controlling their behaviour. All WildMotes have the same hardware design. We will first consider the hardware design of the WildMotes, whereafter we will consider the hardware design of the base stations and repeaters.

### 4.1 WildMotes

The hardware design of the WildMotes formed a very important part of the investigation covered by this thesis. The focus was more on the functionality, adaptability and expandability of the motes, as opposed to the size of the motes. This was decided because it was still uncertain which combination of sensors would work the best for specific applications. After determining the best combination of sensors and electronic components, the motes can be optimized for size. Figure 4.1 shows the block diagram of the WildMotes. The design was divided into five basic categories namely:

- Processing unit and storage
- GPS and sensors
- Wireless communication
- Power source and regulation
- External access

For each category the component selection and design was done individually, taking the design constraints of the other categories into consideration. This eased the integration of the motes as a whole and ensured that the system worked correctly. We shall now consider the design of the various categories. Altium Designer was used to design the PCBs. Figure D.1 in Appendix D shows the complete schematic diagram of the WildMotes. The following subsections explain how the PCB design was carried out for the entire unit.

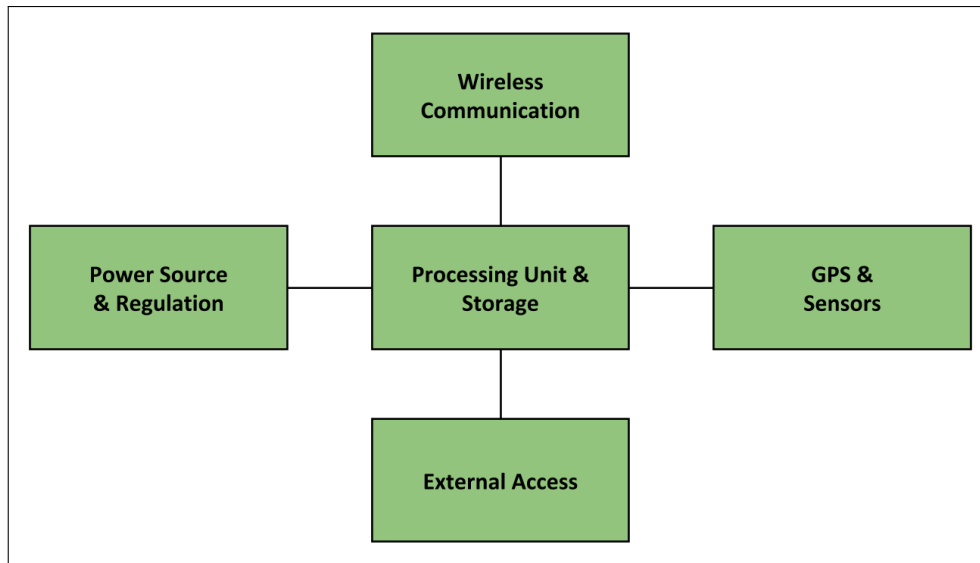


Figure 4.1: WildMotes block diagram.

#### 4.1.1 Processing Unit and Storage

The processing unit selected was the MSP430FR5739 from Texas Instruments (TI). This mixed signal microcontroller was mainly selected due to its excellent ultra-low power capabilities. From the datasheet [22] it was determined that the MSP430FR5739 has four main power modes: Active mode ( $81.4\mu\text{A}/\text{MHz}$ ), Standby mode ( $6.3\mu\text{A}$ ), Real-Time Clock mode ( $1.5\mu\text{A}$ ) and Shutdown mode ( $0.32\mu\text{A}$ ). These low power modes are optimized to achieve extended battery life in portable and wireless sensing applications [22]. It is a 16-Bit RISC Architecture microcontroller, supporting clock frequencies of up to 24 MHz. TI uses ultra-low-leakage (ULL) proprietary technology with embedded Ferroelectric RAM (FRAM) which supports up to 16 KB of nonvolatile memory and provides fast memory writes at 125 ns per Word (16 KB in 1 ms). Peripherals include 10-bit analog-to-digital converter (ADC), 16-channel comparator with voltage reference generation and hysteresis capabilities, three enhanced serial channels capable of I2C (with multiple slave addressing), SPI (at rates up to 10 Mbps), or UART (with automatic baud-rate detection) protocols, an internal Direct Memory Access (DMA) controller, a hardware multiplier, real-time clock (RTC), five 16-bit timers, and more [22]. Furthermore, the MSP430FR5739 has a wide supply voltage range of 2 V to 3.6 V. It has 40 pins and comes in small packages, typically 6.15 mm x 6.15 mm for the MSP430FR5739RHA (S-PVQFN-N40) package. Taking the above-mentioned specifications into consideration, it was clear that this was a good choice of microcontroller for this project. Figure 4.2 shows the schematic diagram of the MSP430FR5739 and how the other system components are connected to it. Figure B.1 in Appendix B shows the PCB footprint design of the microcontroller.

The onboard storage of the system was expanded by making use of a microSD card holder and two ferroelectric nonvolatile RAM modules. Figure 4.3 shows the schematic diagram of the hinged DM3CS microSD card holder made by Hirose Connector [23] and how it is connected to the microcontroller.  $C18$  is a 100 nF decoupling capacitor. Based on the electrical connection of the microSD card and the software used (mainly the memory addressing techniques) to interface the card, the system is limited to a 2 GB microSD card. However this proved to be more than enough storage for all the experiments done during

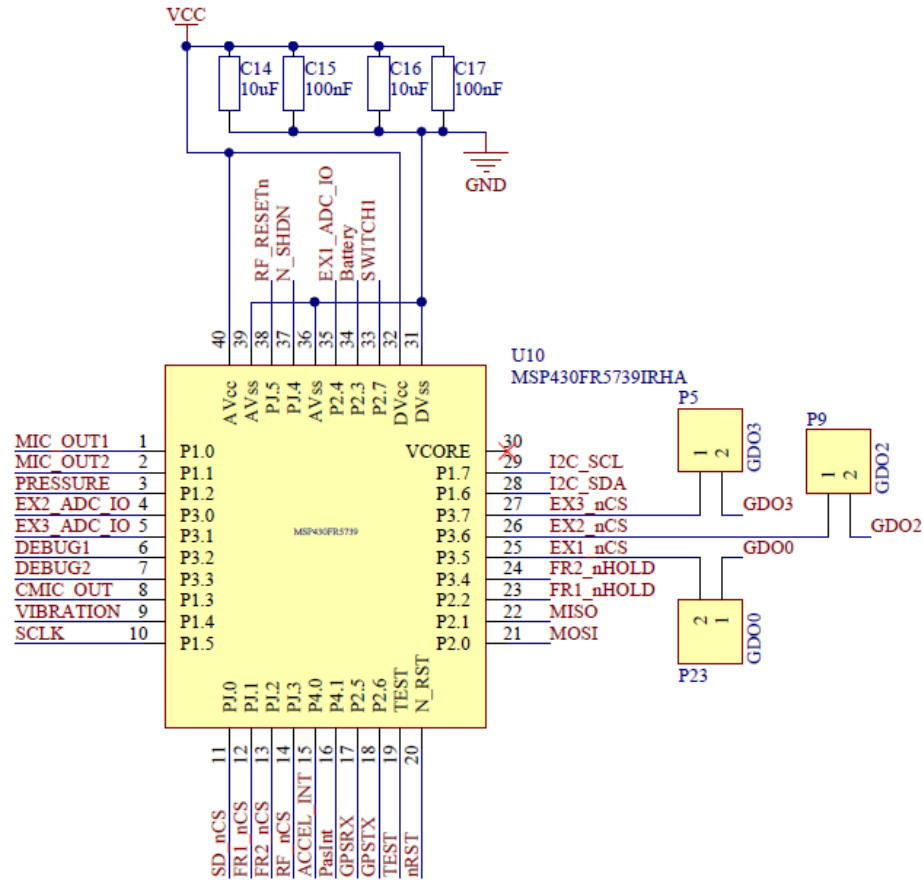


Figure 4.2: MSP430FR5739: Ultralow-power mixed signal microcontroller.

the project. Figure B.2 in Appendix B shows the PCB footprint design of the microSD card holder. Furthermore, two ferroelectric nonvolatile RAM modules of 2Mbit each were used as the fast memory access part of the system to achieve high data sample rates. The module chosen was the FM25V20 manufactured by Ramtron. From the datasheet [24] it is clear that the FM25V20 performs write operations at bus speed. No write delays are incurred. Data is written to the memory array immediately after it has been transferred to the device. Furthermore the modules can operate at very high SPI bus speeds of up to 40 MHz. It has a wide supply voltage range of 2 V to 3.6 V. The FM25V20 module consumes 100  $\mu$ A in Standby mode and 3  $\mu$ A in Sleep mode. Figure 4.4 shows the schematic diagram of the two FM25V20 modules and how they are connected to the microcontroller.  $C_{21}$  and  $C_{23}$  are 100 nF decoupling capacitors. Figure B.3 in Appendix B shows the PCB footprint design of the FM25V20 FRAM modules.

#### 4.1.2 GPS and Sensors

The following sensors were added to the system: a GPS receiver, an accelerometer, a nano power tilt and vibration sensor, a piezoelectric cantilever-type vibration sensor, an ultra-low noise analog microphone, an analog barometric pressure sensor and an external contact microphone. Some of the sensors were used in the experiments in this project and the others are available for further research.

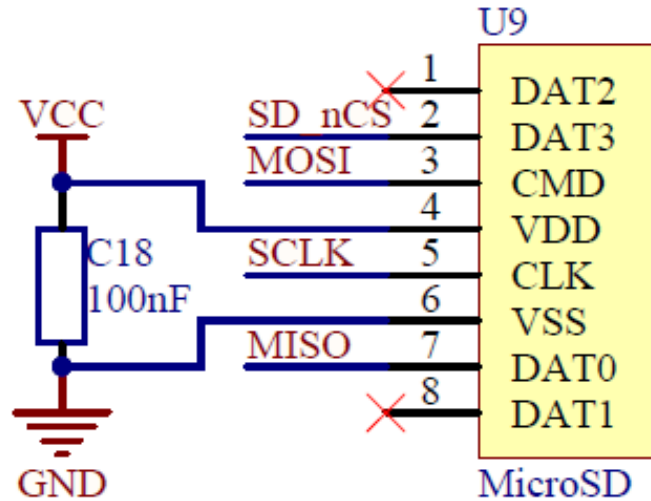


Figure 4.3: DM3CS: MicroSD card connector.

The GPS receiver used was the GNS602 manufactured by Global Navigation Systems (GNS). The two motivators for the GNS602 module were the small size of the module and the low power consumption modes. The datasheet [25] states the power requirements as 66 mW at 3.3 V (acquisition: 25 mA, tracking: 20 mA and low backup current consumption 7  $\mu$ A). The GNS602 offers the industry's highest level of navigation sensitivity up to  $-165$  dBm [25]. Furthermore it has a small form factor of 16 mm x 16 mm x 6.7 mm. Figure 4.5 shows the schematic diagram of the GNS602 and how it is connected to the microcontroller. The GNS602 communicated to the microcontroller over the serial UART channel (Baud rate of 9600, Data length: 8 bits, Stop bit: 1, Parity: None). *C19* is a 100 nF decoupling capacitor. Header pins *P6*, *P7* and *P11* are used to electrically connect and disconnect the power supply, UART receive- and transmit lines to the GNS602 respectively. This provides the ability to disconnect the GNS602 completely from the system. Figure B.4 in Appendix B shows the PCB footprint design of the GNS602 module.

The accelerometer used was an ADXL345 Digital Accelerometer manufactured by Analog Devices. The datasheet [26] states the ADXL345 has a small form factor of 3 mm x 5 mm x 1 mm. The supply voltage ranges from 2 V to 3.6 V. It's power consumption is as low as 23  $\mu$ A in measurement mode and 0.1  $\mu$ A in standby mode. The sampling is done at 13 Bit resolution at  $\pm 16$  g while maintaining a scale factor of 4 mg/LSB [26]. Figure 4.6 shows the schematic diagram of the ADXL345 and how it is connected to the microcontroller. The ADXL345 communicated to the microcontroller over the serial *I2C* bus (Bitrate: 400 Kbps). *R14* and *R15* are 4.7 k $\Omega$  pull-up resistors used to ensure good logic levels while communication takes place between the ADXL345 and the microcontroller. *ACCEL\_INT* is used to interrupt the microcontroller based on events generated by the ADXL345. The combination of *R18* (100  $\Omega$ ), *C12* (10  $\mu$ F), *C13* (100 nF) and *L3* (100  $\mu$ H) is used for the decoupling and noise reduction of the supply voltage lines of the ADXL345 and was strongly recommended by the datasheet [26]. Figure B.5 in Appendix B shows the PCB footprint design of the ADXL345 module.

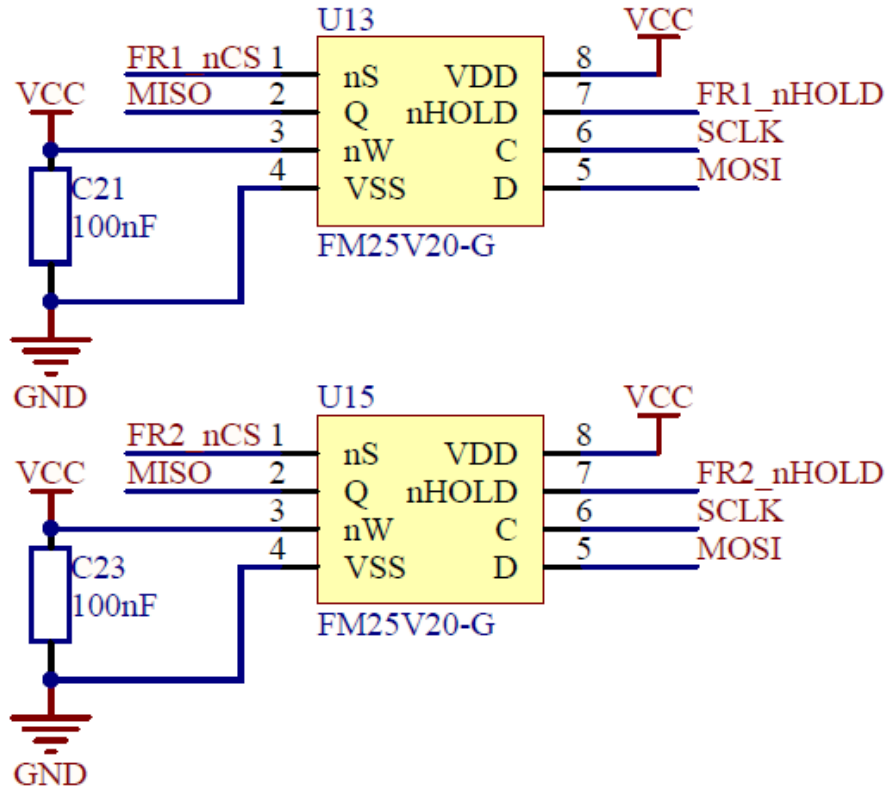


Figure 4.4: FM25V20: 2 Mbit Ferroelectric RAM modules.

A SQ-SEN-200 nano-power tilt and vibration sensor manufactured by Signal Quest was used as a means of ultralow power movement detection. According to the datasheet [27] the SQ-SEN-200 series sensor acts like a normally closed switch which chatters open and closed as it is tilted or vibrated. It consumes as little as 50 nA when triggered and has a wide supply voltage range of 0.5 V to 12 V. Figure 4.7 shows the schematic diagram of the SQ-SEN-200 and how it is connected to the microcontroller. *PasInt* is used to interrupt the microcontroller when the sensor is triggered. Passive components  $C10$  (62 pF),  $R13$  (4.75 M $\Omega$ ) and  $R16$  (1 M $\Omega$ ) were recommended by the SQ-SEN-200 application note [28] as the best choice for a simple, flexible interface circuit. Figure B.6 in Appendix B shows the PCB footprint design of the SQ-SEN-200 nano-power tilt and vibration sensor.

A MiniSense 100 piezoelectric cantilever-type vibration sensor manufactured by Measurement Specialties Inc. was implemented as a possible means of impact or movement detection. The analog output of the MiniSense 100 offers a high voltage sensitivity of 1 V/g [29]. Figure 4.8 shows the schematic diagram of the MiniSense 100 and how it is connected to the microcontroller. Header pins  $P13$  and  $P25$  can be bridged to directly connect the MiniSense 100 to the microcontroller or an external analog signal conditioning circuit can be placed in-line to ensure good signal quality for the microcontroller.  $R24$  is an 1  $\Omega$  resistor that can be changed for further signal conditioning. Figure B.7 in Appendix B depicts the PCB footprint design of the MiniSense 100 piezoelectric cantilever-type vibration sensor.

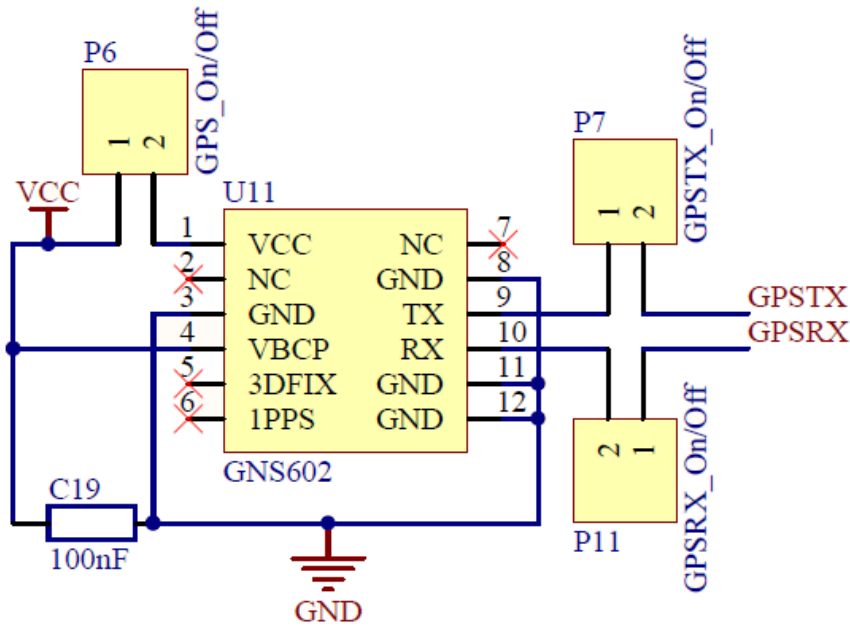


Figure 4.5: GSN602: GPS receiver.

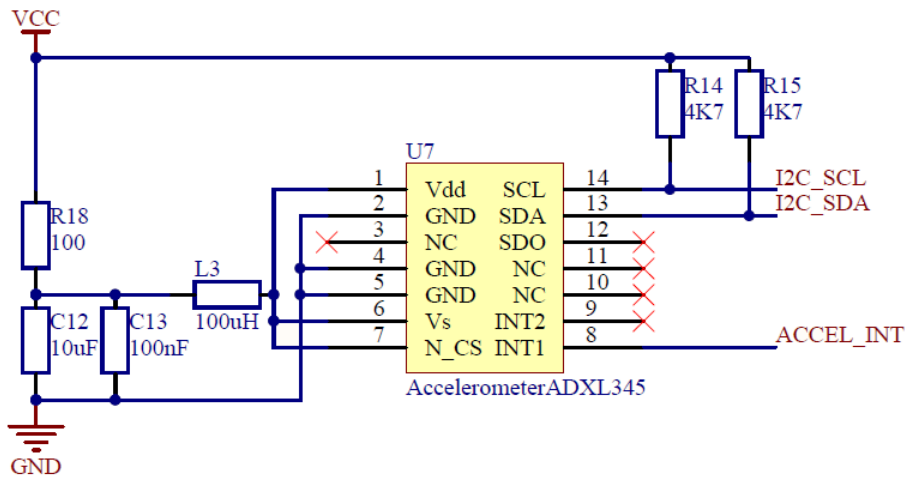


Figure 4.6: ADXL345: Digital accelerometer.

An ADMP504 ultralow noise analog microphone manufactured by InvenSense was used as a possible means of audio sensing and recording. According to the datasheet [30] the ADMP504 has a flat frequency response from 200 Hz to 15 kHz. It has dimensions of 3.35 mm x 2.50 mm x 0.88 mm. The ADMP504 has a supply voltage of 1.6 V to 3.3 V and a low current consumption of smaller than 250  $\mu$ A. Figure 4.9 shows the schematic diagram of the ADMP504. C20 is a 100 nF decoupling capacitor. Header pins P8 and P10 can be bridged to directly connect the ADMP504 to the microcontroller or an external analog signal conditioning circuit can be placed in-line to ensure good signal quality for the microcontroller. R21 and R22 are 1  $\Omega$  resistors that can be changed for further



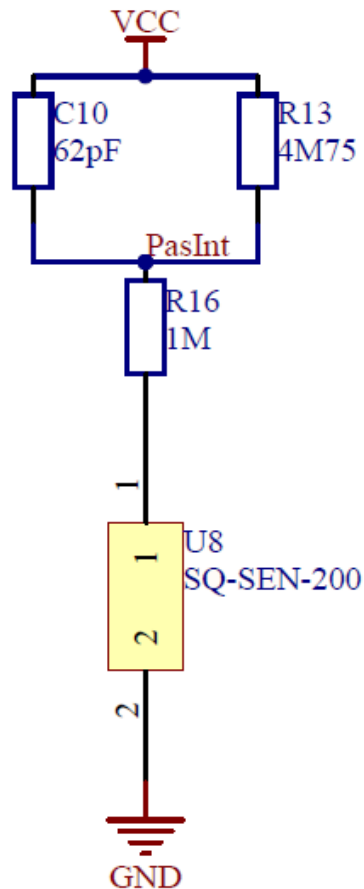


Figure 4.7: SQ-SEN-200: Nano-power tilt and vibration sensor.

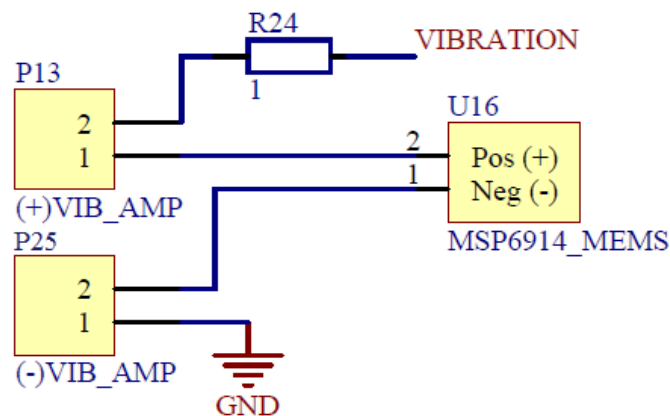


Figure 4.8: MiniSense 100: Piezoelectric cantilever-type vibration sensor.

signal conditioning. The PCB footprint design of the ADMP504 ultralow noise analog microphone can be seen in Figure B.8 in Appendix B.

An ABS 1200 analog barometric pressure sensor manufactured by Epcos, served as a possible means of altitude and pressure sensing. According to the datasheet [31] the ABS 1200 is based on a piezoresistive silicon pressure sensor and produces a voltage between

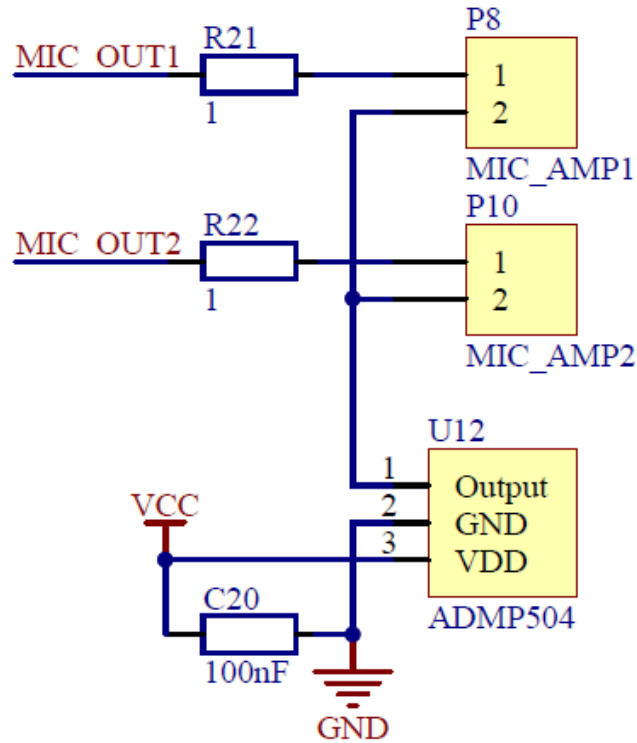


Figure 4.9: ADMP504: Ultralow noise analog microphone.

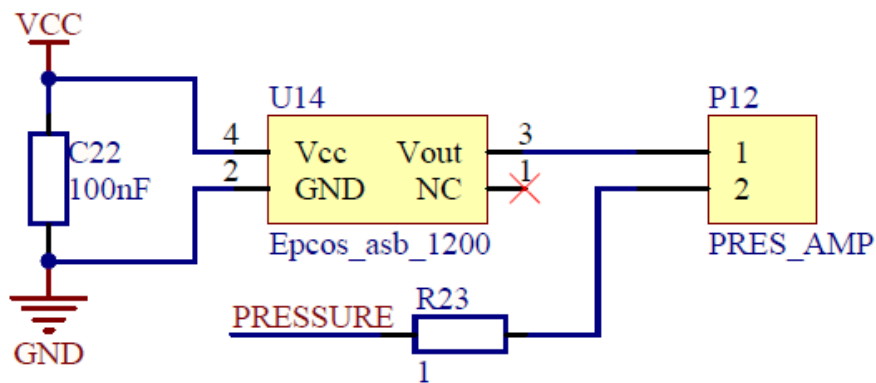


Figure 4.10: ASB1200V1: Analog barometric pressure sensor.

0 V and 1 V proportional to the pressure. The ABS 1200 has a wide supply voltage range of between 2.7 V and 5.5 V and consumes less than 5 mA. It has dimensions of 4 mm x 4 mm. Figure 4.10 shows the schematic diagram of the ABS 1200 and its connections to the microcontroller.  $C22$  is a 100 nF decoupling capacitor. Header pins  $P12$  can be bridged to directly connect the ABS 1200 to the microcontroller, or an external analog signal conditioning circuit can be placed in-line to ensure good signal quality for the microcontroller.  $R23$  is a 1  $\Omega$  resistor that can be changed for further signal conditioning. The PCB footprint design of the ABS 1200 analog barometric pressure sensor is depicted in Figure B.9 in Appendix B.

Provision was made to connect an external contact microphone or any other analog sensor that operates from 5V, to the PCB. The contact microphone selected was the CM-01B manufactured by Measurement Specialties. According to the datasheet [32], the CM-01B's design minimizes external acoustic noise while offering extremely high sensitivity to vibration applied to the central rubber pad. The CM-01B is ideal for detecting body sounds. The CM-01B has a wide supply voltage range of 4V to 30V and has low power consumption of 100  $\mu$ A. Furthermore, the CM-01B has a small form factor. Figure 4.11 shows the schematic diagram and how the sensor should be connected to U6. C11 is a 100 nF decoupling capacitor. Header pins P4 can be bridged to directly connect the CM-01B to the microcontroller, or an external analog signal conditioning circuit can be placed in-line to ensure good signal quality for the microcontroller. R17 is a 1  $\Omega$  resistor which can be changed for further signal conditioning. Figure B.10 in Appendix B shows the PCB footprint design of the solder pads provided for the CM-01B contact microphone.

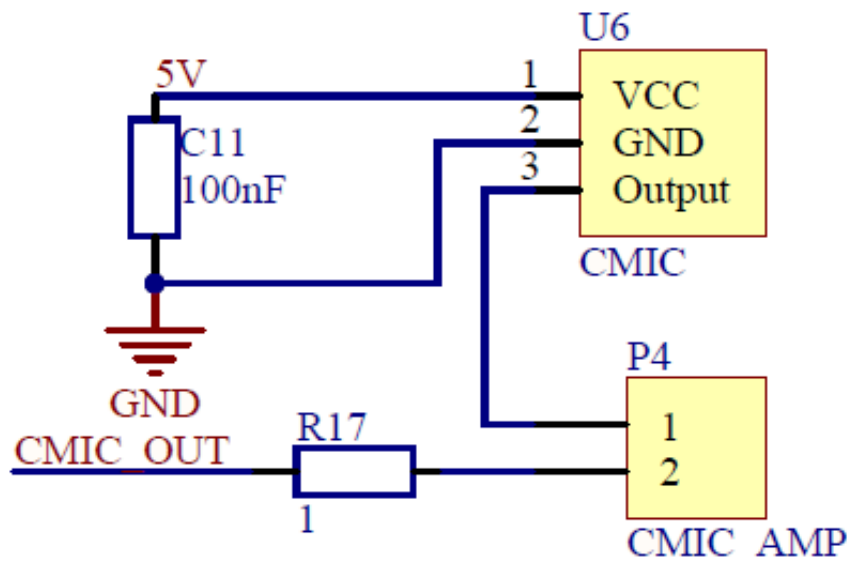


Figure 4.11: PCB contacts for external contact microphone.

### 4.1.3 Wireless Communication

The wireless communication part of the system serves as a means of wireless data collection and message delivery. A previous investigation carried out in the Stock Position Tracking and Theft Prevention System [33] [34] was performed to determine the required RF operational domain for the system. The investigation considered the following three operational frequencies: 433 MHz, 915 MHz and 2.4 GHz. Equation 4.1.1 shows the equation for Free-space path loss as obtained from the Handbook of Antennas in Wireless Communications [35]:

$$L_s(d) = \left(\frac{4\pi d}{\lambda}\right)^2 \quad (4.1.1)$$

where

$$\lambda = \frac{c}{f} \quad (4.1.2)$$

In Equation 4.1.2,  $\lambda$  is the wavelength of the propagating signal,  $c$  is the speed of light and  $d$  is the distance between the transmitter and receiver. The decibel form of the Free-space path loss equation was derived as follows:

$$FSPL(dB) = 10 \log_{10} \left( \left( \frac{4\pi df}{c} \right)^2 \right) \quad (4.1.3)$$

$$FSPL(dB) = 20 \log_{10} \left( \frac{4\pi df}{c} \right) \quad (4.1.4)$$

$$FSPL(dB) = 20 \log_{10}(d) + 20 \log_{10}(f) + 20 \log_{10} \left( \frac{4\pi}{c} \right) \quad (4.1.5)$$

$$FSPL(dB) = 20 \log_{10}(d) + 20 \log_{10}(f) - 147.55 \quad (4.1.6)$$

From Equation 4.1.6 it is clear that the path loss is affected by the operational frequency ( $f$ ) and the distance ( $d$ ) between the transmitter and receiver. A lower  $f$  will result in less path loss over the same communication distance. Hence 433 MHz was chosen as the operational frequency.

The RF communication module chosen was the Texas Instruments CC1101 low-power sub-1 GHz RF transceiver. The datasheet [36] states that the CC1101 is mainly intended for the Industrial, Scientific and Medical (ISM) frequency bands. It has an input voltage range of 1.8 V to 3.6 V with a low current sleep mode of 200 nA. The CC1101 has a high sensitivity of 116 dBm at 0.6 kBd at 433 MHz. It supports programmable output power settings of up to 12 dBm for all supported frequencies and programmable data rates from 0.6 Kbps to 600 Kbps. Furthermore the CC1101 supports automatic Clear Channel Assessment (CCA) for listen-before-talk systems. A modular design approach was used to incorporate the CC1101 into the system. The latter was achieved by using TFC-110-02-F-D-A connectors manufactured by Samtech. This means that different RF evaluation modules made by Texas Instruments can be plugged into and be used in the system. The 433 MHz CC1101 Evaluation Module (CC1101EMK433) was plugged into the system. Figure 4.12 shows the schematic diagram of the connection to the CC1101EMK433 and how it is connected to the microcontroller. The microcontroller communicates to the CC1101EMK433 over the SPI. Furthermore, the three input-output (IO) pins of the CC1101EMK433 was connected to the microcontroller for monitoring and control purposes. Figure B.11 in Appendix B shows the PCB footprint design of the TFC-110-02-F-D-A connectors used to provide the female connection side of the Texas Instruments RF evaluation modules.

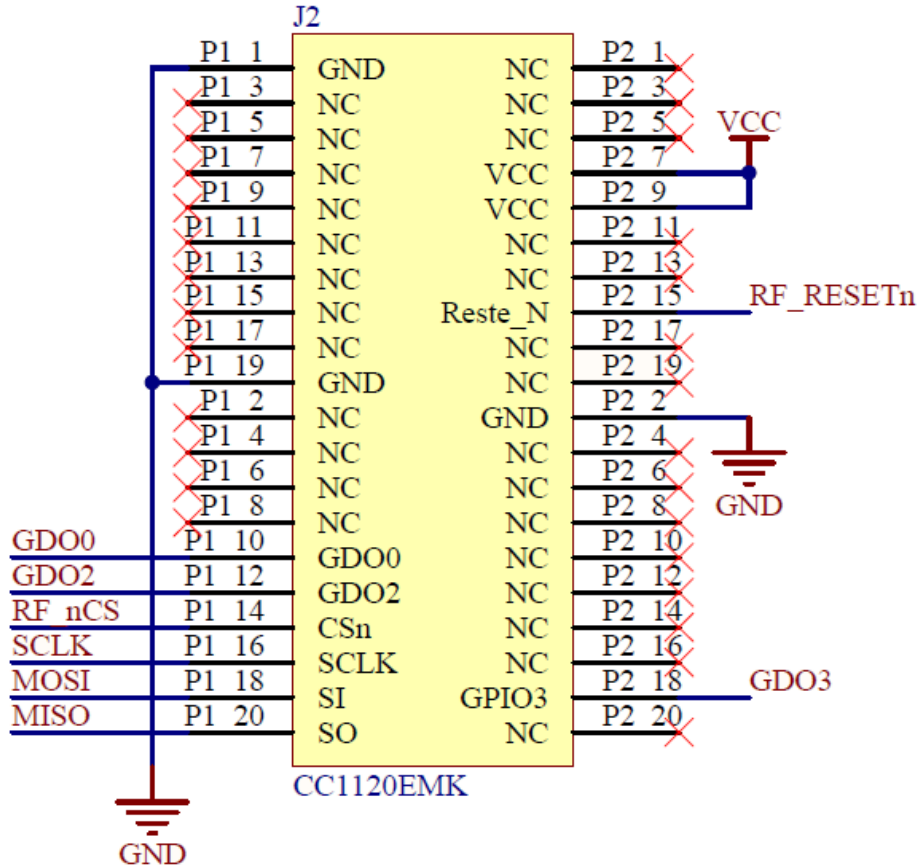


Figure 4.12: CC1101: Low-power sub - 1 GHz RF transceiver.

#### 4.1.4 Power Source and Regulation

The design requirements in Chapter 3.1 clearly states that the system should be battery powered. The system was designed to support a wide input voltage range of 2 V to 14 V. The input voltage is regulated to 3.3 V and serves as the  $V_{cc}$  voltage rail of the system. The later is achieved by using the L6932H1.2 high performance 2 A ultralow drop out (ULDO) linear regulator manufactured by STMicroelectronics. The datasheet [37] reports the L6932H1.2 to have an adjustable output voltage of between 1.2 V to 5 V. The output voltage is adjusted by calculation and changing a few passive components. Figure 4.13 shows the schematic diagram of the L6932H1.2. The values of passive components  $C8$  (22  $\mu$ F),  $C9$  (22  $\mu$ F) and  $R2$  (100 k $\Omega$ ) were provided by the datasheet [37].  $R12$  4.7 k $\Omega$  resistor and  $U5$  a light emitting diode (LED) were added to serve as a power on indicator. Header pins  $P2$  are used to connect or disconnect the battery from the system. The values of  $R6$  and  $R10$  were calculated from a formula provided in the datasheet [37]:

$$V_{cc} = \frac{1.2}{R10} (R6 + R10) \quad (4.1.7)$$

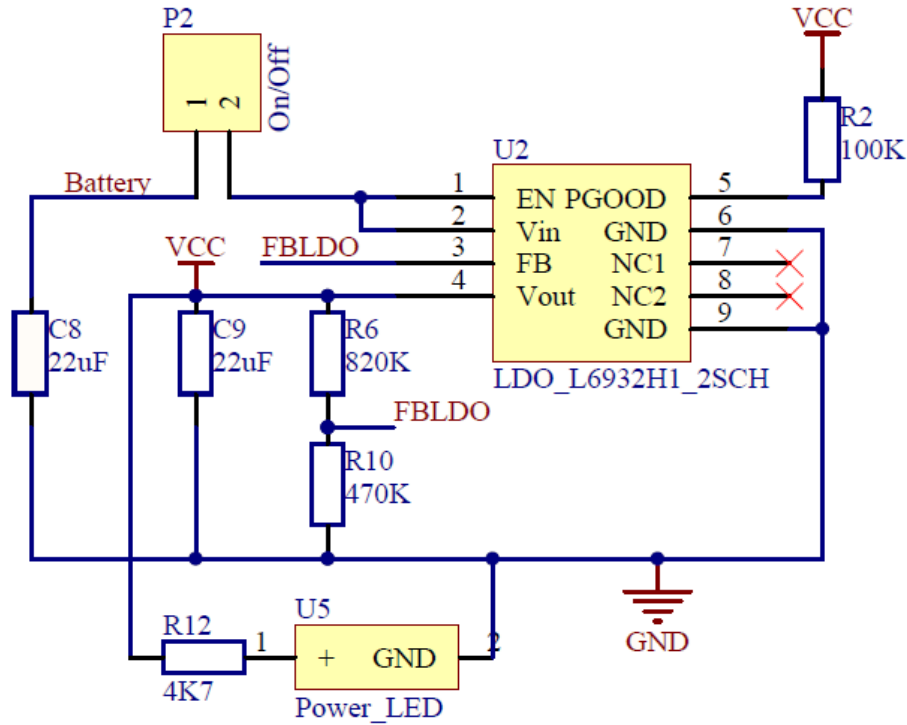


Figure 4.13: L6932: Ultralow drop output linear regulator.

If you consider Equation 4.1.7 and choose  $V_{cc} = 3.3\text{ V}$  and  $R_{10} = 470\text{ k}\Omega$  it is trivial to solve  $R_6$ :

$$V_{cc} = 3.3 = \frac{1.2}{470000} (R_6 + 470000) \quad (4.1.8)$$

$$3.3(470000) = 1.2R_6 + 1.2(470000) \quad (4.1.9)$$

$$1551000 = 1.2R_6 + 564000 \quad (4.1.10)$$

$$1.2R_6 = 98700 \quad (4.1.11)$$

$$R_6 = 822\,500\ \Omega \quad (4.1.12)$$

$R_6 = 822\,500\ \Omega$  is not a standard resistor value, however if  $R_6$  is chosen to be  $820\text{ k}\Omega$  and substituted back into Equation 4.1.7,  $V_{cc}$  becomes  $3.29\text{ V}$ , when component tolerances are neglected. Figure B.12 in Appendix B shows the PCB footprint design of the L6932H1.2 high performance ULDO linear regulator.

The TPS61222 low input voltage boost converter manufactured by Texas Instruments was used to provide stable power for the external contact microphone CM-01B. The datasheet [38] states that the TPS61222 can be up to 95% efficient with a low quiescent current of  $5\ \mu\text{A}$ . It has an input voltage range of  $0.7\text{ V}$  to  $5.5\text{ V}$  and a programmable output voltage of  $1.8\text{ V}$  to  $6\text{ V}$ . Figure 4.14 shows the schematic diagram of the TPS61222. The values of passive components  $C_3$  ( $10\ \mu\text{F}$ ),  $C_4$  ( $10\ \mu\text{F}$ ) and  $L_2$  ( $4.7\ \mu\text{F}$ ) were provided by the datasheet [38]. The latter recommends that the value of  $R_8$  should be kept below  $500\text{ k}\Omega$  and states the value of node  $5V_{FB}$  as  $0.5\text{ V}$ . To set the output voltage at  $5\text{ V}$ ,  $R_8$

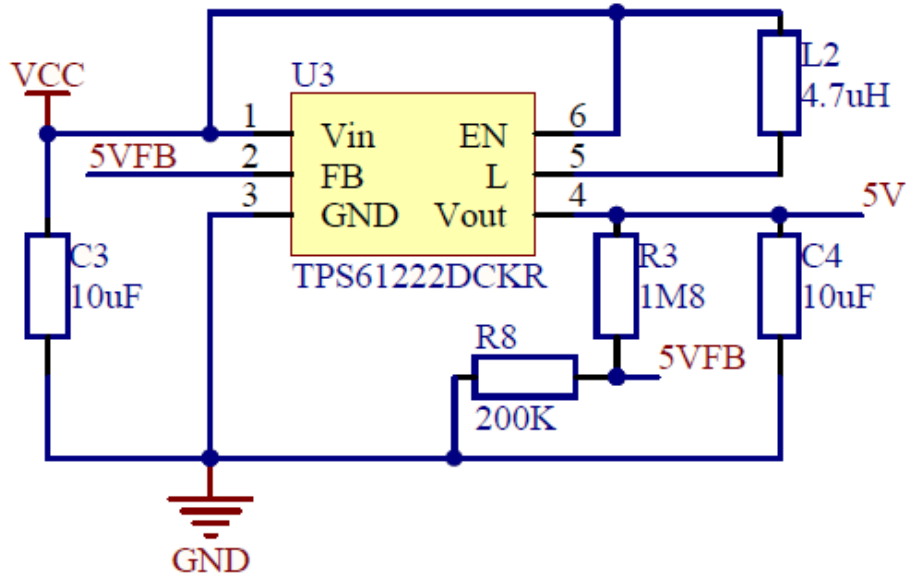


Figure 4.14: TPS61222: Low input voltage step-up converter.

was chosen as  $200\text{ k}\Omega$ . The value of  $R3$  can be obtained from the following formula provided by the datasheet [38]:

$$R3 = R8 \left( \frac{V_{out}}{5V_{FB}} - 1 \right) \quad (4.1.13)$$

$$R3 = 200000 \left( \frac{5}{0.5} - 1 \right) \quad (4.1.14)$$

$$R3 = 200000 (10 - 1) \quad (4.1.15)$$

$$R3 = 1.8\text{ M}\Omega \quad (4.1.16)$$

Figure B.13 in Appendix B shows the PCB footprint design of the TPS61222 low input voltage boost converter.

A LTC3105 step-up DC/DC converter with Maximum Power Point Control (MPPC) manufactured by Linear Technologies, was implemented to provide a possible means of energy harvesting. According to the datasheet [39], the LTC3105 is a high efficiency step-up DC/DC converter that can operate from input voltages as low as  $225\text{ mV}$ . The integrated MPPC enables operation directly from low voltage, high impedance alternative power sources such as photovoltaic cells [39]. The LTC3105 has a wide input voltage range of  $225\text{ mV}$  to  $5\text{ V}$  and an adjustable output voltage range of  $1.5\text{ V}$  to  $5.25\text{ V}$ . It has a peak current limit of  $500\text{ mA}$  and a valley current limit of  $350\text{ mA}$ . Furthermore the LTC3105 has a small footprint of  $3\text{ mm} \times 3\text{ mm} \times 0.75\text{ mm}$ . Figure 4.15 shows the schematic diagram of the LTC3105 step-up DC/DC converter. The datasheet [39] provides a typical application circuit for a single photovoltaic cell Li-Ion trickle charger that sets the output voltage to  $4.1\text{ V}$ . The values of passive components  $C1$  ( $10\text{ }\mu\text{F}$ ),  $C5$  ( $4.7\text{ }\mu\text{F}$ ),  $C6$  ( $1\text{ }\mu\text{F}$ ),  $C7$  ( $10\text{ }\mu\text{F}$ ) and  $L1$  ( $10\text{ }\mu\text{H}$ ) were provided by the datasheet [39]. Connecting  $FBLDO$  to ground sets the LDO output voltage to  $2.2\text{ V}$  which can be accessed from

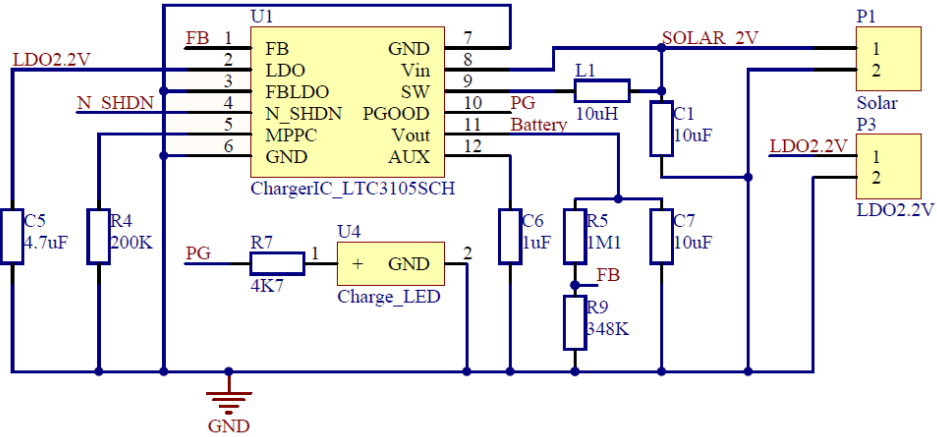


Figure 4.15: LTC3105: Step-up DC/DC converter.

header pin  $P3$  to provide external circuits with a regulated 2.2 V and a maximum current of 12 mA. Header pin  $P1$  is the input of the LTC3105 and can be connected to an energy source.  $R7$  a 4.7 k $\Omega$  resistor and  $U4$  a LED were added to serve as a power good indicator. The value of  $R4$  sets the MPPC of the LTC3105. For a MPPC of 2 V the value of  $R4$  was calculated based on a formula provided by the datasheet [39]:

$$V_{mppc} = 10 \mu\text{A} * R4 \quad (4.1.17)$$

$$2 \text{ V} = 10 \mu\text{A} * R4 \quad (4.1.18)$$

$$R4 = 200 \text{ k}\Omega \quad (4.1.19)$$

The output voltage was adjusted to be slightly higher than the voltage provided by the application circuit. The values of  $R5$  and  $R9$  were chosen as 1.1 M $\Omega$  and 348 k $\Omega$  respectively (based on component availability) and the voltage was calculated based on a formula provided in the datasheet [39] as follows:

$$V_{battery} = 1.004 \left( \frac{R5}{R9} + 1 \right) \quad (4.1.20)$$

$$V_{battery} = 1.004 \left( \frac{1.1 \text{ M}}{348 \text{ k}} + 1 \right) \quad (4.1.21)$$

$$V_{battery} = 1.004 (4.1609) \quad (4.1.22)$$

$$V_{battery} = 4.18 \text{ V} \quad (4.1.23)$$

The PCB footprint design of the LTC3105 step-up DC/DC converter is provided as per Figure B.14 in Appendix B.

#### 4.1.5 External Access

The MSP-FET430UIF USB debug interface was used to program the MSP430FR5739 on the PCB. This was achieved via a female 14 pin 2 row 2.54 mm header connector that mates with the MSP-FET430UIF. Figure 4.16 contains the schematic diagram of



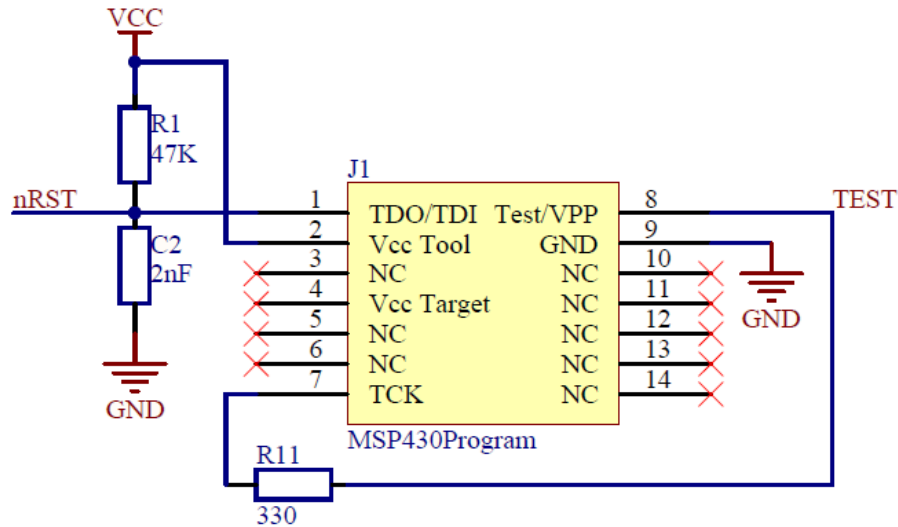


Figure 4.16: MSP430: 4 - Wire JTAG interface.

the MSP-FET430UIF female connector and the connections to the microcontroller. A 4-Wire JTAG connection coupled the MSP-FET430UIF to the MSP430FR5739. A typical application circuit of how to connect the MSP-FET430UIF to the MSP430FR5739 was provided in the MSP-FET430UIF User's Guide [40]. The values of passive components  $C2$  (2 nF),  $R1$  (47 k $\Omega$ ) and  $R11$  (330  $\Omega$ ) were similarly obtained. Figure B.15 in Appendix B depicts the PCB footprint design of the MSP-FET430UIF female connector.

Circuitry was added to ease the debugging process and allow external user induced interrupts to the microcontroller. Figure 4.17 shows the schematic diagram of these components and their connections to the microcontroller.  $R27$  and  $R28$  are two 4.7 k $\Omega$  pull-up resistors connected to two tactile switches  $S1$  and  $S2$ .  $S1$  is used as the system reset and is connected to the main hardware reset pin of the MSP430FR5739.  $S2$  is connected to a hardware interrupt pin on the microcontroller and the function of  $S2$  varies, depending on the software implemented on the microcontroller.  $R25$  and  $R26$  are two 4.7 k $\Omega$  resistors connected to two LEDs  $U17$  and  $U18$  respectively. The two LEDs are connected to IO pins on the microcontroller and can be toggled to indicate various system states.

Various header pins were added to improve the adaptability and scalability of the system. This was done to ease the process of adding new sensors to the system and to ease the debugging process by providing access to all the communication lines. Figures 4.18 and 4.19 shows the schematic diagrams of the various header pins that provide external access to the PCB.  $P14$  and  $P16$  provide a way to easily connect an external power source to the system and to measure the voltage on the *Battery* line.  $P15$  provides a way to easily connect an external component to the regulated 3.3 V line of the PCB or to measure the  $V_{cc}$  voltage.  $P20$ ,  $P21$  and  $P22$  provides a way to connect sensors to the SPI bus of the microcontroller with their own dedicated chip select lines  $EX1\_nCS$ ,  $EX2\_nCS$  and  $EX3\_nCS$  respectively. The latter also provides an easy way to monitor the microcontroller's SPI bus.  $P17$  allows the user to add sensors to the UART lines of the microcontroller and to monitor these lines.  $P18$  and  $P19$  provides a means to connect sensors to the  $I2C$  bus of the microcontroller and to monitor the microcontroller's  $I2C$  bus. Finally,  $P24$  provides access to three digital IO pins on the microcontroller with

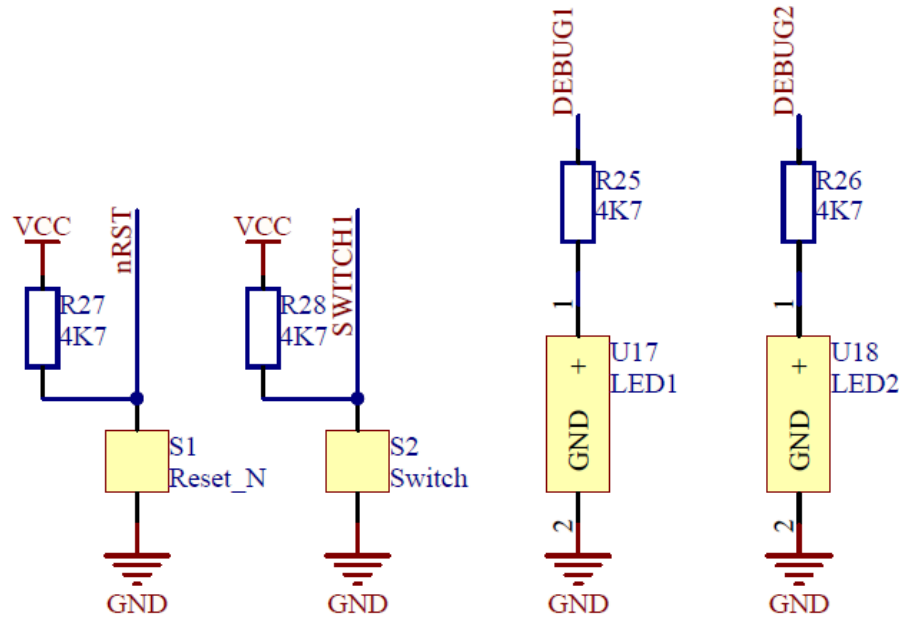


Figure 4.17: Tactile switch and LED's used for debugging.

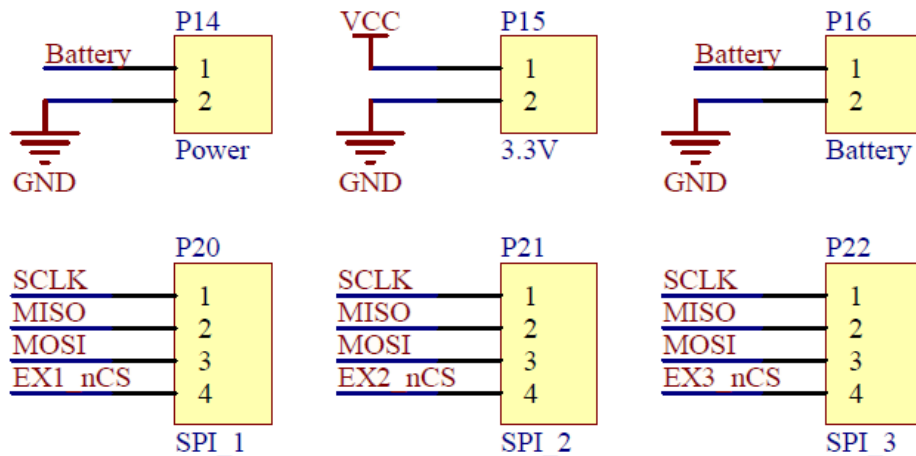


Figure 4.18: PCB headers for external access 1.

ADC and port interrupt capabilities. *P24* also provides access to the  $V_{cc}$  voltage rail for monitoring or external access purposes.

#### 4.1.6 Assembled WildMotes

After the hardware design of the WildMotes, the design files were sent to a company called Trax to manufacture the PCBs. The PCBs along with all the various components were sent to Barracuda Holdings to assemble the PCBs by means of pick and place machines. Figure C.1 in Appendix C depicts the PCB design of the WildMotes. Figure 4.20 shows a single assembled WildMote and Figure 4.21 shows closeup photos of the various

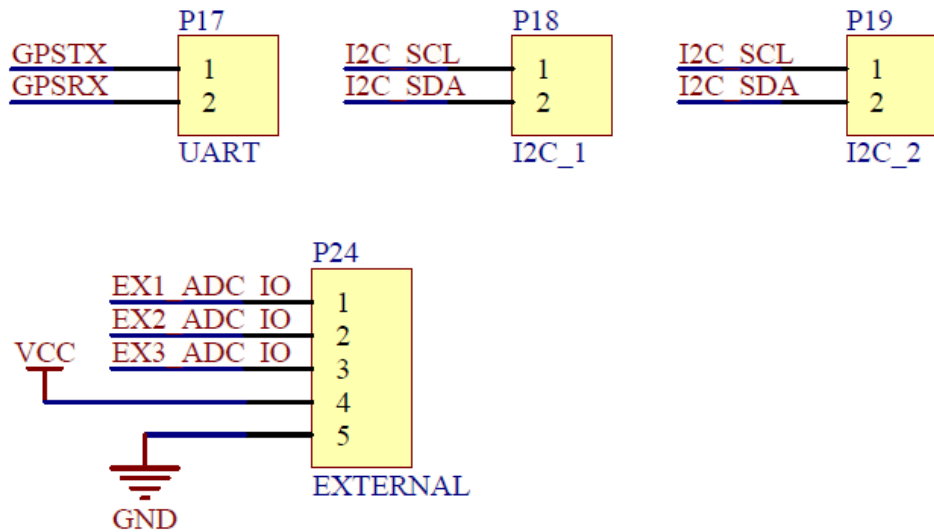


Figure 4.19: PCB headers for external access 2.

components of the WildMotes. In the figure, (a) depicts the SQ-SEN-200 nano-power tilt and vibration sensor, (b) the RF communication module, (c) the accelerometer, (d) the LTC3105, (e) the GPS module and (f) the FRAM modules.

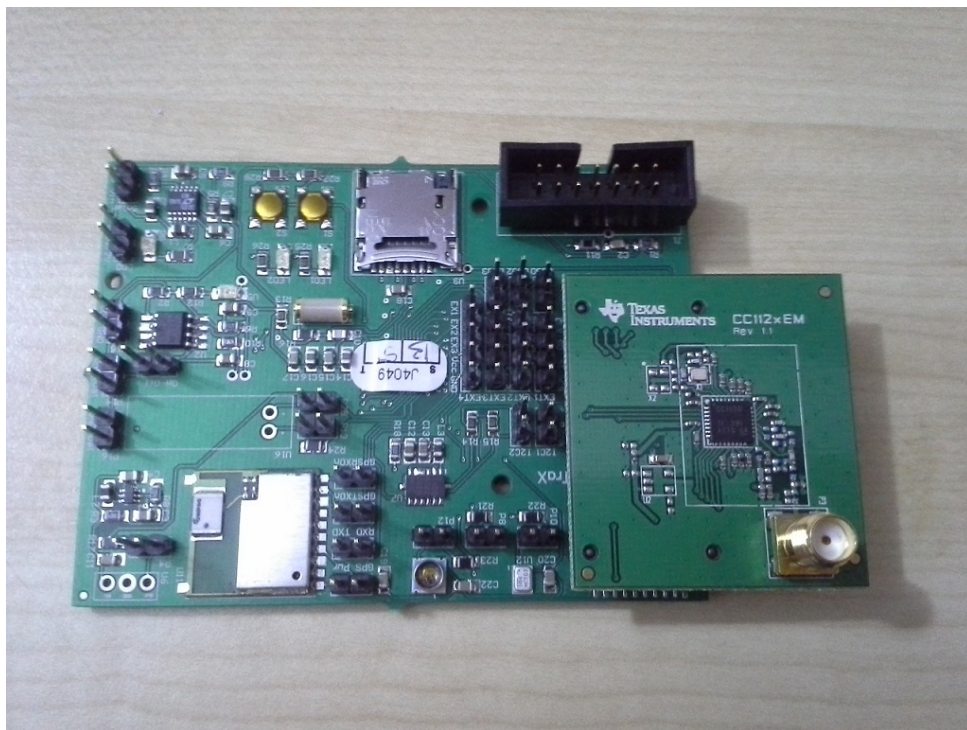


Figure 4.20: Assembled WildMote.

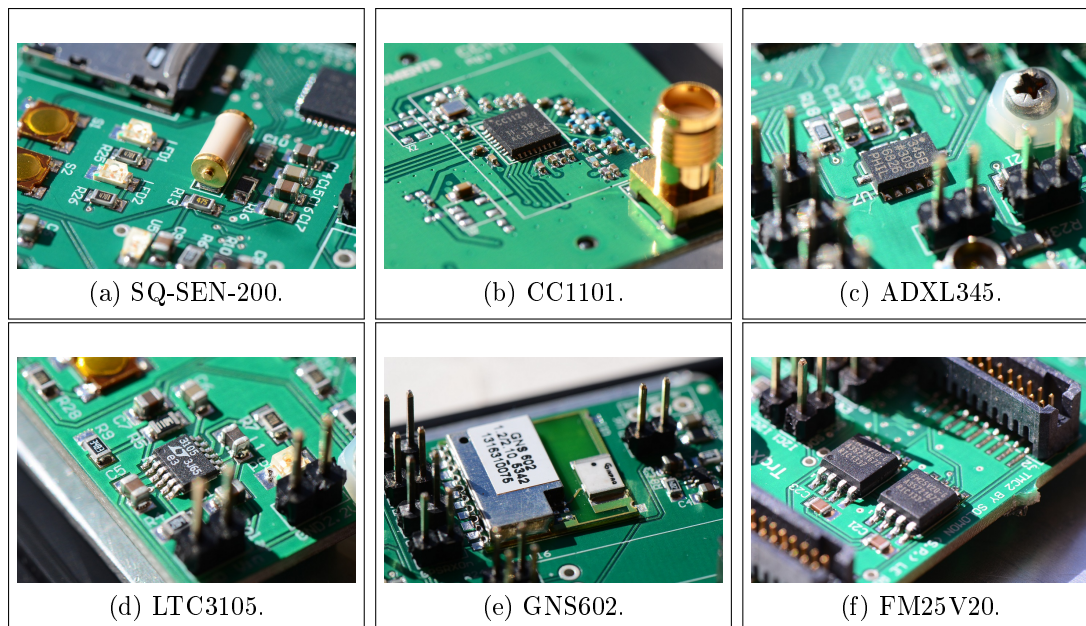


Figure 4.21: WildMote closeup photos.

## 4.2 Base Station and Repeaters

The base station and repeaters both had the same hardware design but run different network protocols. Raspberri Pi B+ units were utilised to simplify the hardware design of the base station and repeaters. A RF Shield was designed to provide the base station and repeaters with the same RF capabilities as the WildMotes. Figure D.2 in Appendix D shows the schematic diagram of the RF Shield. A modular design approach served to incorporate the CC1101 onto the RF Shield. This was achieved by using TFC-110-02-F-D-A connectors manufactured by Samtech. This means that different RF evaluation modules made by Texas Instruments can be plugged into the RF Shield. The 433 MHz CC1101 Evaluation Module (CC1101EMK433) was selected as the radio for the RF Shield. Figure 4.22 shows the RF Shield and the assembled Raspberri Pi with the RF Shield and CC1101EMK433 module can be seen in Figure 4.23. In total one base station and two repeaters were manufactured.

## 4.3 Collar and Casing Design

Two versions of collars were designed during the extent of the project. The first collar called Sheep Collar, had no strict design requirements apart from keeping the PCB safe and in the same orientation all the time. A combination of standard off the shelf casings and belts were used as the collar. Sheep Collar had dimensions of 146 mm x 80 mm x 65 mm and weighed 281 g including the PCB, battery and the belt. The reason for the large size of this collar was that no other smaller off the shelf casing could be found before the practical testing had to begin. Despite the size of the collars, the sheep still acted naturally and hardly even noticed the collars pressing into their thick wool. In total, five Sheep Collars were used for the experiments with sheep.

The second collar called Rhino Collar had strict design requirements. It had to be as small as possible while housing two batteries and the PCB. It had to be durable and



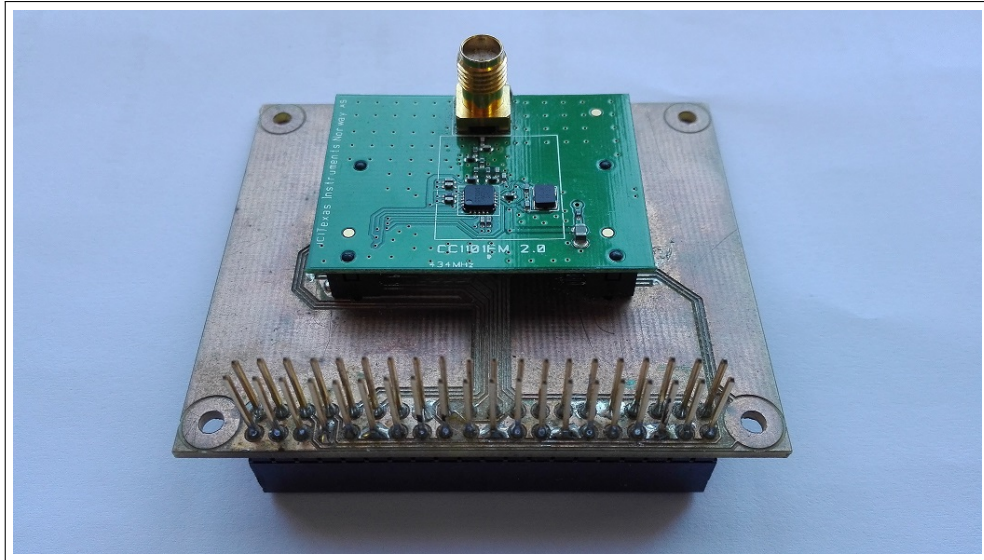


Figure 4.22: RF Shield and CC1101EMK433.

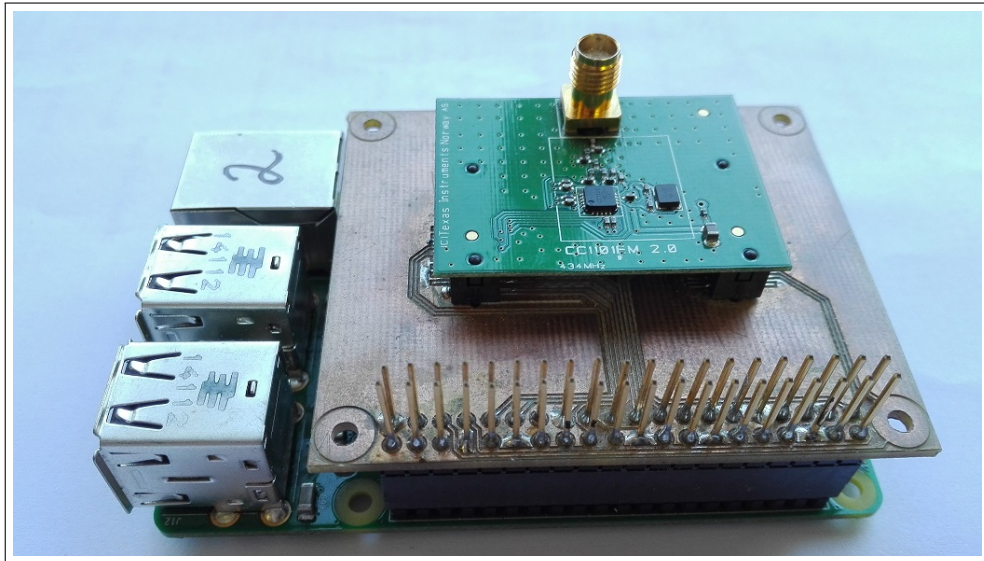


Figure 4.23: Raspberri Pi with RF Shield and CC1101EMK433.

waterproof in order to protect the PCB when the rhinos take mud baths and rub themselves against rocks. A three part casing was designed and manufactured from a strong plastic called Acetal. Figures E.1, E.2 and E.3 in Appendix E shows the design drawings of the casing and Figure E.4 some 3D views of the casing. The casing has dimensions of 117 mm x 77 mm x 38 mm. The base housing the PCB had a wall thickness of 6 mm. The waterproofing was achieved by applying gasket sealer between the Base and Lid 1. Two nylon webbing straps (50 mm wide) were used per casing as collar straps. The latter was chosen because nylon webbing is soft (it will not harm the rhino), flexible and very strong. A combination of two tri-glides per strap were used to quickly fasten the collar around the back legs of semi-tame rhinos. The Rhino Collars weighed 371 g including the PCB, batteries, nylon webbing and tri-glides.

The aim of Rhino Collar was more to design a collar that can quickly be fastened to an unsedated rhino, as opposed to a collar design that should last for long periods of time. In total, two Rhino Collars were manufactured for this experiments.

This chapter dealt with the hardware design of the ABBMS. The details relating to the hardware design of the WildMotes, base stations and repeaters were discussed. The collar- and casing design of the Wildmotes were also considered. The next chapter explains the software design of the ABBMS.

# Chapter 5

## Software Design

The software design formed a big part of this project and constituted two main software categories. The first category called Initialisation Software was required for the development of the backbone software structure of the system. The latter included a combination of low level instructions to interface with the various system components in order to produce higher level functions. The Functional Software category utilised the higher level functions developed in the Initialisation Software category in order to run software routines to achieve very specific goals.

### 5.1 Initialisation Software

All software for this project, was developed under the Code Composer Studio 5.5 (CCS5) environment. Texas Instruments provides a standard *MSP430.h* library within CCS5 along with a Graphical Peripheral Configuration Tool (called GRACE) which is useful to easily configure the MSP430FR5739 microcontroller. GRACE allows you to configure various system peripherals in a graphical manner, as seen in Figure 5.1, and then updates the system registers accordingly. This was especially useful to set up the system clock, direction of IO ports, interrupt pins, timers and to set the UART (Figure 5.2 shows an example of the UART setup), I2C and SPI to work on specific ports. Once GRACE has configured the low level registers it then provides you with GRACE Snippets which are small software components used to interface the various MSP430 peripherals with. For example the GRACE Snippet `eSPI_transmitData (_MSP430_BASEADDRESS_EUSCI_A0_, single_byte)` allows you to send a single byte over the SPI interface. GRACE Snippets were used to interface the various system peripherals. This however, was sometimes experienced as slightly limiting when very specific low level tasks had to be performed, but a work around was always available and easy to do. Code Composer Studio with the combination of GRACE and GRACE Snippets were great learning tools and reduced the initial setup time of the MSP430FR5739 significantly. Once the MSP430FR5739 was up and running the focus then turned towards developing functions to interface the various system components.

#### 5.1.1 MSP430RF5739 Configuration

The MSP430RF5739 was configured with the following settings. The main system clock was set on 20 MHz and the auxiliary clock was set on 8.3 kHz. An internal voltage reference of 2.5 V was used as the reference for the ADC sampling. Timer0A3 was configured to

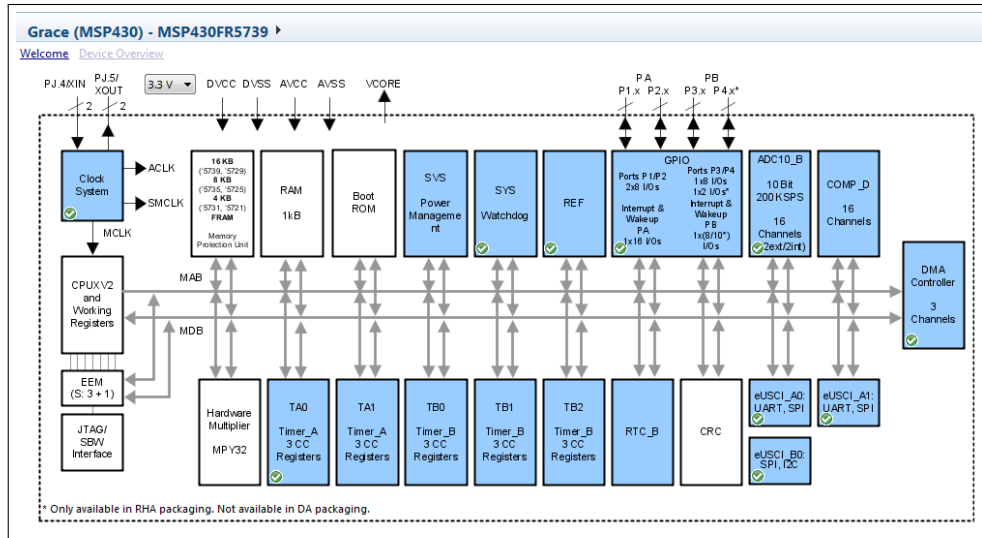


Figure 5.1: GRACE summary of MSP430FR5739.

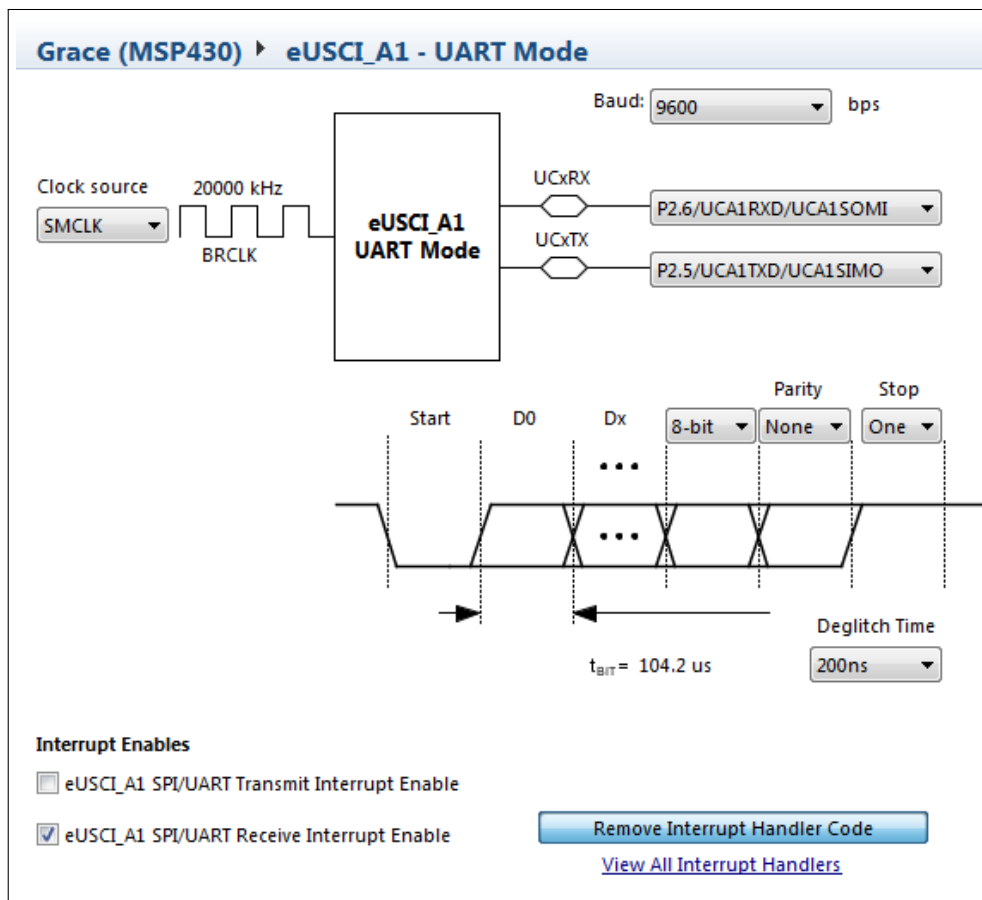


Figure 5.2: GRACE UART setup.



generate an interrupt every 10 ms (100 Hz). The SPI bus was set to run on 10 Mbps and configured on pins *P1.5* (CLK), *P2.0* (SIMO) and *P2.1* (MOSI). The chip select lines of the various SPI devices were driven by normal IO pins. The UART was set up at 9600 bps (Data length: 8 Bits, Stop bit: 1, Parity: None) and configured on pins *P2.5* (TX) and *P2.6* (RX). The I2C bus was set up on 400 Kbps and configured on pins *P1.6* (SDA) and *P1.7* (SCL). The following pins were set as input pins: *P1.0*, *P1.3*, *P1.4*, *P2.3*, *P2.7* (with interrupt on falling edge), *P3.0* (with internal pull-down resistor), *P3.1* (with internal pull-down resistor), *P3.4*, *P3.5*, *P4.0* and *P4.1* (with interrupt on falling edge). The following pins were set as output pins with a default logic low level output: *P2.2*, *P2.4*, *P3.2*, *P3.3*, *P3.6*, *P3.7*, *PJ.0*, *PJ.1*, *PJ.2*, *PJ.3*, *PJ.4* and *PJ.5*.

### 5.1.2 SPI Interface

The following list of functions were developed to interface the SPI bus:

- `int spi_transmit(char spi_tx_buffer[], int num_tx_bytes);` Transmit a number of bytes over the SPI bus
- `char spi_receive(void);` Receive a character over the SPI bus

### 5.1.3 UART Interface

The following list of functions were developed to control the UART interface:

- `int uart_setup(char enable_uart);` Initialize the UART interface
- `int uart_transmit(char uart_tx_buffer[], int num_tx_bytes);` Transmit a number of bytes over the UART interface
- `char uart_receive_interrupt_handler(void);` Receive a character over the UART interface
- `void uart_gps_data_ready(void);` Monitor GPS data ready flag
- `char uart_gps_receive_interrupt_handler(void);` Receive a valid GPS string over the UART interface

### 5.1.4 I2C Interface

The I2C was one example where Grace Snippets struggled to update the I2C interrupt handlers effectively. As a result, an I2C library specifically designed for MSP430s was obtained from Mikrocontroller.net [41]. The following functions were used in the implementation:

- `void init_iic(void);` Initialize the I2C bus
- `int iic_receive(unsigned int address, unsigned int COMMAND, unsigned char Bytes_expected);` Receive a number of bytes over the I2C bus
- `int i2c_tx_interrupt_handler(void);` Handles the I2C transmit interrupt
- `int i2c_rx_interrupt_handler(void);` Handles the I2C receive interrupt

### 5.1.5 CC1101 Interface

The following list of functions were developed to interface the CC1101 RF transceiver:

- `void cc11xx_initialise(void)`; Initialise the CC1101 RF transceiver
- `void cc11xx_sleep(void)`; Put the CC1101 in sleep mode
- `void cc11xx_enable_ChipSelect(void)`; Enable the CC1101 for communication
- `void cc11xx_disable_ChipSelect(void)`; Disable the CC1101 for communication
- `void cc11xx_reset_chip_power_cycle(void)`; Perform CC1101 hardware reset
- `void cc11xx_tx_command(char cmd)`; Transmit a single command to the CC1101
- `void cc11xx_tx_byte(char cmd)`; Transmit a single byte to the CC1101
- `void cc11xx_TXFIFO_Write(char tx_buffer[], char num_tx_bytes)`; Write a number of bytes to the CC1101's Transmit FIFO
- `char cc11xx_RXFIFO_Read(char rx_buffer[], int num_rx_bytes)`; Read a number of bytes from the CC1101's Receive FIFO
- `char cc11xx_tx_data(char tx_buffer[], int num_tx_bytes)`; Transmit a number of bytes using the CC1101
- `char cc11xx_rx_data(char rx_buffer[], int num_rx_bytes)`; Receive a number of bytes from the CC1101
- `char cc11xx_get_state(void)`; Get the current state of the CC1101
- `void cc11xx_read_all_registers(void)`; Read all registers from the CC1101

### 5.1.6 FRAM Interface

The following list of functions were developed to interface the FM25V20 FRAM modules:

- `void fram_sleep(void)`; Put the FRAM modules in sleep mode
- `int fram_write(char fram_tx_buffer[], int num_tx_bytes, char addrMSB, char addrMID, char addrLSB)`; Write a number of bytes to a specific address in the first FRAM module
- `void fram2_write(char fram_tx_buffer[], int num_tx_bytes, char addrMSB, char addrMID, char addrLSB)`; Write a number of bytes to a specific address in the second FRAM module
- `int fram_read(char fram_rx_buffer[], int num_rx_bytes, char addrMSB, char addrMID, char addrLSB)`; Read a number of bytes from a specific address in the first FRAM module
- `void fram2_read(char fram_rx_buffer[], int num_rx_bytes, char addrMSB, char addrMID, char addrLSB)`; Read a number of bytes from a specific address in the second FRAM module

### 5.1.7 MicroSD Card Interface

The following functions were used from a Multi Media Card (MMC) library produced by Texas Instruments [42] [43] to interface the MicroSD card of the system:

- `char mmcInit(void)`; Initialise the MMC
- `char mmcPing(void)`; Check if MMC card is present
- `void mmcSendCmd (const char cmd, unsigned long data, const char crc)`; Send command to MMC
- `char mmcGoIdle()`; Set MMC in Idle mode
- `char mmcSetBlockLength (const unsigned long)`; Set MMC block length
- `char mmcReadBlock(const unsigned long address, const unsigned long count, char *pBuffer)`; `#define mmcReadSector(sector, pBuffer) mmcReadBlock(sector*512ul, 512, pBuffer)` Read a 512 Byte big block beginning at the address
- `char mmcWriteBlock (const unsigned long address, const unsigned long count, char *pBuffer)`; `#define mmcWriteSector(sector, pBuffer) mmcWriteBlock(sector*512ul, 512, pBuffer)` Write a 512 Byte big block beginning at the address
- `char mmcReadRegister(const char, const unsigned char, char *pBuffer)`; Read MMC register into the buffer
- `unsigned long mmcReadCardSize(void)`; Read the card size from the CSD register

### 5.1.8 Useful Functions

The following list of useful functions were developed to perform specific tasks throughout the project:

- `void accelerometer_sleep(void)`; Put accelerometer in sleep mode
- `void accelerometer_init(void)`; Initialise the accelerometer
- `void accelerometer_getData(void)`; Get the x, y and z accelerometer data
- `void gps_init(void)`; Initialise the GPS
- `void gps_sleep(void)`; Put GPS in sleep mode
- `void gps_wakeup(void)`; Wakeup the GPS
- `void get_gps_location(void)`; Get the latest coordinate with 3D Fix
- `void RF_transmit_buffer(char buffer[], char num_bytes)`; Transmit a data buffer over the wireless RF link
- `void SDCard_Reset_Sector_Pointer()`; Reset the microSD card address pointer
- `void SDCard_Logging_Routine()`; Log data to microSD card

- void Fram\_Logging\_Routine(); Log data to FRAM
- void Fram2\_Reading\_Routine(); Read data from FRAM
- char calc\_feachers\_fram(char start\_addr, int number\_samples); Calculate live behaviour classification

## 5.2 Functional Software

Functional software routines were developed to run specific experiments. Apart from software routines developed for testing purposes, four main software routines were developed for the main experiments performed during the project. The four routines were as follows:

- Routine 1: 100 Hz Accelerometer Logger for Sheep Behaviour Classification
- Routine 2: 40 Hz Accelerometer Logger for Rhino Behaviour Classification
- Routine 3: On-animal Behaviour Classification System
- Routine 4: Network Protocol

### 5.2.1 Routine 1: 100 Hz Accelerometer Logger for Sheep Behaviour Classification

The first practical experiments for the project took place on a farm called Rooivlei in the Northern Cape. The goal of the experiment was to fit five Sheep Collars (as described in Chapter 4.3) around the necks of five sheep and to record the behavioural, or movement patterns of the animals based on the different forces experienced by the on-board accelerometer. Routine 1 was developed to control the hardware for this experiment. For signal processing purposes it was very important that the sampling of the accelerometer was done constantly at 100 Hz.

The following flow diagrams give an overview of the workings of Routine 1 and can briefly be explained as follows. Routine 1 is an interrupt driven timing based routine. This means that the system mainly waits for interrupts to occur before entering the following system state. The main interrupt event is generated by an 100 Hz timer thus interrupting the microcontroller every 10 ms. Figure 5.3 shows the three main system interrupts that effect the flow of the software. The software starts by initialising the system (as seen in Figure 5.4) where after it has two main states based on the presence of a jumper between pins *P24.3* and *P24.4* on header pin *P24* (refer to Figure 4.19 in Chapter 4.1.5). If the jumper is present, the system transmits the number of 512 Byte sectors logged on the microSD card, to the computer. This number is used when downloading the data from the microSD card to a computer. If the jumper is absent it means that the system is in logging mode. Figures 5.5 and 5.6 illustrate how the system constantly (at 100 Hz) acquires and logs the *x*-, *y*- and *z*-axis acceleration data from the accelerometer together with the environmental temperature and vibration sensor data.

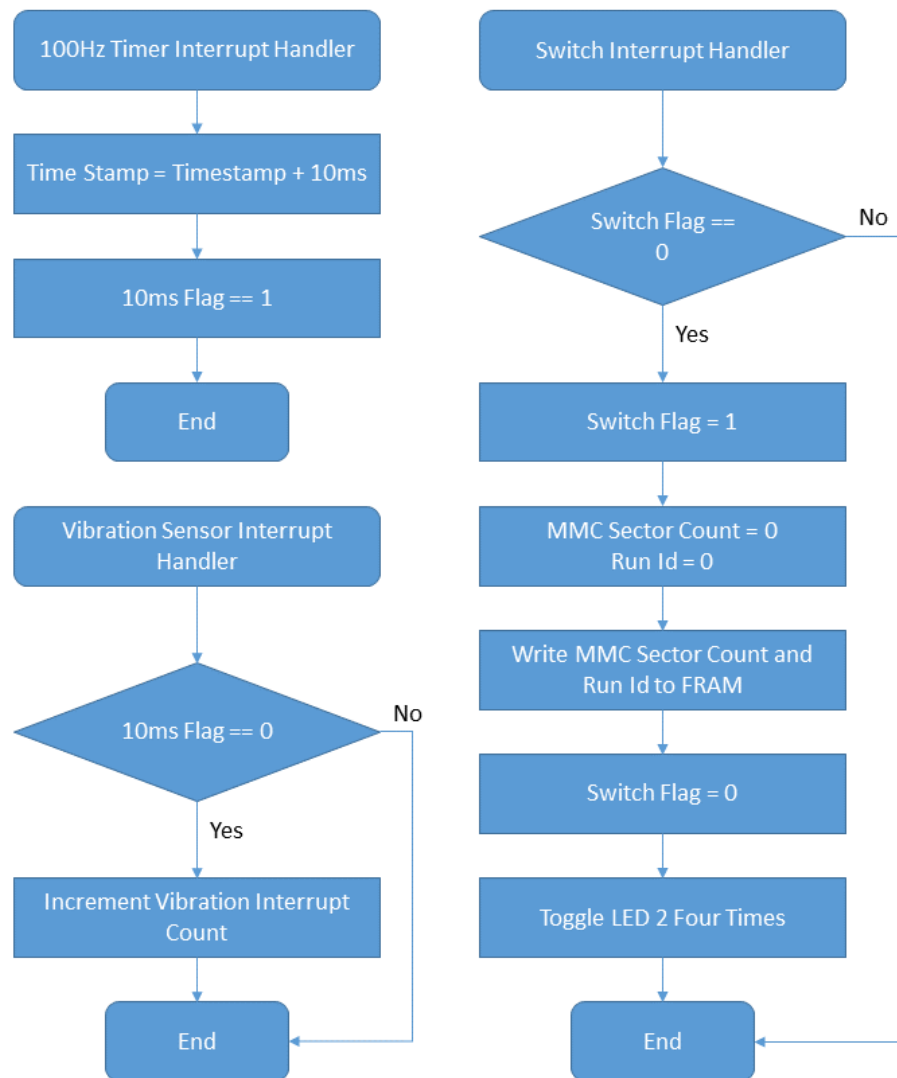


Figure 5.3: Routine 1: Interrupt handlers.

### 5.2.2 Routine 2: 40 Hz Accelerometer Logger for Rhino Behaviour Classification

The second practical experiment took place on a rhino rehabilitation farm. The goal of the experiment was to fit two Rhino Collars (as described in Section 4.3) around the back legs of rhinos and to record the behavioural or movement patterns of the animals based on the different forces experienced by the on-board accelerometer. Routine 2 was developed to control the hardware for this experiment. For signal processing purposes it was very important that the sampling of the accelerometer was done constantly at 40 Hz.

The following flow diagrams provide an overview of the workings of Routine 2 and can briefly be explained as follows. Routine 2 is an interrupt driven timing based routine. This means that the system mainly waits for interrupts to occur before entering the following system state. The main interrupt event is generated by an 40 Hz timer, thus interrupting the microcontroller every 25 ms. Figure 5.7 shows the three main system interrupts that

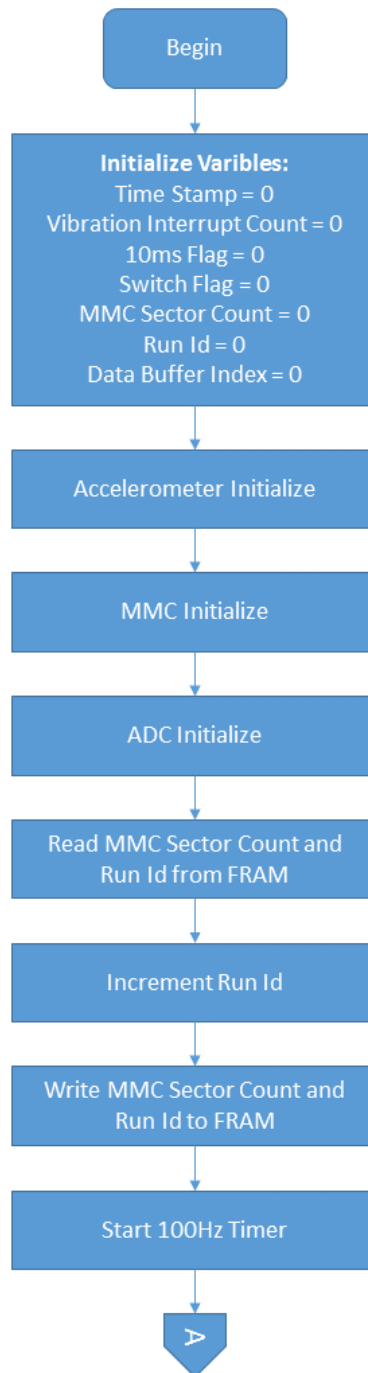


Figure 5.4: Routine 1: Initialisation.

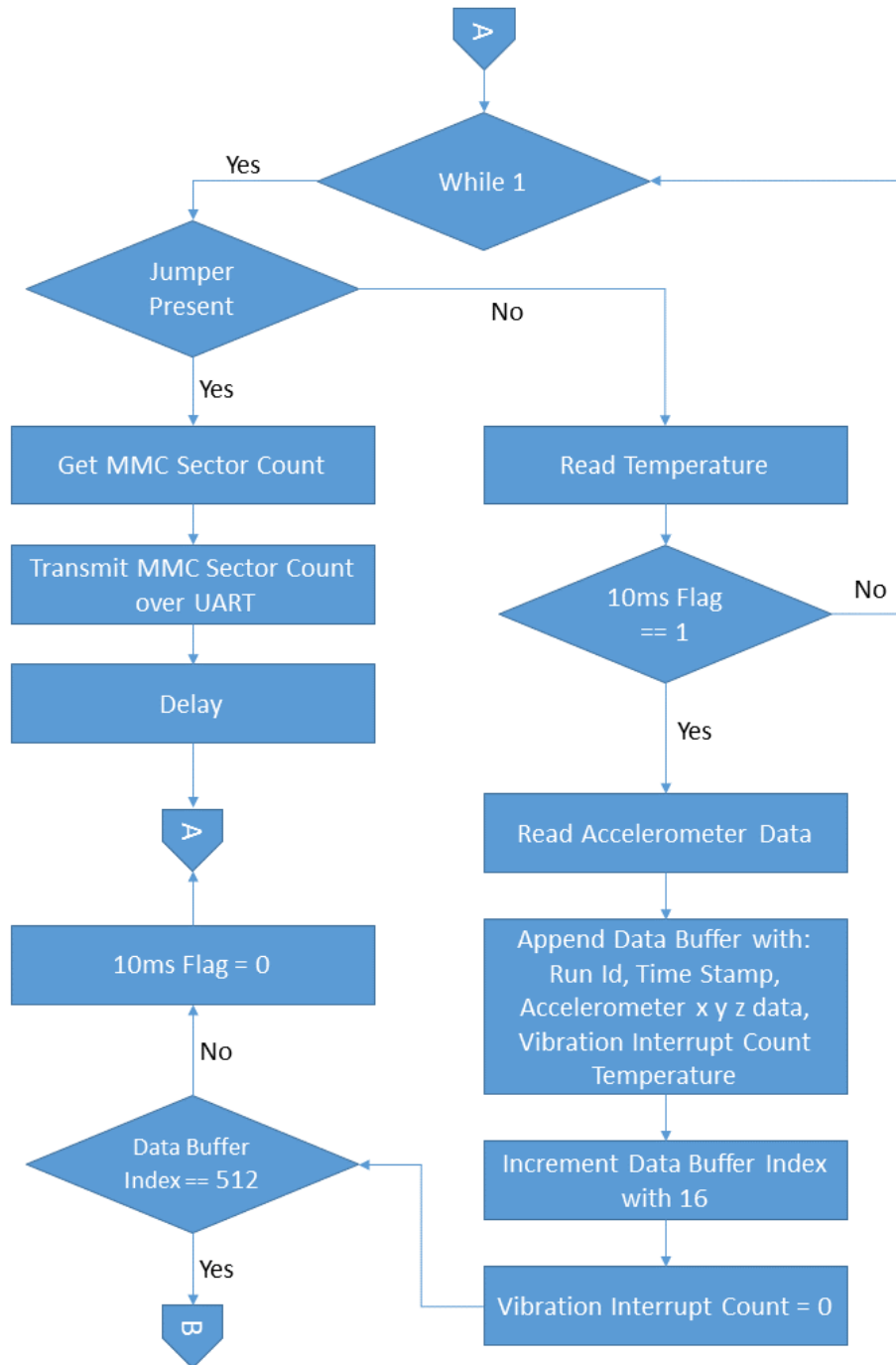


Figure 5.5: Routine 1: Main flow A.

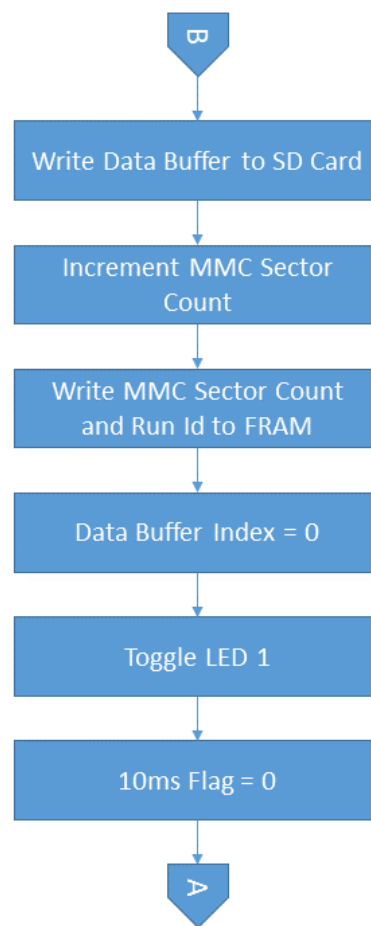


Figure 5.6: Routine 1: Main flow B.

effects the flow of the software. The software starts by initialising the system (as seen in Figure 5.8) where after it has two main states based on the presence of a jumper between pins  $P24.3$  and  $P24.4$  on header pin  $P24$  (refer to Figure 4.19 in Section 4.1.5). If the jumper is present the system transmits the number of 512 Byte sectors logged on the microSD card, over a wireless RF Link. This number is used when downloading the data from the microSD card to a computer. If the jumper is absent it means that the system is in logging mode. Figures 5.9 and 5.10 show how the system constantly (at 40 Hz) acquires and logs the  $x$ -,  $y$ - and  $z$ -axis acceleration data. Figure 5.11 shows how the system waits 90s before waking the GPS module and waits until the *GPS Timer* is at 180s before acquiring the GPS location. Waking the GPS 90s before actually acquiring the location allows the system to continue with the accelerometer logging while the GPS gets a 3D fix. The GPS location is appended to the data buffer and transmitted over a wireless RF link at approximately 3 min intervals.

### 5.2.3 Routine 3: On-animal Behaviour Classification System

The OABCS as developed for sheep and rhinos has, to the best of our knowledge, never been implemented before. It is, therefore, seen as a significant contribution of the work as set out in this thesis. Consequently, the associated software routine is considered as



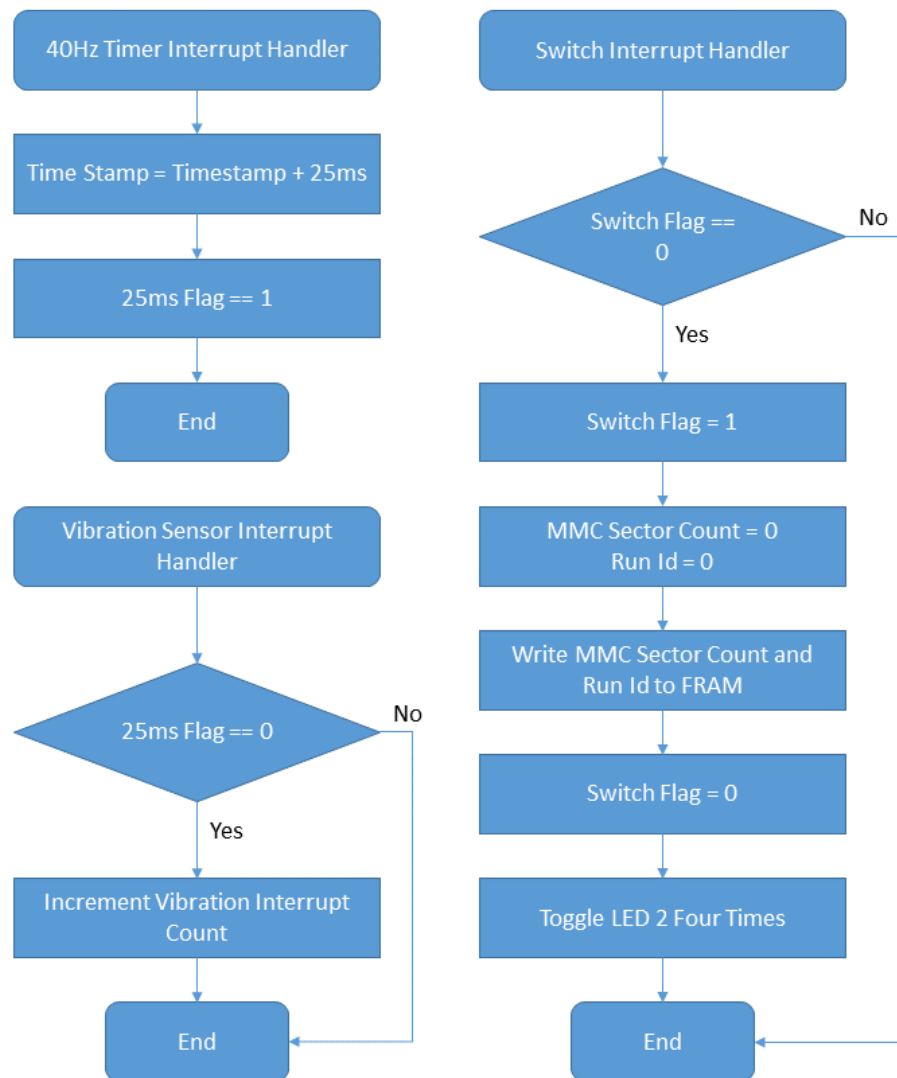


Figure 5.7: Routine 2: Interrupt handlers.

the most important routine developed therein. It resulted in automatic behaviour classification being implemented on the animal in real time, using DSP techniques on small low power microcontrollers and not during post processing as is current practice world wide [10], [13], [14], [15], [16] and [17]. The technique is very powerful and simple to reconfigure for different animals, as it is only necessary to update the sampling frequency and the values of four matrices to effect the change.

There are two stages involved in implementing the OABCS. The first part is to collect training data for the off-line automatic behaviour classifier done on a PC, along with video footage of the animal to ensure the classifier is trained based on ground truth values. Once the classifier is trained, four matrices are obtained for the application in the OABCS. Once the four matrices and a classification routine are programmed onto the WildMotes, it can perform live automatic behaviour classification while fitted on the animals.

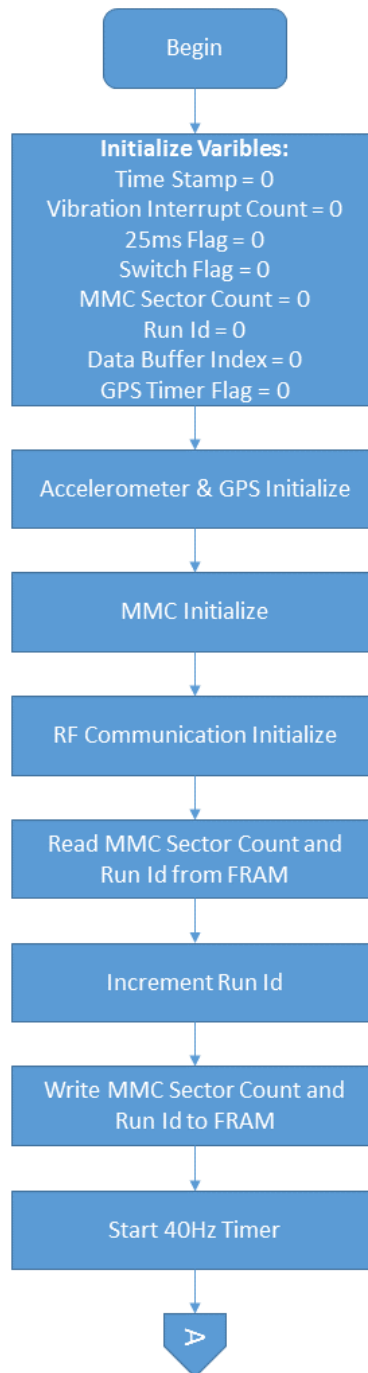


Figure 5.8: Routine 2: Initialisation.

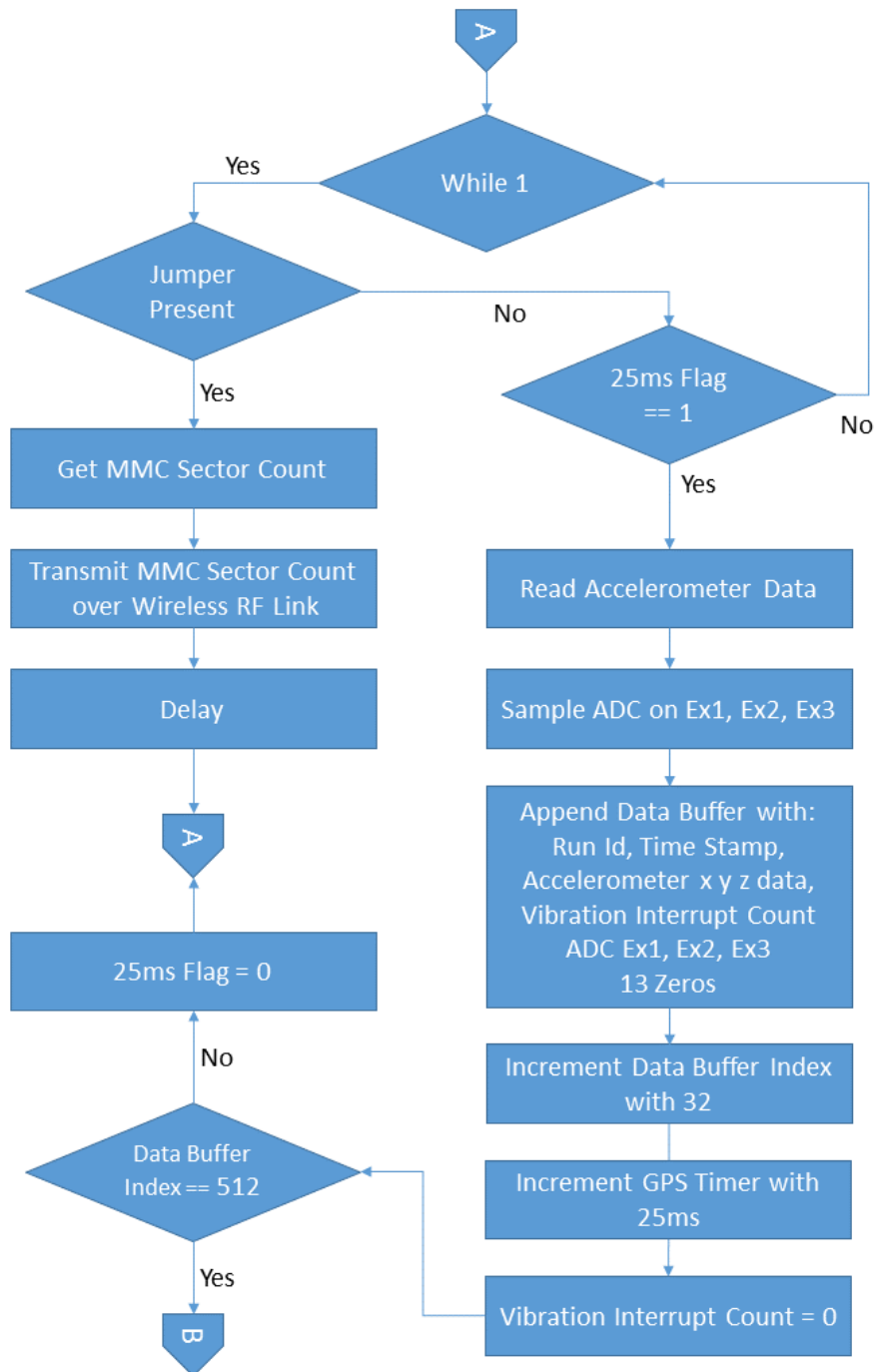


Figure 5.9: Routine 2: Main flow A.

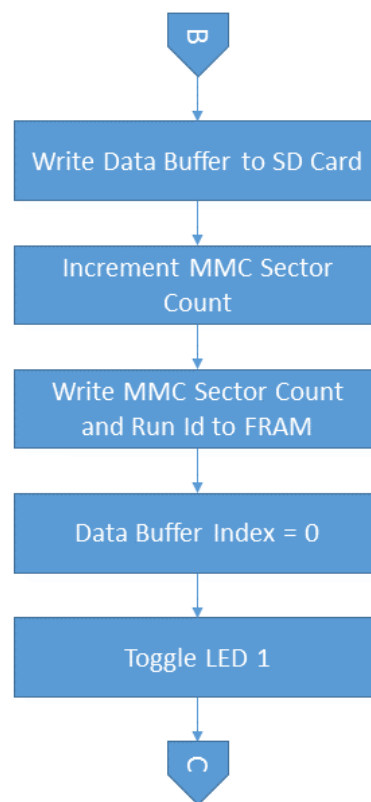


Figure 5.10: Routine 2: Main flow B.

The practical testing of the OABCS took place on a farm called Rooivlei in the Northern Cape. The goal of the experiment was to fit two Sheep Collars (as described in Section 4.3) around the necks of sheep and test the OABCS. Routine 3 was developed to control the hardware for this experiment. The training of the classifier was done using a 100 Hz sampling frequency. It is therefore important to set the sampling frequency of the classification software to sample the accelerometer constantly at 100 Hz.

The following flow diagrams present an overview of the operation of Routine 3 and can briefly be explained as follows. Routine 3 is an interrupt driven timing based routine. This means that the system primarily waits for interrupts to occur before entering the following system state. The main interrupt event is generated by a 100 Hz timer, thus interrupting the microcontroller every 10 ms. Figure 5.12 illustrates the 100 Hz timer interrupt handler. The switch interrupt handler can be followed in Figures 5.13 and 5.14. Both these interrupts can affect the flow of the software. The software starts by initialising the system (as seen in Figure 5.15) where after it has two main states based on the presence of a jumper between pins  $P24.3$  and  $P24.4$  on header pin  $P24$  (refer to Figure 4.19 in Section 4.1.5). If the jumper is present the system waits for the user to press the switch to either transmit all the recorded data over the RF link to the base station ( $P24.3$  is high), or to format the data ( $P24.2$  is high). If the jumper is absent the behaviour classification software is running. Figures 5.16 and 5.17 contain the major details of the software routine. The classification can be summarised as follows.

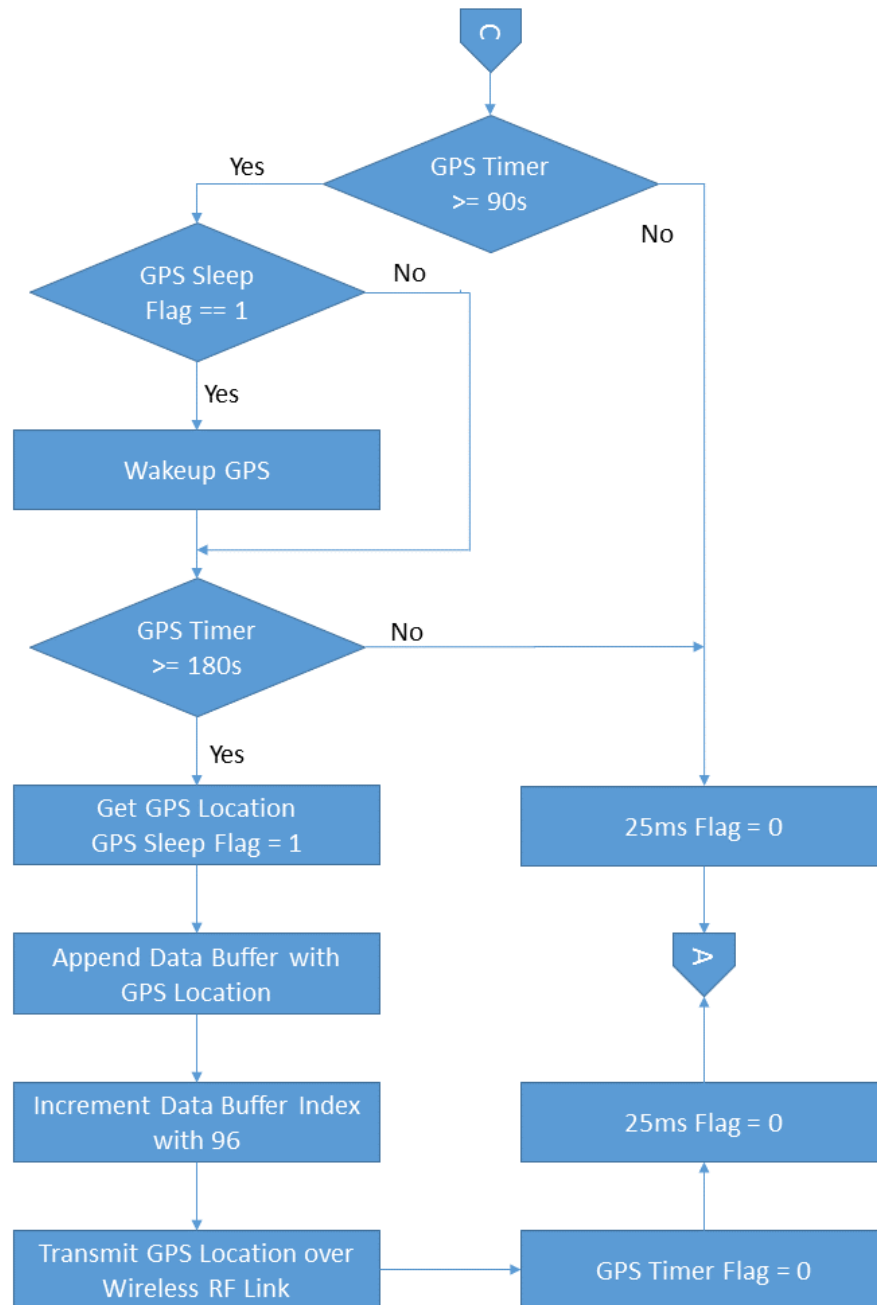


Figure 5.11: Routine 2: Main flow C.

512 Bytes of accelerometer data are sampled constantly at 100 Hz and stored. The 512 Bytes are passed to the classifier which performs a number of mathematical operations on the data before returning the classification result (standing, grazing, walking, running or laying down). The mathematical operations include calculating the maximum, minimum, mean and variances for the  $x$ -,  $y$ - and  $z$ -axis of the 512 Bytes accelerometer data. This data is then placed in a vector as follows:

$$\mathbf{X} = (X_{max} \ Y_{max} \ Z_{max} \ X_{min} \ Y_{min} \ Z_{min} \ X_{mean} \ Y_{mean} \ Z_{mean} \ X_{var} \ Y_{var} \ Z_{var}) \quad (5.2.1)$$

Mr. Jacques Marais is part of the DSP group of the project and did the signal processing aspects of the project. The training data along with the video footage were given to Mr. Marais, after which he implemented the automatic behaviour classification of the data on a PC. After extensive research by Mr Marais, he provided the following four matrices ( $M$ ,  $S$ ,  $C$  and  $I$ ) and the mathematical model to be used in the OABCS.

$$\mathbf{M} = (M_1 \ M_2 \ M_3 \ M_4 \ M_5 \ M_6 \ M_7 \ M_8 \ M_9 \ M_{10} \ M_{11} \ M_{12}) \quad (5.2.2)$$

Where:

$$\begin{aligned} M_1 &= 154.691461498 \\ M_2 &= 102.042954426 \\ M_3 &= -42.793085385 \\ M_4 &= -74.2378208486 \\ M_5 &= -249.544787847 \\ M_6 &= -367.024096386 \\ M_7 &= 35.4355264127 \\ M_8 &= -67.3318307196 \\ M_9 &= -191.920002701 \\ M_{10} &= 5215.28634943 \\ M_{11} &= 15777.4450325 \\ M_{12} &= 13944.7530849 \end{aligned}$$

$$\mathbf{S} = \begin{pmatrix} S_{1,1} & S_{1,2} & S_{1,3} & S_{1,4} \\ S_{2,1} & S_{2,2} & S_{2,3} & S_{2,4} \\ S_{3,1} & S_{3,2} & S_{3,3} & S_{3,4} \\ S_{4,1} & S_{4,2} & S_{4,3} & S_{4,4} \\ S_{5,1} & S_{5,2} & S_{5,3} & S_{5,4} \\ S_{6,1} & S_{6,2} & S_{6,3} & S_{6,4} \\ S_{7,1} & S_{7,2} & S_{7,3} & S_{7,4} \\ S_{8,1} & S_{8,2} & S_{8,3} & S_{8,4} \\ S_{9,1} & S_{9,2} & S_{9,3} & S_{9,4} \\ S_{10,1} & S_{10,2} & S_{10,3} & S_{10,4} \\ S_{11,1} & S_{11,2} & S_{11,3} & S_{11,4} \\ S_{12,1} & S_{12,2} & S_{12,3} & S_{12,4} \end{pmatrix} \quad (5.2.3)$$

Where:

$$\begin{array}{ll}
 S_{1,1} = 1.05979668929e - 05 & S_{1,2} = 0.00160743576802 \\
 S_{1,3} = 0.0153541415865 & S_{1,4} = -0.000243452038942 \\
 S_{2,1} = -0.00369117195468 & S_{2,2} = -0.000615271756311 \\
 S_{2,3} = 0.000431993218797 & S_{2,4} = -0.00104162689555 \\
 S_{3,1} = -0.0113546495864 & S_{3,2} = 1.19571473322e - 05 \\
 S_{3,3} = -1.36361545956e - 06 & S_{3,4} = -3.12513842869e - 05 \\
 S_{4,1} = -0.000971607559402 & S_{4,2} = 0.00273583455925 \\
 S_{4,3} = -0.00469895766451 & S_{4,4} = -0.000188987000027 \\
 S_{5,1} = 0.00300247482829 & S_{5,2} = 0.000666035315289 \\
 S_{5,3} = 0.00405956123222 & S_{5,4} = -0.0195524475692 \\
 S_{6,1} = 0.0108582090951 & S_{6,2} = -1.48142017664e - 06 \\
 S_{6,3} = 5.2306584414e - 07 & S_{6,4} = 2.81007410694e - 05 \\
 S_{7,1} = -0.00104975804661 & S_{7,2} = 8.79124354261e - 05 \\
 S_{7,3} = 0.0114018458356 & S_{7,4} = -0.00232251858368 \\
 S_{8,1} = 0.00337825757433 & S_{8,2} = -0.00751885897916 \\
 S_{8,3} = 0.00583914459435 & S_{8,4} = -0.00527119595752 \\
 S_{9,1} = -0.00276732729515 & S_{9,2} = -2.60495485253e - 05 \\
 S_{9,3} = -1.78846141391e - 05 & S_{9,4} = -9.43011549554e - 05 \\
 S_{10,1} = 0.000811871209361 & S_{10,2} = -0.0156927727573 \\
 S_{10,3} = 0.00442091302599 & S_{10,4} = -0.00192869728358 \\
 S_{11,1} = -0.00827705259017 & S_{11,2} = -0.000666918855429 \\
 S_{11,3} = 0.00979687568814 & S_{11,4} = 0.0238016550155 \\
 S_{12,1} = -0.00503374843062 & S_{12,2} = -1.8683484067e - 05 \\
 S_{12,3} = 1.53180352492e - 05 & S_{12,4} = -1.35983581079e - 05
 \end{array}$$

$$\mathbf{C} = \begin{pmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} & C_{1,5} \\ C_{2,1} & C_{2,2} & C_{2,3} & C_{2,4} & C_{2,5} \\ C_{3,1} & C_{3,2} & C_{3,3} & C_{3,4} & C_{3,5} \\ C_{4,1} & C_{4,2} & C_{4,3} & C_{4,4} & C_{4,5} \end{pmatrix} \quad (5.2.4)$$

Where:

$$\begin{array}{lll}
 C_{1,1} = -2.1642907511 & C_{1,2} = 1.63546003399 & C_{1,3} = -0.180074165815 \\
 C_{1,4} = 0.383958394791 & C_{1,5} = -2.80694302471 & C_{2,1} = -1.70985231351 \\
 C_{2,2} = -0.621782908322 & C_{2,3} = -0.0740915598075 & C_{2,4} = -0.909034669952 \\
 C_{2,5} = -1.25276916502 & C_{3,1} = 0.784542677644 & C_{3,2} = 0.119329327151 \\
 C_{3,3} = 7.74279662782 & C_{3,4} = -0.0262567776369 & C_{3,5} = -0.15474435567 \\
 C_{4,1} = 0.00398019359186 & C_{4,2} = -1.61967258489 & C_{4,3} = 1.39018645637 \\
 C_{4,4} = 0.16252744407 & C_{4,5} = -0.444515729817 &
 \end{array}$$

$$\mathbf{l} = (I_1 \ I_2 \ I_3 \ I_4 \ I_5) \quad (5.2.5)$$

Where:

$$I_1 = -5.36786482036$$

$$I_2 = -7.19840530057$$

$$I_3 = -3.10351613679$$

$$I_4 = -31.6232163173$$

$$I_5 = -4.0120731712$$

After obtaining these values, this project had to translate the math into a form that can be implemented on the limited resource microcontrollers of the WildMotes. This was done by means of careful processing and memory management techniques, data set reduction and classification system state management. The following calculations had to be performed to calculate the behaviour of the animals:

$$X_{scaled} = X - M \quad (5.2.6)$$

$$X_{dot} = X_{scaled} \bullet S \quad (5.2.7)$$

$$C_{dot} = X_{dot} \bullet C \quad (5.2.8)$$

$$R = C_{dot} + I \quad (5.2.9)$$

The result matrix (R) had the following elements:

$$R = (R_1 \ R_2 \ R_3 \ R_4 \ R_5)$$

The final behaviour is determined by calculating the maximum value of the  $R$  matrix and associating the index, where the maximum value occur, with behaviours as follows:

$R_1$  can be associated with Laying Down

$R_2$  can be associated with Standing

$R_3$  can be associated with Walking

$R_4$  can be associated with Running

$R_5$  can be associated with Grazing

For example, if the maximum value occurred at  $R_4$ , the sheep was running. The implementation of this technique is further presented in Figures 5.16 and 5.17.

#### 5.2.4 Routine 4: Network Protocol

Another substantial piece of software was required to implement the Network Protocol. Amongst others, this routine determines the flow and routing of data between nodes and from nodes to base stations. The details are discussed in under Network Design, Chapter 6.

#### 5.2.5 Other Software Routines and Python Scripts

Other software routines were also developed throughout the course of this project, as utilities, or to assist with other Masters projects. The routines were very similar to Routine



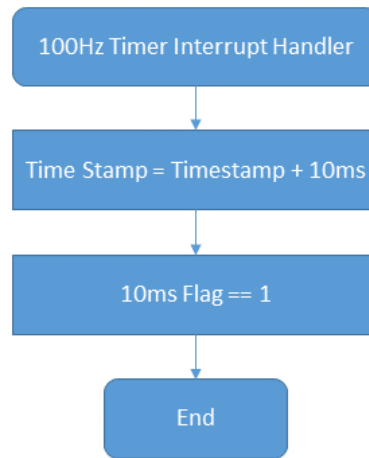


Figure 5.12: Routine 3: Timer interrupt handler.

1 and applied on two different occasions. Mr. Michael Struwig, as part of his Masters research at Stellenbosch University, was developing a kinetic energy generator as a means of energy harvesting for the sensor WildMotes developed in this project. He had to measure and log voltages and currents generated by the kinetic generator as a result of the acceleration experienced by the unit. One of the motes as discussed earlier, was programmed with the first utility routine and successfully utilised to log the acceleration, voltages and currents of the kinetic generator at a sampling frequency of 100 Hz. The second software routine was developed to assist Mr. Luca Lategan, another Masters student at the same institution, with logging GPS coordinates and altitude of trains travelling on specific routes in the Western Cape, South Africa. One of the motes was again appropriately set up to successfully record the required parameters. The latter two examples serve to illustrate the adaptability of the system. It is clear that the same system can also be applied as a convenient research tool for different objectives.

Various Python scripts were developed to aid specific outcomes of this project. The most comprehensive script developed, was the one controlling the hardware of the base stations and allowing it to interface with the CC1101 RF communication module (as described in Chapter 4.2). The latter formed the basis for the scripts developed to receive the GPS information (as described in Chapter 5.2.2) and classification information (as described in Chapter 5.2.3) over the wireless RF link. A script developed by Mr. Marais was used to download the RAW data stored on the microSD card to a computer. In order to download the data from the microSD cards, the script required the number of sectors logged on the microSD card. These were easily obtained as described in Chapter 5.2.1 and 5.2.2. Furthermore scripts were developed to plot the various parameters measured during the practical tests. These results can be seen in Chapter 8. The latter were used to plot the acceleration (in the  $x$ -,  $y$ - and  $z$ -axis), the temperature, measured sampling frequencies, results of ADC conversions and to extract and plot GPS coordinates on Google Maps.

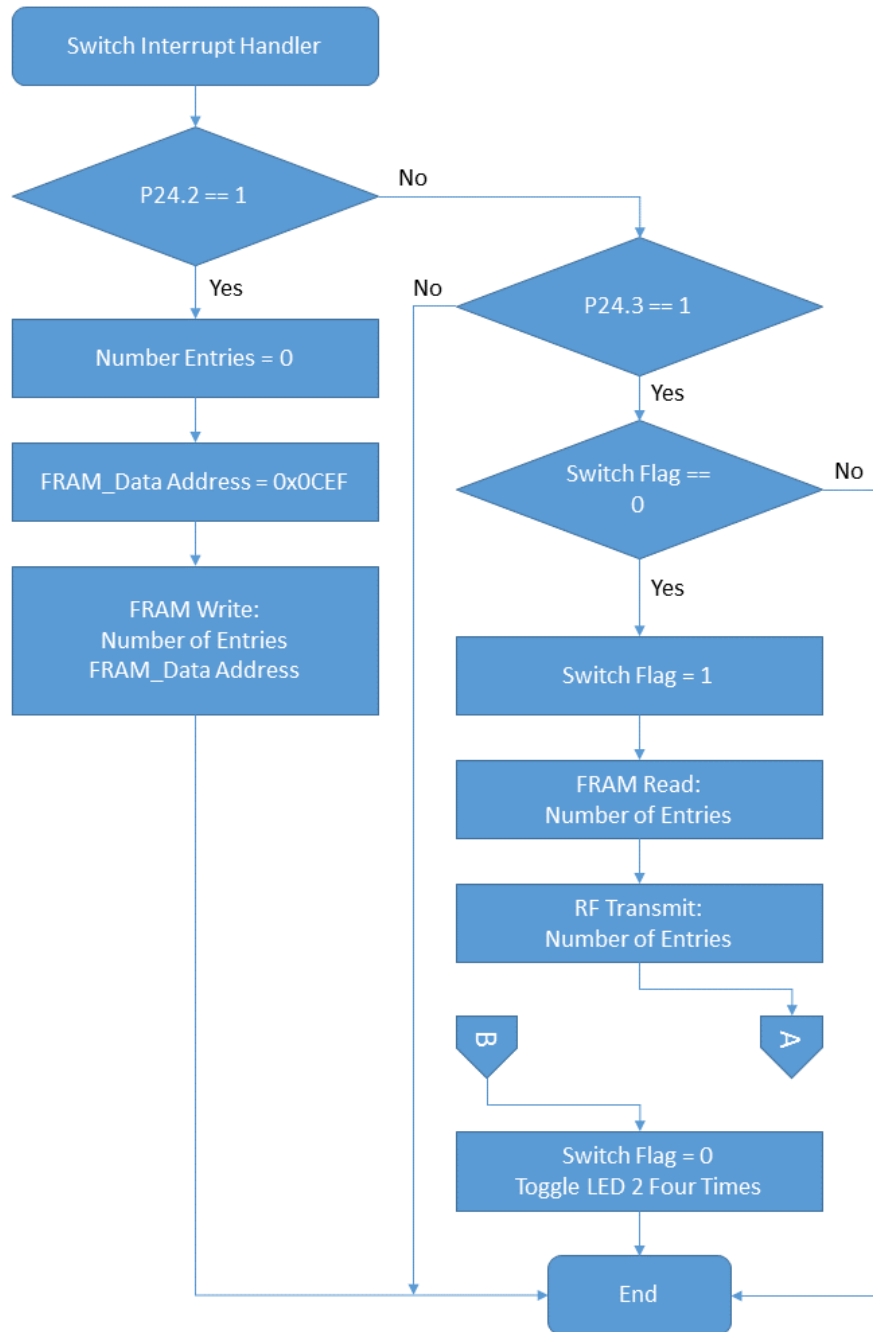


Figure 5.13: Routine 3: Switch interrupt handler A.

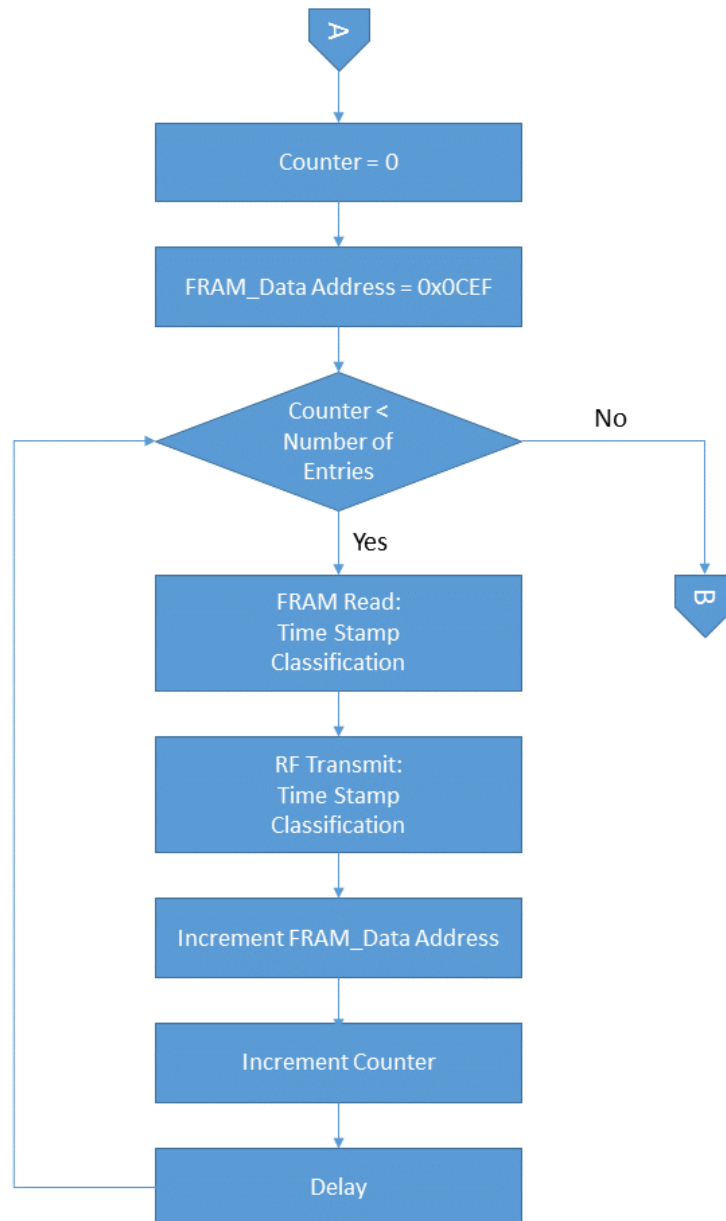


Figure 5.14: Routine 3: Switch interrupt handler B.

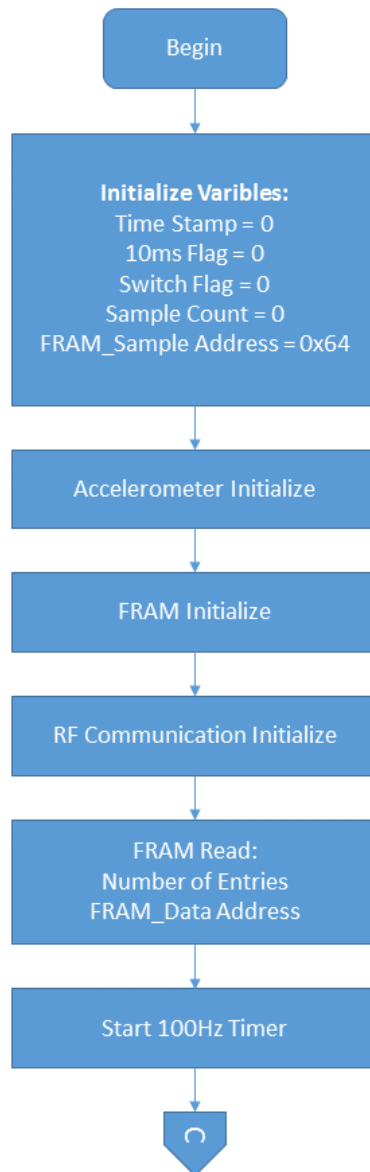


Figure 5.15: Routine 3: Initialisation.

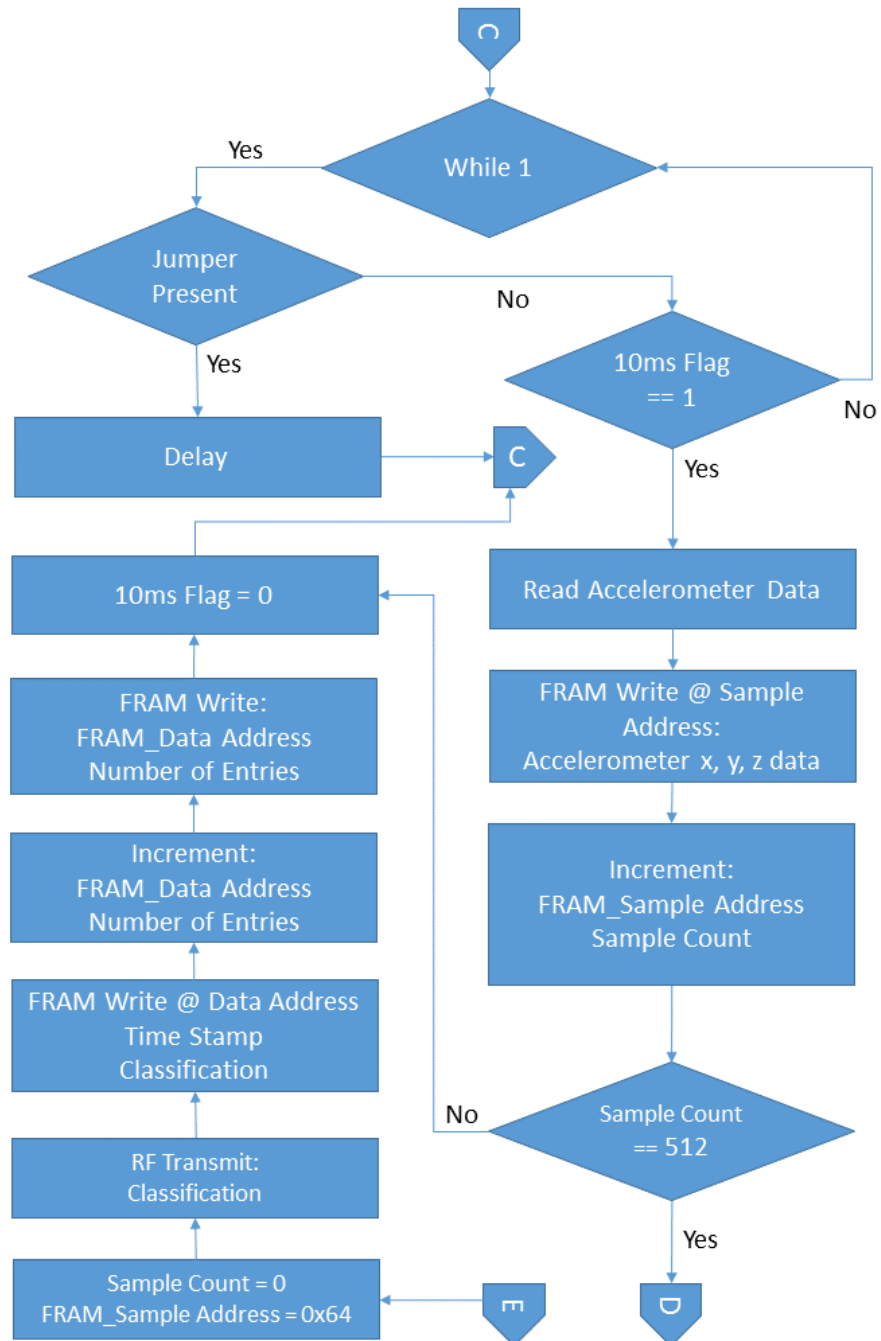


Figure 5.16: Routine 3: Main flow A.

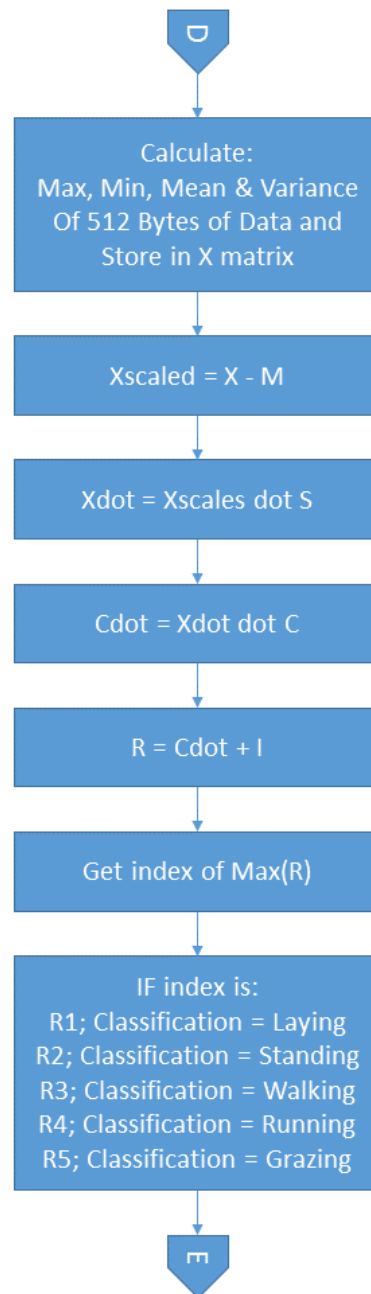


Figure 5.17: Routine 3: Main flow B.

This chapter considered the software design of the ABBMS. The initialisation- and functional software of the system was thoroughly explained. The details of the three main software routines, implemented in this project, were also discussed. The next chapter provides information regarding the WSN implementation of the ABBMS.

# Chapter 6

## Network Implimentation

This section of the thesis describes the design requirements for the WSN of the project. It describes the natural environment where the WSN typically needs to operate and describes some challenges faced in the practical implementation of the network and some techniques that can be followed to overcome some of the problems. Furthermore this section describes the network typology and protocols proposed for the project. The design and evaluation of two practical implementations are also covered.

### 6.1 Network Design Requirements

The network design requirements are mainly derived from the environment where the WSN will be deployed. As stated earlier in Chapter 1.1 most of South Africa's rhinos are located south of the Olifants river in the KNP. Figure 1.2 shows a map of South Africa and indicates where the KNP is located in the country. The KNP is the biggest game reserve in South Africa and stretches over 19 340 km<sup>2</sup>. Since the WSN needs to be deployed in the area south of the Olifants river one can assume the total coverage area of interest is roughly 9670 km<sup>2</sup>. The vegetation and terrain in the KNP are very variable and can range in anything from very dense vegetation with high trees and a number of small rocky hills in between, to wide open areas with less dense vegetation. From a RF communication point of view, the Southern part of the KNP proves to be a very difficult and variable environment to achieve robust and reliable communication using low power RF technologies. Apart from the extremely cluttered environment the placement of the collars further attenuates the signal as the collars are placed on the back legs of rhinos and therefore very close to the ground. The large body of the rhino and the ground around the animal both absorb some RF energy. Knowing that the communication distance of a WSN is greatly affected by the height of both the base stations and collars above ground it is no surprise that the low collar placement reduces the communication distance. From a RF communication point of view the collars are currently placed in probably the worst location on the rhinos. However, until other methods are found to fit the collars to rhinos, this is the only practical location where the collar can be placed on the animal to provide maximum comfort and the least irritation. Taking the abovementioned facts into consideration, the network requirements based on the current hardware design of the WildMotes, can be stated as follows:

- Investigate network protocols that would possibly increase network coverage
- The network protocol must be lightweight and able to run on the MSP430 family of processors
- The network protocol must be compatible with the TI CC1101 RF communication modules
- Implement and test the protocols on a small number of nodes
- The network protocol must be able to easily scale, in order to incorporate a large possible no. of nodes. This is necessary as coverage is required for the largest possible area South of the Olifants river in the KNP. Further research would be necessary to investigate such topologies.

## 6.2 Network Protocol

Extensive research has been done by various research groups on the topic of network protocols for WSNs [44] [45] [46] [47]. However for the purposes of this project, we made use of this extensive knowledge to find a lightweight protocol that will increase the network coverage of our WSN by means of a multi-hop routing protocol. The advantage of using a multi-hop protocol is that it enables each node within the network to potentially contribute to the global network coverage of the system. This allows the network coverage to grow as new nodes join the network. Figure 6.1 presents a classical star network typology and illustrates how the nodes outside the communication radius of the base station are not connected to the network. Figure 6.2 again, a multi-hop network typology and illustrates how each node contributes to increase the network coverage. Further to investigations regarding possible suitable WSN protocols, we propose two network strategies for implementation and if necessary, adaptation, in a multi-hop configuration. These are RIME and RPL.

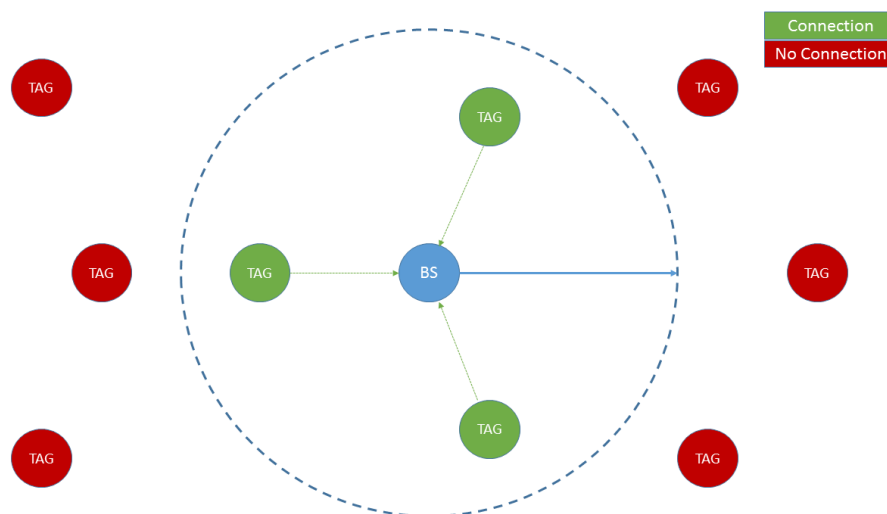


Figure 6.1: Star network typology.



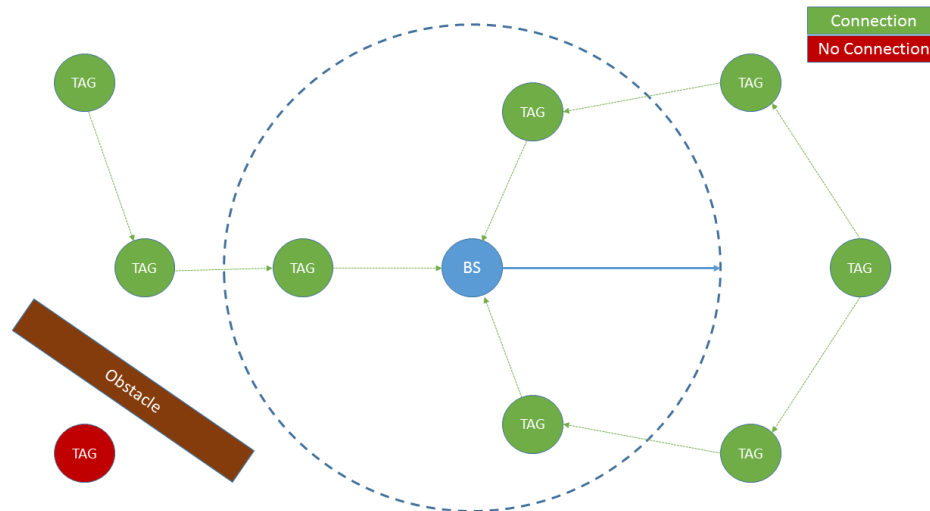


Figure 6.2: Multi-hop network typology.

### 6.2.1 RIME

The first protocol of interest is called RIME. RIME is a lightweight communication stack for WSNs [45]. “RIME was designed to make use of unusually thin layers, to simplify the implementation of WSNs and to facilitate code reuse” [45]. RIME is organised in layers. Each layer adds its own headers to outgoing messages, which are very small and typically a couple of bytes each. The lowest and most basic layer in the stack is the Anonymous Best-effort Single-hop Broadcast (ABC), which forms the basis for all other RIME primitives. The following information provides a brief description of the various primitives in the RIME stack, based on information from [46]:

- Anonymous Best-effort Single-hop Broadcast (ABC): “provides a way for upper layers to send a data packet to all neighbours that listen to the channel on which the packet is sent. No information about who sent the packet is included” [46].
- Identified Best-effort Single-hop Broadcast (IBC): “uses the ABC primitive and adds a single-hop sender address as a packet attribute to the outgoing packet” [46].
- Best-effort Single-hop Unicast (UC): “the UC primitive uses IBC primitive and adds a single-hop receiver address as a packet attribute to the outgoing packet” [46].
- Stubborn Single-hop Unicast (STUC): “repeatedly sends a packet to a single hop neighbour using the UC primitive. The STUC sends and resends a packet until a upper layer primitive or protocol cancels it” [46].
- Reliable Single-hop Unicast (RUC): “reliably sends a packet to a single hop neighbour. The RUC primitive uses acknowledgements and retransmissions to ensure that the data was delivered successfully” [46].
- Polite Single-hop Broadcast (POLITE): “Avoids that multiple copies of a specific set of packet attributes is sent on a specified logical channel in the local neighbourhood during a time interval” [46].

- Identified Polite Single-hop Broadcast (IPOLITE): “works in the same way as the POLITE primitive but adds the IBC primitive to include the identity of the sender in the packet” [46].
- Best-effort Multi-hop Unicast (MH): “sends a packet to an identified node in the network by using multi-hop forwarding at each node in the network. The application or protocol that uses the MH primitive supplies a routing function for selecting the next-hop neighbour” [46].
- Hop-by-hop Reliable Multi-hop Unicast (RMH): “the RMH primitive is similar to the MH primitive except that it uses the RUC primitive for the communication between two single-hop neighbours” [46].
- Best-effort Network Flooding (NF): “sends a single packet to all nodes in the network using the polite primitive for broadcasts at every hop to reduce the number of redundant transmissions. The NF primitive adds the end-to-end sender and end-to-end packet ID attributes on the outgoing packets. A forwarding node saves the end-to-end sender and packet ID of the last packet it forwards and does not forward a packet if it has the same end-to-end sender and packet ID as the last packet. This reduces the risk of routing loops, but does not eliminate them entirely as the NF primitive saves the attributes of the latest packet seen only. Therefore, the NF primitive also uses the time to live attribute, which is decreased by one before forwarding a packet. If the time to live reaches zero, the primitive does not forward the packet” [46].

A practical implementation as described in Section 6.3.1, implemented the Best-effort Multi-hop Unicast primitive of the RIME stack.

## 6.2.2 RPL

The second protocol of interest is called RPL (pronounces rippel). “RPL is a IPv6 Routing Protocol for Low power and Lossy Networks (LLNs)” [44]. For this project the User Datagram Protocol (UDP) transport layer was used on top of RPL. RPL allows multiple clients to send data to a server through multi-hop or single-hop paths. It also allows the server to send a message to multiple clients. A more detailed description of the RPL protocol can be found in [44]. LLNs are normally constrained in areas such as memory capacity, power consumption and processing power. RPL was designed to be used with LLNs to decrease the memory footprint, power consumption and processing power of the protocol. For this reason RPL should be a good routing protocol for the WSN in this project.

The COOJA network simulator for Contiki was used to run RPL network simulations. The first simulation used the RPL-UDP protocol and proved how a multi-hop typology can increase the communication range of the network. Nine nodes were placed in a line. Each node could only communicate to the nodes directly next to the node. Figure 6.3 indicates the node placement and the communication radius (green circle) and interference radius (gray circle) of the base station (node 1). In this experiment each node had to transmit 100 data packets to the base station by means of the multi-hop connections that exist within the network. The blue arrows in Figure 6.3 shows that the data was able to flow even from the furthest node (node 9) to the base station. Figure 6.4 contains

an example of the data flow. 100% of all packets were received at the base station proving that the RPL-UDP protocol can be a good implementations strategy to increase the communication range of a network.

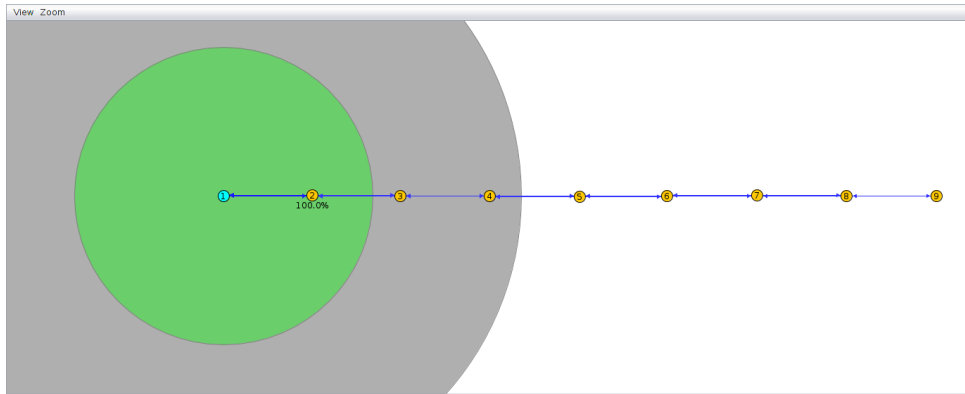


Figure 6.3: RPL-UDP simulation 1: Nodes placement.

Time	Note	Message
1:39:51.903	ID:1	DATA recv 'Hello 99 from the client' from 5
1:40:02.772	ID:5	DATA send to 1 'Hello 100'
1:40:03.134	ID:1	DATA recv 'Hello 100 from the client' from 5
1:40:04.339	ID:6	DATA send to 1 'Hello 100'
1:40:04.874	ID:1	DATA recv 'Hello 100 from the client' from 6
1:40:22.990	ID:7	DATA send to 1 'Hello 100'
1:40:23.614	ID:1	DATA recv 'Hello 100 from the client' from 7
1:40:31.261	ID:3	DATA send to 1 'Hello 100'
1:40:31.470	ID:1	DATA recv 'Hello 100 from the client' from 3
1:40:32.133	ID:9	DATA send to 1 'Hello 100'
1:40:32.960	ID:1	DATA recv 'Hello 100 from the client' from 9
1:40:40.380	ID:8	DATA send to 1 'Hello 100'
1:40:40.937	ID:4	DATA send to 1 'Hello 100'
1:40:40.951	ID:1	DATA recv 'Hello 100 from the client' from 8
1:40:41.316	ID:1	DATA recv 'Hello 100 from the client' from 4
1:41:00.293	ID:2	DATA send to 1 'Hello 100'

Figure 6.4: RPL-UDP simulation 1: Data flow.

A second simulation was done to investigate the working of the RPL-UDP protocol with multiple base stations and nodes. Figure 6.5 shows the placement of 5 base stations (nodes 1, 2, 3, 4, and 5) and 50 nodes. The protocol easily allowed all nodes to send all their data to one of the five base stations. The simulation proved how the RPL-UDP protocol can be applied in a more practical outdoor environment where multiple base stations will be used and where large numbers of nodes require network access.

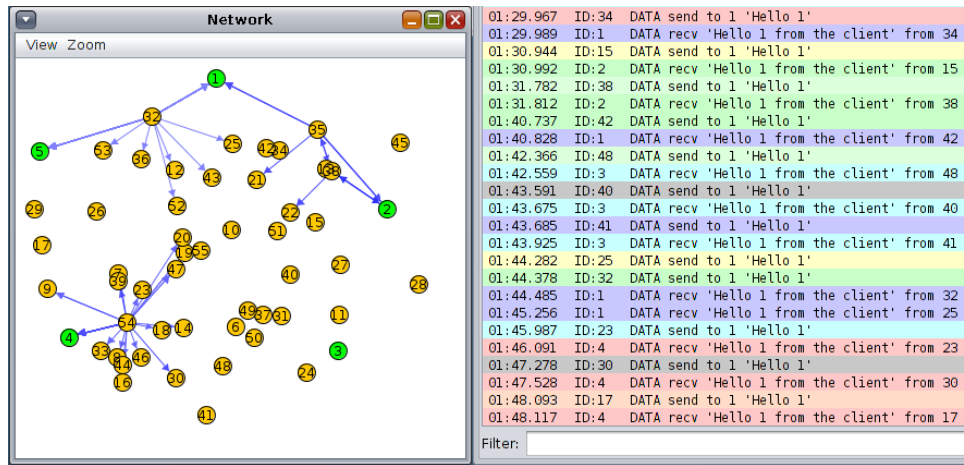


Figure 6.5: RPL-UDP simulation 2: Node placement and data flow.

## 6.3 Network Implementation

The practical implementation of the WSN was completed whilst on a research visit to Inria, Lille, France. Inria partially hosts and utilises the Internet of Things LAB (IoT-LAB), which is part of the Future Internet of Things (FIT) platform. FIT is in turn part of OneLab: a consortium consisting of five different higher education and research institutions. The FIT IoT-LAB is a very large scale open testbed that provides an infrastructure facility suitable for testing small wireless sensor devices. The FIT IoT-LAB features over 2700 wireless nodes spread across six different sites in France. The FIT IoT-Lab provides a means for scientists and engineers to perform WSN experiments in an internationally recognised manner. For our implementation we chose to deploy on the Grenoble testbed located at Inria Grenoble. The Grenoble testbed has 256 wsn430 nodes. The main hardware components of the wsn430 nodes are very similar to the WildMotes used in this project. Both the wsn430 nodes and the WildMotes are based on the MSP430 family of processors and CC1101 RF communication modules. This means that a protocol developed on the testbed can easily be translated from the testbed to the WildMotes with some minor adjustments to the port mapping of the processor. The Grenoble testbed only supports 860 MHz and 2.4 GHz as operational frequencies for the wsn430 nodes. We chose to work with the 860 MHz configuration as it was the closest to our 433 MHz application. However the difference in frequency does not influence our tests as the aim of the tests are only to prove that a multi-hop protocol can increase the network coverage of the proposed system. The protocols investigated under this thesis can be used on different operational frequencies. Both the RIME Best-effort Multi-hop Unicast stack layer and the RPL-UDP protocol were implemented on the Grenoble testbed to evaluate the functionality of the protocols under combination of the MSP430 and CC1101 hardware architectures.

### 6.3.1 RIME Implementation

The Best-effort Multi-hop Unicast stack layer was tested by means of a practical implementation of the stack. 8 Nodes were placed in a line and their communication ranges restricted so that a specific node could only communicate to the nodes directly next to it. Figure 6.6 shows the node placement and the communication ranges of the nodes. Node 202 was configured as the base station and collected all the data transmitted by the

other nodes. The remainder of the nodes were configured to transmit data to the base station on specific intervals and to forward packets received from other nodes to the base station. Figure 6.7 shows the data flow of the first implementation where node 181 was commanded to send a single data packet to the base station. Nodes 184 and 199 forwarded the data to the base station. Figure 6.8 shows the data flow of the second implementation where node 199 was commanded to send a single data packet to the base station. Node 200 forwarded the data to the base station. The gray nodes were active during the implementations, but no data flowed through them. The latter two implementations proved that the RIME Best-effort Multi-hop Unicast stack layer can be considered to develop multi-hop network protocols for the MSP430 and CC1101 architectures in order to increase the network coverage of a WSN, such as under investigation.

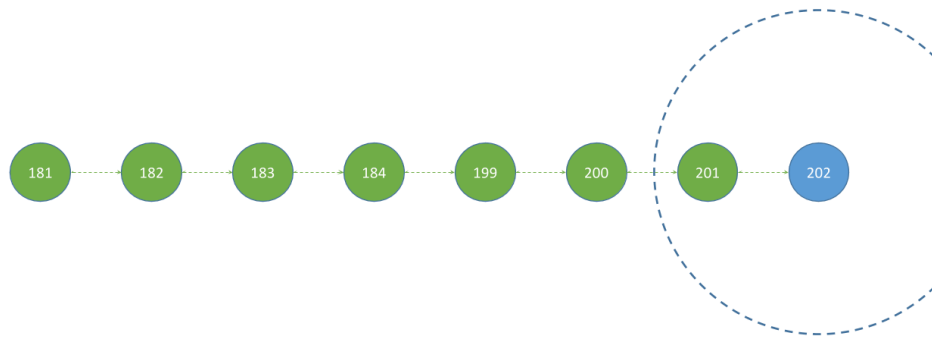


Figure 6.6: RIME implementation: Node placement.

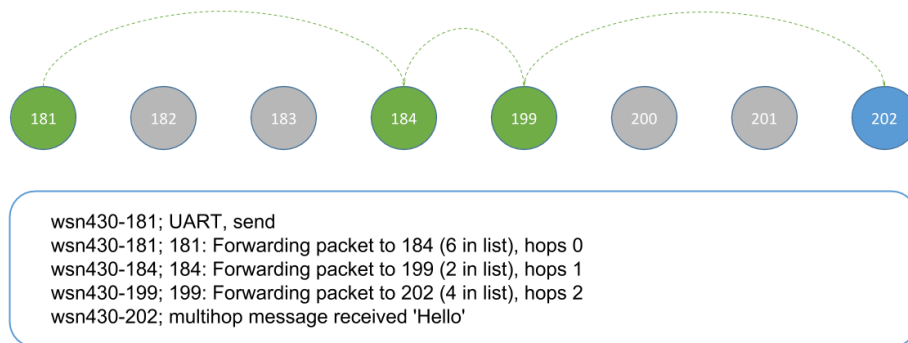


Figure 6.7: RIME implementation 1.

### 6.3.2 RPL Implementation

The RPL-UDP protocol was tested by means of five different practical implementations. 9 Nodes were placed in a line and their communication ranges restricted so that only specific nodes could communicate with the base station. Figure 6.9 depicts the node layout for the implementation of the RPL-UDP protocol. Node 185 was configured as the base station. Nodes 181 to 184 were placed within the communication range of the base station and nodes 177 to 180 were placed outside this range. If nodes 177 to 180 were to communicate to the base station, they would have to make use of a multi-hop path through

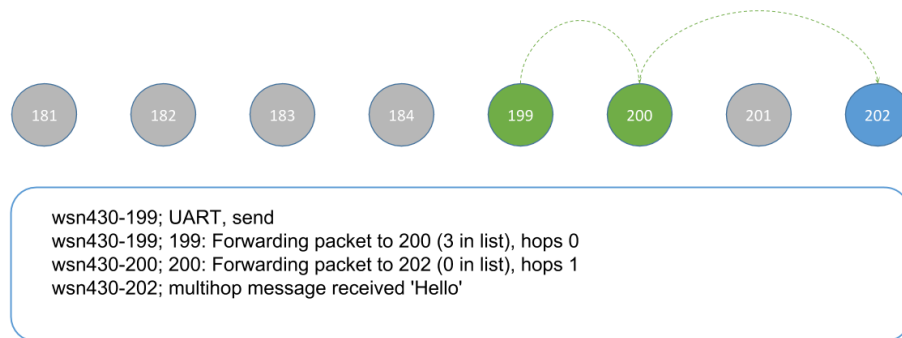


Figure 6.8: RIME implementation 2.

nodes 181 to 184. Figures 6.10 to 6.14 illustrate the five different implementations tested. In the figures, the blue node represents the base station, green nodes show that the node is connected to local neighbours or to the base station and gray nodes means the nodes were inactive during the specific implementation. The figures also show the first 10 data packets sent from the nodes to the base station. If the color of a specific packet is green, it means the base station successfully received the packet. If the color of a specific packet is red, it means the base station did not receive the packet. The implementation considered the first 10 packets as sufficient to represent alarm signals arriving at the base station.

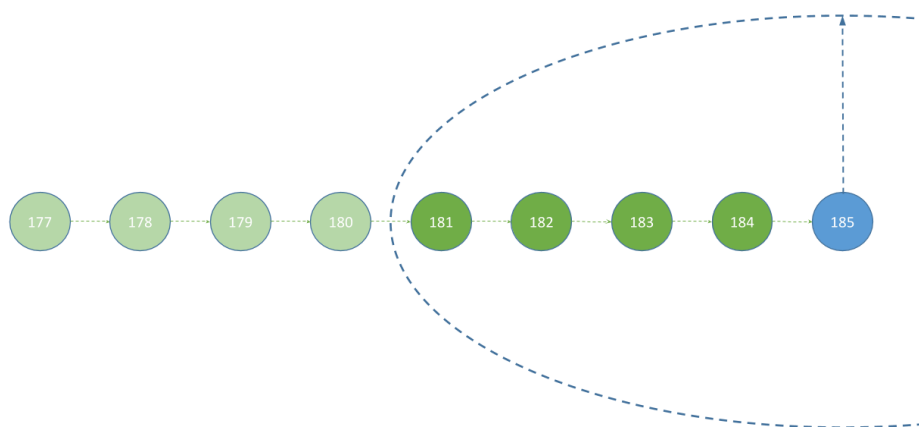


Figure 6.9: RPL implementation: Node placement.

The first implementation was done to prove that all data packets will be lost if the nodes (177 and 180 in this case) cannot establish a multi-hop path to the base station. Figure 6.10 demonstrates that no data packets from either nodes 177 or 180 were received by the base station. The purpose of the second implementation was to test if node 179 could communicate to the base station through node 181. Figure 6.11 proves that the base station was able to receive all the data sent by both nodes 179 and 181. The third implementation was intended to test how the system would act when node 177 had to send data to the base station by means of three hops. Figure 6.12 proves that node 177 was able to successfully send 80% of its data packets to the base station. Node 179 was able to send 90% of its data packets and node 181 all its data packets. The reason for the loss

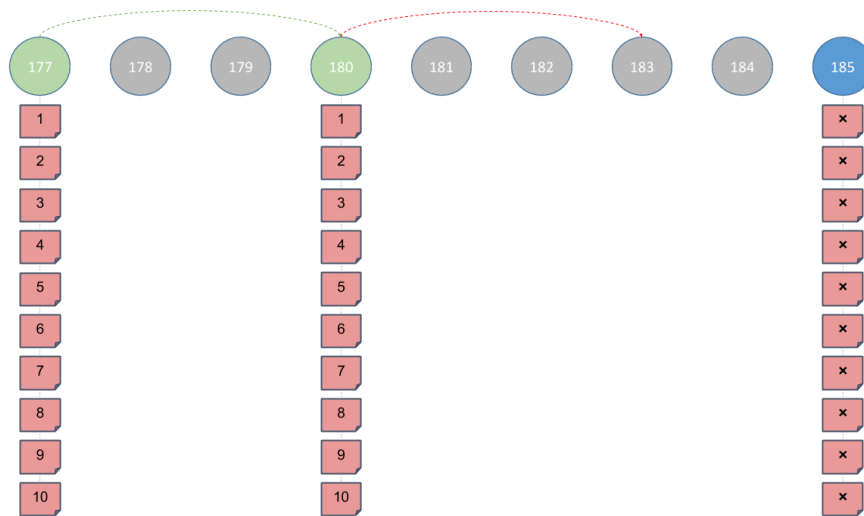


Figure 6.10: RPL implementation 1.

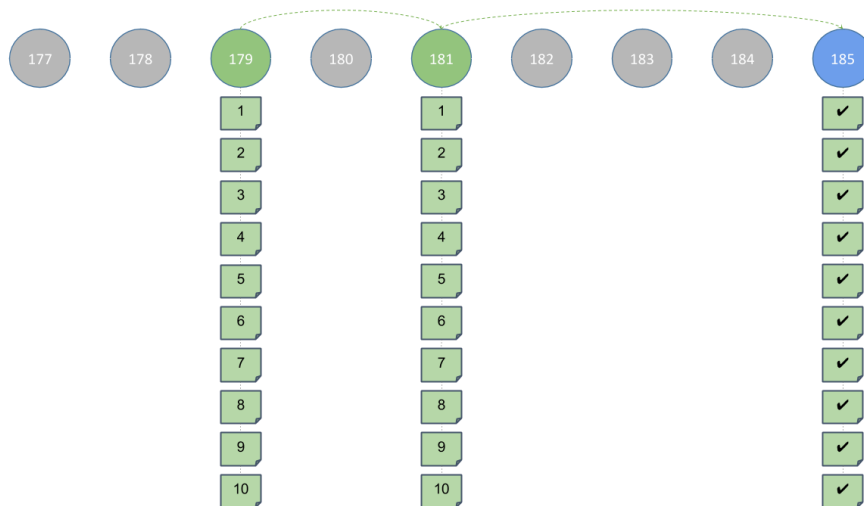


Figure 6.11: RPL implementation 2.

in packets, is network collisions as each node transmits its data on a random time with the system not optimised for collision avoidance. This loss of packets can further be seen in Figure 6.13, which illustrates the fourth implementation. This tested the functionality of the network when all nodes were activated on the same time and configured to transmit data at very short random intervals. The packet loss was greatly decreased by means of a simple change in the network startup sequence. In implementation five, each node was started shortly after the other, as opposed to starting the nodes at the same time. This increased the probability that each node will attempt to communicate when the medium was free. Figure 6.14 proves that the slight change in the startup sequence had a great effect on the number of packets received by the base station which was able to receive 97.5% of all data packets. Taking the abovementioned results into consideration proved that the RPL-UDP protocol can be implemented on the MSP430 family of processors along with the CC1101 RF communication modules, to accommodate a protocol that will extend the network coverage of the system. It is however advisable to make better use of

the various available networking layers, to optimise the system for collision avoidance and increase of the robustness of the system, to ensure higher probabilities of packet delivery.

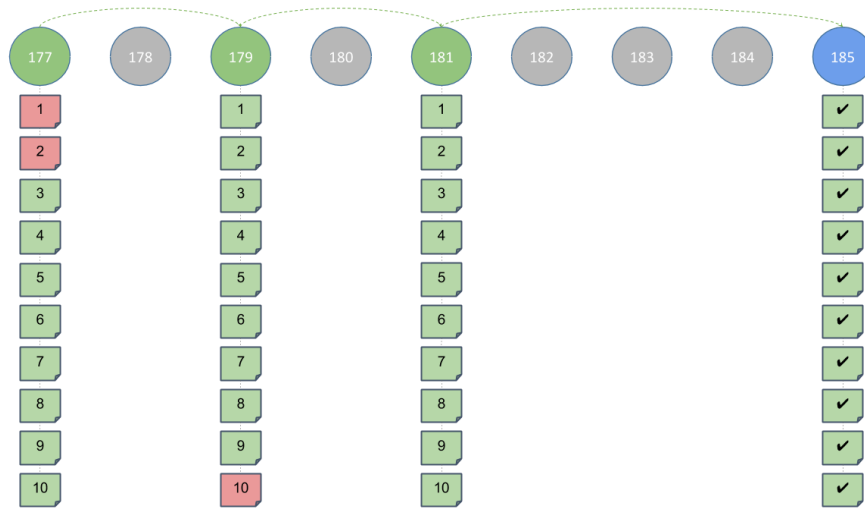


Figure 6.12: RPL implementation 3.

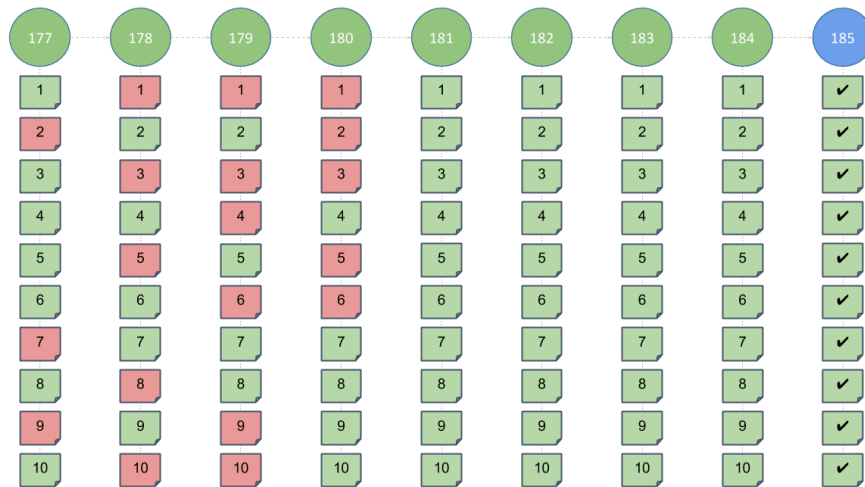


Figure 6.13: RPL implementation 4.



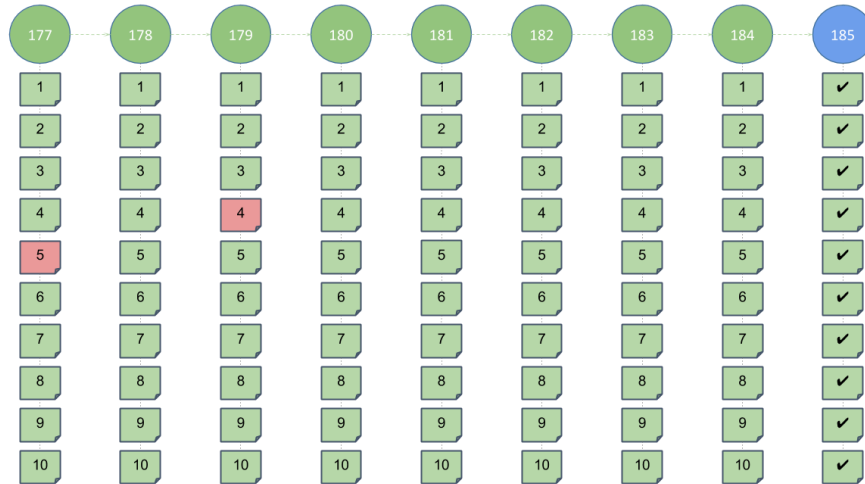


Figure 6.14: RPL implementation 5.

This chapter considered the various design considerations for the WSN of the ABBMS. It focused on two network implementations, that can be used in a multi-hop configuration, to extend the network coverage of the system. Both RIME and RPL were implemented successfully, however it is advised to use RIME in LLNs as it is a lightweight protocol. RPL has more overhead due to the IPv6 nature of the protocol. No practical range tests were performed. The latter will be done in future work within the KNP. The next chapter provides design considerations for energy harvesting systems to be used with the ABBMS.

# Chapter 7

## Energy Harvesting

Energy Harvesting (EH) has become a hot topic in the last couple of years and is certainly also of great interest for this project. Although EH is not within the primary scope or focus area of this thesis, it is worth some brief overview as it does have a direct bearing on some topological and hardware design considerations. It can also be of interest for other researchers in the PREDNET domain and possibly be considered for future work on similar projects.

The energy problems relating to wireless sensor networks are very well known and can most of the time be derived from the same problem statement: WSNs need to be as small as possible and require very long lives. The latter two design requirements are naturally opposed. Designing small sensors forces the use of small batteries, which in turn have less capacity. It is common knowledge that some forms of EH can be utilised to extend the battery life of WSNs or even replace the batteries completely. Extensive research has been carried out on the topic of EH for WSNs by different research groups [48] [49] [50] and it will remain the focus for many groups during the next couple of years.

This discussion will touch on some practical considerations for EH technology selection when designing animal borne WSNs for specific animal species and their behavioural patterns. There are different sources of energy available for EH [50]. These sources include:

- Mechanical energy resulting from vibration, stress and strain [50];
- Thermal energy from heat sources [50];
- Solar energy from all forms of light sources [50];
- Electromagnetic energy captured via inductors, coils and transformers [50];
- Wind and fluid energy resulting from air and liquid flow [50];
- Body energy depending on foot movement, blood flow, skin potential and electrochemical energy from naturally recurring, or biological processes [50].

As solar energy is a very common source of energy for EH, it is worthwhile elaborating a little more on the topic. It is readily available and can provide usable power densities of up to 15 mW/cm<sup>2</sup> [50]. Solar energy is a particularly good choice for EH on fixed nodes within a network, where nodes can be placed in optimal positions and utilise small solar

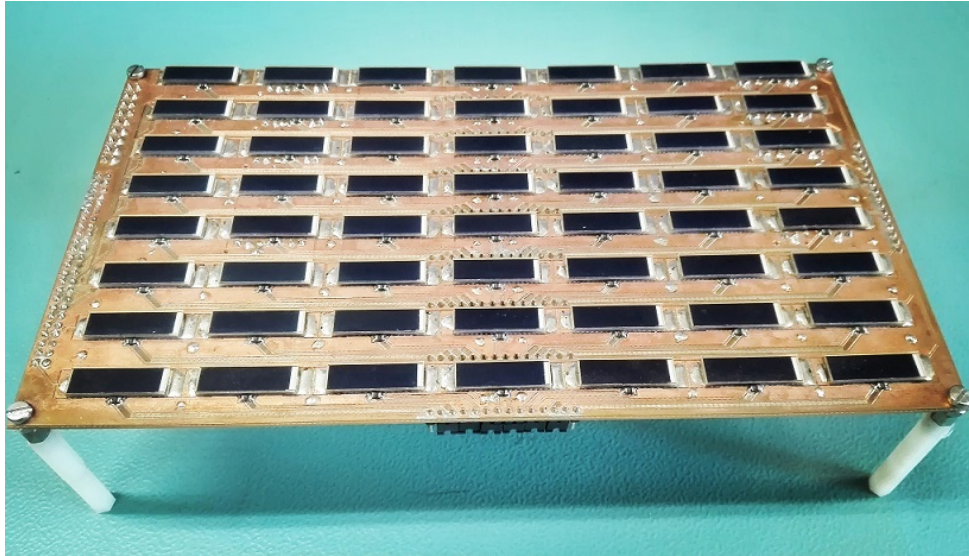


Figure 7.1: 56 Cell solar array.



Figure 7.2: Sheep with collars.

cells to sustain energy demands for long periods of time. During a research collaboration between ECE Paris and Stellenbosch University, Mr. J.B. de Chaisemartin from ECE was assisted in designing a small solar cell array intended for energy harvesting testing purposes. The solar array can be seen in Figure 7.1 and consisted of 56 small solar cells that could be configured in either series, parallel or a combination of series and parallel to achieve specific output voltage and current ranges. The solar array used the same energy harvesting IC as the one discussed in Chapter 4.1.4. Various tests were carried out to measure the performance of the solar cells and energy harvester for high- and low intensity light conditions, while evaluating the effect of different MPPC settings. The results and specific details of these experiments and tests can be found in [51]. From the report it is clear that solar is a good potential source of energy for outdoor applications on fixed nodes.



Figure 7.3: Rhino wallowing.

If placement of sensor tags on animals is considered, the practical implementation of EH techniques becomes more complicated as opposed to that in fixed nodes. Solar energy is a good option for diurnal animals that can be fitted with collars where the solar cells can be directed towards the sun, allowing the cells to be irradiated during the day. This is clearly not possible for nocturnal animals. Figure 7.2 presents an example where sheep are fitted with collars. In the same configuration, solar cells could possibly be placed at the top of the collars to face towards the sun. Some cases where solar EH may prove to be very difficult, if not impossible, is where the collars with cells cannot be placed facing the sun, or where the solar cells will get dirty very quickly. In our case where the collars were attached to the back legs of rhinos, the collars got covered in mud (as seen in Figures 7.3 and 7.4) very rapidly. Another case which might be difficult, is where collars are placed on elephants, as they often use water as a means to cool down (as seen in Figure 7.5) and can be submersed completely. They also love mud- and dust baths. With them, however, battery size might not be such a problem. Where solar EH techniques are not an option, other techniques have to be considered. Mechanical energy as a result of the physical movement of animals seems to be the only sensible option until researchers find more practical ways to attach and use solar-, thermal- (Peltier effect), wind- and chemical EH techniques. Micro kinetic generators could possibly be designed and optimised for specific animal types. One current idea is to use the force generated by the gait of a rhino, to convert from kinetic- to electrical energy. As shown in Chapter 8.3 the forces experienced by collars attached on the legs of rhinos are very high. These forces can be used to design optimal kinetic energy harvesters for rhinos. Mr Struwig is doing great work in designing micro kinetic generators and is also developing software tools to calculate the optimal design parameters for these micro kinetic generators.





Figure 7.4: Collar covered in mud.



Figure 7.5: Elephants swimming.

Taking all of the above mentioned information into consideration, it is clear that the type of animals and the behavioural patterns of the animals of interest, greatly affect the design considerations for animal borne EH systems. Engineers must therefore study the behaviour of the animals before designing EH systems. The next chapter presents the various measurements and results of the project.

# Chapter 8

## Measurements and Results

This section of the thesis evaluates the various measurements and results obtained throughout the extent of the project. Initial system tests were performed to determine if all the components worked as expected and to eliminate problems that might affect later results. The recording and evaluation of measurements obtained from three different subsequent experiments performed during the project are presented in detail, where after the results of two different trial techniques to measure a rhino's pulse rate, are discussed.

### 8.1 Initial Hardware Tests

Initial hardware tests were done to determine if all hardware components, worked as expected. The following hardware components had to be configured, for the experiments performed in this project: The microcontroller, accelerometer, vibration sensor, GPS, microSD card, FRAM modules, RF communication module, power regulators and all debugging circuitry. After configuring the hardware, the software routines, as described in Chapter 5.1, were used to perform various tests to evaluate each individual component. The tests showed that all components, except the GPS, functioned as expected. A thorough investigation revealed that the GPS was placed on the PCB, with a rotational error of 90 degrees. The error is depicted in Figure 8.1. According to the datasheet [25], the GPS module requires a ground plane of at least 2 cm, on either sides of the patch antenna. It further requires the top of the GPS module to be placed on the edge of the PCB. From Figure 8.1 it is clear that, due to the rotational error, the top was no longer at the edge of the PCB and only the right side of the module had a ground plane. Furthermore, the left side of the GPS was aligned with the edge of the PCB. This error greatly affected the Time to First Fix (TFF), which in some cases easily exceeded 40 min.

The problem was quickly resolved by utilising the modular design of the WildMotes. The on-board GPS modules were simply unplugged and external GPS expansion cards, with improved designs, were plugged into the WildMotes. The PCB layout of the GPS expansion cards can be seen in Figure C.3 in Appendix C. The GPS expansion cards reduced the TFF to similar values as mentioned in the datasheet. A further problem was encountered regarding the accuracy of the obtained coordinates, which in some cases, were more than 100 m away from the actual place of measurement. Upon investigating the problem, it was found that the horizontal delusion of coordinate precision values, was excessive. This was true for at least the first coordinate, after performing a GPS warm start. The problem was resolved by discarding the first 10 coordinates obtained, after

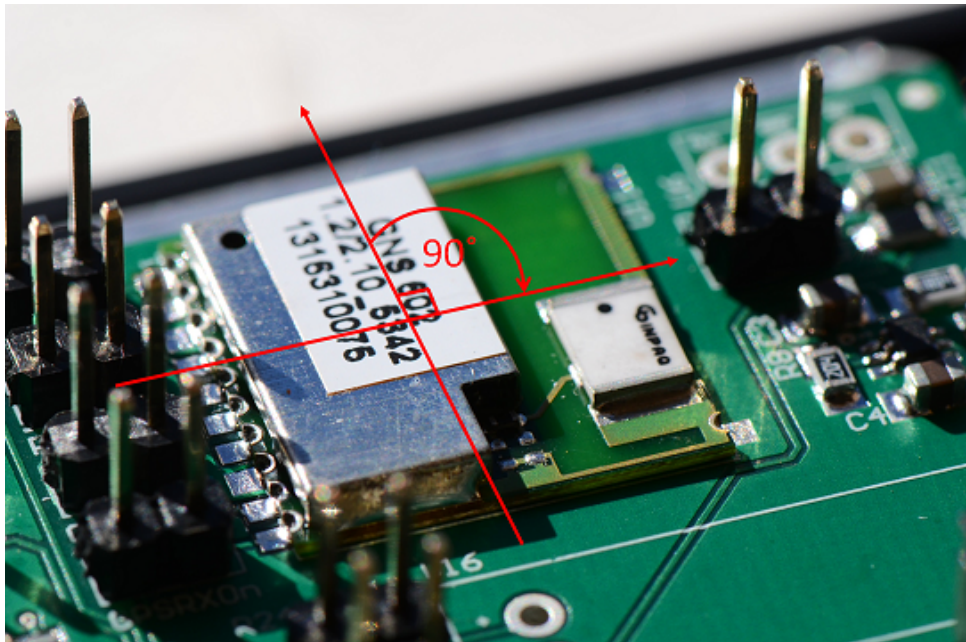


Figure 8.1: 90 Degree GPS rotational error.

a cold- or warm start. This technique ensured that the coordinates had a precision of  $\pm 3$  m. Further tests proved that the WildMotes were ready for the rest of the various experiments as required by the project. These commenced subsequently.

## 8.2 Experiment 1: 100 Hz Accelerometer Logger for Sheep Behaviour Classification

The objective of the first experiment was to configure five Sheep Collars with Software Routine 1 (as described in Chapter 5.2.1) and to determine if the collars can successfully log acceleration data at a rate of 100 Hz. Five Sheep Collars were prepared and clearly marked to indicate the collar's number and the accelerometer's  $x$ -,  $y$ - and  $z$ -axis orientation. A typical experiment started by fitting the collars on the sheep as illustrated in Figure 8.2, after which the sheep were moved to a camp where they could be observed. Figure 8.3 shows how the accelerometers were orientated on the sheep.

The  $x$ -axis of the accelerometer was positioned to correspond with sway motion (left-right acceleration), the  $y$ -axis with heave motion (up-down acceleration) and the  $z$ -axis with surge motion (front-back acceleration). The sheep were observed throughout the day and their behaviour documented as either walking, standing, grazing, running or laying down. Tables A.1 to A.10 in Appendix A, list all the observations made with regards to the behaviour of the sheep. At the end of the day, the sheep were collected and the collars removed, after which each collar's data was downloaded to a computer. The experiments were repeated for a period of 11 days. A total volume of 10.174 GB (roughly 431 M samples and 136 h) of data were collected throughout this period. Table 8.1 shows an example of the RAW data saved on the microSD card. Six data fields were logged namely: *Run Id*, *Time Stamp*, *Int Counter*,  $x$ -,  $y$ - and  $z$ -axis acceleration. The *Run Id* starts at 1 after formatting the system and automatically increments every time when



Figure 8.2: Sheep Collar placement.

there is a system failure or reset. Although this hardly ever happened, it was used as a backup strategy to realign the *Time Stamps* (which starts at 0 after system reset), with post processing techniques, if a system failure occurred. The *Time Stamps* were sampled at the beginning of the accelerometer data request event and were necessary to confirm that the sampling frequency remained constant. It was further utilised to align the data with the time of specific observations made. The *Int Counter* kept record of the number of interrupts caused by the SQ-SEN-200 nano-power tilt and vibration sensor during each 10 ms window (resetting the *Int Counter*) between samples. Columns  $x$ ,  $y$  and  $z$  held the  $x$ -,  $y$ - and  $z$ -axis acceleration of each data sample.

Figure 8.4 shows a plot of the acceleration data collected throughout the day of the 4<sup>th</sup> of July 2014. Figure 8.5 proves that the system was able to constantly log data at 100 Hz. Table 8.2 is an extract of Table A.2 in Appendix A and indicates that the sheep were running between 11:59 and 12:01. Figure 8.6 shows the acceleration data of this time period. The change in acceleration is apparent and large forces were experienced in the negative  $z$ -axis as expected for running behaviour. Figure 8.7 depicts the number of interrupts caused by the vibration sensor during each 10 ms window, between samples. The increase in interrupts corresponds with the time and increase of acceleration. Both these results confirm that the data is in line with the observations made.

After the data collection phase, the data were provided to the DSP group, to determine if the data and observations are suitable to classify the behaviour of the sheep, by means of automatic behaviour classification algorithms. This was indeed the case as



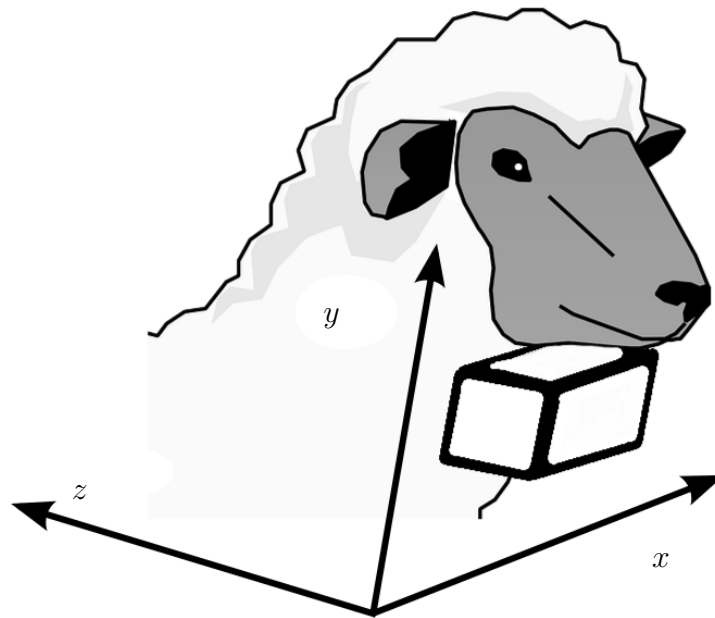


Figure 8.3: Accelerometer orientation on sheep.

Table 8.1: RAW data stored on the microSD card.

Run Id	Time Stamp	Int Count	x	y	z
01	4483470	0	40	142	-159
01	4483480	0	53	106	-153
01	4483490	0	55	84	-157
01	4483500	0	54	66	-129
01	4483510	1	57	35	-117
01	4483520	2	43	12	-72
01	4483530	0	42	-7	-62
01	4483540	1	16	-25	-46
01	4483550	2	3	-19	-51
01	4483560	0	12	-18	-39
01	4483570	3	43	-1	-40

Mr. Marais successfully used machine learning techniques to automatically classify the data. A research paper titled: “Automatic classification of sheep behaviour using 3-axis accelerometer data” [52], provides a comprehensive report of the results. From [52] it can be determined that linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) classifiers were trained using the following ten features calculated from the RAW data: Mean values, standard deviation, variance, skewness, kurtosis, maximum and minimum, energy, frequency-domain entropy, pairwise correlation between the axis and the average vector length. A greedy selection algorithm was applied to determine the best features for the highest accuracy. From [52] it was determined that the most important feature was the maximum and minimum values. If it is combined with the next two features selected by the greedy algorithm, only 1% of accuracy is sacrificed, as opposed to using all ten features. The LDA had an overall accuracy of 87.1% and the QDA had an

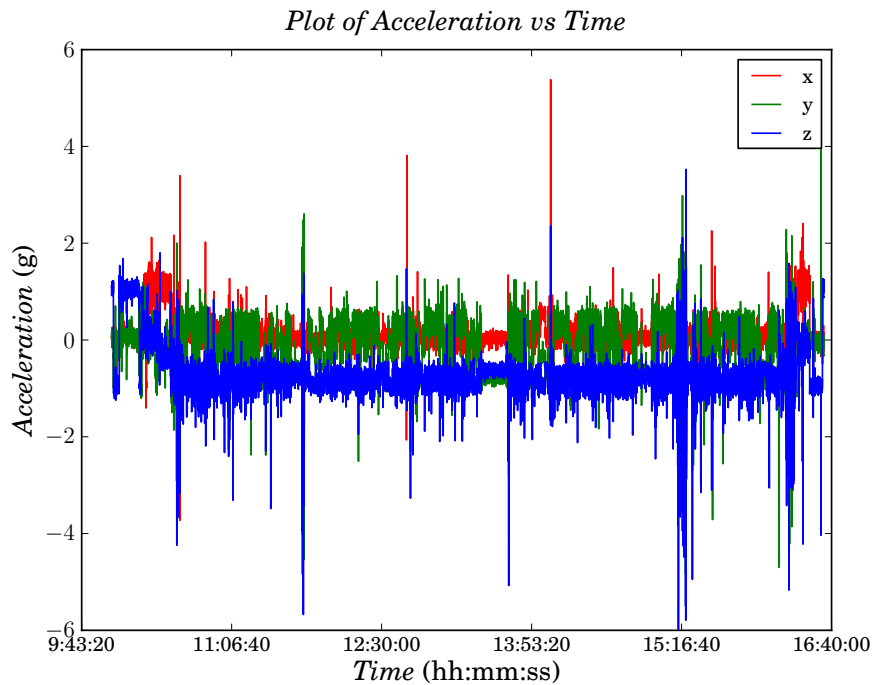


Figure 8.4: Acceleration data for the 4th of July 2014.

Table 8.2: Running behaviour observation time stamps on the 4th of July 2014.

Time 1	Time 2	Description
10:51	11:59	Grazing
11:59	12:01	Run
12:01	12:10	Stand

overall accuracy of 89.7%. The classifier could accurately distinguish between running, walking, standing and laying down with a certainty of 99.5%, 93.7%, 95.2%, 93.5% respectively. Grazing could however only be classified with an accuracy of 66.4%, as it was confused with laying down due to the similarities in the accelerometer orientation [52].

After obtaining the labeled data from Mr. Marais, some statistics were calculated for each class (running, walking, grazing, standing and laying down). Tables 8.3 to 8.7 are summaries of the mean values, variances and standard deviations of the  $x$ -,  $y$ - and  $z$ -axis acceleration (measured in  $g$ ) and the number of interrupts caused by the SQ-SEN-200 nano-power tilt and vibration sensor (*Int Counter*) during each 10 ms window. These statistics can be valuable for scientists, biologists and other researchers, requiring acceleration data as experienced by sheep for different behaviour. Although the mean values are quite smaller than initially expected, the variances and standard deviations describe that variability between the different classes. This should however not influence your expectation for the maximum and minimum values experienced, as it can easily be higher than expected. Figure 8.6 shows that the negative  $z$ -axis acceleration easily exceeded the  $-4g$  mark. However prior to this knowledge being available, the first experiment carried out, on the 3rd of July 2014, had the accelerometer's sensing range set to  $\pm 4g$  as seen in Table A.1. The latter caused the data to clip on  $\pm 4g$  and valuable insights were lost. The

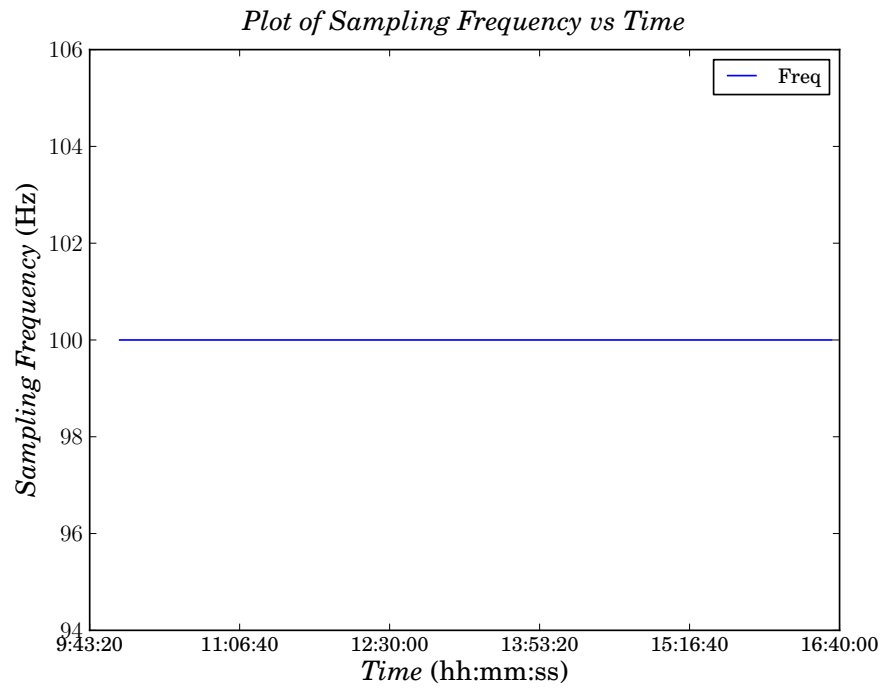


Figure 8.5: 100 Hz constant sampling frequency.

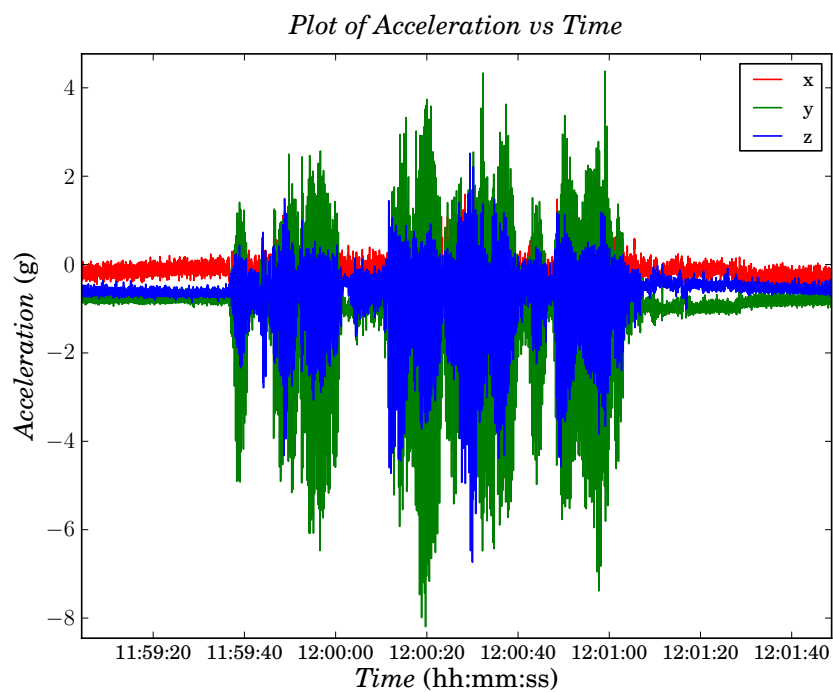


Figure 8.6: Acceleration of a running sheep.

sensing range was increased to  $\pm 16$  g and used a 13 Bit resolution (3.9 mg/LSB). This ensured that no data were lost due to clipping and a very good resolution was maintained.

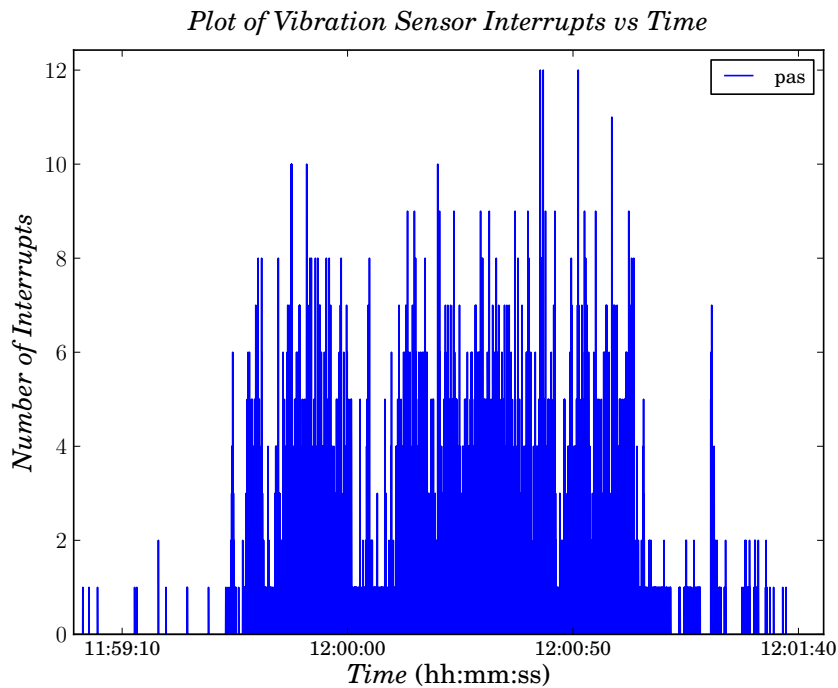


Figure 8.7: Vibration sensor interrupt distribution for a sheep running.

Table 8.3: Table of classification statistics: Sheep running.

Sheep: Running			
Description	Mean	Variance	Std Deviation
x acceleration (g)	0.104629	0.492670	0.701905
y acceleration (g)	-0.619420	1.260289	1.122626
z acceleration (g)	-0.805081	1.218342	1.103785
<i>Int Counter</i>	1.099027	2.634358	1.623070

Table 8.4: Table of classification statistics: Sheep walking.

Sheep: Walking			
Description	Mean	Variance	Std Deviation
x acceleration (g)	0.135030	0.079632	0.282192
y acceleration (g)	0.074005	0.226755	0.476188
z acceleration (g)	-0.767731	0.056136	0.236931
<i>Int Counter</i>	0.268335	0.389574	0.624158

Experiment 1 was implemented successfully, yielding good results. The collars were placed on sheep without harming them, or affecting their normal behaviour. The hardware configured with Software Routine 1, worked as expected. It was able to constantly log accelerometer- and the vibration sensor data. The DSP group successfully used the data and observations, to accurately classify at least 4 out of the 5 behavioural classes. Feedback from the DSP group mentioned that the resolution of hand documented observations were not fine enough. They suggested using video recordings as a future means of collecting ground truth data. They also suggested investigating a possible means of

Table 8.5: Table of classification statistics: Sheep grazing.

Sheep: Grazing			
Description	Mean	Variance	Std Deviation
x acceleration (g)	0.138419	0.062630	0.250259
y acceleration (g)	-0.474570	0.088228	0.297032
z acceleration (g)	-0.717264	0.031732	0.178135
<i>Int Counter</i>	0.070883	0.092390	0.303957

Table 8.6: Table of classification statistics: Sheep standing.

Sheep: Standing			
Description	Mean	Variance	Std Deviation
x acceleration (g)	0.057693	0.045536	0.213393
y acceleration (g)	0.294290	0.057402	0.239587
z acceleration (g)	-0.787956	0.033585	0.183261
<i>Int Counter</i>	0.074673	0.090509	0.300848

Table 8.7: Table of classification statistics: Sheep laying down.

Sheep: Laying down			
Description	Mean	Variance	Std Deviation
x acceleration (g)	0.231584	0.063224	0.251445
y acceleration (g)	-0.572203	0.058226	0.241300
z acceleration (g)	-0.693353	0.027605	0.166147
<i>Int Counter</i>	0.014646	0.018004	0.134179

reducing the noise levels of accelerometer readings, to improve the quality of the data. Finally, statistics gathered from the labelled data added valuable insights regarding the acceleration experienced by sheep for different behaviour classes.

### 8.3 Experiment 2: 40 Hz Accelerometer Logger for Rhino Behaviour Classification

The second practical experiment took place on a rhino rehabilitation farm in the Northern parts of South Africa. The first goal of the experiment was to determine if the hardware and software designed in this project, can be used to collect sensible data, which can be used for the automatic behaviour classification of rhinos. The second goal was to test the GPS tracking abilities of the collars. Two Rhino Collars (as described in Chapter 4.3) were configured with Software Routine 2 (as described in Chapter 5.2.2). They were set to record the behavioural- or movement patterns of the animals based on the different forces experienced by the on-board accelerometer at a constant rate of 40 Hz. The sampling rate of 40 Hz, was specified by the DSP group, as calculations showed that no aliasing will occur at this rate.

A typical experiment started in the afternoon during feeding time. The rhinos were distracted with food, in order to fit the collars, as shown in Figure 8.8. Once the collars were fastened, the rhinos were left undisturbed in their natural environment. The accelerometer's  $x$ -axis was positioned to correspond with heave motion (up-down acceleration), the  $y$ -axis with surge motion (front-back acceleration) and the  $z$ -axis with sway motion (left-right acceleration). The rhinos were observed throughout the day and video recordings were made to serve as high resolution ground truth data for the DSP team. The collars could typically only be removed the next day during feeding time. Each collar's data was downloaded to a computer. The experiments were repeated for a period of 6 days. A total number of 725 MB (roughly 17.5 M samples and 121 h) of data were collected throughout this period. The RAW data saved on the microSD card had a very similar structure to the data collected in Experiment 1.

After the data collection phase, the data was provided to the DSP group to determine if it can be used, along with the video recordings, to classify the behaviour of the rhinos by means of automatic behaviour classification algorithms. Mr. Marais successfully used machine learning techniques to automatically classify the data. Similar to Experiment 1 different classification techniques were implemented and the accuracy of each calculated. The implementation of a LDA classifier resulted in an overall classification accuracy of 85.98 %, with an accuracy of 93.33 %, 98.97 % and 65.64 % for standing, walking and laying down respectively. Unfortunately running could not be evaluated, due to the lack of enough video recordings of the class. The available recordings of running were however used in other investigations and calculations. The implementation of decision trees resulted in an overall classification accuracy of 95.81 %, with an accuracy of 99.48 %, 97.18 % and 90.77 % for standing, walking and laying down respectively. The confusion matrices for both the LDA- and the decision tree classifiers can be seen in Figures 8.9 and 8.10, respectively. Decision trees are easy to implement and in this case proved to be more accurate than using LDA classification.

Similar to Experiment 1, statistics were calculated for each class (running, walking, standing and laying down). Tables 8.8 to 8.11 are summaries of the mean values, variances and standard deviations for the  $x$ -,  $y$ - and  $z$ -axis acceleration (measured in  $g$ ) and the number of interrupts caused by the SQ-SEN-200 nano-power tilt and vibration sensor (*Int Counter*) during each 25 ms window, between samples. These statistics can provide scientists, biologists and other researchers with key insights regarding the acceleration experienced by rhinos for different behaviours.

Table 8.8: Table of classification statistics: Rhino running.

Rhino: Running			
Description	Mean	Variance	Std Deviation
x axis	-0.156502	10.941626	3.307813
y axis	0.202081	7.804492	2.793652
z axis	-1.713355	10.553719	3.248649
<i>Int Counter</i>	2.853421	10.156330	3.186900



Figure 8.8: Rhino Collar placement.

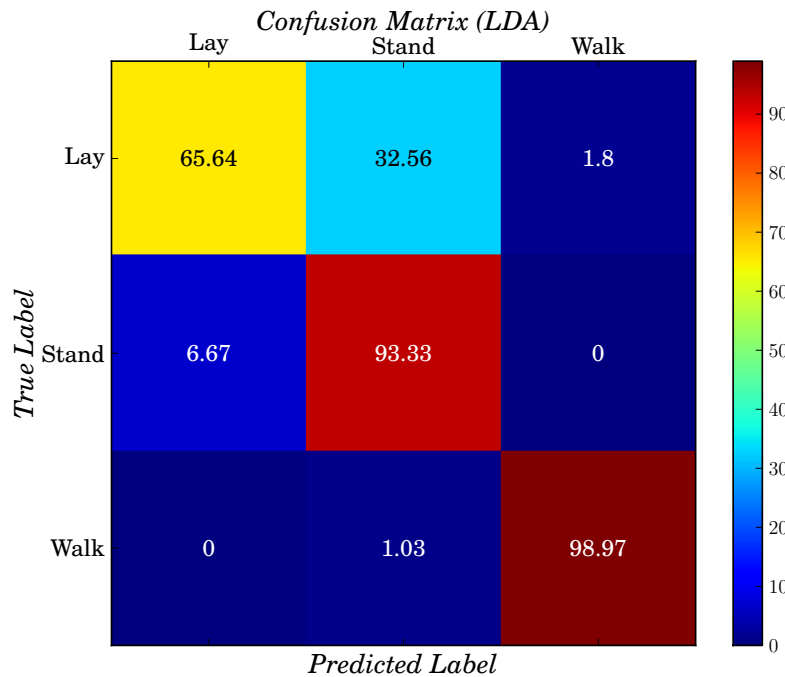


Figure 8.9: Confusion matrix of the LDA classifier: Rhino behaviour.

Figure 8.11 depicts the acceleration experienced by a walking rhino and Figure 8.12 depicts the same for a running rhino. The impulses created whilst walking and running is clearly visible. The acceleration experienced by the rhinos are much higher than that of the sheep. While walking, forces in both the  $x$ - and negative  $z$ -axis easily exceeded  $\pm 2g$  and  $\pm 10g$  while running. From the same figures it is apparent that slight modifications in the accelerometer's configuration registers and the use of the *Data Ready* function,

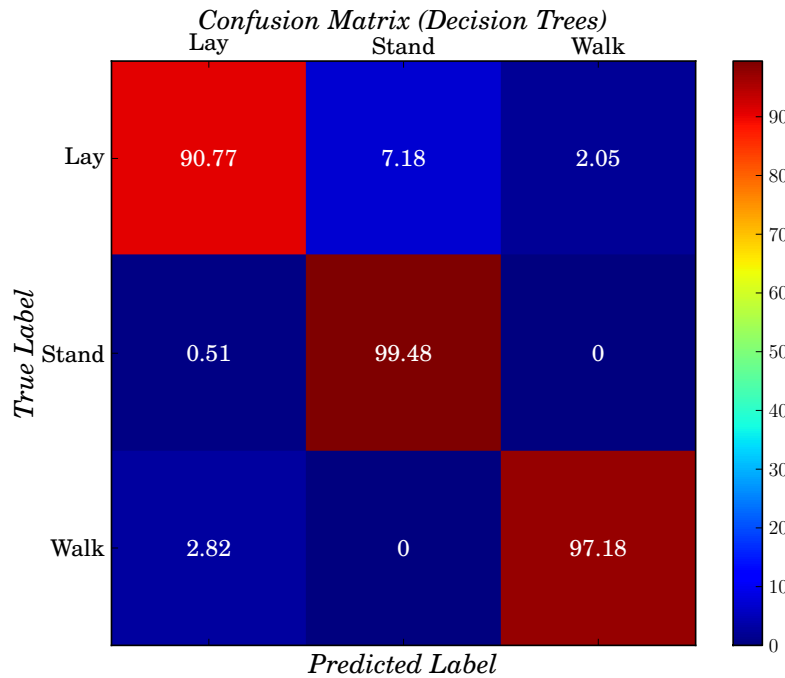


Figure 8.10: Confusion matrix of the decision tree classifier: Rhino behaviour.

Table 8.9: Table of classification statistics: Rhinos walking.

Rhino: Walking			
Description	Mean	Variance	Std Deviation
x axis	0.643475	0.520880	0.721720
y axis	-0.218298	0.274448	0.523878
z axis	-0.574131	0.547095	0.739659
<i>Int Counter</i>	2.501149	13.838283	3.719984

Table 8.10: Table of classification statistics: Rhino standing.

Rhino: Standing			
Description	Mean	Variance	Std Deviation
x axis	0.596448	0.013526	0.116300
y axis	-0.320286	0.066400	0.257682
z axis	-0.439497	0.131004	0.361945
<i>Int Counter</i>	0.046767	0.198756	0.445821

significantly decreased the signal noise. The latter is visible when comparing this data, with the data shown in Experiment 1. The improved signal quality equipped the DSP group to classify the behaviour of the rhinos with greater precision.

As mentioned earlier, the second goal of this experiment was to test the GPS tracking functionalities of the collars. The collars configured with Software Routine 2, sampled GPS coordinates every 3 min. Once a valid GPS coordinate was received, it was saved to the microSD card and transmitted via a RF link to the base station. Constant reception



Table 8.11: Table of classification statistics: Rhino laying down.

Rhino: Laying down			
Description	Mean	Variance	Std Deviation
x axis	-0.096536	0.014051	0.118536
y axis	-0.524731	0.039046	0.197602
z axis	0.134386	0.413884	0.643338
<i>Int Counter</i>	0.046252	0.229833	0.479409

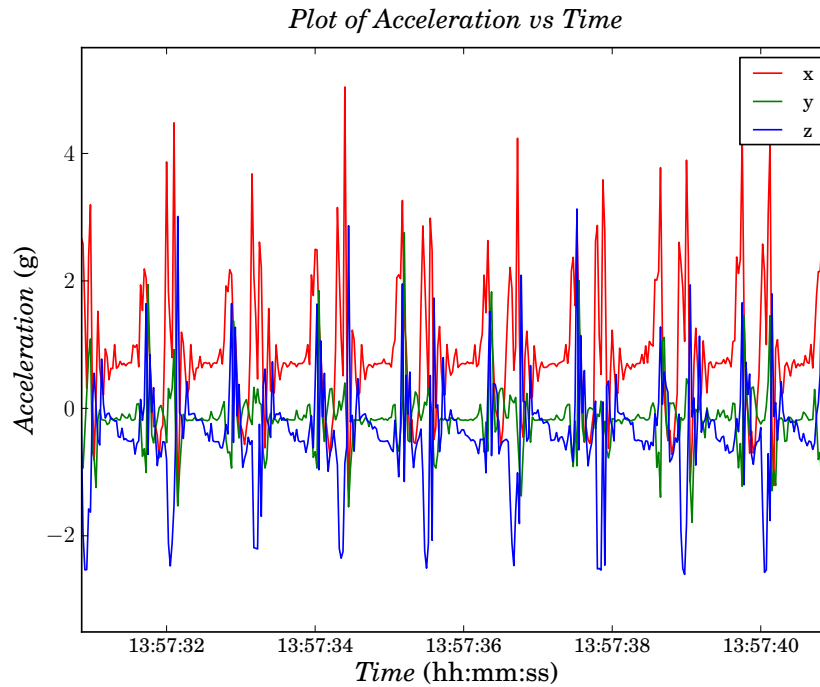


Figure 8.11: Acceleration of a walking rhino.

of GPS coordinates at the base station, gave the hardware team a clear indication of both where the rhinos were and that the collars functioned properly. The collected GPS coordinates can be seen in the following two maps: Figure 8.13 depicts a GPS localisation plot and Figure 8.14 is a heatmap representation of the data. Please note that, due to security reasons, the data does not reflect the immediate location of where the experiment took place, however, the relative distances between coordinates were maintained to correctly illustrate the movement of the animals. Live GPS localisation plots, such as in Figure 8.13, can be very useful to determine the exact location of a rhino showing abnormal behaviour. Heatmaps provide valuable insights in historical data and can assist in analysing the movement patterns of animals. Although only 121 h of data were collected, Figure 8.14, already reveals some repetitive movement patterns. In the figure, the areas marked in red, indicate where the rhinos spend a lot of time. By utilising this information, it is possible to run more strategic anti-poaching operations, as resources can be focused on specific areas.

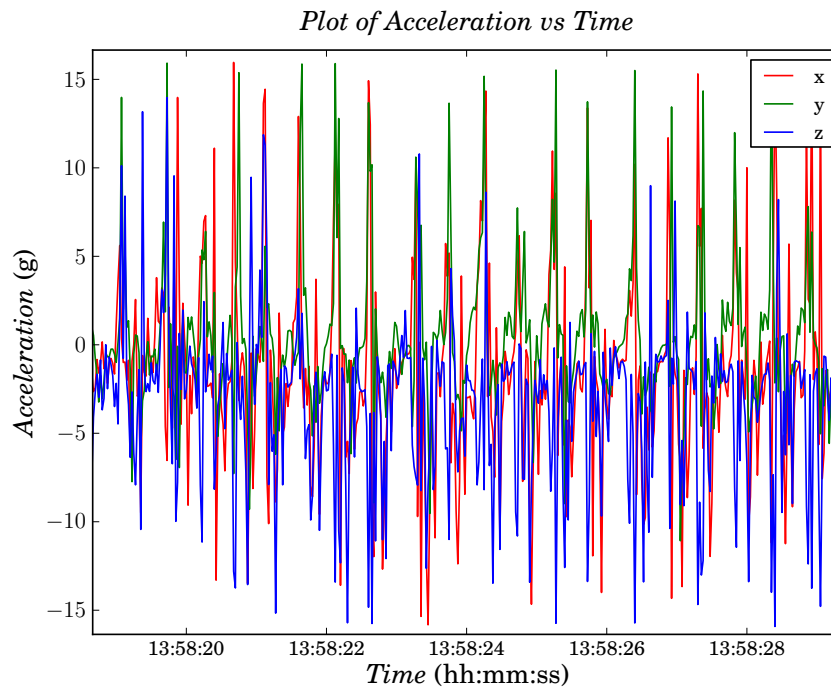


Figure 8.12: Acceleration of a running rhino.

Experiment 2, was implemented successfully, yielding good results. The Rhino Collars were successfully used to collect accelerometer data, which was utilised in the automatic behaviour classification of rhinos, with high accuracy. GPS coordinates and the heatmap representation thereof, revealed some insights of how the data can be used to assist in the process of tracking a specific animal, or tracking a collective group's movement patterns.

## 8.4 Experiment 3: On-animal Behaviour Classification System

The third experiment evaluated the performance of the OABCS. To our knowledge it is the first implementation of its kind. The OABCS consisted of three WildMotes configured with the software as described in Chapter 5.2.3. The WildMotes were placed on sheep and video recording were made, to be used as ground truth data. The goal of the experiment was to determine how accurately the OABCS can calculate the behaviour of animals in real time, by means of limited resource microcontrollers as typically found in animal tracking applications.

For the experiment the WildMotes sampled 512 Bytes of accelerometer data at 100 Hz. It then calculated the maximum, minimum, mean and variances for the  $x$ -,  $y$ - and  $z$ -axis acceleration. These 4 features were fed into a LDA classifier implemented on the WildMotes. Using the features as input, the classifier further had to calculate the dot product of two floating point matrices. The microcontroller classified the behaviour as either running, walking, standing, grazing or laying down (as described in Chapter 5.2.3). The behaviour was then stored and transmitted over the RF link to the base station. The classifier took 150 ms to calculate the behaviour from 512 Bytes of data. The FRAM

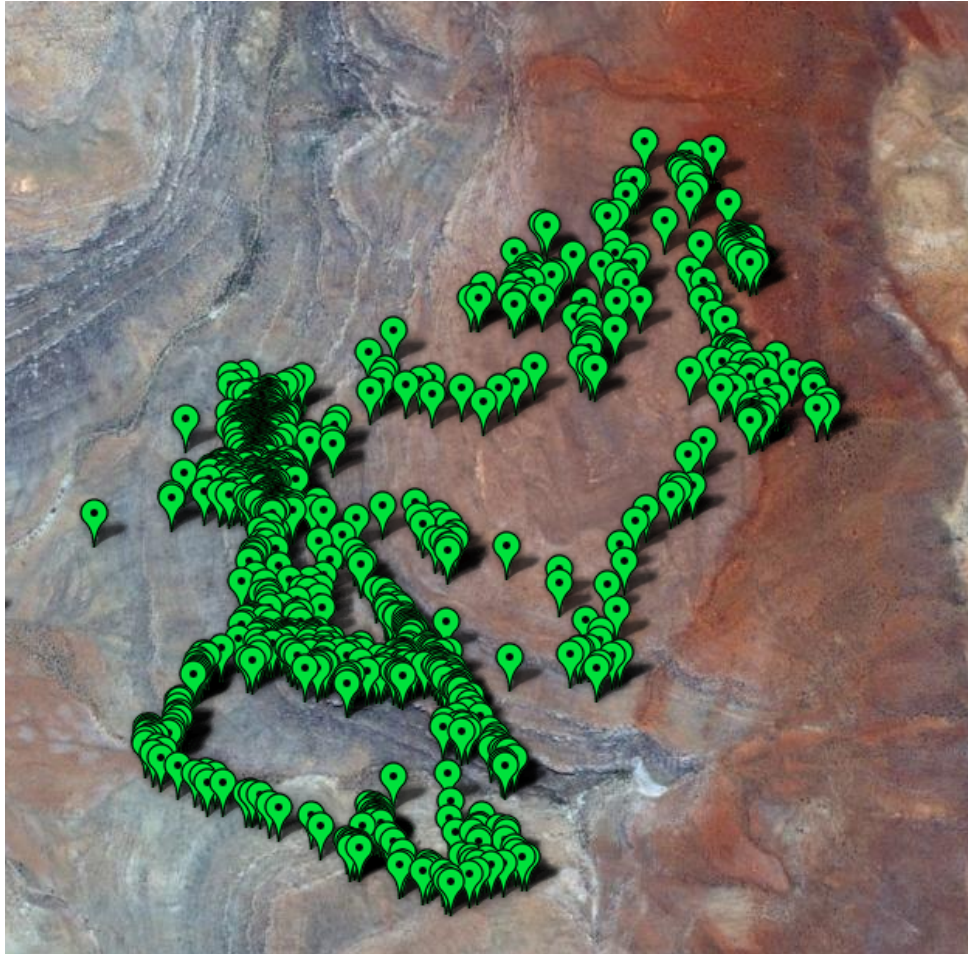


Figure 8.13: GPS coordinates obtained from free roaming rhinos.

modules were used as the primary means of data storage. The FRAM modules allowed memory access at 10 Mbps, while consuming less than 1.4 mA. Transmitting the data at an output power of 10 dBm, consumed 32.4 mA for a period of 30 ms. The WildMotes thus took 5120 ms to sample-, 150 ms to classify and store- and 30 ms to transmit the data. Accordingly, the classification result was stored at a rate of 5.3 s. Only 5 bytes were used to store each data entry. One byte for the classification and 4 bytes from the time stamp. The OABCS spent 99.434% of the time in classification mode (consuming 3 mA) and 0.566% in RF transmit mode (consuming 32.4 mA). The average current consumption is thus 3.17 mA. The system is limited in terms of battery lifetime and storage capacity. If one assumes that no power saving techniques are implemented and the OABCS logged the behaviour every 5.3 s, these limitations can be calculated as follows:

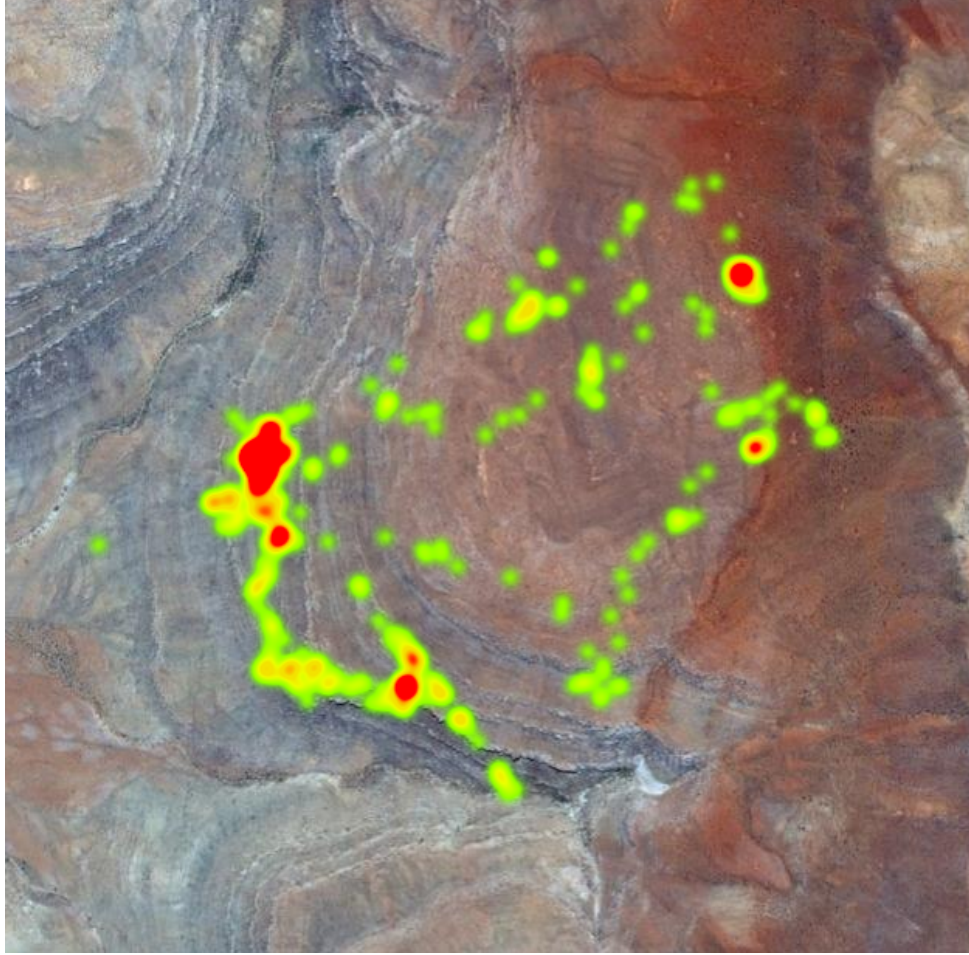


Figure 8.14: Heatmap representation of the GPS coordinates.

$$\text{Battery life} = \left( \frac{\text{Battery capacity (mAh)}}{24\text{h} * \text{Energy consumption (mA)}} \right) \quad (8.4.1)$$

$$= \left( \frac{3600}{24 * 3.17} \right) \quad (8.4.2)$$

$$= 47.32 \text{ Days} \quad (8.4.3)$$

$$\text{Storage life} = \left( \frac{\text{Storage capacity (bits)}}{\text{Data entry size (bits)}} \right) * \left( \frac{\text{Sampling period (s)}}{24\text{h} * 3600} \right) \quad (8.4.4)$$

$$= \left( \frac{4096000}{5} \right) * \left( \frac{5.3}{24 * 3600} \right) \quad (8.4.5)$$

$$= 50.25 \text{ Days} \quad (8.4.6)$$

During the experiment approximately 99 h of data (1 888 920 samples), were collected by the WildMotes. The video recordings were processed to correctly compare the classified behaviour with the observed behaviour. We found it difficult to monitor the behaviour of the sheep from the video footage. The first reason was the camera had to be placed far from the animals, to ensure the whole enclosure could be observed. Secondly the sheep with WildeMotes were placed back in the herd to ensure normal behavioural patterns. This, however, made it very difficult to identify the individual sheep. However, the observations were made to the best of our abilities and could still be used with a reasonable

degree of confidence. A Python script was written to compare the observed behaviour with the classified version. The comparator divided the data into running, walking and passive behaviour (which included: grazing, standing and laying down). Overall the OABCS had an accuracy of 82.37%, by only using 4 features as input to the system. This is 4.73% lower than the computer based LDA implementation, using 10 features as input to the system. It was able to correctly classify running, walking and passive behaviour, with respective precisions of 76.92%, 80.15% and 90.03%. Figure 8.15, depicts these results and shows how the inaccuracies were misclassified. It is believed that the accuracy of the system can be improved by placing a WildMote on a sheep, which is left alone in a camp, where it can more easily be observed. This will eliminate the difficulties of identifying the sheep's behaviour and lead to more accurate observations.

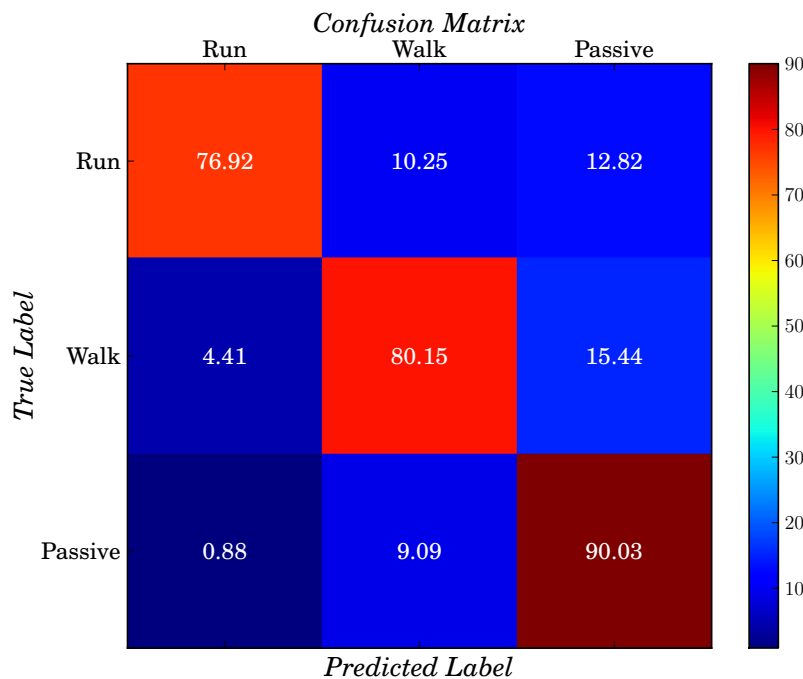


Figure 8.15: Confusion matrix of the OABCS: Sheep behaviour.

Unfortunately, there was no opportunity during the project, to attach the WildMotes on rhinos, for a second time. This meant that the accuracy of the OABCS could not be calculated for rhinos. However all the necessary steps were taken to ensure that the OABCS is ready for deployment on rhinos. As explained in Chapter 5.2.3, it is easy to reconfigure the OABCS for different animals. For rhinos, the sampling frequency must be changed to 40Hz and the values of the  $M$ ,  $S$ ,  $C$  and  $I$ , matrices, need to be updated to the following values:



$$\mathbf{M} = (M_1 \ M_2 \ M_3 \ M_4 \ M_5 \ M_6 \ M_7 \ M_8 \ M_9 \ M_{10} \ M_{11} \ M_{12}) \quad (8.4.7)$$

Where:

$$\begin{aligned} M_1 &= 389.38501292 \\ M_2 &= 151.899224806 \\ M_3 &= 90.8527131783 \\ M_4 &= -140.453488372 \\ M_5 &= -279.872093023 \\ M_6 &= -401.399224806 \\ M_7 &= 85.344373385 \\ M_8 &= -83.0532299742 \\ M_9 &= -163.05247416 \\ M_{10} &= 11664.2969858 \\ M_{11} &= 6302.61404709 \\ M_{12} &= 9273.66358401 \end{aligned}$$

$$\mathbf{S} = \begin{pmatrix} S_{1,1} & S_{1,2} \\ S_{2,1} & S_{2,2} \\ S_{3,1} & S_{3,2} \\ S_{4,1} & S_{4,2} \\ S_{5,1} & S_{5,2} \\ S_{6,1} & S_{6,2} \\ S_{7,1} & S_{7,2} \\ S_{8,1} & S_{8,2} \\ S_{9,1} & S_{9,2} \\ S_{10,1} & S_{10,2} \\ S_{11,1} & S_{11,2} \\ S_{12,1} & S_{12,2} \end{pmatrix} \quad (8.4.8)$$

Where:

$$\begin{aligned} S_{1,1} &= 0.000468037069056 & S_{1,2} &= 0.00217402749391 \\ S_{2,1} &= -0.000736736570192 & S_{2,2} &= -0.00395316463751 \\ S_{3,1} &= -0.00259674633596 & S_{3,2} &= -0.00195454005099 \\ S_{4,1} &= 0.0303776596278 & S_{4,2} &= 0.00331344414192 \\ S_{5,1} &= -0.00129163306642 & S_{5,2} &= -3.8806218439e - 05 \\ S_{6,1} &= -4.96501958827e - 05 & S_{6,2} &= 2.23524757987e - 05 \\ S_{7,1} &= -0.00105121871319 & S_{7,2} &= 0.00240585335335 \\ S_{8,1} &= 0.00160957270801 & S_{8,2} &= -0.00172201697702 \\ S_{9,1} &= -0.00514145213285 & S_{9,2} &= -0.00205526867576 \\ S_{10,1} &= -0.018024526105 & S_{10,2} &= 0.00345973381615 \\ S_{11,1} &= 0.00384185828433 & S_{11,2} &= 5.99719214852e - 05 \\ S_{12,1} &= -0.000119367278081 & S_{12,2} &= -8.49801094278e - 05 \end{aligned}$$

$$\mathbf{C} = \begin{pmatrix} C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,1} & C_{2,2} & C_{2,3} \end{pmatrix} \quad (8.4.9)$$

Where:

$$C_{1,1} = -5.034$$

$$C_{1,2} = 1.309$$

$$C_{1,3} = 0.049$$

$$C_{2,1} = -2.658$$

$$C_{2,2} = 4.985$$

$$C_{2,3} = 1.348$$

$$\mathbf{l} = (I_1 \ I_2 \ I_3) \quad (8.4.10)$$

Where:

$$I_1 = 8.83157782792$$

$$I_2 = -6.67244769299$$

$$I_3 = -35.8483433993$$

The OABCS was successfully tested on sheep, yielding good results. It achieved an overall average of 82.37 %, by using 4 features as input to the system. The WildMotes had a battery lifetime of 47.32 d, while constantly logging behaviours every 5.3 s. The OABCS managed to increase the storage lifetime of the system from 11.38 h to 50.25 d. This was achieved by doing calculations on the animal and only saving 5 bytes per entry as opposed to saving 512 bytes, for use in post processing techniques. For the first time a system is able to provide live behavioural data to scientist, biologists and nature conservationists. These new insights could definitely aid research efforts in terms of increased knowledge regarding different species behaviour and possible threats.

## 8.5 Behaviour Classification using an Omni-directional Vibration Sensor

An investigation was done to determine if data collected from a SQ-SEN-200 nano-power tilt and vibration sensor, can also be used to classify the behaviour of animals. The SQ-SEN-200 nano-power tilt and vibration sensor (referred to as the vibration sensor), acts like a normally closed switch which chatters open and closed as it is tilted or vibrated [27]. It consumes as little as 50 nA when triggered [27]. The vibration sensor responds to omnidirectional movements. The latter, in-effect reduces three dimensional movements to merely a couple of impulses.

The vibration sensors were active during both Experiments 1 and 2. It was set to interrupt the microcontroller whenever an impulse was generated by the sensor. These interrupts were counted over a period of 10 ms for Experiment 1 and 25 ms for Experiment 2. The total number of interrupts were saved to the microSD card, where after the interrupt counter was set to 0. A data frame from Experiment 2, is shown in Figure 8.16

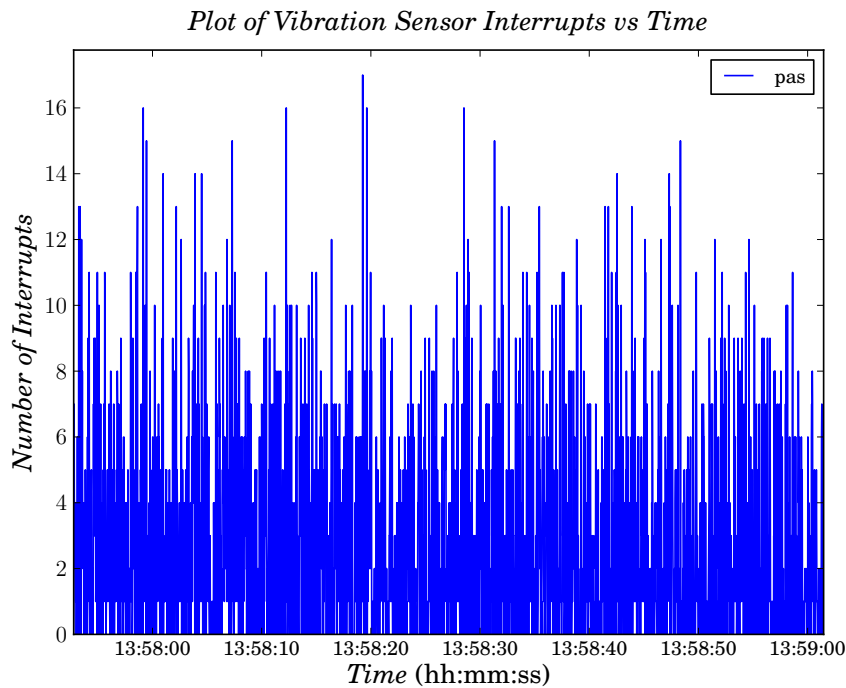


Figure 8.16: Vibration sensor interrupt distribution for rhino behaviour.

and illustrates how the number of interrupts varied over time. The data contains numerous samples and it is not easy to derive any information regarding the movement of the animals. Based on the fact that the number of interrupts were collected during each data frame, we suspect that the moving average of the data could reveal some information regarding the behaviour of the animals.

### 8.5.1 Behaviour Classification of Sheep using a Vibration Sensor

To start with, moving averages were calculated for each of the 5 classes of sheep behaviour. The data was analysed to get an idea of the values, for the different classes. The moving averages for running, walking and laying down, are depicted in Figures 8.17 to 8.19, respectively. The moving average of running behaviour had a mean value of 1.09905, with variance of 0.02813 and standard deviation of 0.16772. From Figure 8.17, it is clear that if a *Runing Threshold* was chosen between 0 and 0.8, most of the values above the *Running Threshold*, could be classified as running. However, lower threshold values would start to interfere with walking behaviour, as it had a mean value of 0.26835, with variance of 0.02379 and standard deviation of 0.15423. Figure 8.18 shows that if a *Walking Threshold* was chosen between 0 and 0.15, most of the values above the *Walking Threshold* and below the *Running Threshold*, could be classified as walking. By analysing Figure 8.19, it can be said that any values below the *Walking Threshold*, can be classified as *passive behaviour*, which in this case, includes: standing, grazing and laying down. Confusion between the 3 classes are inevitable, but, the *Running Threshold* and *Walking Threshold* can be fine tuned to provide application specific classification results. A more detailed discussion on application specific classification, follows later in this section.



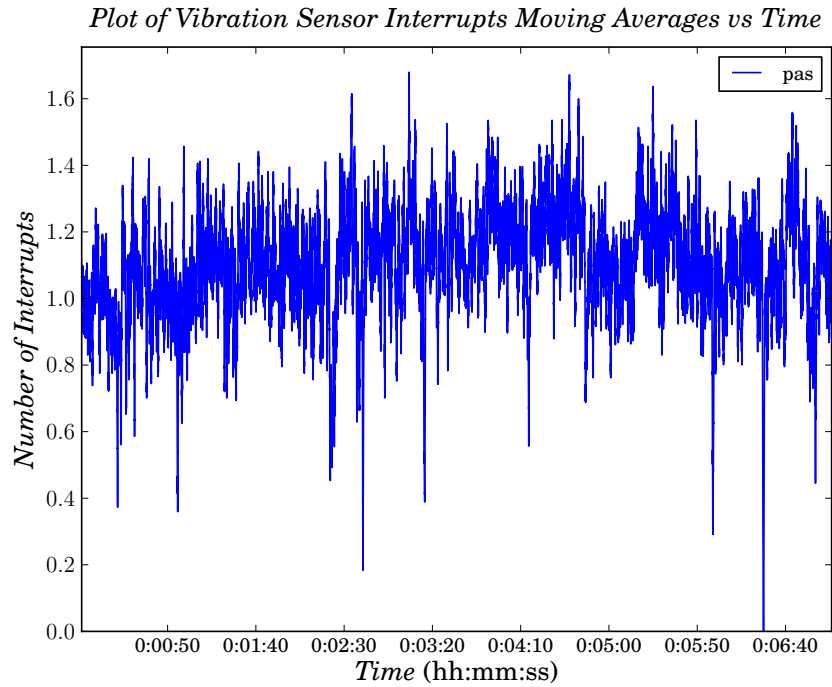


Figure 8.17: Moving average of vibration sensor interrupts: Sheep running.

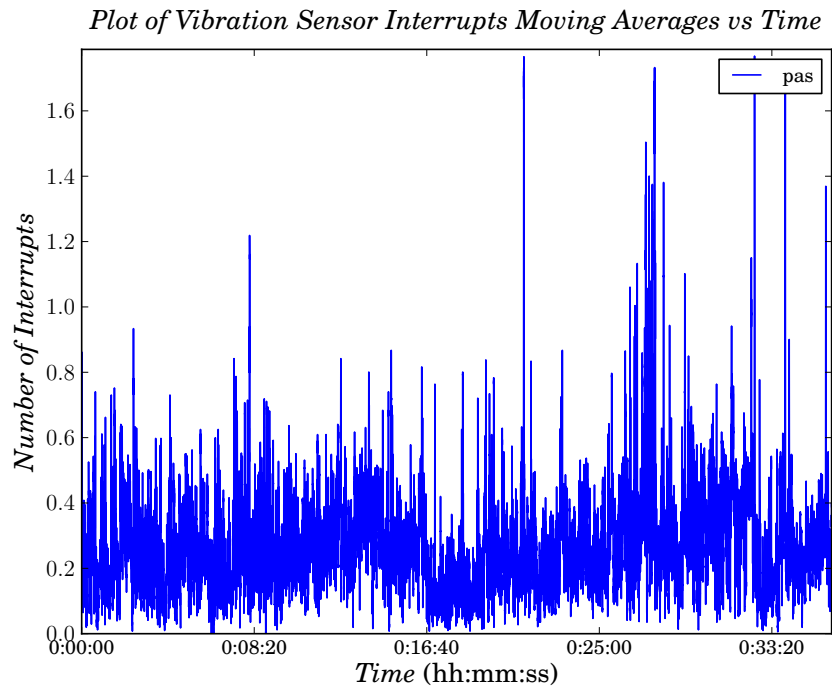


Figure 8.18: Moving average of vibration sensor interrupts: Sheep walking.

An investigation was done to evaluate how different values of the *Walking Threshold*, affects the classification accuracy. For the investigation the *Running Threshold* was kept constant at 0.55, while the *Walking Threshold* was increased from 0 to 0.15. The classification accuracies for each class can be seen in Figure 8.20. From the figure it is clear that

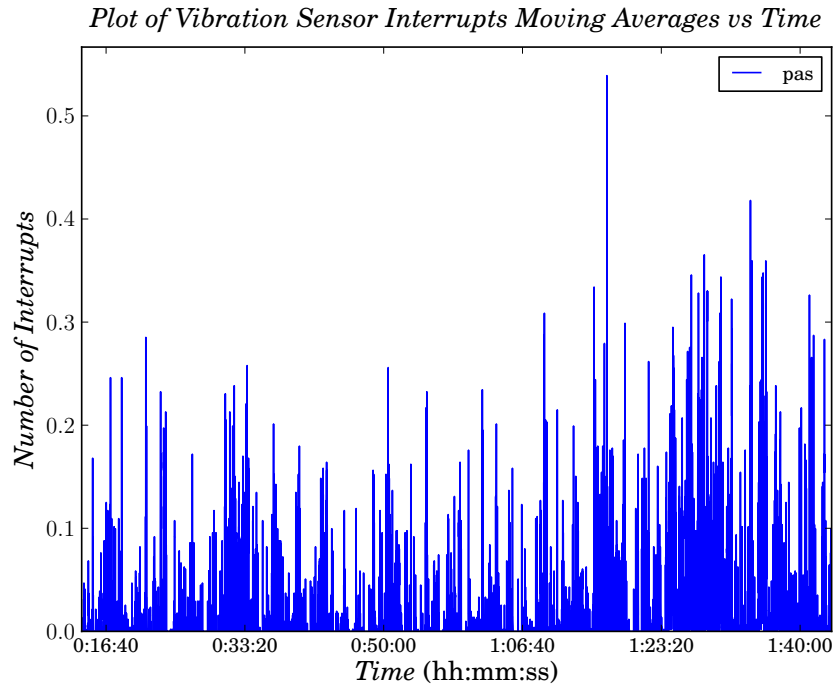


Figure 8.19: Moving average of vibration sensor interrupts: Sheep laying down.

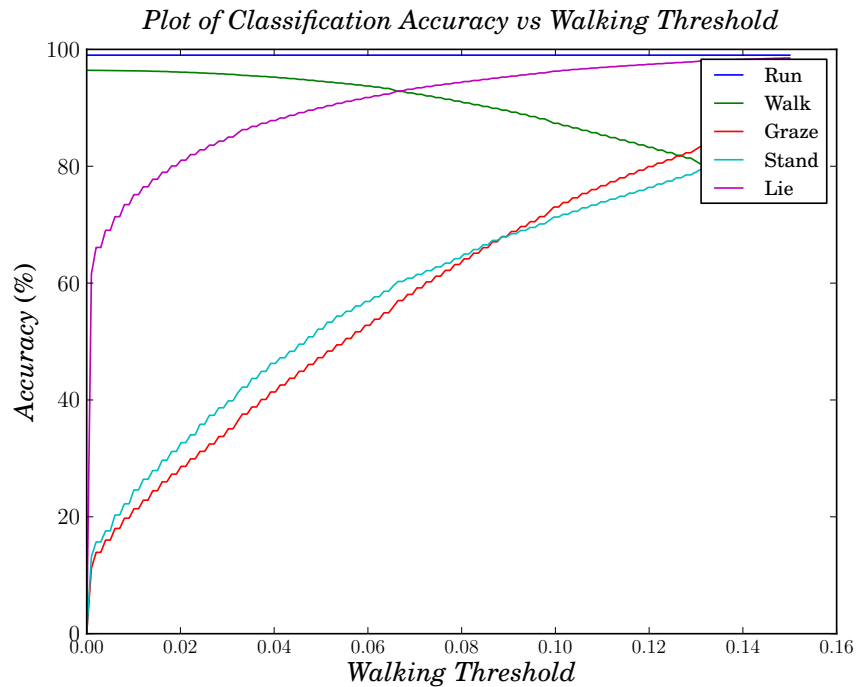


Figure 8.20: Classification accuracy vs walking threshold for sheep behaviour.

the *Walking Threshold* does not affect the classification accuracy of running. It is also clear, that increasing the threshold, results in a decrease in the ability to classify walking behaviour and an increase in passive behaviour. It is thus necessary to determine the importance of each class, for a specific application and to set the thresholds accordingly.

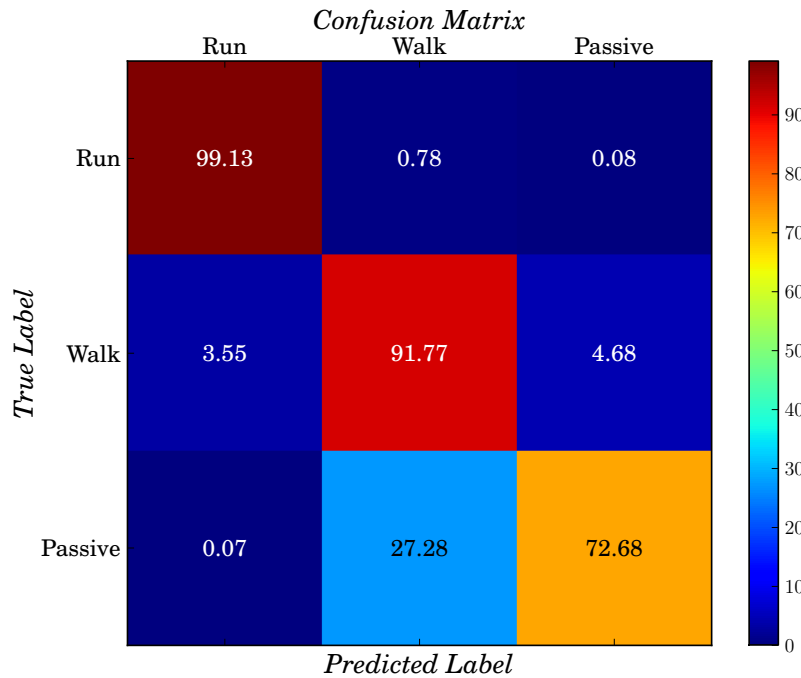


Figure 8.21: Confusion matrix of the vibration sensor classifier: Sheep behaviour.

To calculate the accuracy of using a vibration sensor to classify the behaviour of sheep, the *Running Threshold* was set to 0.55 and the *Walking Threshold* set to 0.075. The following decision tree was implemented, by means of nested IF-statements:

```

if (value >= Running Threshold)
{
  Classification = Running
}
elseif ((value >= Walking Threshold)
        and (value < Running Threshold))
{
  Classification = Walking
}
else
{
  Classification = Passive
}

```

The classifier could accurately distinguish between running, walking and passive behaviour, with a precision of 99.13%, 91.77% and 72.68%, respectively. The confusion matrix, depicted in Figure 8.21, illustrates how the inaccuracies were misclassified.

## 8.5.2 Behaviour Classification of Rhinos using a Vibration Sensor

A similar investigation was done to determine if the behaviour of rhinos can be classified by utilising a vibration sensor. The moving averages were calculated for each of the 4 classes of behaviour. The data was analysed once more to get an idea of the values, for the different classes. The moving averages for running, walking, standing and laying down, are depicted in Figures 8.22 to 8.25, respectively. The moving average of running behaviour had a mean value of 2.84536, with variance of 0.09833 and standard deviation of 0.31357. From Figure 8.22, it is clear that if a *Running Threshold* was chosen between 0 and 2.2, all of the values above the *Running Threshold* could be classified as running. However, due to the similarities of walking behaviour, which had a mean value of 2.50207, with variance of 0.43434 and standard deviation of 0.65905, a large percentage of confusion was expected between the two classes. Figure 8.23 shows, that if a *Walking Threshold* was chosen between 0 and 1.5, most of the values above the *Walking Threshold* and below the *Running Threshold*, could be classified as walking but, confusions are however expected. By analysing Figures 8.24 and 8.25, it can be said that any values below the *Walking Threshold*, can be classified as *passive behaviour*, which in this case includes: standing and laying down.

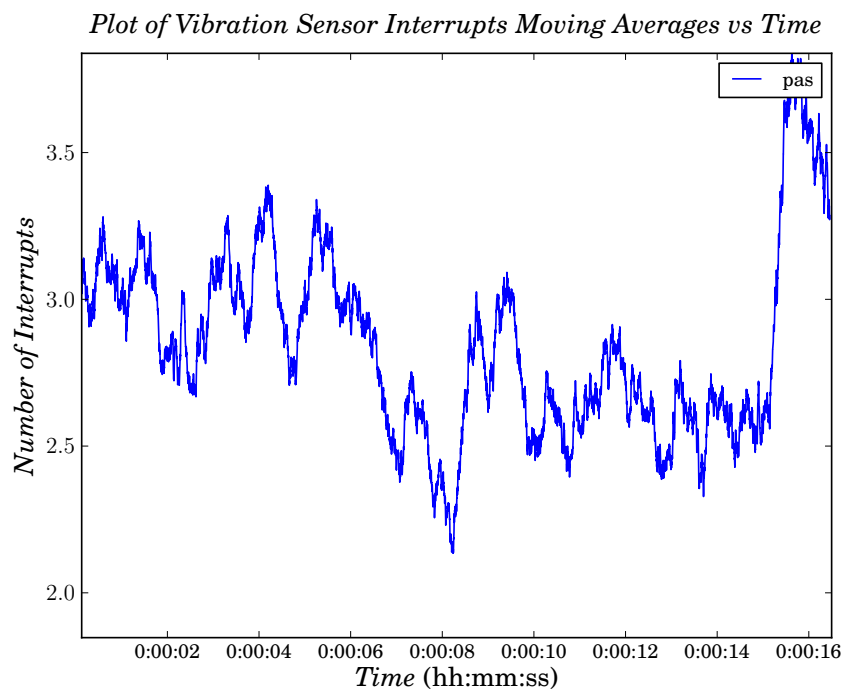


Figure 8.22: Moving average of vibration sensor interrupts: Rhino running.

An investigation was done to evaluate how different values of the *Running Threshold*, affects the classification accuracy. For the investigation the *Walking Threshold* was kept constant at 0.7, while the *Running Threshold* was increased from 0 to 3.5. The classification accuracies for each class can be seen in Figure 8.26. From the figure it is clear that the *Running Threshold* does not affect the classification accuracy of standing or laying

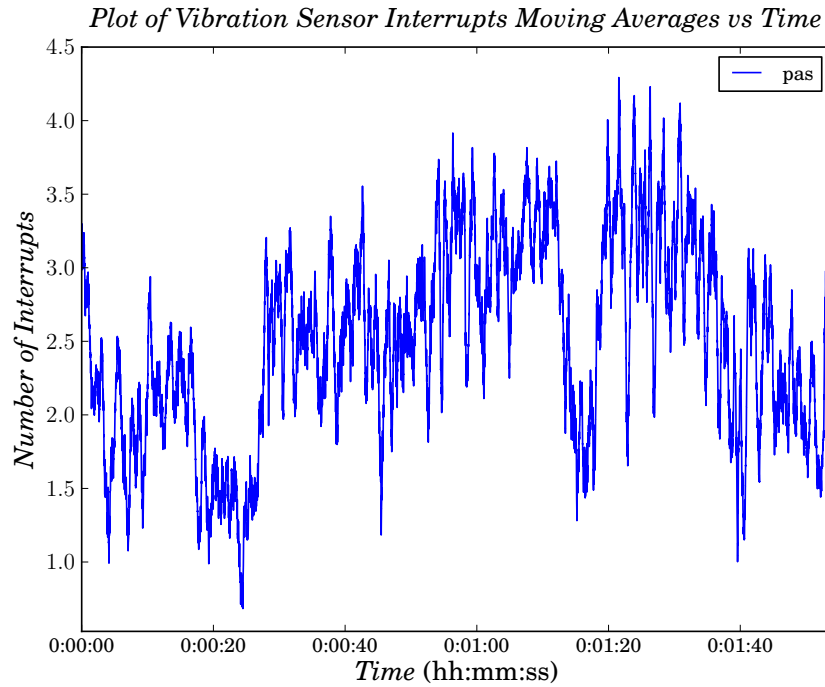


Figure 8.23: Moving average of vibration sensor interrupts: Rhino walking.

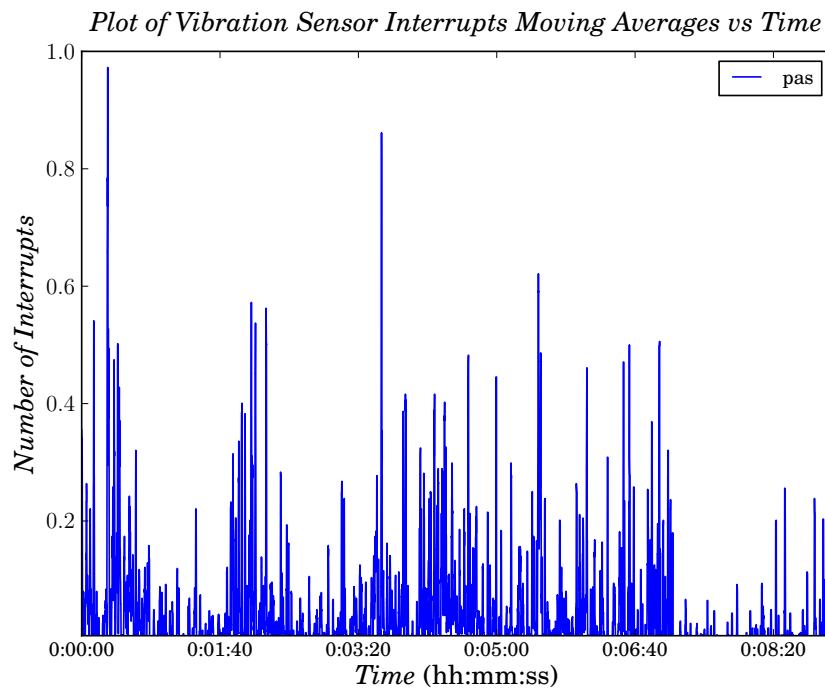


Figure 8.24: Moving average of vibration sensor interrupts: Rhino standing.

down. However increasing the *Running Threshold*, results in a decrease in the ability to classify running behaviour and an increase in walking behaviour. It is thus necessary to determine the importance of each class, for a specific application, and to set the thresholds accordingly.

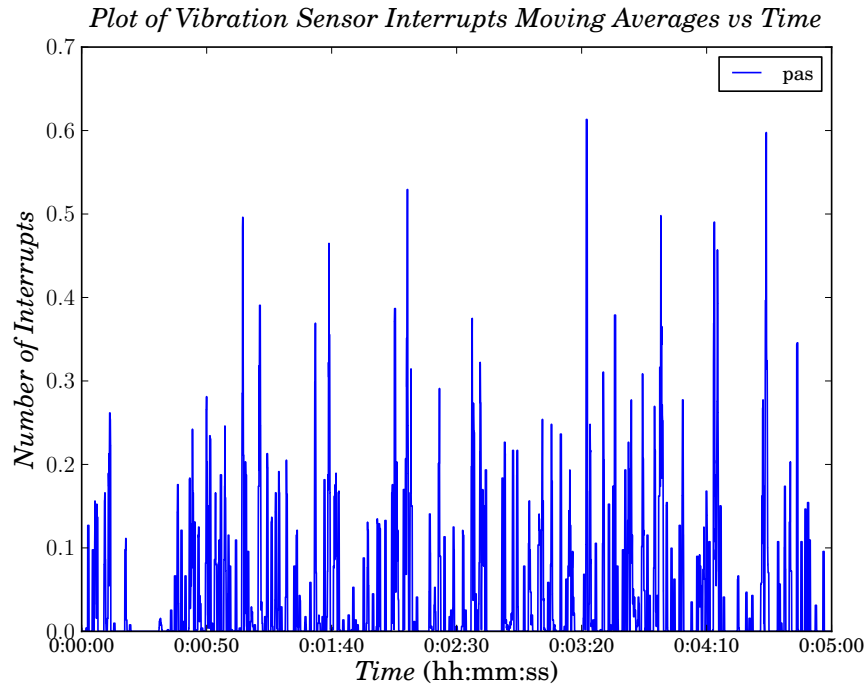


Figure 8.25: Moving average of vibration sensor interrupts: Rhino laying down.

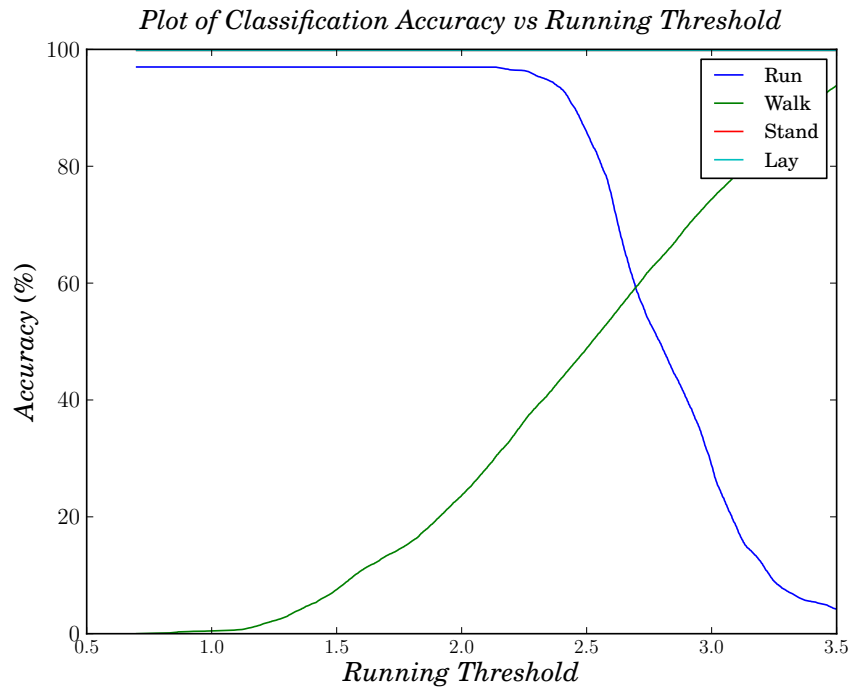


Figure 8.26: Classification accuracy vs running threshold for rhino behaviour.

To calculate the accuracy, of using a vibration sensor to classify the behaviour of rhinos, the *Running Threshold* was set to 2.133 and the *Walking Threshold* set to 0.7. The decision tree shown in Section 8.5.1 was implemented, obtaining the following results.

The classifier could accurately distinguish between running and passive behaviour, with a precision of 100 % and 99.96 %, respectively. Walking behaviour could however, only be accurately classified, 30.29 % of the times. The latter is a result of the expected confusion between the running- and walking classes. The confusion matrix, depicted in Figure 8.27, illustrates how the inaccuracies were misclassified. The results can be improved by using longer signal time averages. This will drive the margins between classes further apart. A test was done using 6000 samples for the moving average as opposed to the normal 512 samples. This increased the ability to detect walking with 28.6 %. Figure 8.28, shows the confusion matrix of these results. The number of samples used for the moving average can also be thought of as an approximate time threshold for each class. In the previous examples, a 512 point and 6000 point moving average based on a 40 Hz sampling frequency, means that the average is calculated over 12.8s and 150s, respectively. For the latter of the two, the animal would have to run for approximately 150s, before the moving average will reach the *Running Thershold*.

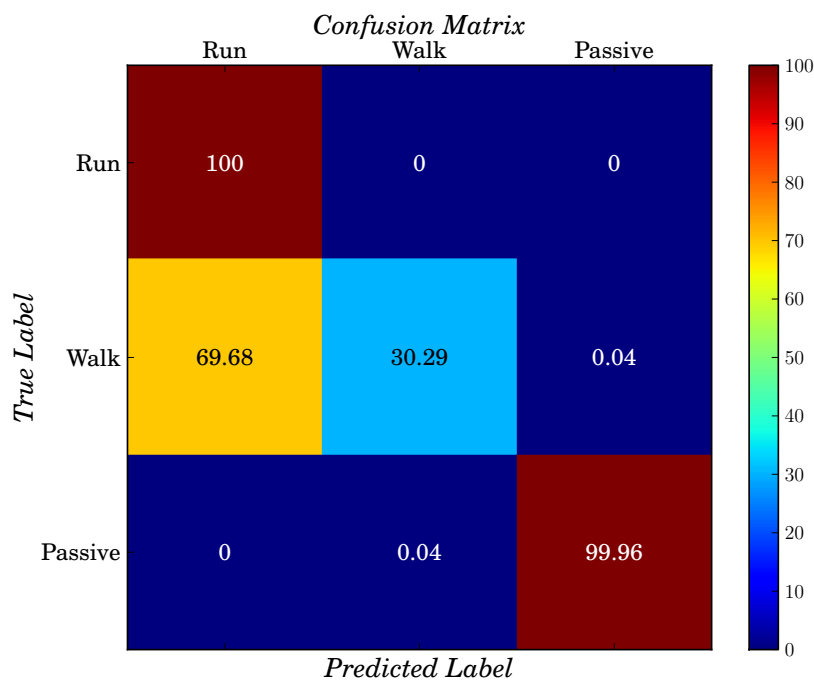


Figure 8.27: Confusion matrix of the vibration sensor classifier: Rhino behaviour.

This classification technique is strongly advised as a resource management and power saving tool. For example, in the case where an OABCS is used to detect abnormal behaviour of rhinos, for anti-poaching efforts, the abnormality would probably include running behaviour. The vibration sensor classifier can be used as a trigger for the more advanced classification techniques. This will allow the system to keep all power intensive components (including the microcontroller) in sleep mode, while a simple interrupt handler can determine the behaviour of the animal. If running behaviour is suspected, the microcontroller and accelerometer can be requested to perform more advanced classification techniques, after which decisions can be made accordingly. Maximizing the sleep

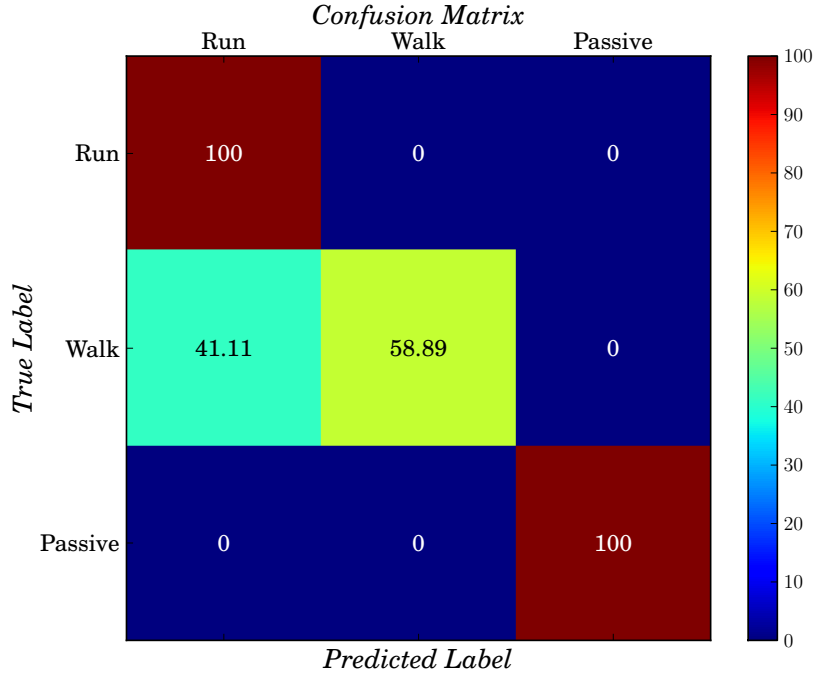


Figure 8.28: Confusion matrix of the vibration sensor classifier, using a 6000 point moving average: Rhino behaviour.

duration for various components, will accordingly extend the battery life of the device.

This is an important advantage and can be proved by means of a simple calculation. If one assumes that the OABCS implemented on the sheep (as explained in the previous section) had an overall accuracy of 100%, the classified data shows that the sheep respectively spent 1.89 %, 12.26 % and 85.85 % of their time running, walking and being passive. If the aim is only to detect running behaviour the system will be active for 1.89 % (true running) plus 8.54 % (walking classified as running), of the time. The remainder of the time the system is kept in sleep mode. The theoretical power consumption of the OABCS in sleep mode is  $\pm 18.8 \mu\text{A}$ . If this value is increased to  $250 \mu\text{A}$ , to allow for system imperfections, the system will consume  $250 \mu\text{A}$ , 89.57 % of the time and  $3.17 \text{ mA}$  for remaining 10.43 %. The system will then on average consume  $554.56 \mu\text{A}$ . The battery lifetime can approximately be calculated as:

$$\text{Battery life} = \left( \frac{\text{Battery capacity (mAh)}}{24\text{h} * \text{Energy consumption (mA)}} \right) \quad (8.5.1)$$

$$= \left( \frac{3600}{24 * 0.55456} \right) \quad (8.5.2)$$

$$= 270.48 \text{ d} \quad (8.5.3)$$

Taking the abovementioned results into consideration, it is apparent that a vibration sensor can be used to accurately classify at least some behavioural classes of both sheep and rhinos. Although this technique is not specifically advised as a primary classification method, it can be used to improve the performance of the OABCS and similar systems that will follow. It is strongly advised to use this technique as an application specific tool, especially to extend the battery lifetime for low power applications.



## 8.6 Pulse Rate Detection

An investigation was also done to evaluate two different techniques of pulse rate detection, which could be applied in future studies on rhinos. A pulse oximeter was used in the first investigation. Texas Instruments, manufactures an AFE4490SPO2EVM pulse oximeter demonstration kit [53]. The demonstration kit comes with fully integrated hardware and configuration software, that makes it easy to measure and analyse pulse rates. Using the demonstration kit was a good choice. It saved time and money, by avoiding the hardware- and software design processes. Although this technology was tried out in the investigation, it was anticipated that using a pulse oximeter as a means of pulse rate detection of rhinos, will not work, due to the practical implementation thereof. The basic components of a pulse oximeter consists of a red LED, infrared LED, photodetector, analog to digital converter and a processing unit. Normally a human finger is placed between the LEDs and the photodetector, as illustrated in Figure 8.29. The LEDs and photodetector are shielded from external light to reduce interference. In simple terms, a pulse oximeter calculates a persons heart rate by means of the differences in light intensities, of both red- and infrared light, received at the photodetector.



Figure 8.29: Finger placement for pulse oximeter measurements.

For humans, this process is normally easy, as it is very controlled in terms of accurate finger placement, very little to no hair on the finger tips and avoiding movements while measurements are in progress. Rhinos, however, are not so accommodating, therefore the following difficulties were anticipated: A very limited number of places exist where a small pulse oximeter can permanently be attached to a rhino. The ear seems to be a good location, but attaching the sensor in a sensible way would prove to be difficult. This is due to the fact that the photodetector needs to stay in alignment with the LEDs, to ensure good measurements. Furthermore, the motion of the ear would greatly affect the chance of accurate pulse rate detection. Finally, the ability to keep the photodetector and LEDs clean would be very difficult, as rhinos often take mud baths and are exposed to large amounts of dust. Despite these concerns, the technology was still tested to evaluate its performance.

The initial tests were performed on humans. The pulse rate, of a 25 year old man, was accurately obtained as 66 beats/min. Further tests involved placing the pulse oximeter on a lamb's ear, as illustrated in Figure 8.30. After decreasing the light intensities, for both the red- and infrared LEDs, to avoid saturation of the photodetector, the lamb's pulse rate was easily detected. Figure 8.31, contains a segment of the measured pulse rate of the lamb. Various tests were performed on young semi-tame rhinos, placing the



Figure 8.30: Pulse oximeter on the ear of a lamb.

sensor on their ears and tails, but no continuous pulse rate could be detected. This was mainly because the tests did not involve sedating the animals. The rhinos kept on moving their ears during the measurements. However, the couple of times they kept still, did not produce any good results.

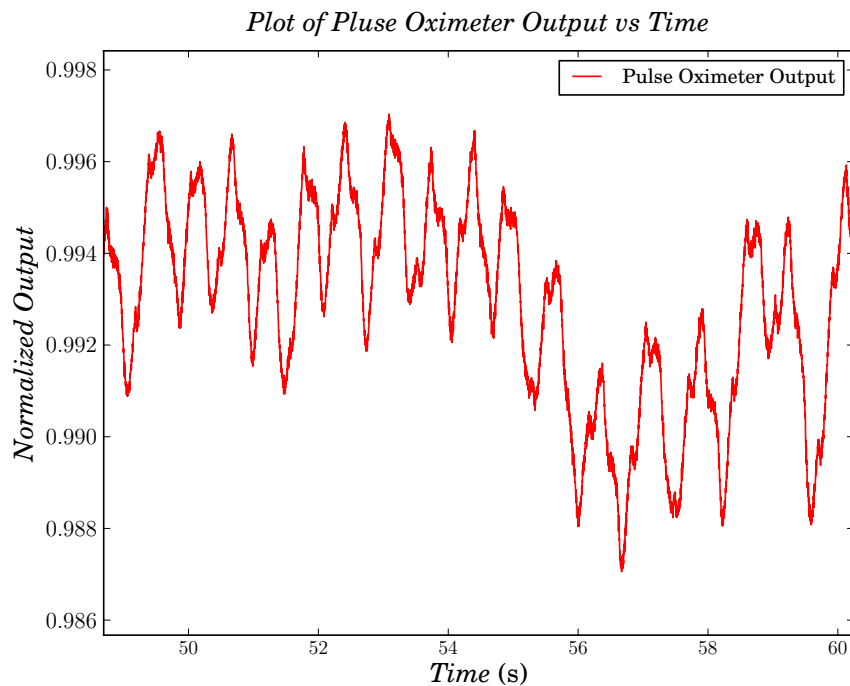


Figure 8.31: Pulse oximeter output as measured on the ear of a lamb.

The second investigation utilised a contact microphone as measuring device. The idea was to use something similar to stethoscopes as used by medical professionals, to examine internal body sounds of humans. The CM-01B analog contact microphone, seemed to be a good choice to detect internal body sounds of rhinos. The sensor has a cylindrical shape with a diameter of 18 mm and height of 11 mm. The sensor minimizes external acoustic noise while offering extremely high sensitivity to vibrations applied to the central rubber pad. The CM-01B is ideal for detecting body sounds. The contact microphone was connected to a laptop by means of a 3.5 mm audio jack and powered from the USB ports or

from an external 9 V battery. The sensor consumed 100  $\mu$ A whilst active. Professional audio recording software was used to sample the output of the sensor at a rate of 1 kHz. While recording, the output of the sensor could be heard using earphones. This ensured that good recordings were made, as the sensor could be placed in optimal positions. The best results were obtained by pressing the sensor against the skin, in areas close to the heart. The first measurements were taken from a 4 month old rhino. The rhino was distracted with food, while the measurements were taken. The baby rhino had a heart rate of 60 beats/min, which although higher than the resting heart rate of 39 beats/min, still indicates that the rhino is healthy and not under stress.

Further measurements were done on two fully grown, sedated rhinos during a conservation effort involving the removal of the horns from the rhinos. The rhinos were sedated and very little time was available to perform the tests. Successful recordings were made of both rhinos. Figure 8.32, depicts the RAW data, as obtained from the contact microphone, of a single heart beat. The contact microphone produces high quality signals. Figure 8.33, shows 10s of recorded data. The figure indicates that 26 beats occurred during this time. Although the RAW data can be hard to interpret, a spectrogram may reveal interesting information. The cross-correlation of the 10s of data (as show in 8.33) and an isolated pulse (similar to the one in 8.32) was calculated, to better select the pulse information. The spectrogram of the correlated signal was calculated and is depicted in Figure 8.34. The correlation was optimised for time resolution and the 26 beats can be seen where energy spikes occur. The spectrogram further indicated that the energy of each pulse starts to decline above 80 Hz. The heart rate of 156 beats/min, clearly indicates that the animal was under stress. Although the process was carefully monitored by veterinarians, the rhinos still experience high stress levels, due to the noise caused by chainsaws cutting through their horns. This results prove that the pulse rate of rhinos can directly be used as key stress level indicators to determine their well being.

The use of contact microphones are also subject to the problems of attaching the device to the rhinos. Solutions of how to permanently attach such a device still need to be derived and evaluated. Some veterinarians are of the opinion that the sensor is small enough to be used as an implant. This idea is not so far fetched, but more electronics would have to be combined with the sensor to provide a wireless solution. This will increase the size of the sensing device, reducing the chance that it might be utilised as an implant. This can still be a worth wile avenue of investigation.

The utilisation of both pulse oximeters and contact microphones where evaluated as a means to detect the pulse rate of rhinos. Both the technologies are limited in terms of permanently attaching the device on animals, in a way that will ensure robust functionality. Although some veterinarians successfully used pulse oximeters on sedated rhinos, it is firmly believed, that the abovementioned results are a good indication that it might not be a sustainable solution. Contact microphones proved to be much easier to work with and produced high quality signals. These signals can be studied by means of DSP techniques in future work, to gain more insights. The spectrogram revealed interesting information regarding the frequency content of the signal. The latter can be used together with analog- or digital filters and amplifiers, to further suppress external noise and enhance signal quality.

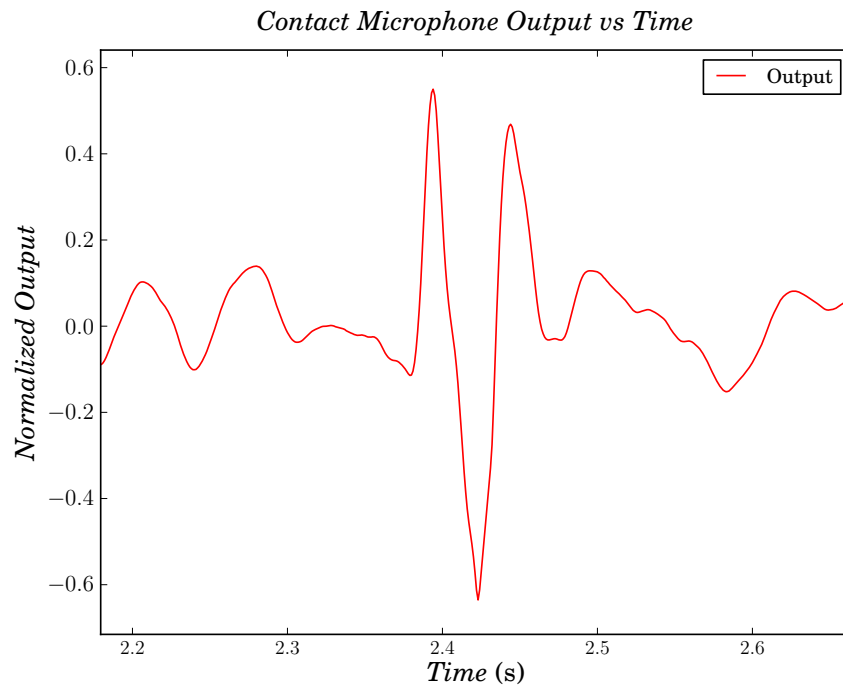


Figure 8.32: Single heart beat of a rhino.

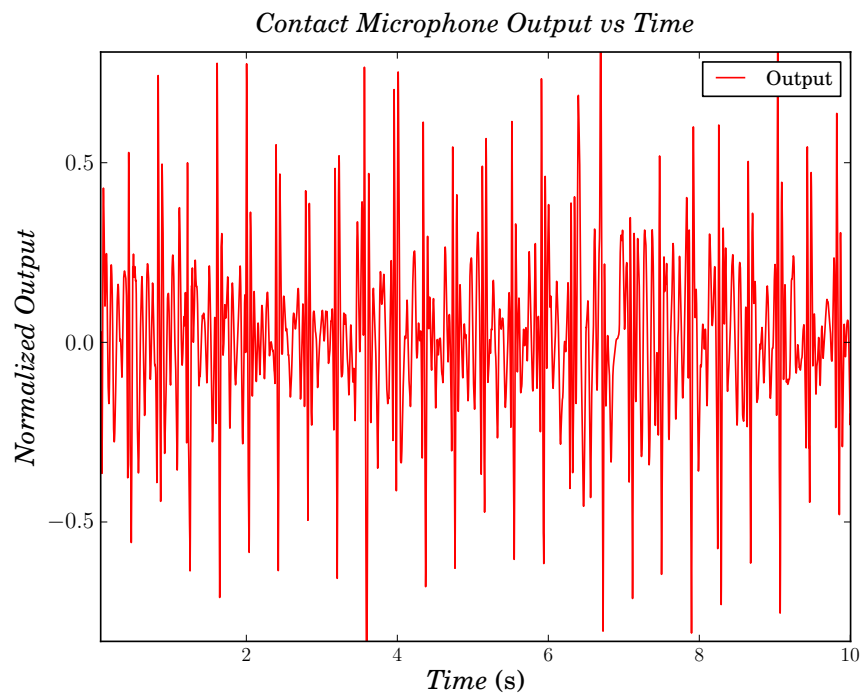


Figure 8.33: 26 Heart beats of a rhino.

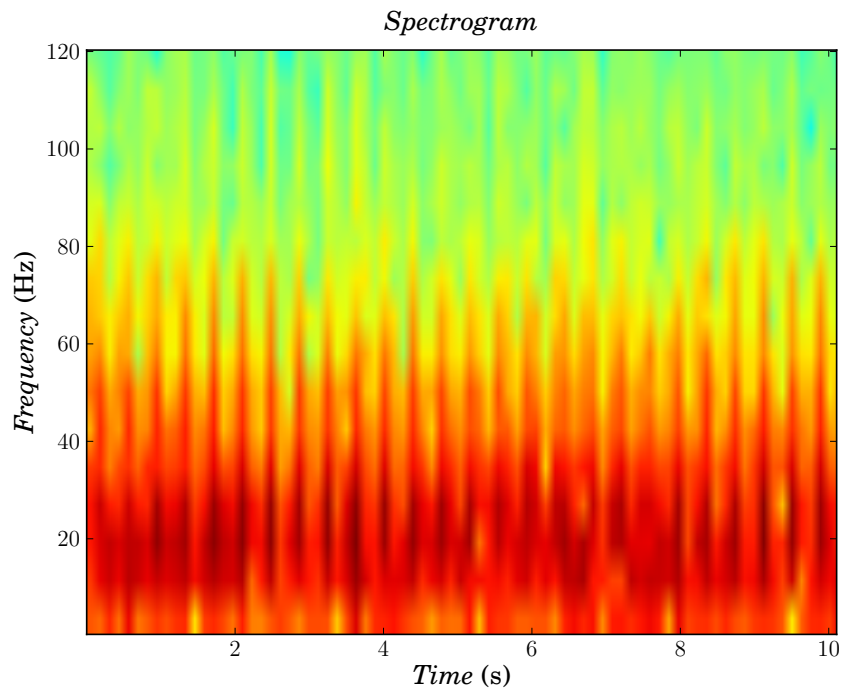


Figure 8.34: Spectrogram of a rhino's pulse rate at 156 beats/min.

This chapter thoroughly presented the measurements and results obtained, from the various tests performed, during this project. WildMotes were successfully used to collect data, that was used in the automatic behaviour classification of both sheep and rhinos. The OABCS accurately classified the behaviour of sheep in real time. The use of vibration sensor classifiers, proved to be an effective technique, to extend the battery life of the WildMotes. Finally, contact microphones were used to record the pulse rates of rhinos and sheep. The next chapter makes some final remarks and recommendations, where after the thesis is concluded.

# Chapter 9

## Conclusion

### 9.1 Project Motivation

As stated in the beginning, wildlife trafficking is one of the main forces driving many wildlife species towards extinction, of which rhinos are a prime and unfortunate example. Despite all the efforts, rhino poachers still have the upper-hand and the rhino population is very rapidly decreasing. In the fight against poaching, a variety of techniques are being implemented, some more effective and drastic than others, such as “*shoot to kill*”. Research also indicates that there are very few effective technologically based anti-poaching methods. New technologies would greatly assist scientists, biologists and nature conservationists with the provision of key information regarding the real time behaviour, location and well being of animals. This information can be an aid in anti-poaching operations. Further research, to the best of our knowledge, revealed that no current technology exists, that can effectively provide these kinds of insights. Furthermore, it was recognized that the design and implementation of such a system would not be trivial and subject to many difficulties. However, this project had the main objective to develop a first prototype for such a biological sensing platform.

### 9.2 Work Done Under the Project

A prototype Animal Borne Behaviour Monitoring System (ABBMS) that could serve as a technological data acquisition platform for scientists and engineers, was developed. The ABBMS is intended for scientific data collection and further development of advanced anti-poaching systems, particularly aimed at endangered wildlife species. Central to the ABBMS was the design and testing of animal borne sensing devices, known as WildMotes. In total five WildMotes were developed. The most important hardware components included among others: an ultra-low power microcontroller, GPS, tri-axial accelerometer, nano-power vibration sensor, microSD card and FRAM modules. The hardware design was based on extensive prior investigations and it is felt that the present design is fairly optimal, in terms of present component performance.

Furthermore, the WildMotes had wireless connectivity capabilities, through which all WildMotes were connected to a wireless sensor network (WSN). The WSN provided the WildMotes with the means to forward on-animal sensed data to a central data collection point via an on-board RF link in real time. This included GPS coordinates, which were utilised to reveal repetitive movement patterns by means of heat maps. Two multi-hop

network strategies were implemented, ie. RIME and RPL. The latter should expand the conventional single transmitter animal tracker propagation footprint, which is a general and known constraint with this type of application.

Various software routines were developed to configure and test the WildMotes. The routines served to enable specific implementations and experiments performed throughout the extent of the project. Initially the WildMotes were fixed to a number of sheep, using collars. The WildMotes constantly logged acceleration- and vibration sensor data, with the behaviour of the sheep simultaneously documented by human observations. The data and observations collected during the field tests were used by research team members to develop a set of classification algorithms. These were subsequently implemented in the Wildmotes as part of this project, to successfully classify the behaviour of the sheep as either running, walking, standing, laying down or grazing, to a very good level of precision, in real time. If the limited resources of the on-board hardware is taken into account, the techniques developed to do this, should prove very useful in future work. Battery life is a key issue in all forms of animal monitoring and tracking. It was, therefore, investigated how this could possibly be extended by innovative application of a simple movement sensor, as opposed to a more power hungry accelerometer. Classification accuracy could acceptably be retained, while battery life was extended significantly.

Abnormal stress level indication is required as an early warning signal to anti-poaching coordinators. To this effect some initial pulse rate measurements were carried out on immobilised rhinos. Two techniques were tried, of which one, using contact microphones, appears to be promising for future development.

In conclusion, the prototype ABBMS was successfully implemented, yielding good results. These should be eminently applicable in the development of the next generation of on- and off-animal behavioural sensing network. The information so obtained, should hopefully be useful to technology inputs for advanced anti-poaching strategies. It should, however, almost certainly be valuable to scientists and biologists in the particular field of research, providing exact real time animal behavioural- and localisation information over extended periods of time.

### 9.3 Project Outcomes and Contributions

The work resulted in the following specific outcomes and contributions:

1. The design, implementation and testing on both sheep and rhinos, of on-animal Wildmote sensing devices. These devices present a state of the art hardware design, as could be satisfactorily confirmed during a recent visit to the network group at Trento University, in Italy. The units and prototype packaging have also proved suitably robust during field trials.
2. The Wildmotes were actively tested in field trials, where acceleration- and vibration sensor data were sampled and continuously logged for both sheep and rhino. In the case of rhinos, the ABBMS' networking capabilities were utilised to transmit GPS coordinates to a base station, every 3 min.

3. Two different multi-hop networking protocols were implemented to prove that it can provide the WildMotes with an effective way to route data to the base station and at the same time increase the network coverage of the system.
4. The collected data was successfully used in different implementations of animal movement/behavioural classifiers. For rhinos, the implementation of decision tree classifiers resulted in an overall classification accuracy of 95.81 %, with an accuracy of 99.49 %, 97.18 % and 90.77 % for standing, walking and laying down respectively. Video recordings were made to verify classifier performance.
5. The definition of the classifier algorithms was part of a research team effort, but a challenge was presented in this project to implement these techniques in a resource constrained low power microcontroller environment. The success thereof depended on careful processing and memory management, data set reduction and classification system state management.
6. The live transmission of on-animal classified behaviour has not been done before, to the best of our information and could make a significant contribution to research and real time abnormal behaviour recognition. The ABBMS successfully implemented an on-animal behaviour classification system. In the case of sheep it could accurately distinguish between running, walking and passive behaviour, with respective precisions of 76.92 %, 80.15 % and 90.03 %. It provided live updates of animal behaviour, every 5.3s.
7. An alternative and simplified route to effective classification and particularly battery life extension, was also explored. It was proved that data collected from a nano-power tilt and vibration sensor, could be utilised to this effect. A single decision tree was implemented using moving averages from the simple sensor and overall classification accuracy was acceptably close to the results from the full accelerometer approach. The battery life was extended from an estimated 47 to 270 days. This is important in an application type where battery life is a prime concern.
8. An exploratory audio based recording of rhino pulse rate was made in order to correlate this with other behavioural information, as a step in the process to provide the necessary stress related information to biologists.

## 9.4 Future Work and Recommendations

In summary a few recommendations could be presented:

1. The size of the WildMotes and the collars should be reduced. This can be done by removing all the unused sensors and external access headers from the system. Components can also be placed on both sides of the PCB.
2. It could also be considered to add a gyroscope to the system to increase the classification accuracy. This will have to be offset against power consumption.
3. It is recommended to keep on collecting more ground truth data of rhinos and other endangered species, to increase the classification accuracy over a wider range of species.



4. The use of the vibration sensor classification technique is strongly recommended along with an ultra-low power mode for the WildMotes. This will allow the ABBMS to accurately identify and investigate possible alarm conditions, while maintaining a very low power state in all other cases.
5. It is further advised to investigate more effective methods of pulse rate and stress level detection.
6. Further research is required to develop a robust network and protocol. This will ensure data delivery in extreme environments. Such research is in fact, being planned for the immediate future.

# List of References

- [1] Inria, “Self-Organizing Future Ubiquitous Networks (FUN),” accessed on 30 Sep 2015. [Online]. Available: <https://team.inria.fr/fun/research/>
- [2] US Department of State, “National Strategy For Wildlife Combating Wildlife Trafficking,” pp. 0–13, 2014. [Online]. Available: <https://www.whitehouse.gov/sites/default/files/docs/nationalstrategywildlifetrafficking.pdf>
- [3] South African Department of Environmental Affairs, “Rhino Issue Management,” pp. 1–46, 2013. [Online]. Available: [http://www.rhinorage.org/Downloads/rhinoissue\\_managementreport%20\(1\).pdf](http://www.rhinorage.org/Downloads/rhinoissue_managementreport%20(1).pdf)
- [4] International Rhino Coalition, “Assessing the risks of rhino horn trade,” *A journal of arguments presented at the April 2014 conference in South Africa*, pp. 1–86, 2014.
- [5] P. Cheteni, “An analysis of antipoaching techniques in africa: a case of rhino poaching,” *Environmental Economics*, vol. 5, no. 3, pp. 1–25, February 2014. [Online]. Available: <http://mpra.ub.uni-muenchen.de/59031/>
- [6] J. Rademeyer, *Killing for profit: exposing the illegal rhino horn trade*. Cape Town: Zebra Press, 2012.
- [7] C. K. Harper *et al.*, “Extraction of nuclear dna from rhinoceros horn and characterization of dna profiling systems for white (*ceratotherium simum*) and black (*diceros bicornis*) rhinoceros,” *Forensic Science International: Genetics*, vol. 7, no. 4, pp. 428 – 433, July 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1872497313001038>
- [8] Lindberghfoundation.org, “Lindbergh Foundation,” accessed on 12 Aug 2015. [Online]. Available: <http://lindberghfoundation.org/>
- [9] Habib *et al.*, “Three decades of wildlife radio telemetry in india: a review.” *Animal Biotelemetry*, vol. 2, no. 4, 2014.
- [10] G. Bishop-Hurley *et al.*, “An investigation of cow feeding behavior using motion sensors,” in *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, May 2014, pp. 1285–1290.
- [11] G. Santha and G. Hermann, “Accelerometer based activity monitoring system for behavioural analysis of free-roaming animals,” in *Intelligent Systems and Informatics (SISY), 2013 IEEE 11th International Symposium on*, Sept 2013, pp. 199–203.
- [12] G. Santha *et al.*, “Miniature wireless sensor design for behavioral analysis on free-roaming laboratory animals,” pp. 239–241, Sept 2012.

- [13] A. Mason and J. Sneddon, "Automated monitoring of foraging behaviour in free ranging sheep grazing a biodiverse pasture," in *Sensing Technology (ICST), 2013 Seventh International Conference on*, Dec 2013, pp. 46–51.
- [14] E. Shepard *et al.*, "Identification of animal movement patterns using tri-axial accelerometry," *Endangered Species Research*, vol. 10, no. 1, pp. 47–60, March 2008. [Online]. Available: <http://www.int-res.com/abstracts/esr/v10/p47-60/>
- [15] D. W. McClune *et al.*, "Tri-axial accelerometers quantify behaviour in the Eurasian badger (*Meles meles*): towards an automated interpretation of field data," *Animal Biotelemetry*, vol. 2, no. 1, p. 5, 2014. [Online]. Available: <http://www.animalbiotelemetry.com/content/2/1/5>
- [16] J. Tanha *et al.*, "Multiclass semi-supervised learning for animal behavior recognition from accelerometer data," in *Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on*, Nov 2012, pp. 690–697.
- [17] J. A. Vazquez Diosdado *et al.*, "Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system," *Animal Biotelemetry*, vol. 3, no. 5, June 2015. [Online]. Available: <http://www.animalbiotelemetry.com/content/3/1/15>
- [18] Rhinoresourcecenter.com, "RRC: White Rhino," accessed on 20 Sep 2015. [Online]. Available: <http://www.rhinoresourcecenter.com/species/white-rhino/>
- [19] D. B. Allbrook *et al.*, "Temperature regulation in the white rhinoceros," *Journal of Physiology*, vol. 143, pp. 51,52, July 1958. [Online]. Available: [http://www.rhinoresourcecenter.com/pdf\\_files/124/1240048153.pdf](http://www.rhinoresourcecenter.com/pdf_files/124/1240048153.pdf)
- [20] W. Boomsma and M. van der Sijde, "Concept husbandry guidelines for the white rhinoceros (*ceratotherium simum*)," October 2010. [Online]. Available: <http://www.samhao.nl/webopac/MetaDataEditDownload.csp?file=2:123216:1>
- [21] R. B. Mark Pilgrim, "Eaza best practice guidelines black rhinoceros (*diceros bicornis*)," December 2013. [Online]. Available: <http://www.eaza.net/assets/Uploads/CCC/2014-Black-Rhino-EAZA-Best-Practice-Guidelines-Approved.pdf>
- [22] *MSP430FR573x Mixed-Signal Microcontrollers*, Texas Instruments Datasheet SLAS639J, July 2011, revised June 2014. [Online]. Available: <http://www.ti.com/lit/ds/symlink/msp430fr5739.pdf>
- [23] [www.mouser.com](http://www.mouser.com), "microSD Card Connectors," accessed on 30 Sep 2015. [Online]. Available: <http://www.mouser.com/ds/2/185/e60900232-38395.pdf>
- [24] *FM25V20 2Mb Serial 3V F-RAM Memory*, Ramtron Datasheet, Rev. 3.0, August 2012. [Online]. Available: [http://www.impedanca.si/media/FM25V20\\_ds.pdf](http://www.impedanca.si/media/FM25V20_ds.pdf)
- [25] *GPS receiver GNS 602*, Global Navigation Systems Datasheet, Rev. 1.0, April 2013, datasheet.
- [26] *Digital Accelerometer ADXL345*, Analog Devices Datasheet, 2010, rev. E. [Online]. Available: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>

- [27] *SQ-SEN-200 Nano-Power Tilt and Vibration Sensor*, Signal Quest Datasheet, January 2014. [Online]. Available: <http://signalquest.com/download/Tilt%20and%20Vibration%20Sensor%20SQ-SEN-200.pdf>
- [28] *SQ-SEN-200 Application Circuits*, Signal Quest Application Note, January 2014. [Online]. Available: [http://signalquest.com/download/Simple%20Interface%20Circuits%20\(SQ-SEN-200,%20SQ-MIN-200\).pdf](http://signalquest.com/download/Simple%20Interface%20Circuits%20(SQ-SEN-200,%20SQ-MIN-200).pdf)
- [29] *MiniSense 100 Vibration Sensor*, Measurement Specialties Inc Datasheet, Rev. 1, March 2008. [Online]. Available: [www.meas-spec.com](http://www.meas-spec.com)
- [30] *Ultralow Noise Microphone with Bottom Port and Analog Output*, InvenSense Datasheet, Rev. 1.0, November 2013. [Online]. Available: [www.invensense.com](http://www.invensense.com)
- [31] *Pressure Sensors ABS Series*, Epcos Datasheet, Rev. 2, Augustus 2009. [Online]. Available: [http://en.tdk.eu/inf/57/ds/asb\\_1200\\_v1\\_vr.pdf](http://en.tdk.eu/inf/57/ds/asb_1200_v1_vr.pdf)
- [32] *CM-01B Contact Microphone*, Measurement Specialties Inc Datasheet, Rev. 1.0, January 2011. [Online]. Available: [http://www.meas-spec.com/downloads/Contact\\_Microphone.pdf](http://www.meas-spec.com/downloads/Contact_Microphone.pdf)
- [33] S. P. le Roux, *Stock Position Tracking and Theft Prevention System*, Stellenbosch University Thesis, Rev. 1.0, December 2013. [Online]. Available: [https://dl.dropboxusercontent.com/u/98050418/Stock\\_Position\\_Tracking\\_Theft\\_Prevention\\_System.pdf](https://dl.dropboxusercontent.com/u/98050418/Stock_Position_Tracking_Theft_Prevention_System.pdf)
- [34] S. P. le Roux and R. Wolhuter, *Stock Position Tracking and Theft Detection System*, in Proceedings of Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Conference Paper ISBN: 978-0-620-61965-3, August 2014. [Online]. Available: [http://www.satnac.org.za/proceedings/2014/SATNAC%202014%20Conference%20Proceedings\\_USB\\_edition.pdf](http://www.satnac.org.za/proceedings/2014/SATNAC%202014%20Conference%20Proceedings_USB_edition.pdf)
- [35] Godara, *Handbook of Antennas in Wireless Communications*, L. C. Godara, Ed. CRC Press LLC, 2002.
- [36] *Low-Power Sub-1 GHz RF Transceiver*, Texas Instruments Datasheet, 2013. [Online]. Available: [www.ti.com/lit/ds/symlink/cc1101.pdf](http://www.ti.com/lit/ds/symlink/cc1101.pdf)
- [37] *L6932H1.2 High Performance 2A ULDO Linear Regulator*, STMicroelectronics Datasheet, Rev. 2.0, May 2007. [Online]. Available: [www.st.com](http://www.st.com)
- [38] *TPS6122x Low Input Voltage, 0.7V Boost Converter With 5.5uA Quiescent Current*, Texas Instruments Datasheet, November 2014. [Online]. Available: <http://www.ti.com/lit/ds/slvs776b/slvs776b.pdf>
- [39] *LTC3105 400mA Step-Up DC/DC Converter with Maximum Power Point Control and 250mV Start-Up*, Linear Technologies Datasheet, Rev. A, 2010. [Online]. Available: <http://cds.linear.com/docs/en/datasheet/3105fa.pdf>
- [40] *MSP-FET430 FLASH Emulation Tool (FET)*, Texas Instruments User's Guide SLAU138L, Rev. 3+, February 2009. [Online]. Available: [http://www.ti.com/ww/cn/uprogram/share/pdfs/MSP-FET430\\_flash\\_emulation\\_tool\\_user\\_guide.pdf](http://www.ti.com/ww/cn/uprogram/share/pdfs/MSP-FET430_flash_emulation_tool_user_guide.pdf)

- [41] [www.mikrocontroller.net](http://www.mikrocontroller.net/attachment/138077/main.c), accessed on 14 Jun 2014. [Online]. Available: <http://www.mikrocontroller.net/attachment/138077/main.c>
- [42] C. Speck, *Library to access a MultiMediaCard functions: init, read, write*, Texas Instruments Source Code 1.1, June 2005, mmc.c. [Online]. Available: [https://github.com/hanzz/qsimkit/blob/master/example\\_apps/mmc/mmc.c](https://github.com/hanzz/qsimkit/blob/master/example_apps/mmc/mmc.c)
- [43] —, *Library to access a MultiMediaCard functions: init, read, write*, Texas Instruments Source Code 1.1, June 2005, mmc.h. [Online]. Available: [https://github.com/hanzz/qsimkit/blob/master/example\\_apps/mmc/mmc.h](https://github.com/hanzz/qsimkit/blob/master/example_apps/mmc/mmc.h)
- [44] T. Tsvetkov, *RPL: IPv6 Routing Protocol for Low Power and Lossy Networks*, Network Architectures and Services, Std. SS2011, July 2011. [Online]. Available: [http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2011-07-1/NET-2011-07-1\\_09.pdf](http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2011-07-1/NET-2011-07-1_09.pdf)
- [45] A. Dunkels, *Poster Abstract: Rime - A Lightweight Layered Communication Stack for Sensor Networks*, The European conference on Wireless Sensor Networks, Swedish Institute of Computer Science Poster, January 2007. [Online]. Available: [http://www.researchgate.net/publication/242380890\\_Poster\\_Abstract\\_Rime\\_\\_A\\_Lightweight\\_Layered\\_Communication\\_Stack\\_for\\_Sensor\\_Networks](http://www.researchgate.net/publication/242380890_Poster_Abstract_Rime__A_Lightweight_Layered_Communication_Stack_for_Sensor_Networks)
- [46] A. Dunkels, F. Osterlind, and Z. He, “An adaptive communication architecture for wireless sensor networks,” in *Proceedings of ACM SenSys 2007*, Sydney, Australia, Sept 2007, pp. 335–349.
- [47] I. Radoi *et al.*, “Evaluation of Routing Protocols for Internet-Enabled Wireless Sensor Networks,” in *2012 8th International Conference on Wireless and Mobile Communications*, Venice, Italy, 2012, pp. 56–61. [Online]. Available: [http://www.thinkmind.org/index.php?view=article&articleid=icwmc\\_2012\\_3\\_30\\_20243](http://www.thinkmind.org/index.php?view=article&articleid=icwmc_2012_3_30_20243)
- [48] N. Mitton and R. Wolhuter, *Energy Harvesting in Wireless Sensor Networks*, ch. 6, pp. 205–220. [Online]. Available: [http://www.worldscientific.com/doi/abs/10.1142/9789814525466\\_0006](http://www.worldscientific.com/doi/abs/10.1142/9789814525466_0006)
- [49] J. M. Gilbert and F. Balouchi, “Comparison of energy harvesting systems for wireless sensor networks,” *International Journal of Automation and Computing*, vol. 5, no. 4, pp. 334–347, October 2008. [Online]. Available: <http://link.springer.com/article/10.1007%2Fs11633-008-0334-2>
- [50] M. K. Stojcev *et al.*, “Power management and energy harvesting techniques for wireless sensor nodes,” in *2009 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services*, 2009, pp. 65–72. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5339410>
- [51] J. B. de Chaisemartin, *Efficient solar energy harvesting for animal tracking systems*, ECE Paris and Stellenbosch University Report, Rev. V3, December 2014. [Online]. Available: <https://dl.dropboxusercontent.com/u/98050418/Report-Jean-Baptiste%20de%20Chaisemartin-Efficient%20solar%20energy%20harvesting%20%20for%20animal%20tracking%20systems%20-.pdf>

- [52] J. Marais, S. P. le Roux, R. Wollhuter, and T. R. Niesler, “Automatic classification of sheep behaviour using 3-axis accelerometer data,” in *Proceedings of the twenty-fifth annual symposium of the Pattern Recognition Association of South Africa (PRASA)*, no. ISBN 978-0-620-62617-0, Cape Town, South Africa, 2014. [Online]. Available: [http://dsp.sun.ac.za/~trn/reports/marais+leroux+wollhuter+niesler\\_prasa14.pdf](http://dsp.sun.ac.za/~trn/reports/marais+leroux+wollhuter+niesler_prasa14.pdf)
- [53] *AFE4400 and AFE4490 Development Guide*, Texas Instruments User’s Guide, Rev. 2, May 2014. [Online]. Available: <http://www.ti.com/lit/ug/slau480c/slau480c.pdf>

# Appendices

# Appendix A

## Measurements and Results Additional Info



Table A.1: Observations made on 03\_07\_2014

Test Name	03_07_2014	Small_Camp
Resolution	4g	
Tag Number	Sheep Type	
Tag 1	Lamb	
Tag 2	Lamb	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
10:02	10:51	Setup
10:51	11:01	Run
11:01	11:06	Stand
11:06	11:14	Stand
11:14	11:50	Unknown (Suspect: Stand)
11:50	12:13	Stand
-	12:03	Tag 2 Stands and turns right way round
12:13	12:14	Walk
12:14	12:15	Run
12:15	12:43	Stand
-	12:27	Tag 5 Lay
12:37	12:41	Tag 4 Lay
12:43	12:52	Grazing
12:52	12:56	Stand
12:56	13:01	Grazing
13:01	13:06	Stand
13:06	13:29	Grazing
13:29	14:22	Stand
13:51	14:11	Tag 5 Lay
14:03	14:06	Tag 4 Lay
14:03	14:11	Tag ? Lay
14:22	15:11	Grazing
15:11	15:18	Stand
15:18	15:24	Stand
15:34	16:23	Grazing
-	16:23	Tag 5,4,1 Lay
16:23	16:33	Collect sheep & remove collars

Table A.2: Observations made on 04\_07\_2014

Test Name	04_07_2014	Small_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Ewe	
Tag 2	Ewe	
Tag 3	Ewe	
Tag 4	Lamb	
Tag 5	Lamb	
Time 1	Time 2	Description
10:13	10:29	Setup
10:29	10:33	Calibrate 6 Axis
10:33	10:50	Setup
10:50	10:51	Run
10:51	11:59	Grazing
11:59	12:01	Run
12:01	12:10	Stand
12:10	12:15	Stand
12:15	12:30	Grazing
	12:20	Tag 4/5 Shake head
	12:27	Tag 1/2/3 Shake head
12:30	12:39	Grazing
12:39	12:52	Stand
12:40	12:45	Tag ? Lay
12:52	12:53	Walk
12:53	12:53	Grazing
12:53	12:54	Tag 5 Walk
13:53	15:27	Lay and Stand
15:27	15:34	Run
15:34	15:38	Stand
15:38	15:40	Stand
15:40	15:43	Grazing
15:43	15:44	Stand
15:44	15:55	Grazing
15:55	16:26	Grazing
16:26	16:30	Walk & Run
16:30	16:35	Collect sheep & remove collars
	16:52	Stop

Table A.3: Observations made on 05\_07\_2014 and 06\_07\_2014

Test Name	05_07_2014	Big_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Lamb	
Tag 2	Lamb	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
10:17	10:40	Setup
10:40	10:43	Move sheep to big camp (run)
10:43	?	Run
06_07_2014		The next day
	10:56	Collect sheep & remove collars
	11:10	Stop

Table A.4: Observations made on 06\_07\_2014, 07\_07\_2014 and 08\_07\_2014

Test Name	06_07_2014	Big_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Lamb	
Tag 2	Lamb	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
16:59	17:34	Setup
17:34	17:36	Move sheep to big camp (run)
07_07_2014		The next day
	17:00	Stand
08_07_2014		The next day
	16:35	Collect sheep & remove collars
	19:43	Stop

Table A.5: Observations made on 10\_07\_2014

Test Name	10_07_2014	Small_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Ewe	
Tag 2	Ewe	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
8:37	9:03	Setup
9:03	9:05	Move sheep to small camp (run)
	16:37	Collect sheep & remove collars
	18:22	Stop

Table A.6: Observations made on 12\_07\_2014 and 13\_07\_2014

Test Name	12_07_2014	Small_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Lamb	
Tag 2	Lamb	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
9:43	10:03	Setup
10:03	10:04	Run
10:04	10:52	Grazing
10:52	11:09	Grazing
11:09	11:15	Stand
11:15	11:17	Lay
11:17	11:51	Lay
11:51	11:58	Stand
11:58	12:42	Lay
12:42	13:00	Lay
13:00	13:38	Lay
13:38	13:48	Unknown (Suspect: Grazing)
13:48	13:54	Grazing
13:54	13:58	Run Anti-Clock Wise
13:58	14:01	Stand
14:01	14:04	Run Clock Wise
14:04	14:05	Stand
14:05	14:07	Grazing
14:07	14:09	Stand
14:09	14:10	Grazing
14:10	14:30	Walk
14:30	14:33	Grazing
14:33	14:40	Stand
14:40	15:30	Grazing
15:30	24:00:00	Unknown
	24:00:00	Lay
13_07_2014		The next day
8:02	8:04	Stand
	8:04	Grazing
	10:14	Stand
	13:39	Grazing
	13:47	Run
	13:50	Collect sheep & remove collars
	14:10	Stop

Table A.7: Observations made on 14\_07\_2014

Test Name	14_07_2014	Big_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Ewe	
Tag 2	Ewe	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
9:14	9:35	Setup
9:35	9:37	Move sheep to big camp (run)
9:37	9:38	Run
	21:48	Fright & Stand
	-	Collect sheep & remove collars
	-	Stop

Table A.8: Observations made on 16\_07\_2014

Test Name	16_07_2014	Big_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Ewe	
Tag 2	Ewe	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
8:09	8:48	Setup
8:48	8:49	Move sheep to big camp (run)
	10:26	Lay
	10:52	Lay
	11:59	Grazing
	12:00	Stand
	12:02	Grazing
	12:08	Stand
	12:09	Walk
	12:10	Grazing
	12:58	Grazing
	13:08	Grazing
	13:14	Grazing
	14:33	Lay
	14:35	Stand
	14:37	Walk
	14:38	Stand
	14:39	Fright, Run, Stand
14:40	14:44	Walk
14:44	14:51	Stand
	14:52	Grazing
	16:39	Stand
	16:44	Run
	16:59	Stop

Table A.9: Observations made on 17\_07\_2014

Test Name	17_07_2014	Small_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Lamb	
Tag 2	Lamb	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
7:54	8:21	Setup
8:21	8:42	Grazing
8:42	9:13	Grazing
9:13	9:35	Grazing
9:35	9:36	Stand
9:36	9:46	Grazing
9:46	10:08	Grazing
10:08	10:45	Grazing
10:45	10:49	Grazing
10:49	10:50	Walk
10:50	10:51	Grazing
10:51	10:57	Stand
10:57	10:58	Stand
10:58	10:59	Walk
10:59	11:09	Grazing
11:09	11:11	Grazing
11:11	11:13	Stand
11:13	11:30	Grazing
11:30	12:18	Stand
12:18	12:39	Stand
12:39	12:41	Fright, Run
12:41	13:11	Stand
13:11	14:28	Unknown (Suspect: Grazing)
	14:29	Collect sheep & remove collars
	-	Stop



Table A.10: Observations made on 17\_07\_2014 and 18\_07\_2014

Test Name	17_07_2014	Very_Big_Camp
Resolution	16g	
Tag Number	Sheep Type	
Tag 1	Lamb	
Tag 2	Lamb	
Tag 3	Ewe	
Tag 4	Ewe	
Tag 5	Ewe	
Time 1	Time 2	Description
16:49	17:02	Setup
17:02	17:11	Run
17:11	17:12	Walk
17:12	17:13	Stand
17:13	17:14	Walk
17:14	17:15	Run
17:15	17:21	Walk
	17:21	Grazing
18_07_2014		The Next Day
	14:25	Collect sheep with horse
	15:05	Sheep in small camp
	-	Remove collars
	-	Stop

# Appendix B

## PCB Footprints

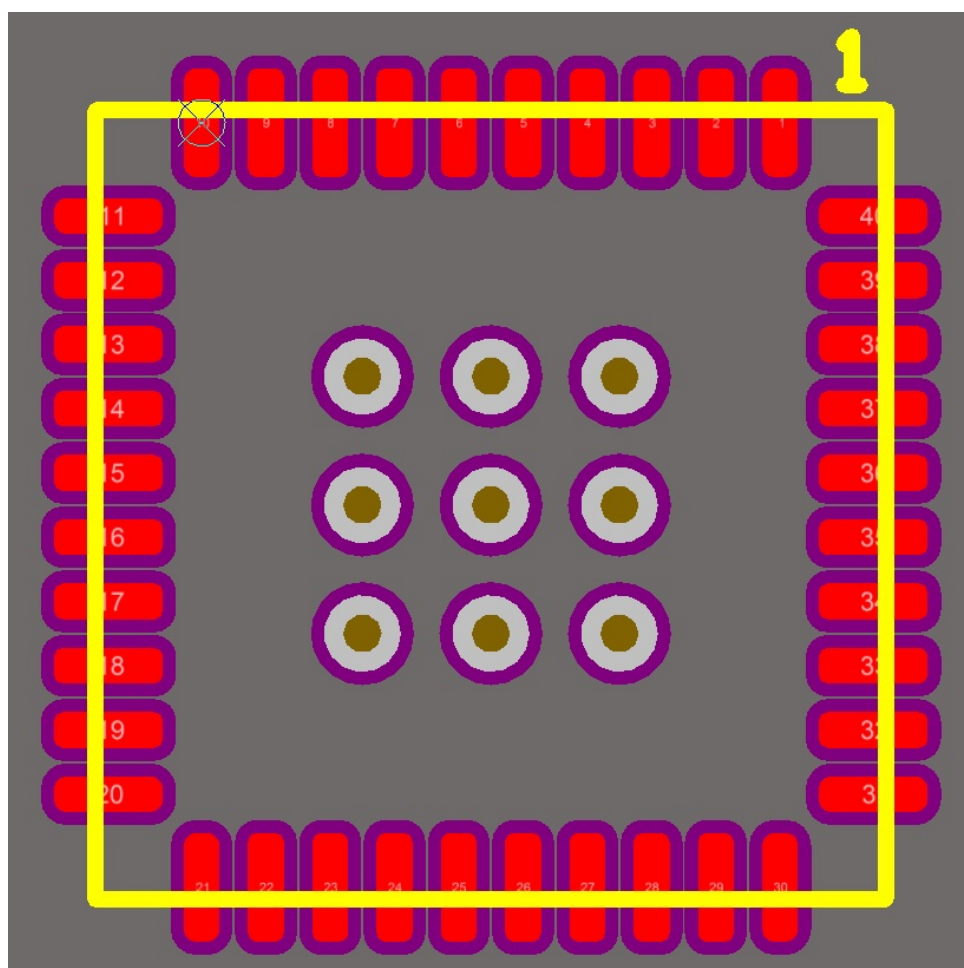


Figure B.1: MSP430FR5739 PCB footprint.

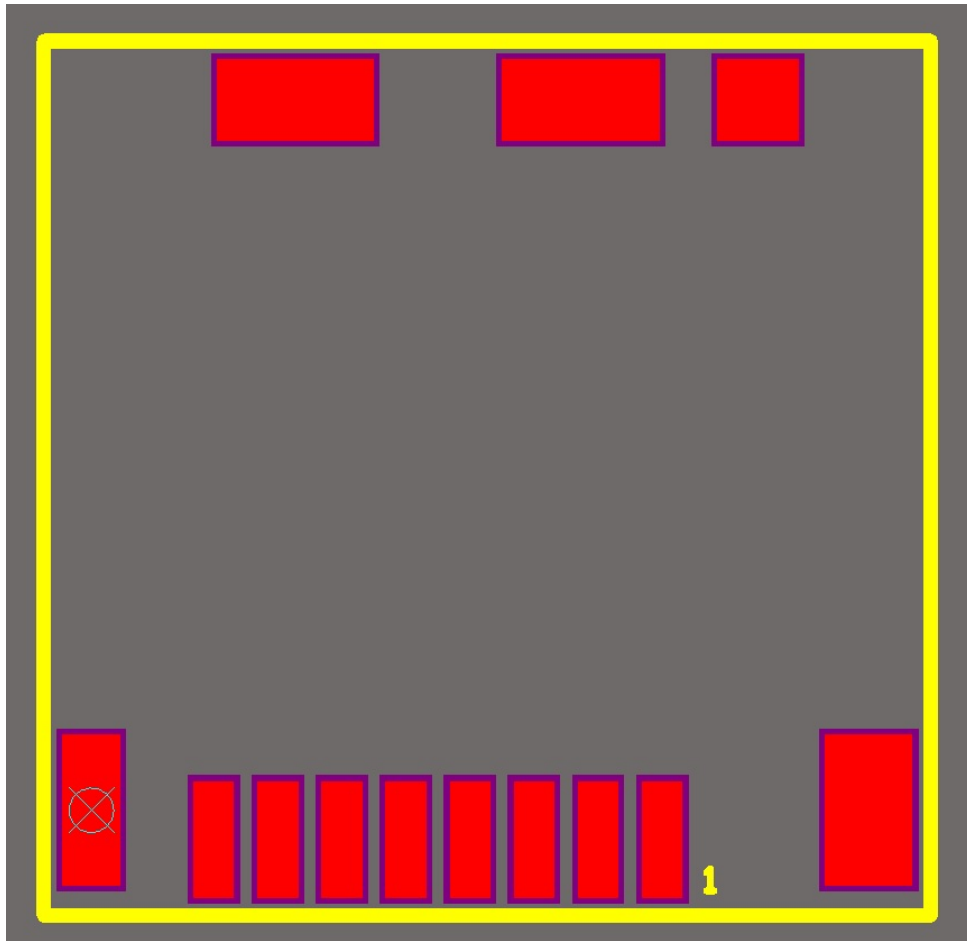


Figure B.2: MicroSD card holder PCB footprint.

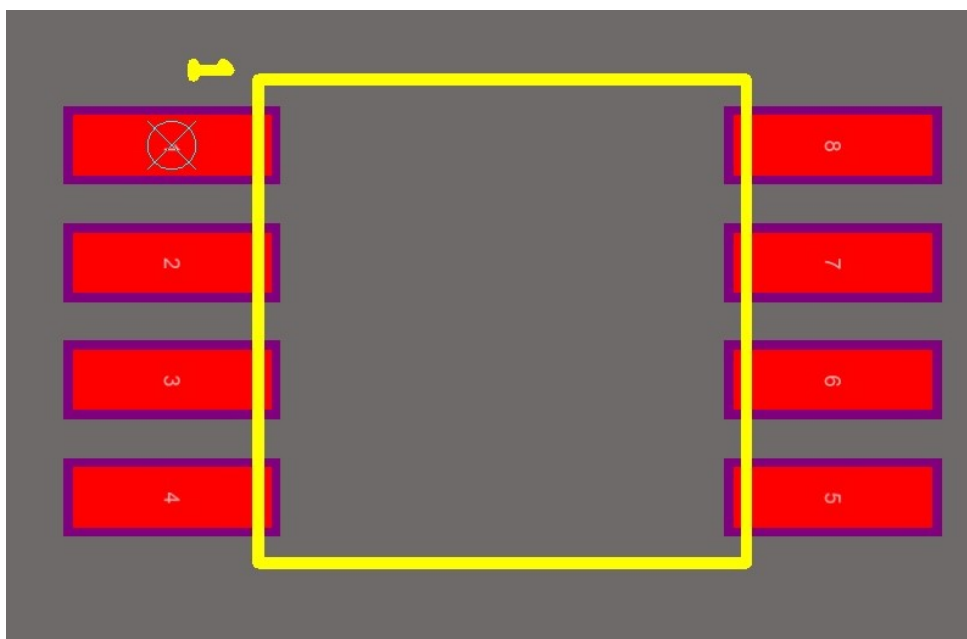


Figure B.3: FRAM PCB footprint.

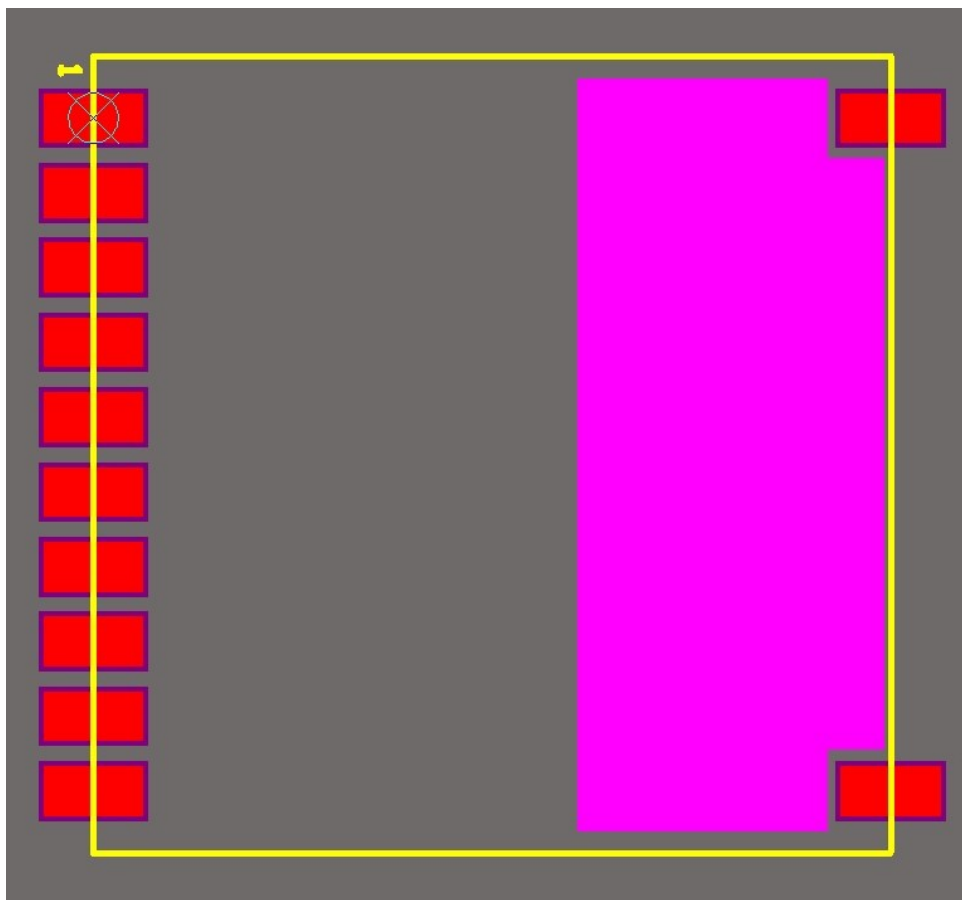


Figure B.4: GNS602 PCB footprint.

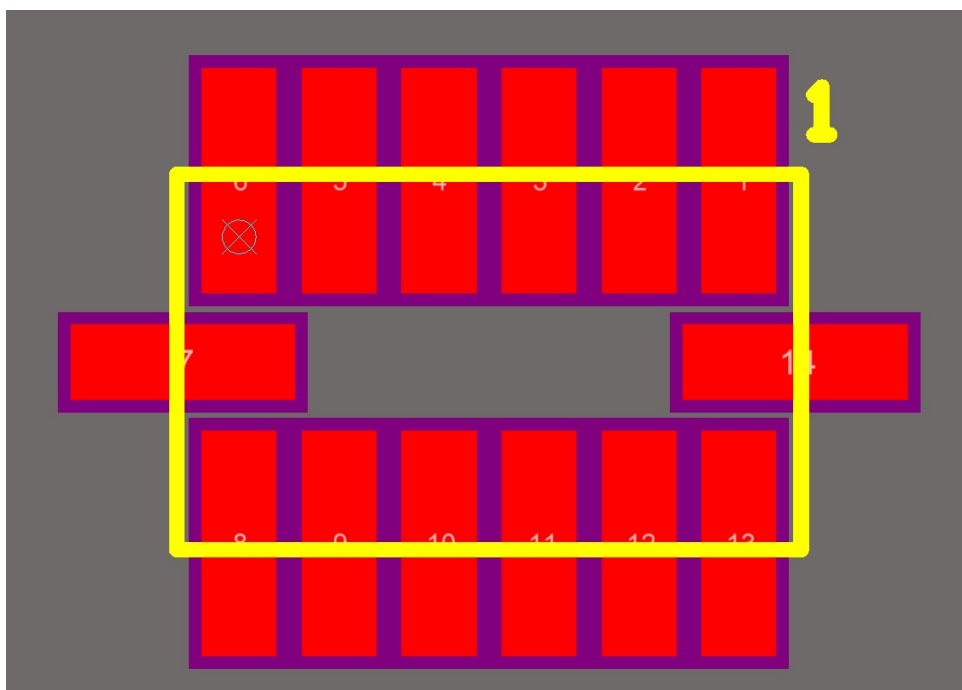


Figure B.5: ADXL345 PCB footprint.



Figure B.6: SQ-SEN-200 PCB footprint.



Figure B.7: MiniSense 100 PCB footprint.

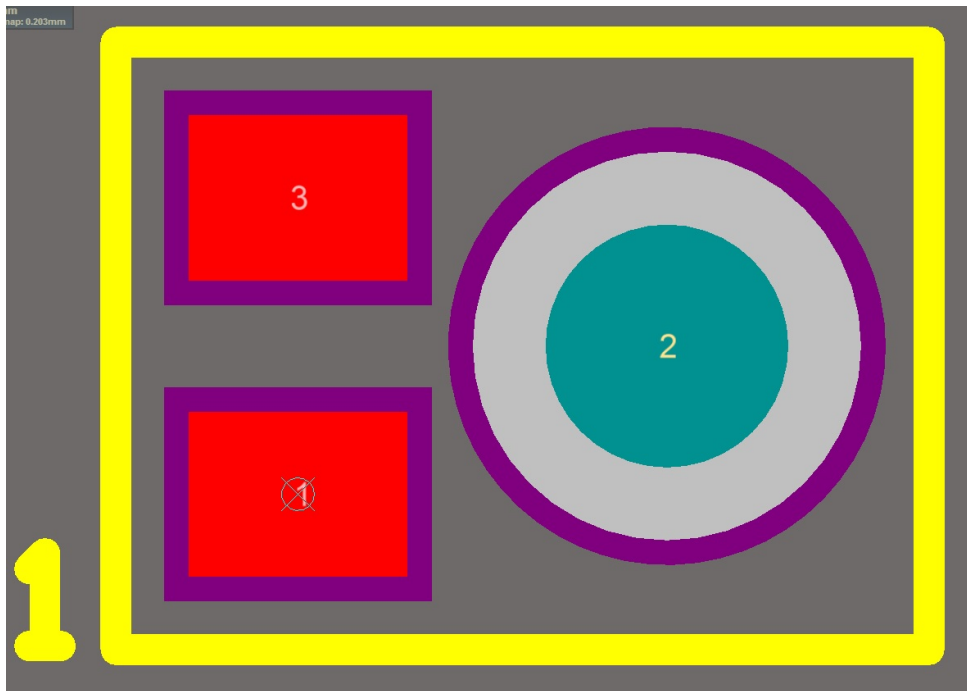


Figure B.8: ADMP504 PCB footprint.

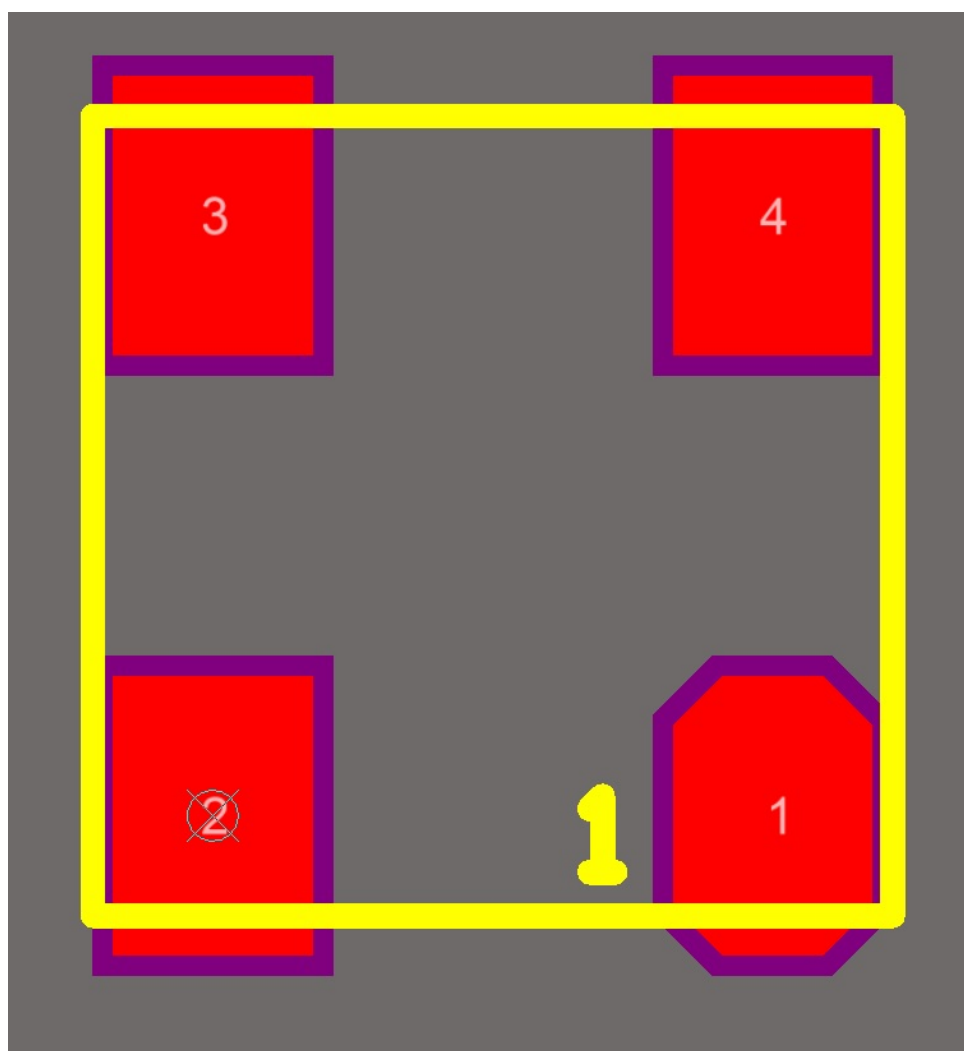


Figure B.9: ABS 1200 PCB footprint.

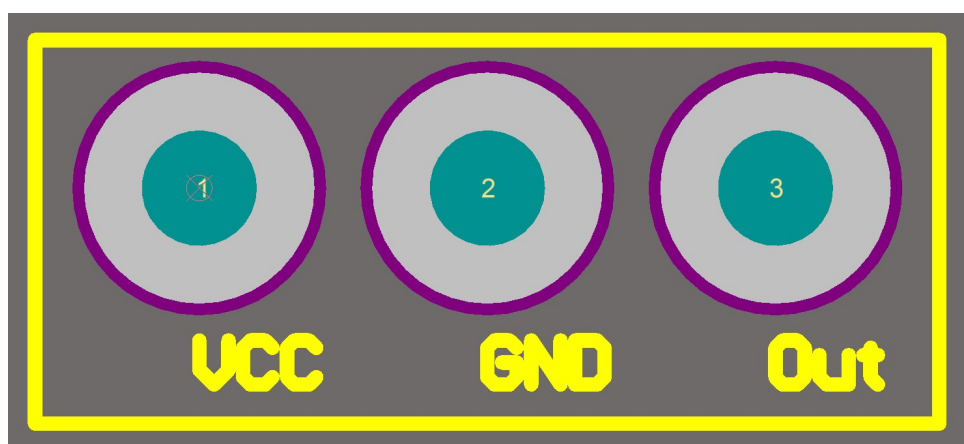


Figure B.10: Contact microphone PCB footprint.

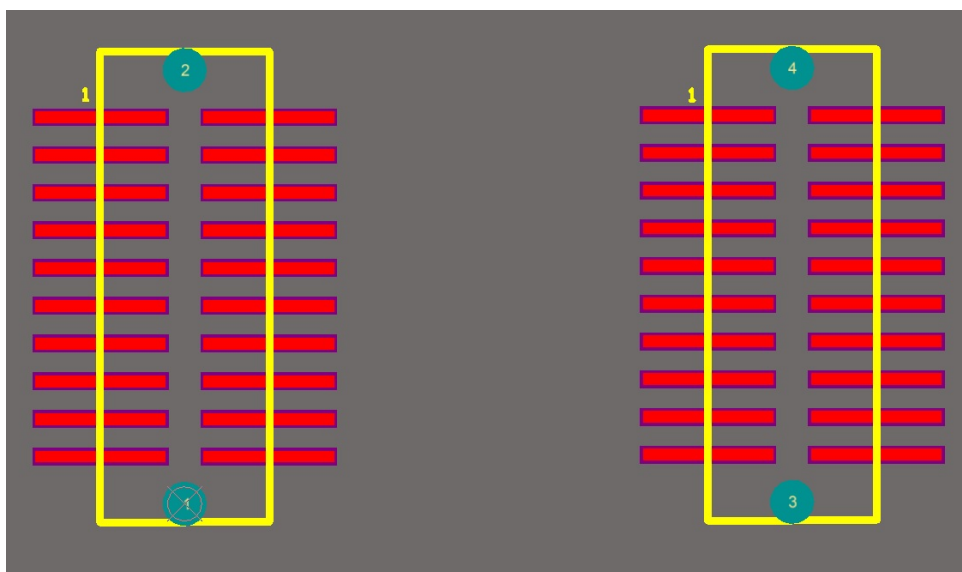


Figure B.11: Samtech connector PCB footprint.

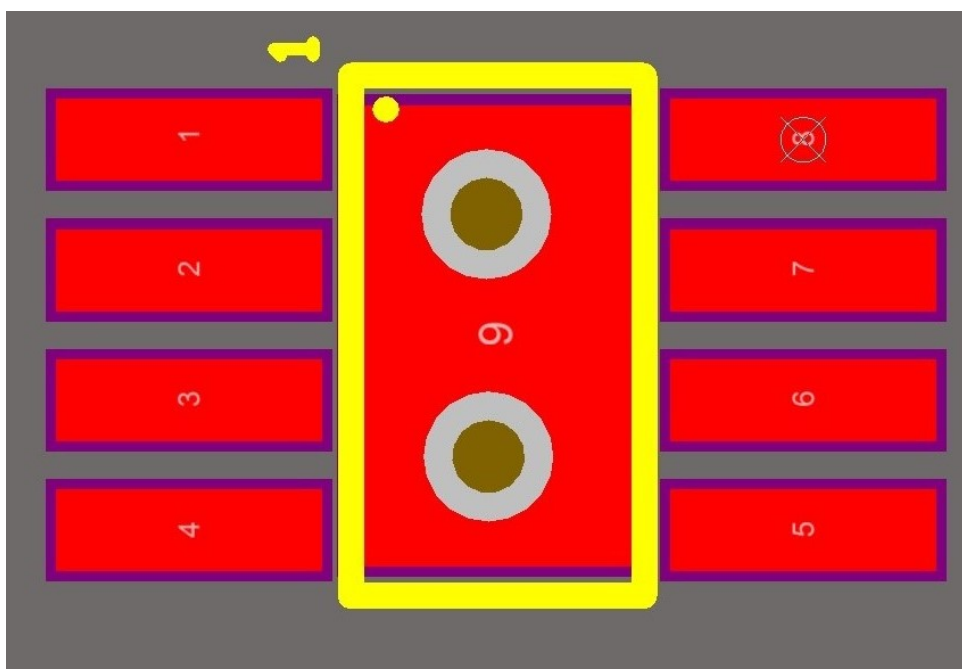


Figure B.12: L6932H1.2 PCB footprint.

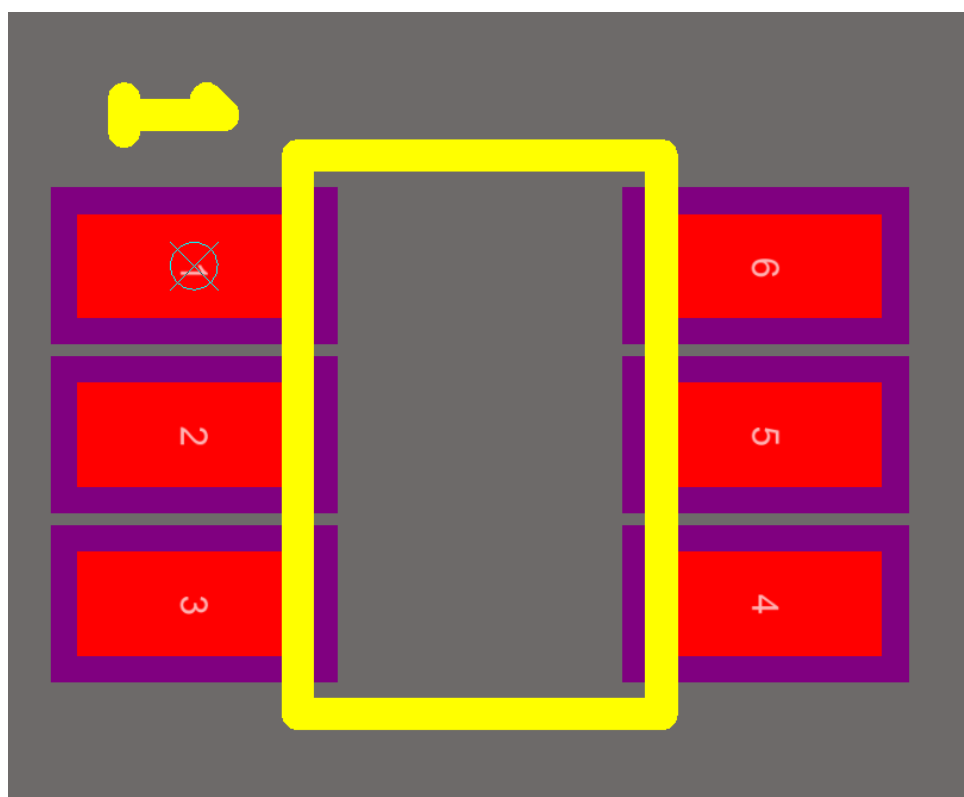


Figure B.13: TPS61222 PCB footprint.



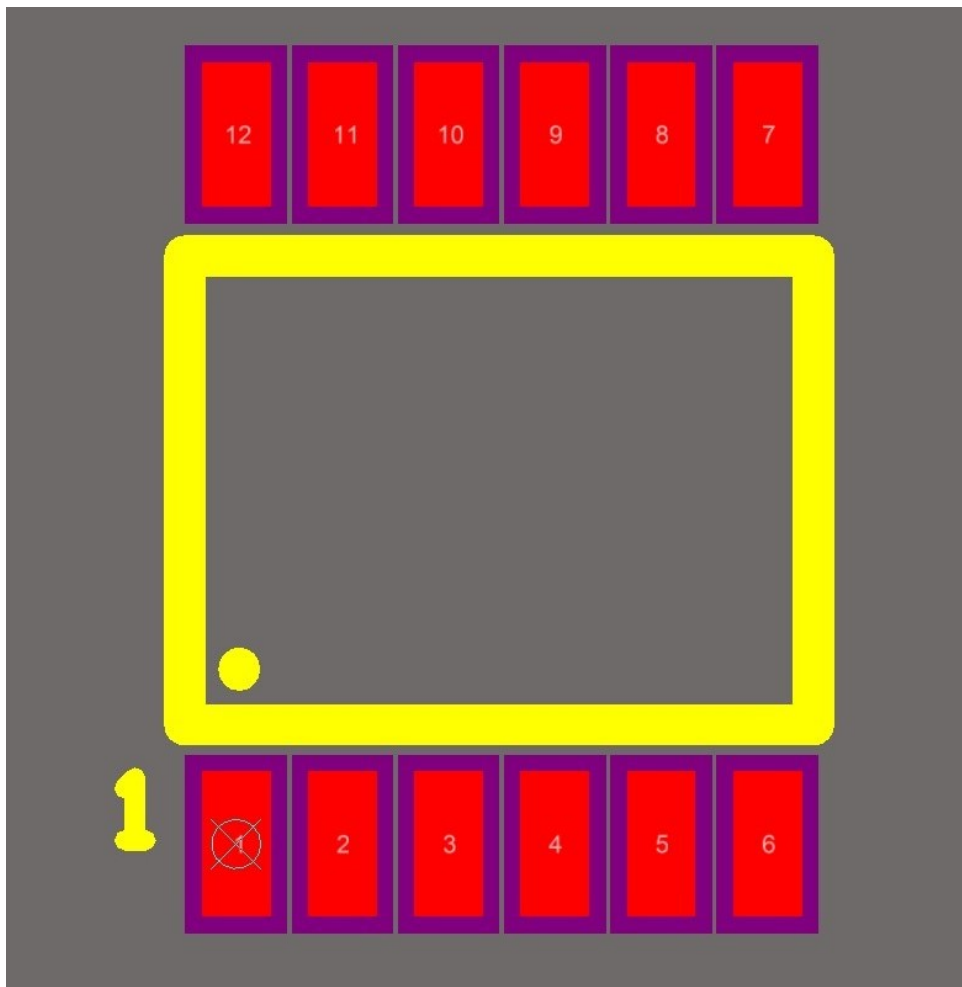


Figure B.14: LTC3105 PCB footprint.

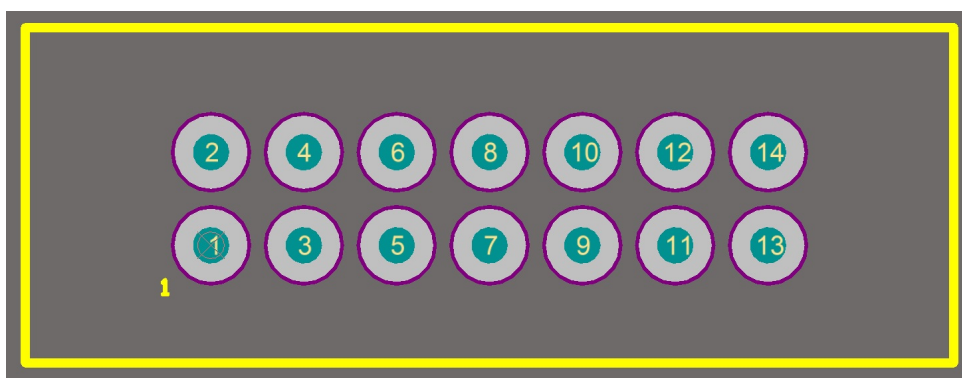


Figure B.15: MSP-FET430UIF female connector PCB footprint.

# Appendix C

## PCB Layouts

### C.1 WildMote PCB Layout

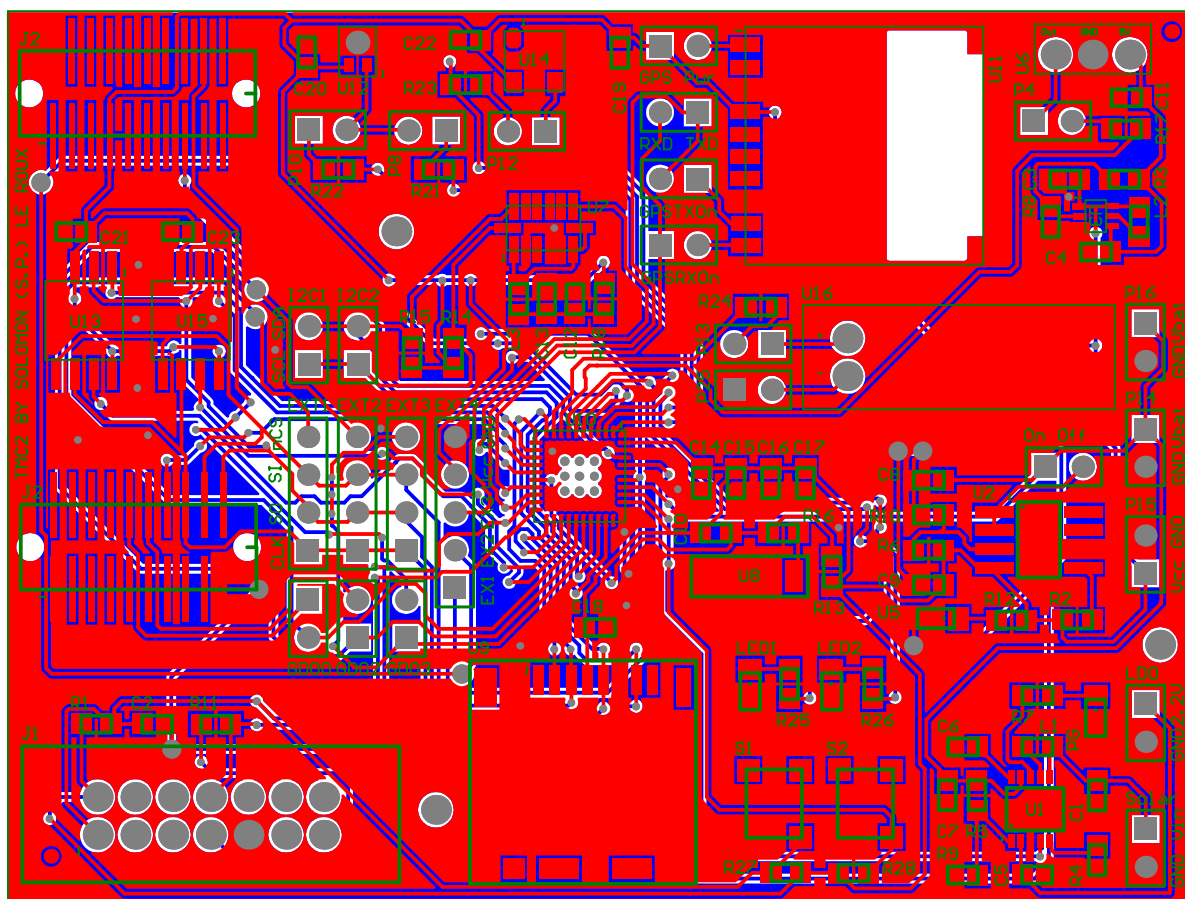


Figure C.1: WildMote PCB layout.

## C.2 RF Shield PCB Layout

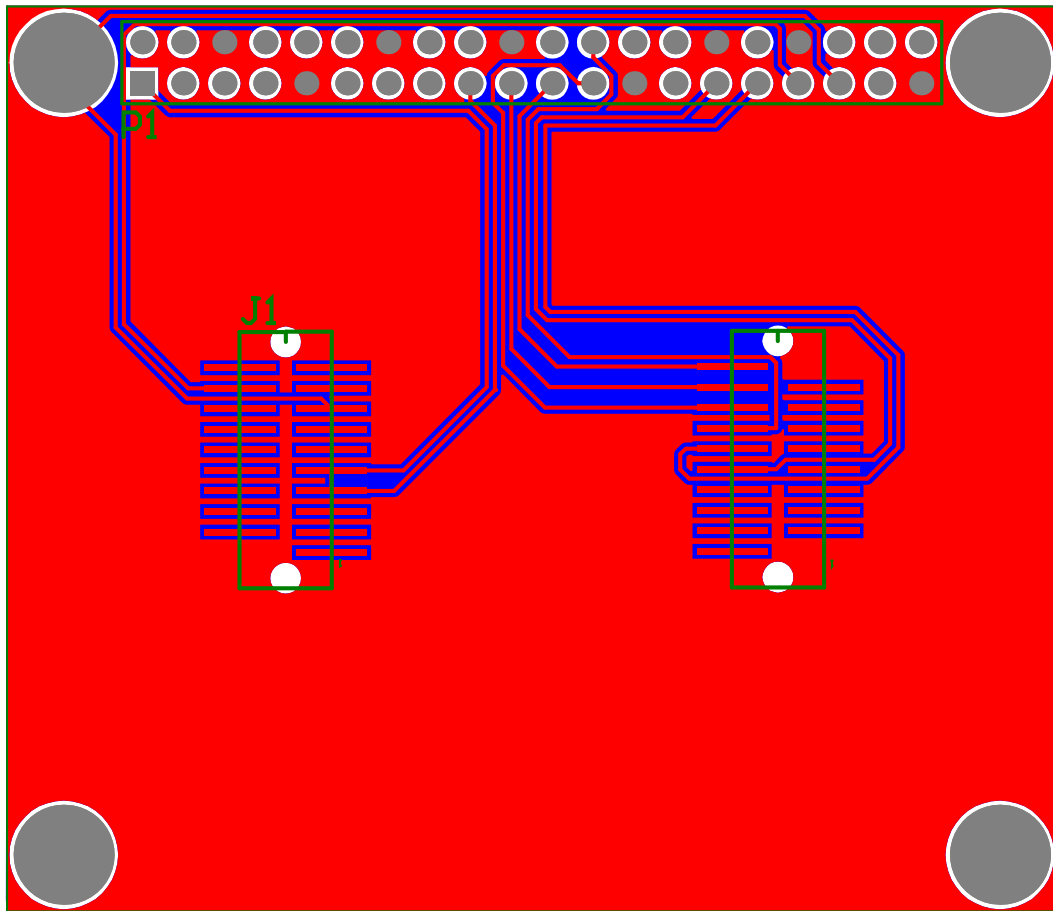


Figure C.2: RF Shield PCB layout.

### C.3 GSP Expansion Card PCB Layout

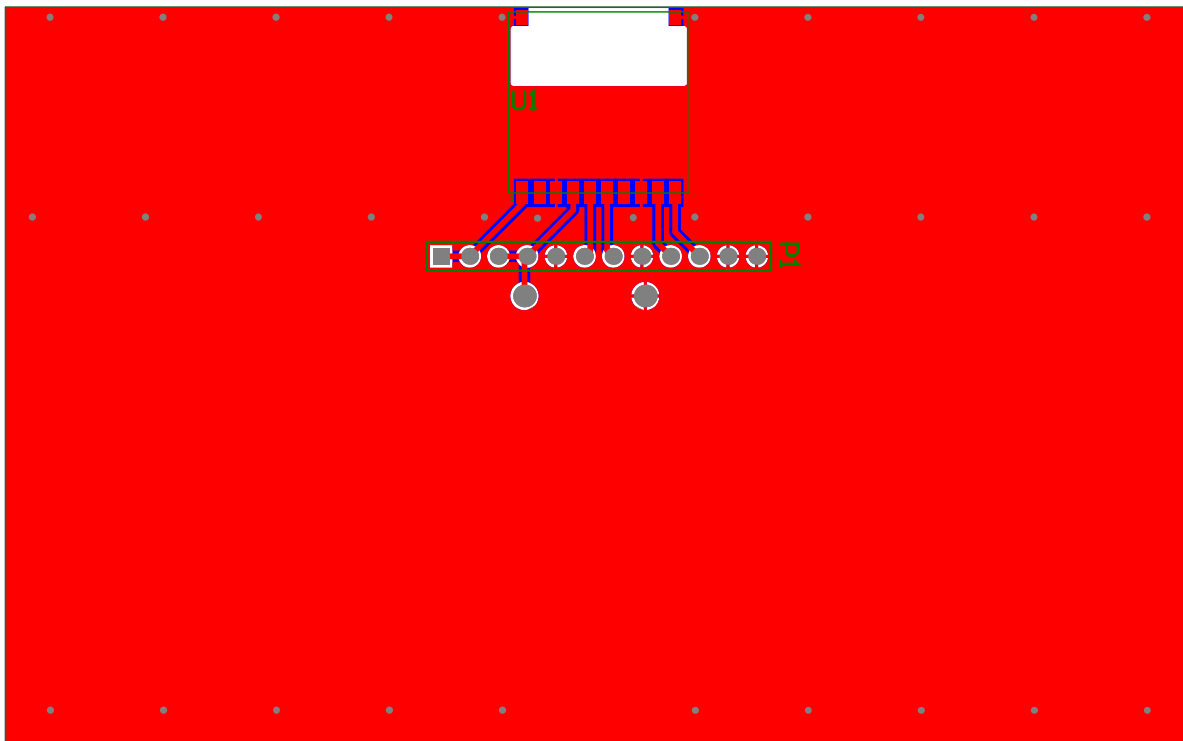


Figure C.3: GSP Expansion Card PCB layout.

# Appendix D

## Schematic Diagrams

D.1 WildMote Schematic Diagram

D.2 RF Shield Schematic Diagram

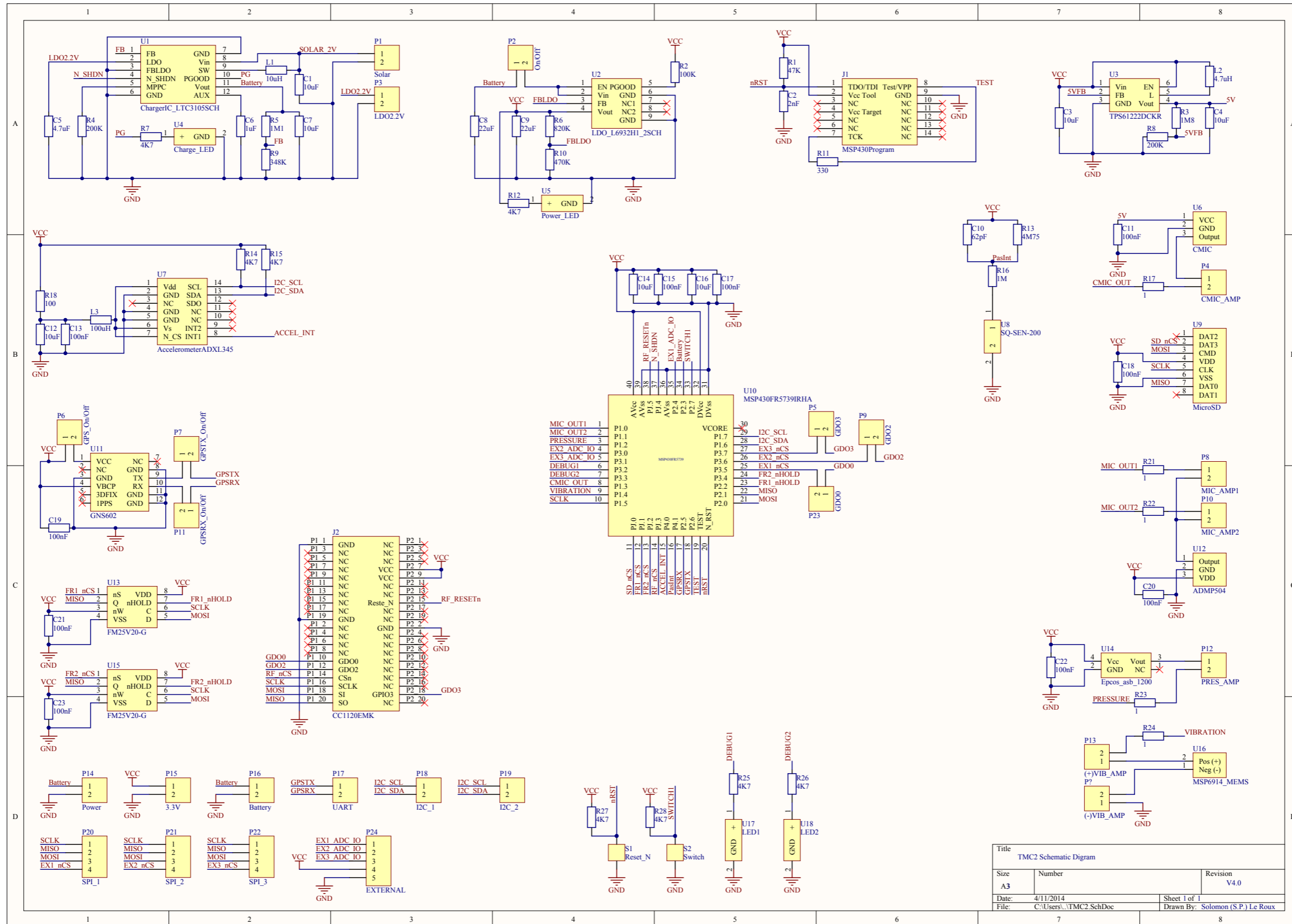
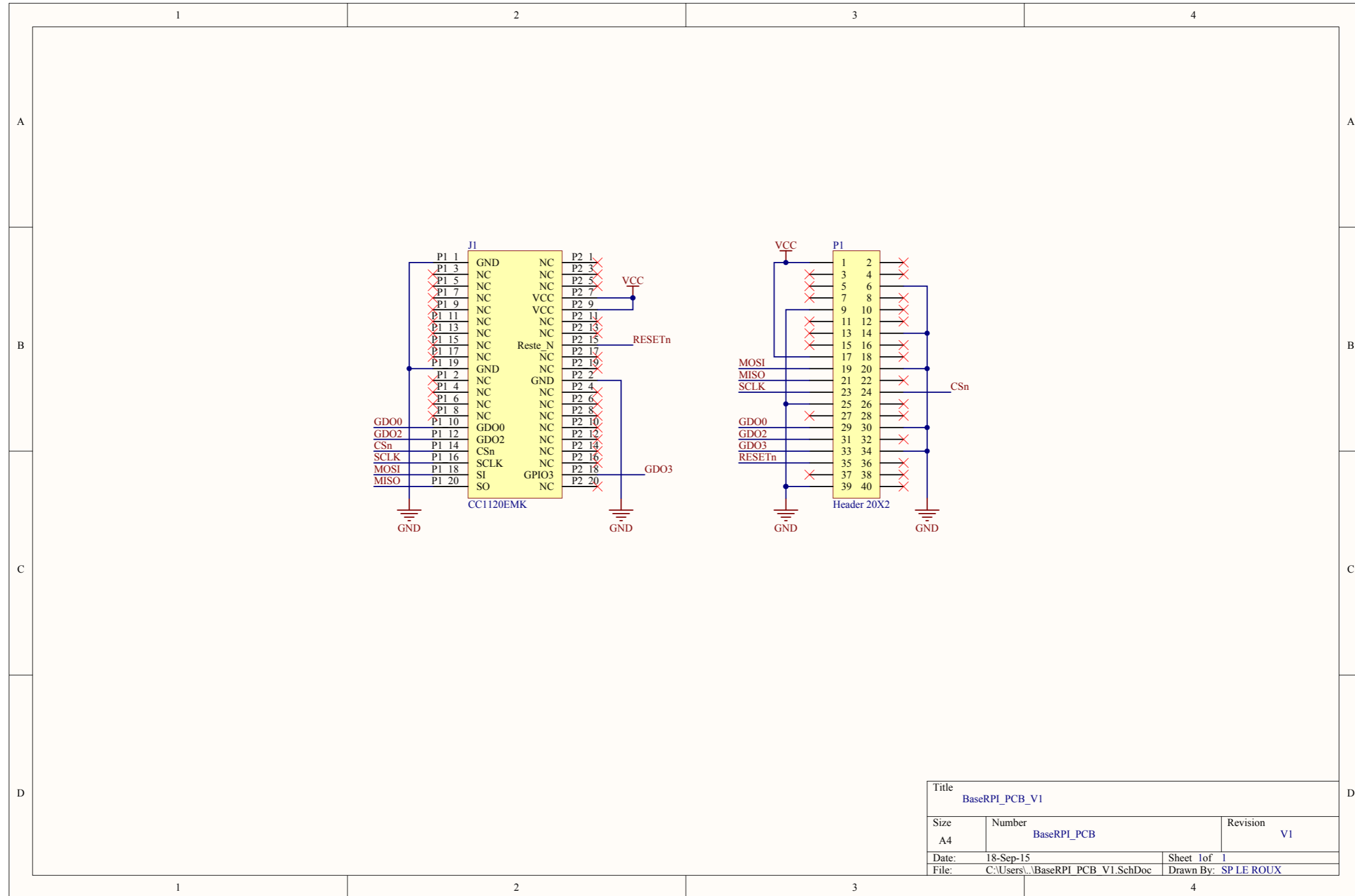


Figure D.1: WildMote schematic diagram.



Title BaseRPI_PCB_V1		
Size A4	Number BaseRPI_PCB	Revision V1
Date: 18-Sep-15	Sheet 1 of 1	
File: C:\Users\...\BaseRPI_PCB_V1.SchDoc	Drawn By: SP LE ROUX	

Figure D.2: RF Shield schematic diagram.

# Appendix E

## Casing Design Drawings



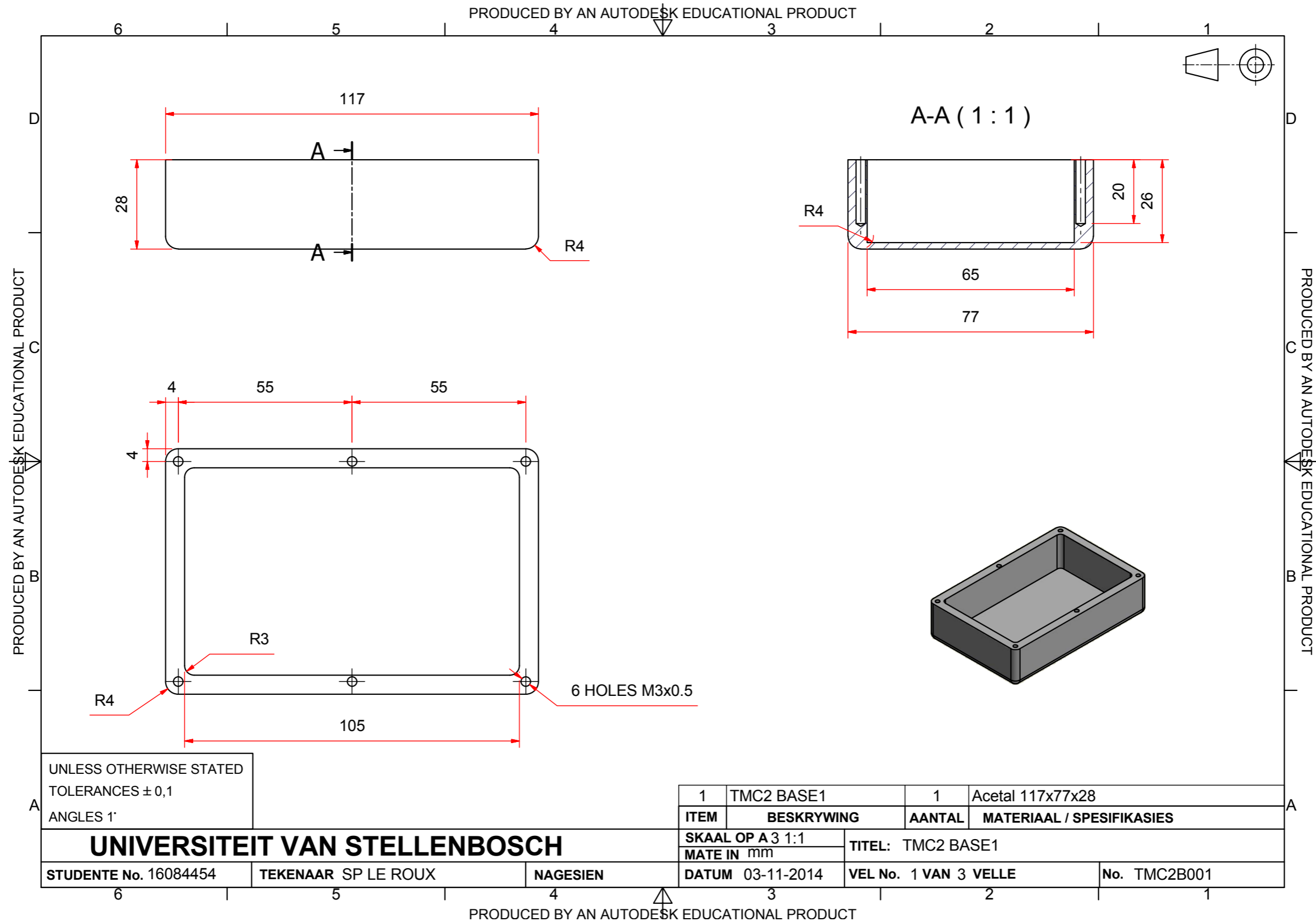


Figure E.1: Casing Base drawing.

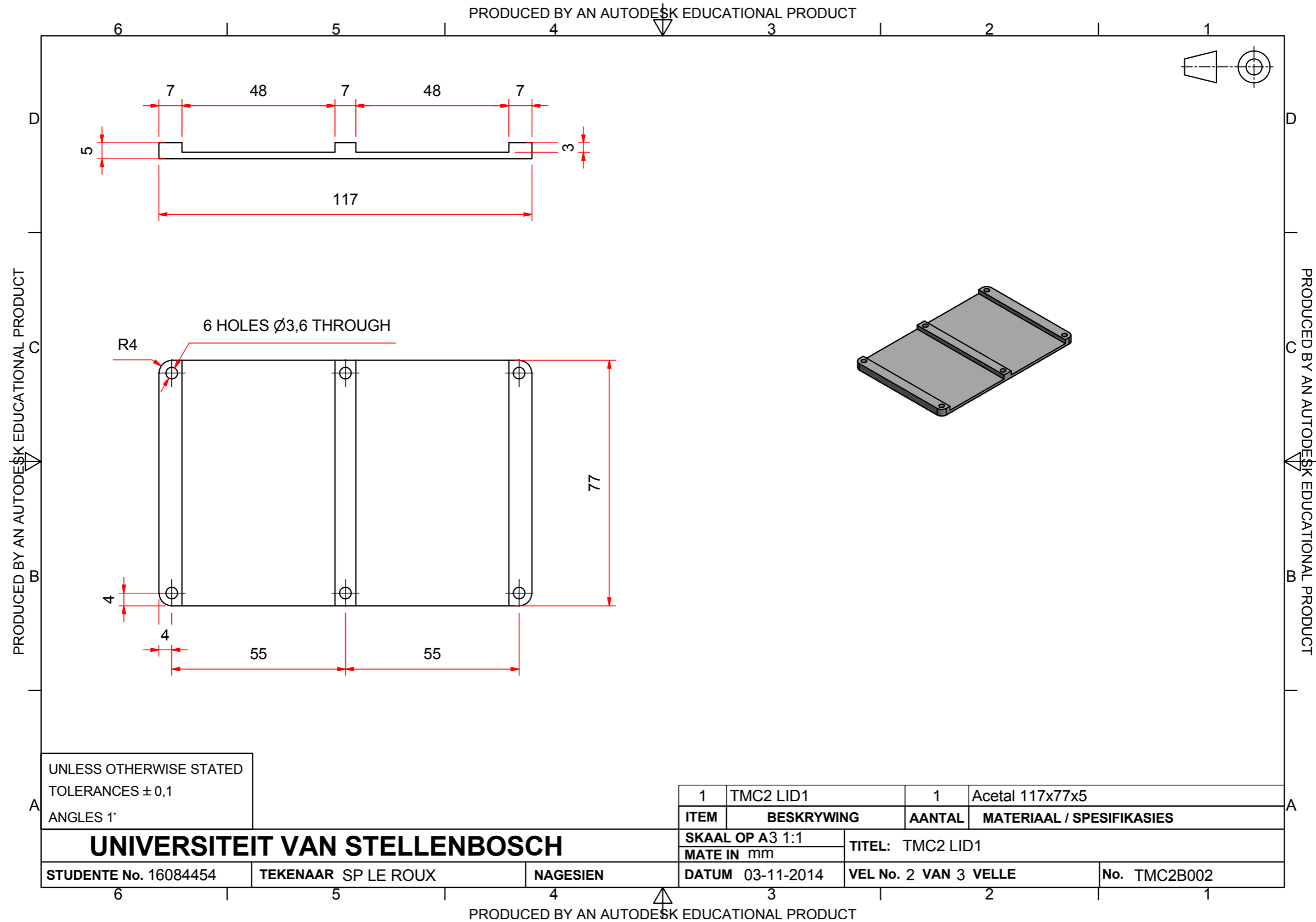


Figure E.2: Casing Lid 1 drawing.

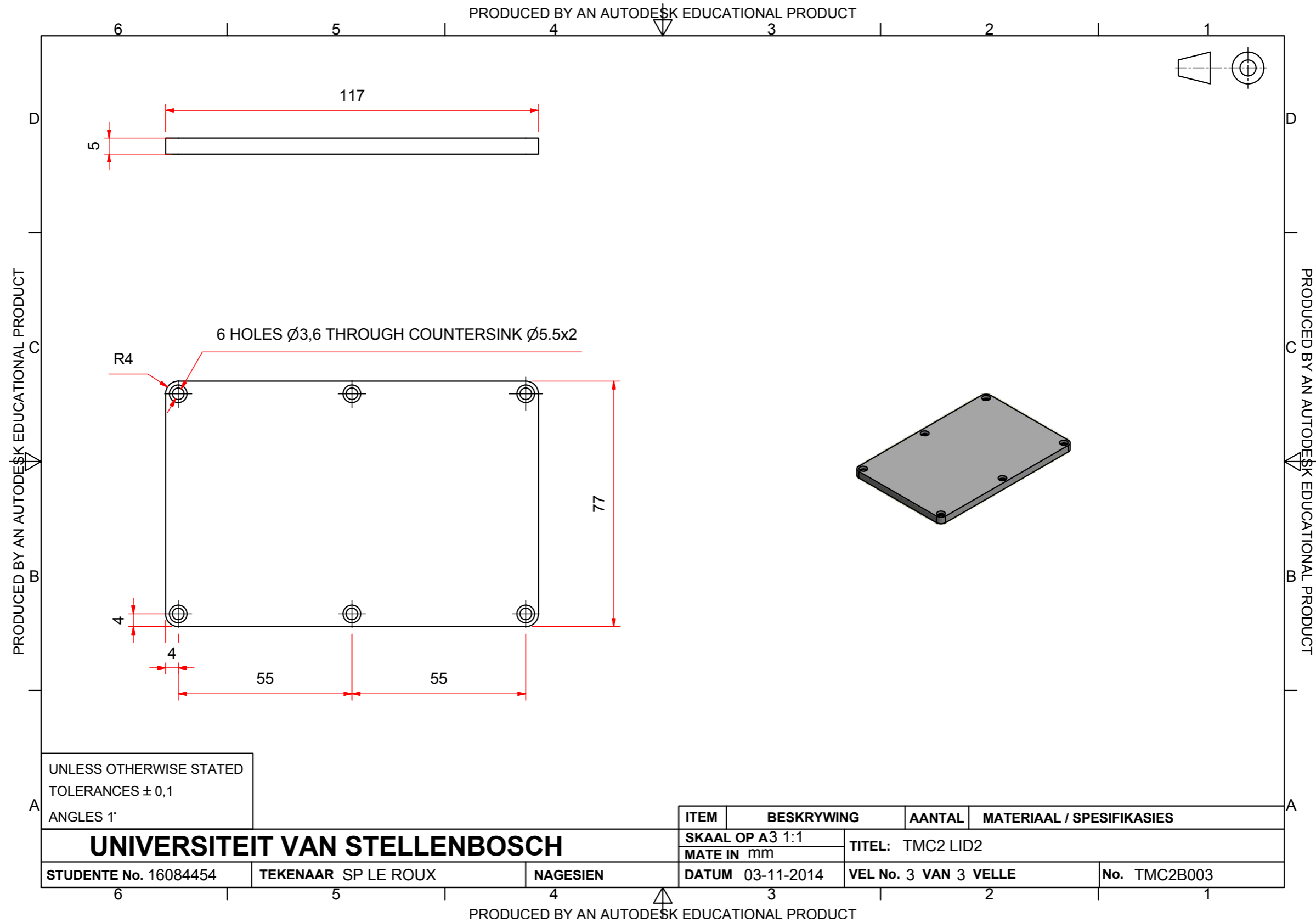


Figure E.3: Casing Lid 2 drawing.

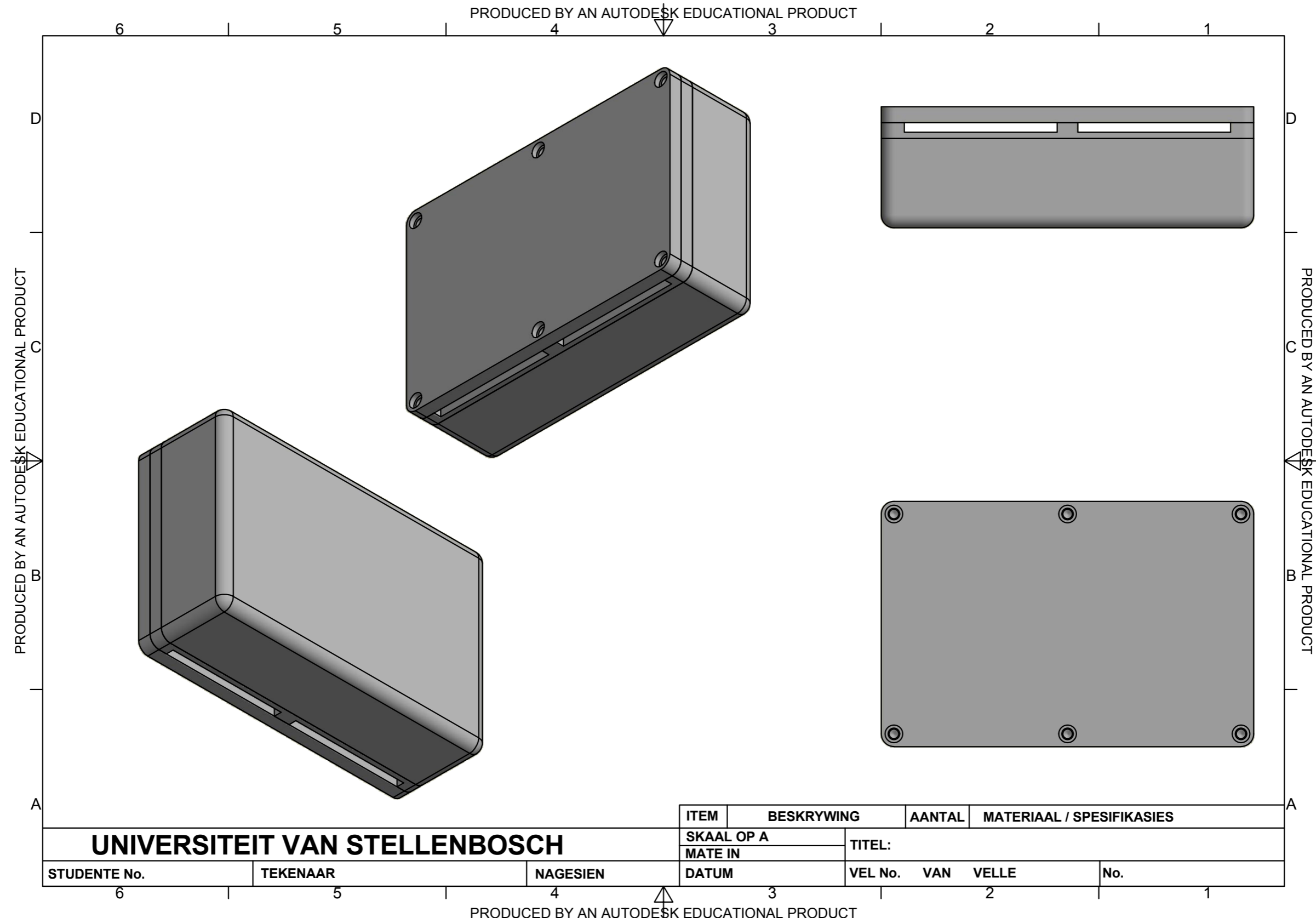


Figure E.4: 3D view of the casing.