



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Design and Development of a Satellite Ground Station for Water Resource Monitoring



Thesis submitted in partial fulfilment of the requirements for the
degree of Master of Science in Electronic Engineering

in the
Faculty of Engineering
Department of Electrical and Electronic Engineering

Supervisor: Dr. Riaan Wolhuter

December 2014

Declaration of Authorship

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.

Signed: Harry D Mafukidze

Date: December 2014

UNIVERSITY OF STELLENBOSCH

Abstract

Faculty of Engineering

Department of Electrical and Electronic Engineering

Master of Science in Electronic Engineering

Design and Development of a Satellite Ground Station for Water Resource Monitoring

by [Harry D Mafukidze](#)

The SU Department of Forestry has the responsibility to monitor, assess and suggest management processes for water resources in some remote areas. The researchers need information on wind speed, wind direction, soil run-off, absorption and soil drainage. Most of the areas they are targeting have no form of GSM/GPRS coverage. This thesis presents the design and development of a Zigbee based wireless sensor network to send data from distributed sensor nodes to a ground station, all in a remote area. The ground station in turn uses a global commercial satellite communications system to send the field data to a centralised host computer. This was accomplished through the integration of the most common and popular open source and commercial electronics prototyping platforms, namely, Arduino, Digi XBee radio, Raspberry Pi and Iridium modem. The system relies on an Arduino Uno working as a sensor node, Digi XBee radios for forming wireless mesh and multi-hop networks, Raspberry Pi being the heart of the ground station and the Iridium modem to send data to the master station through the Iridium gateway. A comprehensive literature study was conducted and a prototype of the system implemented. Various tests were conducted to determine and prove the feasibility of the system.

UNIVERSITEIT VAN STELLENBOSCH

Uittreksel

Fakulteit Ingenieurswese

Departement Elektriese en Elektroniese Ingenieurswese

Meester van Wetenskap in Elektroniese Ingenieurswese

Ontwerp en ontwikkeling van 'n satellite Station vir waterhulpbronbestuur Monitoring

deur Harry D Mafukidze

Die Departement van Bosbou het die verantwoordelikheid om water hulpbronne in afgeleë areas te monitor, evalueer en voorstelle te maak tov. die bestuur daarvan. Die navorsers benodig inligting oor windspoed, windrigting, grondwater afloop, -opname en -dreinerings. Die meeste van die gebiede ter sprake het geen vorm van GSM / GPRS-dekking nie. Hierdie tesis beskryf die ontwerp en ontwikkeling van 'n Zigbee gebaseerde radio sensornetwerk om data vanaf verspreide sensornodes te stuur na 'n grondstasie. Die grondstasie op sy beurt maak gebruik van 'n globale kommersieel satelliet-kommunikasiesstelsel om data van 'n afgeleë plek in die veld te stuur na 'n gesentraliseerde rekenaarstelsel. Dit is gedoen deur van die mees algemene en gewilde prototipe oopbron en kommersieel platforms, naamlik Arduino, Digi XBee radio, Raspberry Pi en Iridium modem te integreer. Die sensornodes is gebaseer op 'n Arduino Uno, met Digi XBee radio's vir die radio- multihop netwerk. Die grondstasie is gebou om die Raspberry Pi en stuur data aan na die meesterstasie, via die Iridium modem en satellietstelsel. Na 'n omvattende literatuurstudie, is 'n prototipe van die stelsel ontwerp en geïmplementeer. Omvattende toetse is gedoen om die korrekte werking en bruikbaarheid van die stelsel te bewys.

Acknowledgements

First and foremost, I would like to express my sincerest gratitude to my supervisor, Dr Riaan Wolhuter, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I would also like to thank the following for their assistance throughout the course of my work:

- Mr. Landiech Matthieu for helping with the Master Station software interface
- Mr. J P Meijers for hardware and software related issues
- Mr. Tim for hardware related services
- Mr. W Croucamp for assistance with fabrication of mounting stands
- Mr. A Kunneke for assistance with weather related information
- My family for their support
- My saviour Jesus

*This thesis is dedicated to my parents who supported me
throughout my studies*

Contents

Declaration of Authorship	i
Abstract	ii
Uittreksel	iii
Acknowledgements	iv
List of Figures	xiii
List of Tables	xvi
Abbreviations	xvii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problem Definition	2
1.3 Project Background	2
1.4 Some existing solutions of radio telemetry	3
1.4.1 Direct Radio (VHF and UHF)	3
1.4.2 Cellular (GSM/GPRS)	3
1.4.3 Bluetooth	3
1.5 Aim	4
1.6 Specific Objectives	5
1.7 Proposed Method	6
1.8 Project Outcome	6
1.9 Thesis Organisation	7
2 Literature Study	8
2.1 Introduction	8
2.2 Characteristics of sensor nodes: requirements	9
2.2.1 Computation power:	9
2.2.2 Power source:	9
2.2.3 Memory:	10

2.2.4	Transceiver:	10
2.2.5	Sensing element:	11
2.3	Random deployments and self-organization:	11
2.4	Primary functions	12
2.5	Application of wireless sensor networks	12
2.5.1	Environment and wildlife monitoring:	12
2.5.2	Water supply and sewer system monitoring:	13
2.5.3	Industrial control and manufacturing:	13
2.5.4	Ocean monitoring:	13
2.5.5	Agriculture:	13
2.5.6	Disaster control	14
2.5.7	Structural health monitoring:	14
2.5.8	Some applications of WSN	14
2.6	WSN Requirements	14
2.6.1	Scalability	14
2.6.2	Adaptability	15
2.6.3	Efficiency	15
2.6.4	Security	15
2.6.5	Reliability	15
2.6.6	Mobility	16
2.6.7	Responsiveness	16
2.6.8	Throughput, bandwidth	16
2.7	Protocol stack	16
2.7.1	Application layer	18
2.7.2	Presentation layer	18
2.7.3	Session layer	19
2.7.4	Transport layer	19
2.7.5	Network layer	19
2.7.6	Datalink layer	21
2.7.7	Mac layer	21
	2.7.7.1 Addressing mechanism	22
	2.7.7.2 Channel access control mechanism	22
2.7.8	Physical layer	22
2.8	Sensor nets operating systems:	23
2.8.1	TinyOS	23
2.8.2	Contiki	23
2.8.3	Nano-RK	23
2.9	Basics of satellite communication systems	23
2.10	Iridium satellite system	24
2.10.1	Iridium communication system.	24
2.10.2	Power supply design consideration:	25
2.10.3	Hardware	26
2.10.4	Types of data services	26
2.11	Globalstar	26
2.11.1	Operating modes	26
2.12	Orbcomm Satellite system	28
2.13	Inmarsat	29

2.14	Zigbee technology	32
2.14.1	IEEE 802.15	32
2.14.2	802.15.4 PHY	33
2.14.3	802.15.4 Device types	34
2.14.4	802.15.4 MAC	35
2.15	Zigbee protocol	35
2.15.1	Zigbee applications:	35
2.15.2	Zigbee Phy layer	36
2.15.3	Zigbee Mac Layer	36
2.15.4	Zigbee Network Layer	37
2.15.5	Application Layer	37
2.15.6	Device Types	38
2.15.7	Star	39
2.15.8	Cluster Tree	39
2.15.9	Mesh	39
2.15.10	Broadcast	40
2.15.11	Zigbee addressing	40
2.15.12	Forming a Zigbee network	41
2.15.13	Joining a Zigbee network	41
2.16	Zigbee routing	41
2.16.1	Mesh routing	41
2.16.2	Proactive mesh routing	42
2.16.3	Reactive mesh routing	43
2.17	Free space propagation model	43
2.17.1	Path loss	44
2.18	Summary	45
3	Proposed Methodology	46
3.1	Project development	46
3.1.1	Project requirements	47
3.1.2	Performance Requirements	47
3.1.3	Integration Requirements	47
3.1.4	Regulatory Requirements	48
3.1.5	Design Phase	48
3.1.6	Architecture Decisions	48
3.1.6.1	Star vs. Mesh Topology	48
3.1.6.2	Considerations for Battery-Powered Systems	49
3.1.7	Implementation	49
3.1.7.1	Testing WSN applications	49
3.1.7.2	Unit Test	49
3.1.7.3	Integration Test	50
3.1.7.4	Performance Test	50
3.1.7.5	Field Test	50
3.1.8	Operation Phase	50
3.2	List of project requirements	51
3.3	Software requirements	51
3.4	Hardware requirements	52

3.5	Hardware development overview	53
3.5.1	Arduino	53
3.5.2	Arduino Uno	53
3.5.3	Raspberry Pi	55
3.6	Power supply design	56
3.6.1	Battery sizing:	57
3.6.2	Battery sizing methodology	57
3.6.2.1	Battery sizing	58
3.6.3	Solar panel sizing	60
3.6.3.1	Sensor node	60
3.6.3.2	Ground station	61
3.6.4	Charging regulator	62
3.6.4.1	Design equations	62
3.6.4.2	Resistor calculations	64
3.6.4.3	Charger specs	65
3.6.5	Low voltage disconnect (Battery monitor)	66
3.7	Hardware integration	67
3.7.1	Soil moisture sensor	67
3.7.2	Soil temperature sensor	69
3.7.3	Rain gauge	70
3.7.4	Cup anemometer and wind vane	71
3.7.5	Air temperature and air humidity sensor	73
3.7.6	Ambient light	74
3.7.7	Battery backup	74
3.7.8	Solar Panel	75
3.8	Summary	76
4	Implementation	77
4.1	Hardware design	77
4.1.1	Node design	77
4.1.2	Radio transceiver	78
4.1.3	Microcontroller transceiver interface	78
4.1.4	XBee and sensor power control	79
4.1.5	Power dissipation	80
4.1.6	Enclosure	80
4.2	System testing	80
4.3	Sensor node development platform	82
4.3.1	ADC Implementation	83
4.3.2	UART and Software Serial Implementation	84
4.3.3	Sensor node software Implementation:	85
4.3.4	Ground station software architecture	86
4.3.5	GUI software	86
4.4	Radio Communication and networking	87
4.4.1	Hardware and electrical characteristics	87
4.5	Data collection in WSN	89
4.5.1	Node discovery	89
4.5.2	Data transfer	89

4.5.3	Routing to the destination	89
4.6	File types	91
4.7	Solar panel tilt angle	92
4.8	Measuring Arduino's power source	93
4.9	Master Station	93
4.10	Installation	93
4.11	Interface Architecture	94
4.12	Connection	94
4.12.1	Index.php	94
4.12.2	userconnection.php	94
4.13	Include	95
4.13.1	Header.php	95
4.13.2	Footer.php	95
4.13.3	Toolbar.php	96
4.13.4	AdminToolBar.php	96
4.14	Main Pages	96
4.14.1	Map.php	96
4.14.2	Station.php	96
4.14.3	Sensor.php	96
4.15	Graphs	97
4.15.1	Minmax.php	97
4.15.2	Minmaxbattery.php	97
4.15.3	Polarclock.php	97
4.15.4	Bargraph.php	97
4.15.5	Reservoir.php	97
4.16	Settings includes	98
4.16.1	AddGs.php	98
4.16.2	DelGs.php	98
4.16.3	Addss.php	98
4.16.4	Delss.php	98
4.16.5	Importform.php	98
4.16.6	AjaxGs.php	98
4.16.7	SensorSetting.php	98
4.17	Settings	99
4.17.1	UserSetting.php	99
4.17.2	StationSetting.php	99
4.17.3	Setting.php	99
4.18	Database actions	99
4.18.1	dbmanager.php	99
4.18.2	Import.php	99
4.18.3	Export.php	100
4.18.4	Exportall.php	100
4.19	Mail	100
4.19.1	Mail.php	100
4.19.2	MailProcessing.php	100
4.20	Database Architecture	100
4.20.1	User	100

4.20.2	Sensor	100
4.20.3	Ground station	102
4.20.4	Gs name	102
4.20.5	Gsname battery	103
4.20.6	Gsnameessname	103
4.21	Mail processing	103
4.21.1	Mail setting	104
4.21.2	Usage	104
4.21.3	Error Flag	105
4.22	User guide	106
4.22.1	Change password	106
4.22.2	Map	106
4.22.3	Station	107
4.22.4	Sensor	107
4.22.4.1	Curve Display	107
4.22.5	Date selection	108
4.23	Display	110
4.23.1	Min Max	110
4.23.2	Bar Graph	111
4.23.3	Polar clock	111
4.24	Administrator	113
4.24.1	Station management: Add a ground station	113
4.24.2	Add a sensor	113
4.24.3	Delete a sensor	114
4.24.4	Add or Delete User	114
4.24.5	Modify a sensor Type	115
4.25	Import Data	116
4.26	Export All	116
4.27	Summary	117
5	Evaluation and results	118
5.1	Power performance	118
5.1.1	Measurement setup	118
5.1.2	Lead acid battery considerations	119
5.1.3	Battery life time estimation	119
5.1.4	Interpretation	123
5.1.5	Experimental set up	123
5.1.6	Results:	124
5.1.7	Interpretation	125
5.2	RSSI	125
5.2.1	Experimental setting.	126
5.3	Throughput analysis	126
5.3.1	Experimental setup	127
5.4	Summary	128
6	Conclusion	129
6.1	Introduction	129

6.2	Future work	131
A	Project images	132
B	XBee RF modules	133
B.1	XBee ZB module configuration	133
B.1.1	Setting up the XBee coordinator	134
C	Project cost breakdown	136
D	Iridium SBD modem	137
D.1	9602 structure	137
D.2	RF interface	140
D.3	9602 AT commands	140
D.3.1	AT+SBDWB	140
D.3.2	AT+SBDWT	141
	Bibliography	143

List of Figures

2.1	WSN Architecture	9
2.2	OSI and Internet model comparison	17
2.3	WSN cross layers	17
2.4	Single hop routing model	20
2.5	Multi hop routing model	20
2.6	Categories of routing protocols	20
2.7	Categories of MAC protocols	22
2.8	Iridium communication architecture [7].	25
2.9	Iridium 9602 modem	26
2.10	Globalstar coverage map	27
2.11	Orbcomm system overview.	28
2.12	Orbcomm coverage map	28
2.13	Inmarsat coverage map	29
2.14	Comparison between Satellite Systems	30
2.15	IEEE WPAN	32
2.16	802.15.4 operating frequency bands	34
2.17	Zigbee stack architecture	36
2.18	Zigbee network topologies	39
2.19	Propagation of a broadcast message in a Zigbee network	40
2.20	AODV routing example	43
2.21	Wave propagation	44
3.1	Stages of project development	46
3.2	Project layout	52
3.3	Arduino Uno.	54
3.4	Raspberry Pi.	56
3.5	Dual level float charger.	63
3.6	Dual level float charger states.	63
3.7	Dual level float charger circuit diagram.	64
3.8	Traco Switching DC-DC converter.	66
3.9	Sensor node block diagram	67
3.10	Davis soil moisture sensor	68
3.11	Davis soil moisture sensor output resistance	69
3.12	Davis soil temperature sensor	69
3.13	Graph of resistance against temperature	70
3.14	Davis rain gauge	70
3.15	Pull up resistor configuration	71
3.16	Davis cup anemometer	72

3.17	Internal schematic of the cup anemometer and connection to controller . .	72
3.18	Graph of output voltage vs degrees	72
3.19	Davis air temperature sensor	73
3.20	Sensirion SHT1x temperature–humidity sensor	74
3.21	TSL257-LF light sensor	74
3.22	12 V 12 Ah Sealed Lead Acid Rechargeable Battery	75
3.23	Solar Panels.	76
4.1	Sensor node architecture.	77
4.2	Microcontroller transceiver interface.	78
4.3	Microcontroller XBee UART interface.	79
4.4	Sensor power control circuit.	79
4.5	Project functional blocks.	81
4.6	Sensor node circuitry	81
4.7	Ground station circuitry	81
4.8	Arduino IDE.	82
4.9	Round robin loop.	83
4.10	ADC block diagram.	83
4.11	ADC Flowchart.	84
4.12	Sensor node flow diagram.	85
4.13	Ground station flow diagram.	86
4.14	XBee pin–out.	87
4.15	Example data format [23].	87
4.16	XBee Internal data flow.	88
4.17	XBee star configuration	90
4.18	Mesh configuration	90
4.19	Power circuitry functional overview.	90
4.20	Example of iridium mail attachment.	91
4.21	Comma separated txt file.	91
4.22	Interface architecture.	95
4.23	Database architecture.	101
4.24	User information.	101
4.25	Sensor information.	102
4.26	Ground station layout.	102
4.27	Sensor name layout.	102
4.28	Date of measurement and state of the sensor.	103
4.29	Value and date of the sensor.	103
4.30	Mail processing architecture.	104
4.31	E-mail credentials.	104
4.32	Mail reception information.	105
4.33	Change password page.	106
4.34	Main page showing map and location of the GS.	106
4.35	GS, battery level and warning indicators.	107
4.36	Page showing available sensors and their respective battery levels.	108
4.37	Curve display dates selection page.	109
4.38	Dates selection layout.	109
4.39	Dates selection page.	109

4.40	Page to export data.	110
4.41	Example of a text file to export.	110
4.42	Min Max display.	111
4.43	Bar graph display.	111
4.44	Polar clock display.	112
4.45	Battery reservoir graph.	112
4.46	Station management.	113
4.47	Add sensor.	114
4.48	Delete sensor.	114
4.49	Administrator's page to add or delete a user.	115
4.50	Modify a sensor type.	115
4.51	Add a sensor type.	115
4.52	Import data feature.	116
4.53	Sample data file.	116
4.54	Export data feature.	116
5.1	Experimental setup	119
5.2	Snapshot of current consumption of a sensor node during transmission of data.	120
5.3	Sensor node power consumption breakdown	121
5.4	Node lifetime of a sensor node	122
5.5	Matlab representation of GS discharge curves	124
5.6	Discharge curves of sensor nodes and ground station	124
5.7	Representation of RSSI against distance for LOS and NLOS conditions	126
5.8	Throughput measurements results for 9600bps	127
5.9	Throughput measurements results for 19200bps	127
A.1	Senor node layout	132
A.2	Ground station layout	132
B.1	XBee star configuration.	133
B.2	XBee mesh configuration.	134
B.3	XCTU screenshots.	134
D.1	9602 Iridium modem.	138
D.2	9602 typical power profile [7].	139

List of Tables

2.1	Comparison of micro-controller memory and power consumption.	10
2.2	Comparison of radio power consumption	11
2.3	Comparison of commercial sensor nodes	12
2.4	Iridium characteristics	24
2.5	Iridium frequency bands	25
2.6	DC Power input specs [7]	25
2.7	Globalstar specs	27
2.8	Typical Orbcomm modem specs [9]	29
2.9	Typical Inmarsat modem specs [9]	30
2.10	Skywave DMR-200 modem specifications	30
2.11	Comparison of message-based satellite systems [34].	31
2.12	Wireless standard comparisons [11]	33
3.1	Arduino Uno specifications	54
3.2	Key parameters for Atmega328P	55
3.3	Raspberry Pi specifications	56
3.4	Beaglebone, Arduino Due and Raspberry Pi comparison	57
3.5	Sensor node load list	58
3.6	Sensor node load profile	59
3.7	GS load profile	60
3.8	Summary	62
3.9	Battery charger specifications	65
3.10	Traco Switching DC-DC converter specs	66
3.11	SHT1x pin specifications	74
3.12	SHT1x pin description	74
3.13	Battery specifications	75
4.1	XBee pin description.	88
5.1	SLA battery specs.	119
5.2	Node current profile.	120
5.3	GS current profile.	120
5.4	current measurements.	121
C.1	Project cost breakdown	136
D.1	9602 power profile [7]	138
D.2	Binary data commands	141
D.3	Text message commands	141

Abbreviations

ADC	A nalog to D igital C onveter
AODV	A d hoc O n- D emand D istance V ector
DC	D irect C urrent voltage
DSDV	D estination- S equenced D istance V ector
DSR	D ynamic S ource R outing
DSSS	D irect S equene S pread S pectrum
FFD	F ull F unction D evice
FSLs	F uzzy S ighted L ink S tate
FSR	F isheye S tate R outing
GPS	G lobal P ositioning S ystem
GS	G round S tation
GSM	G lobal S ystem for M obile C ommunications
ICASA	I ndependent C ommunications A uthority of S outh A frica
ISM	I ndustrial S cientific and M edical band
LOS	L ine of S ight
MAC	M edium A ccess C ontrol
MWSN	M obile W ireless S ensor N etwork
NLOS	N one L ine of S ight
OLSR	O ptimized L ink S tate R outing
OSI	O pen S ystem I nterconnection model
OSPF	O pen S hortest P ath F irst
PAN	P ersonal A rea N etwork
PHY	P hysical layer
RF	R adio F requency
RFD	R educed F unction D evice

RREP	R oute R eply M essage
RREQ	R oute R equest M essage
RU	R emote U nit
SBC	S ingle B oard C omputer
SLA	S ealed L ead A cid battery
SN	S ensor N ode
TBRPF	T opology B roadcast based on R everse P ath forwarding
TCP/IP	T ransmission C ontrol P rotocol / I nternet P rotocol
UART	U niversal A synchronous R eceiver T ransmitter
USB	U niversal S erial B us
VHF	V ery H igh F requency
WPAN	W ireless P ersonal A rea N etwork
WSN	W ireless S ensor N etworks
UHF	U ltra H igh F requency
ZRP	Z one R outing P rotocol

Chapter 1

Introduction

1.1 Background and Motivation

Real time environmental monitoring is the process of collecting environmental data of a particular area by deploying an automated station with different kinds of environmental sensors connected. Parameters such as humidity, rainfall, ambient light, air temperature and wind direction can be measured with those sensors deployed in the field. This thesis provides a comprehensive description of the various steps involved, which are data collection, data transfer and data reception. This work includes the data collection procedure, transferring of data from one data collection unit (sensor node) to the data receiving unit (GS) and finally transmitting all the combined collected data to a web server using a combination of wireless and satellite communication technologies, where once processed, it is made available for researchers on the master station (MS) via the internet.

This work presents an alternative method to the USB/MicroSD and GPRS/GSM solutions, which have been used before for transmitting data from the field to the user's web server. Innovative, low cost and low power Zigbee technology is employed to transmit the collected data from the sensor nodes in the field to a ground station, also in the field, to the web server via a satellite link. Since the new Zigbee technology replaces the GPRS/GSM network, Zigbee technology is a low power wireless technology which means it is suitable for long duration applications.

1.2 Problem Definition

Need exists for the Department of Forestry to obtain near real-time information on water resources in remote areas. By collecting and utilizing data over a period of time from these stations, correct decisions can be made to assist in planning and modeling.

Water is a very important and critical natural resource in most economic and industrial activities and at the same time it's under considerable pressure, hence the need to put in place monitoring mechanisms. Water resource monitoring consists of collection of data at specific points, in specific time intervals and the process provides a way to understand and predict the state of the water resources of a particular area.

The Forestry Department's monitoring program monitors long-term water quantity across specific target areas. To provide the data, a sensor network has been established to monitor soil drainage, soil leakage and soil moisture and other parameters. The data produced by this work intends to provide information and knowledge required to make management decisions based on water quantity levels, flood forecasting and forest fire detection. The advantage of having this system is being able to make decisions on the basis of the most current or real time data from remote areas.

Current data transfer mechanisms employed by the researchers include GSM/GPRS solutions. These conventional technologies rely on already existing telecommunications infrastructure which is not available in remote areas and thus, the researchers need an alternative data transfer mechanism which is reliable and could work in any remote area.

1.3 Project Background

Key aim of the Department of Forestry is to monitor the water resources of various remote areas by collecting and analyzing data of various environmental parameters over a period of time.

Previous solutions included setting up USB/Micro SD data loggers. The data would be collected manually after some time. Ideally, an automatic system is required that sends data automatically to the master station (MS) without human intervention. This will

help reduce the costs of site visits and would also guarantee that data is received in near real time.

1.4 Some existing solutions of radio telemetry

1.4.1 Direct Radio (VHF and UHF)

VHF transmissions make use of the 30 – 300MHz band of the radio spectrum and they provide short range LOS communications. This method is one of the most commonly used methods of transmitting data from remote locations and it requires an un-obscured LOS between the ground station and the master station. Range can be affected by antenna positioning and many terrain induced factors. Using radio telemetry might be costly in that erecting antennae towers and repeater sites would increase the overall cost of the system and also, since the area of interest is generally mountainous, maintaining a clear LOS from the field to the master station will be difficult.

1.4.2 Cellular (GSM/GPRS)

GSM telemetry uses a circuit switched technology that enables data to be transmitted over a GSM network at speeds of 9.6kbps [35]. The biggest disadvantage is that cellular telemetry requires existing telecommunications infrastructure and wide cell phone coverage over the particular area. Also, connections might be dropped during peak hours thereby affecting continuous operation with the remote unit.

1.4.3 Bluetooth

Bluetooth is a wireless technology standard for sending and receiving data over short distances [2]. It operates in the range of 2400 – 2485.5MHz in the unlicensed spectrum. Its low cost, low power parameters makes it ideal for telemetry applications but the disadvantage of using this type of technology is its short range (0 – 100m). This protocol is also limited to a few number of nodes in the network which makes it not ideal for a very large PAN.

From the above-mentioned disadvantages of existing telemetry options, it is thus desirable to create a dedicated, effective and less costly network which collects data from the sensors for display on a master station with minimum human interaction.

The sensor network and ground station comprise of various devices, which would be covered in the thesis. Its main functions are data collection, data storage and data transmission. The heart of the GS used in this work is a Raspberry Pi SBC.

Sensors are the other important components of the station. They measure environmental parameters such as air temperature, wind speed, wind direction, rainfall and soil moisture among others. The remaining components relate to power supply: battery, solar panel, lead acid battery charging IC. The sensor node, battery and charging IC are contained inside a weatherproof IP65 rated enclosure, while all the environmental sensors and solar panel remain outside of the enclosure.

In order to transmit the collected data to the master station, a Zigbee link is established using XBee-ZB radio modules, manufactured by Digi and operates at 2.4GHz. They have a built-in directional antenna, which radiates the signal in line of sight (LOS) direction. Multiple XBee modules were installed to form a wireless mesh network and one XBee device which acted as the network coordinator was installed at the ground station (data collection site). The ground station incorporates an Iridium modem which sends data to the Iridium GS via a constellation of satellites. Data is then sent for display from the Iridium GS to the MS as an email attachment through the internet.

1.5 Aim

The main goal of this project was to design, develop and deploy a weather monitoring system which collects data from the field for display on a centralized monitoring system. The aim of the project is to design a system which is economical and more efficient than the current system employed by researchers in this area.

1.6 Specific Objectives

The scope of this project entails the system level design and implementation of an environmental monitoring system which could be used in practice for general environmental data acquisition in remote areas not serviced by mobile or other terrestrial communication infrastructure. The main objective of developing this system is to provide a direct communication link between remote field sensors and the researcher's laboratories so as to obtain near real-time data and to reduce site visits. The hypothesis is that this is achievable by using a mixture of Zigbee protocols and off the shelf commercial hardware and software platforms. The communication will be based on a mix of terrestrial and satellite solutions.

The main areas of focus will be to:

1. Investigate and define a suitable network topology for the required system.
2. Identify a suitable satellite constellation/service as a store and forward for the data
3. Determine exact parameters to be monitored
4. Determining the design specifications and architecture of the wireless sensor network. This includes the following aspects:
 - (a) Selection of an ICASA approved frequency of operation.
 - (b) The necessary radio coverage required in that area.
 - (c) Suitable locations for the nodes in the network.
 - (d) The type of data to be transferred between nodes.
 - (e) The necessary bandwidth requirements and associated data rate.
 - (f) Selecting or developing appropriate communication protocols.
5. Design / obtain hardware for the various system functions. This includes the radios, modems and IP65 rated enclosures.
6. Develop (or adapt from existing) ground station software
7. Determine power supply requirements to meet design specifications and standalone operation.

8. Determine configuration of master station
9. Set up the master station
10. Build the prototype and deploy in field and test. This includes field testing and verifying that the deployed network meets the design specifications by measuring parameters such as the power consumption, data throughput. The goal is to demonstrate a fully functional ad-hoc network, created by deploying a number of the nodes designed as part of this project.
11. Adapt and modify if necessary and finalise design for say 5 stations.

1.7 Proposed Method

To achieve the objectives set out in the previous section, off the shelf environmental sensors will be networked to a wireless sensor network. Some of this equipment include a soil moisture sensor, soil temperature sensor, rain gauge, wind speed sensor and wind direction sensor. The data is collected by these sensors and sent to a ground station using a high data rate wireless sensor network which operates in the free 2.4GHz unlicensed band. The ground station collects data from the sensor network, stores the data into a text file which will be later transmitted to a master station using an Iridium satellite network. Finally, the data is sent for sorting and display on our master station using the internet via the Iridium network. The diagrammatic representation is shown in [fig 3.2](#).

1.8 Project Outcome

A satellite based water resource monitoring system was developed and its overall performance was examined continuously in an actual deployment situation. It is working well. The system includes 1 GS and 4 sensor nodes which were built using off the shelf and easy to use electronic prototyping platforms. A practical system was realised that could be deployed as required anywhere in any remote area of interest.

1.9 Thesis Organisation

This thesis is arranged in 6 chapters and organized as follows: Chapter 2 conducts a literature review of the current trends in wireless sensor networks, Zigbee technology and the chapter takes a look at the available satellite communications systems. The chapter starts by defining WSN and gives an insight into the possible application areas.

Chapter 3 discusses the methodology. The chapter explores the steps taken to develop the project and it begins by giving an overview of the hardware used and describes different software packages used. Topics include comparison of sensor hardware platforms used and the explanation of their selection criteria is given. The chapter will also give a brief overview of the sensors and the radios used.

Chapter 4 explains the implementation of the project. Basically, this chapter describes how the various hardware and software blocks were integrated to form one working unit.

Chapter 5 presents the results of the project. Power tests, node lifetime, signal strength tests and evaluation of the performance of the sensor nodes and the GS was recorded in this chapter.

Chapter 6 summarises the project and gives an insight into areas that can be improved to enhance the performance of the project in the future.

Chapter 2

Literature Study

This chapter explores the available literature in the fields of WSN, satellite communication systems and Zigbee technology. The literature covers a wide variety of subjects but particular emphasis will be given to aspects which are of paramount importance to our project. Examples include, selection of hardware used, cost of components, power requirements, selection or development of appropriate communication protocols.

2.1 Introduction

A WSN is a network which uses radio waves for data transmission and it consists of sensor nodes which monitor the physical environment and relays their information to the gateway (or coordinator). Each sensor node is typically made up of: micro-controller, radio transceiver, power source and one or more sensors, fig 2.1. The network topologies can be simple star networks or more complex mesh networks which employ different routing protocols to transfer data. Power consumption, scalability and throughput are some of the important factors to consider when designing a WSN. Choosing the right sensors and radio communications equipment requires a thorough knowledge of the problem and application area. Sensors deployed in the environment or integrated into systems combined with efficient transfer and delivery of sensory information could provide great help to researchers. The ideal sensor node has the components described in the next section.

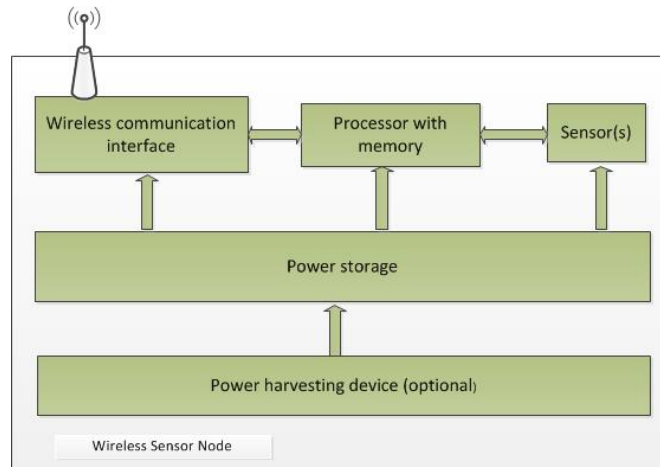


FIGURE 2.1: WSN Architecture.

2.2 Characteristics of sensor nodes: requirements

2.2.1 Computation power:

A controller performs tasks, for example; checking values of analog pins or checking states of digital pins. This module performs data processing functions and controls other components in the sensor node. Among various controllers available on the market today, a micro-controller is often used in many wireless sensor applications. This choice is fueled by the low cost, ease of programming and low power consumption, among other factors. Digital signal processors (DSPs) such as the Texas Instruments C6000 or the Freescale MSC81xx offer better signal processing capabilities than micro-controllers but due to the less complex signal processing required by the project at hand, superior processing capabilities of DSPs and FPGAs is not very important. Our choice of a micro-controller should meet the computational and power requirements of our application.

2.2.2 Power source:

Power is needed for computation, data processing and data transmission. Depending on the power requirements of the sensor node, small lithium-ion batteries can be used to provide power. To eliminate the cost of batteries, rechargeable battery systems and solar panels are now widely in use to provide power in many stand alone systems. New frontiers of research are investigating the possibilities of energy harvesting techniques as alternative to the more bulky and expensive solar panels. This comes in handy when

the sensor node can be used for military purposes where the overall size of the node is of great concern or when the node is deployed in an area which does not receive adequate sunshine.

Power reduction has always been a cause of concern in designing wireless sensor networks, research is now focusing on developing power aware communication protocols. The sensor nodes in this project will be deployed in a remote area where servicing a node may be difficult. Thus, the lifetime of a sensor node greatly depends on the battery capacity and hence our source of power should be more than sufficient to power the sensor node for at least 3 days of autonomy in worst winter cases so as to ensure reliability.

2.2.3 Memory:

Memory also plays a significant role in selecting the processor for the sensor node. Choosing a processor with a small memory may mean that information such as routing tables, application related data and the program code may not fit on the on-chip memory or flash memory thereby requiring external memory. The most commonly used on-chip memory is Flash and EEPROM memory. If, in case of a data logging application, an external memory is required but this also translates to an increase in power consumption of the node.

Table 2.1 compares the power consumption and memories of different microcontroller families available on the market.

TABLE 2.1: Comparison of micro-controller memory and power consumption.

	Power consumption (mA)	Flash (Kbytes)	EEPROM (bytes)	IO pins
Atmega328p	0.2 @ 1MHz	32	1024	23
Atxmega256a3		256	4096	50
MSP430	1.8	48	1024	
PIC18F4620	16			36
Atmega128l	6.0	128	4096	53

2.2.4 Transceiver:

To make the sensor nodes wireless, a transceiver is required. This will enable the sensor nodes to connect and communicate with one another through a wireless medium thus

forming a network. Most transceivers are configured to operate in three modes; receive, transmit and sleep mode. Among other options such as optical and infra-red, radio frequency based communication is the most possible choice as the sensor nodes make use of the license free 2.4GHz ISM band and they do not usually require line of sight operation.

Table 2.2 compares various parameters of the most commonly used transceivers.

TABLE 2.2: Comparison of radio power consumption

Radio	Rx current (mA)	Tx current (mA)	Output power (dBm) / (frequency band)
XBee	40	40	3 / 2.4GHz
XBee Pro - S2	45	295	18 / 2.4GHz
AT86RF230	16.0	17.0	3 / 2.4GHz
AT86RF212	9.0	18.0	10 / 700/800/900 MHz
AT86RF233	6.0	13.8	53
MC13192	42	35	
CC2420	18.8	17.4	4 / 2.4GHz

2.2.5 Sensing element:

A sensing element is a hardware device that measures or detects a physical condition and produces a measurable value. They measure environmental parameters such as air temperature, humidity, wind speed and wind direction among others. The analog or digital signals they produce is manipulated by the ADC of the micro-controller and the resulting digital signal is calibrated by a calibration equation and a meaningful value of the environmental parameter is produced and transmitted for storage or display.

2.3 Random deployments and self-organization:

In some cases, sensor nodes can be deployed in harsh environments or they can be dropped randomly from a moving vehicle, they should be able to coordinate and assemble the network themselves and be able to forward data to the destination flawlessly without the need of a human operator. In case of one node dying or degradation, sensor nodes should be able to dynamically adapt, search for other nodes within range and continue

TABLE 2.3: Comparison of commercial sensor nodes

Sensor node	Idle current (mA)	Operational (mA)	Data rate	Range (m)	Notes
TeleosB	21 idle	23	250	70-100	Onboard light, temp and humidity sensors Tiny OS support
	1u sleep			20-30	IEEE 802.15.4 compatible
Sensnode		Tx - 18.4mA Rx - 20.7mA	250	100	
Waspote	55uA	15	250	500	IEEE 802.15.4 compatible Solar panel interface Longer range
Crossbow	15	Rx - 19.7	250	100m	Onboard light, temp and acceleration sensors

to send data. Moreover, new nodes may need to join the network so the network must be able to accommodate additional nodes and continue functioning.

2.4 Primary functions

The primary function of a WSN is to sample the environment for sensory information. A WSN is made up of nodes, where each node is connected to one or more sensing elements which in turn monitor and sample physical environmental conditions such as humidity. Each node is able to transmit its data to a centralized location for storage or further processing and display.

2.5 Application of wireless sensor networks

Today, WSN find many application areas notably:

2.5.1 Environment and wildlife monitoring:

WSNs can be effectively used to monitor environmental parameters such as temperature, rainfall and wind speed among others. They can monitor and record the occurrence of veld fires, monitor carbon dioxide levels and also monitor the movement of wildlife.

This can be important to researchers who are interested in the breeding, feeding or even protection of certain wildlife.

2.5.2 Water supply and sewer system monitoring:

WSNs can be effectively used to detect leaks in a water supply system, hence, early detection can help prevent loss of the precious liquid and avoid land or water pollution in case a sewer system leaks or bursts.

2.5.3 Industrial control and manufacturing:

Temperature, pressure, position, flow rate are among the most measured parameters in instrumentation and industrial control. Sensors for this application are readily available on the market. It is much more convenient to interface these sensors with a wireless network.

2.5.4 Ocean monitoring:

Over the years, oceanic levels have always been of concern to oceanic researchers. Wireless sensor networks can detect any rise or fall in ocean water level and the information can be used to predict flooding or drought. Sensors can also detect any form of oceanic pollution thereby saving aquatic life if evasive decisions are taken.

2.5.5 Agriculture:

Wireless sensor networks within agriculture enable farmers to do precise monitoring of the crops and to know the status which will enable them to take correct decisions. Wireless sensor networks can detect in real time, soil temperature, air temperature and soil moisture among others. This will aid in taking correct irrigation measures. Pests are also an issue and some wireless sensor networks can detect pests and farmers can take decisive measures towards the eradication of these pests and the protection of crops.

2.5.6 Disaster control

Natural disasters occur and wireless sensor networks can prevent fatalities on human lives. They can be deployed in forests where changes of air temperature, CO₂ levels and smoke can be monitored in real time and this helps to detect wild and uncontrolled fires.

2.5.7 Structural health monitoring:

Civil structures such as bridges and tall buildings are in constant danger of wear and tear. Wireless sensor networks can detect flaws such as cracks and help alleviate the dangers they cause.

2.5.8 Some applications of WSN

- Retail
- Domestic and home automation
- Smart metering
- E-health
- Smart parking

2.6 WSN Requirements

2.6.1 Scalability

Scalability is the ability of a system, network, or process to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth [3]. In other words, this refers to the capability of a system to increase the average throughput and continue performing well when more nodes are introduced to the network.

2.6.2 Adaptability

Communication schemes need to adapt to dynamic topologies and fluctuating traffic rates [4]. The network should be able to keep up with the disappearance of dead nodes or the introduction of additional nodes.

2.6.3 Efficiency

One of the main design challenges for a WSN is power. This network runs on battery power and there is danger if incorrectly sized, batteries may quickly die before reaching the three day autonomy target. [2] suggests that a typical method for designing a low-power wireless sensor network is to reduce the duty cycle of each node. The disadvantage is that as the sensor node stays longer in sleep mode to save power, there is less chance that the node can take part in data routing if the network is configured in an ad-hoc set up. The other drawback, some applications require the sensing element of a sensor node to be in an always on state, for example, measuring the amount of rainfall, checking GPS time or measuring wind speed.

2.6.4 Security

Nodes in a WSN are vulnerable to a variety of attacks including node capture, physical tampering, and denial of service [6]. The WSN should be able to guarantee data confidentiality and integrity if security is a key concern. Designers might consider installing the sensor nodes in a fenced area, this stops animals interfering with the sensor node's enclosure.

2.6.5 Reliability

Sensor networks are dynamic [31]; they are characterized by constant change. By being reliable, the sensor network should guarantee reliable data transmission even when the structure of the network changes. A smart network should attempt to balance scalability and reliability. As the network increases in size with more additional sensor nodes, more control packets will be needed for routing information and data transfer. At some point, this might overwhelm the network which will lead to less reliability of data transmission.

2.6.6 Mobility

This refers to the capability of a sensor network to manage movable nodes. This feature greatly improves the functionality of WSN but also brings its own challenges as well. If nodes are mobile, they can rearrange themselves to increase range of the sense area. Example application is in autonomous robots and unmanned vehicles. However, mobile nodes should overcome hardware challenges in the form of a power source. The power should be sufficient enough for motion and normal sense applications. Mobility affects responsiveness [3], and hence the MWSN should have a high responsiveness.

2.6.7 Responsiveness

Responsiveness of a WSN refers to the ability of the network to quickly adapt itself to changes in the physical arrangement of nodes [4]. The network should manage movable nodes, additional nodes or dead nodes. Neighbouring nodes would need to issue control packets regularly so as to update information like routing tables.

2.6.8 Throughput, bandwidth

Throughput is the average rate of successful message delivery over a communication channel [2]. This is usually expressed in bps and denotes the number of bits arriving at the receiver in a second. Throughput can sometimes be referred to digital bandwidth consumption and it is affected by various parameters which would be covered later in this thesis.

2.7 Protocol stack

The communication system of devices is broken down into layers with the purpose of making networking technologies consistent. Each layer provides services to the upper layers and requests for services from the layer below. This consists of seven logical layers and make up the OSI model. On the contrary, the TCP/IP model is a simplified version of the OSI model and consists of four logical layers.



FIGURE 2.2: OSI and Internet model comparison.

The WSN architecture is built based on the OSI model. [1] states that five layers are needed: application layer, transport layer, network layer, datalink layer and physical layer. Three cross layers are also added to the five layers as shown in fig 2.3.

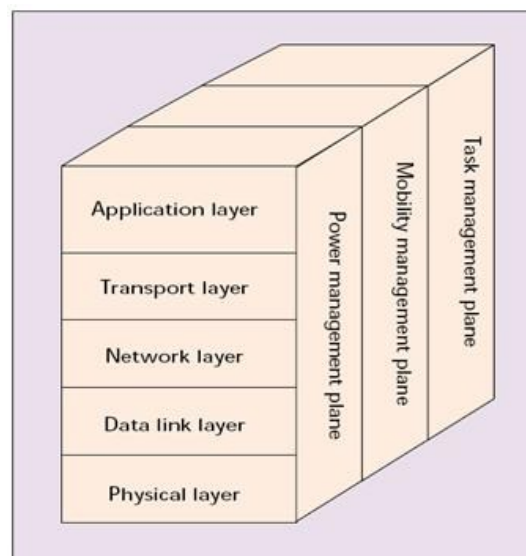


FIGURE 2.3: WSN cross layers.

The three cross layers consists of the power management plane, mobility management plane and task management plane. These layers are used to manage the network and work together to increase the overall efficiency of the network [1].

- Task management plane, the purpose of this plane is to schedule and control the processing of information in the node

- Mobility plane; this plane comes in handy in cases of mobile nodes. The mobility management plane ensures the availability of routes through neighboring nodes.
- Power management plane. This plane is tasked with monitoring the power consumption of the network at each horizontal and vertical layer [1].

2.7.1 Application layer

The role of the application layer is to interface with the user software. This layer also abstracts the physical topology of the WSN for the applications. Moreover, the application layer provides necessary interfaces to the user to interact with the physical world through the WSN [2]. The three most common application layer protocols are:

1. sensor management protocol (SMP): this is a management protocol that provides the software operations needed to perform some administrative tasks for example:
 - setting the rules for data aggregation
 - time synchronization
 - moving sensor nodes [2].
2. Task assignment and data advertisement protocol (TADAP): this is a protocol that allows users to poll for data from the whole network or to trigger an event. Another approach is when a sensor node advertises the availability of data to the users [2].
3. Sensor query and data dissemination protocol (SQDDP): SQDDP provides user applications with interfaces to issue queries, respond to queries and collect incoming replies [2].

2.7.2 Presentation layer

The layer provides a standard interface to data for the application layer [2]. Data encoding, encryption and conversion are handled by the presentation layer.

2.7.3 Session layer

This layer is responsible for establishing, managing and terminating connections within applications [2].

2.7.4 Transport layer

This layer was designed to provide reliability and congestion avoidance, it also comes in handy when we need to connect the system to the internet or other external networks, it also provides reliable and sequential packet delivery through error recovery and flow control mechanisms [2]. Protocols in this layer include:

- STCP (sensor transmission control protocol) : provides reliability, congestion detection and avoidance [2].
- PORT (price oriented reliable transport protocol): this is a downstream protocol which assures that the sink receives enough information from the physical phenomena [2].

2.7.5 Network layer

The main function of the network layer is data routing, this layer will determine the path the data will take. The layer routes packets from source to a sink according to unit network device addresses through one or more relays.

In a simple approach, a direct communication model is followed, fig 2.4, where each source is directly connected to the sink and information is transmitted over a single hop to reach the destination. This model is limited by range and a more flexible multi-hop routing model, as shown in fig 2.5 is chosen. In this scenario, the challenge of the network layer of each node in the network is to determine a path the data will traverse from the source to the destination across multiple nodes configured as routers. This is implemented through routing protocols.

[3] found out that the design of a routing protocol is challenging due to the unique characteristics of WSNs, including resource scarcity or the unreliability of the wireless medium. For example, the limited processing, storage, bandwidth, and energy capacities

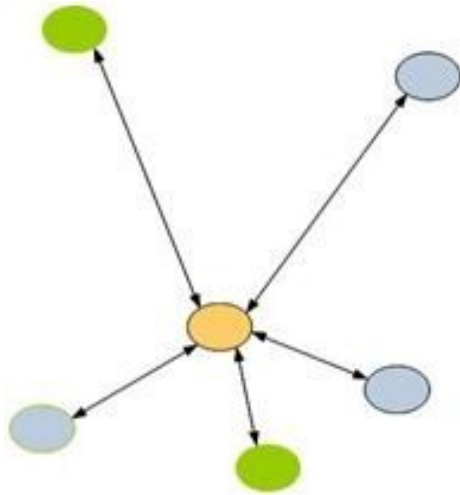


FIGURE 2.4: Single hop routing model.

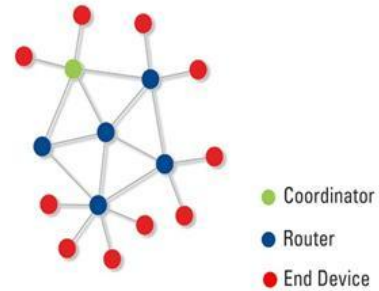


FIGURE 2.5: Multi hop routing model.

require routing solutions that are energy efficient and lightweight in memory. These challenges require the selection/design of routing solutions that are flexible and lightweight. Routing protocols can be classified into 3 different categories based on the route discovery mechanisms, network organisation and protocol operation. This overview will look at routing protocols which are based on route discovery.

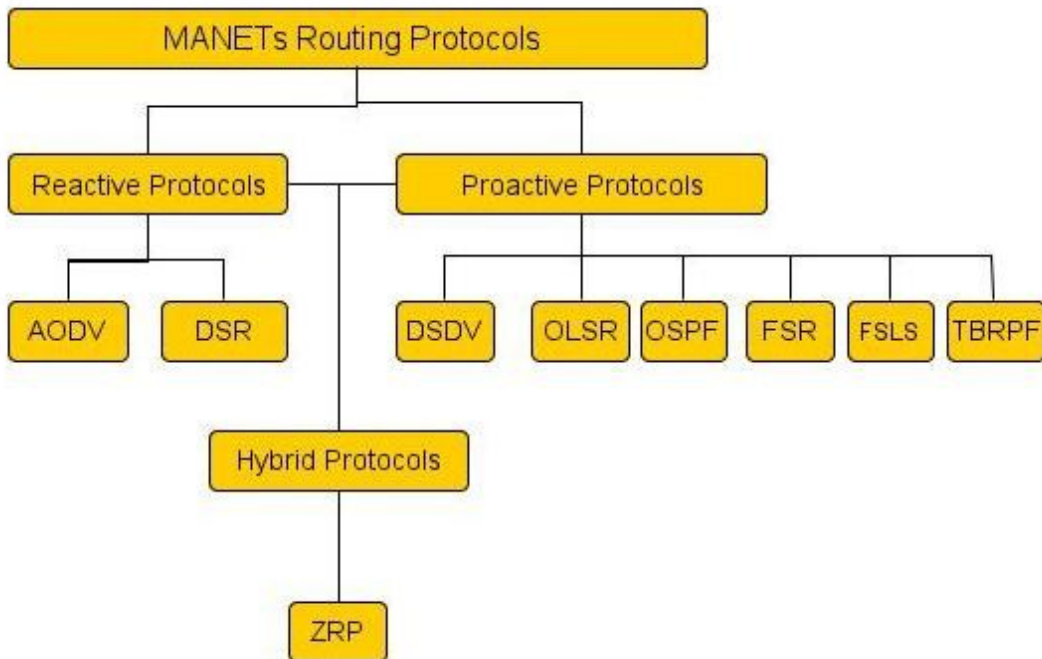


FIGURE 2.6: Categories of routing protocols.

Reactive protocols discover routes on-demand [3]. This means that each time a source

node intends to transmit information to a receiving node and does not have a route established, it discovers its own route when data traverses through the network. Examples include:

- Ad-hoc on demand distance vector (AODV)
- Dynamic source routing (DSR)

Proactive routing protocols establish routes before they are actually needed [3]. These protocols are sometimes referred to as table driven because routes are selected depending on the contents of a routing table that contains information of the destination and the next hop neighbor. The advantage of proactive routing protocols is that they eliminate route discovery delays. Examples include:

- Destination sequenced distance vector (DSDV)
- Optimized link state routing (OLSR)

In hybrid routing protocols, this is where protocols which exhibit both reactive and proactive properties belong. The ZRP protocol falls under this category.

2.7.6 Datalink layer

The datalink layer is responsible for the medium access and error control. It ensures reliable point to point and point to multipoint connections in a communication medium [3]. The datalink layer frames the packets and it defines procedures for operating the communication links.

2.7.7 Mac layer

Refers to the sub layer that provides addressing, framing, flow control and it regulates access to the wireless media. It is responsible for allocating the communication resources between sensor nodes.

2.7.7.1 Addressing mechanism

A MAC address is a local network address used in IEEE802.xx networks [3]. A MAC address is different in all network devices and each device is uniquely identified by this address.

2.7.7.2 Channel access control mechanism

The channel access control mechanisms provided by the MAC layer are also known as a multiple access protocol. This allows multiple wireless devices connected to the same physical medium to share it. If a packet mode contention based channel access method is used, the multiple access protocol may also be used to detect or avoid data packet collisions Existing MAC protocols can be categorized by the way they control access to the medium [3]. Fig 2.6 shows an example. Most MAC protocols are categorised as contention free or contention based protocols. Contention free protocols are further split into fixed assignment and dynamic assignment protocols. Examples include TDMA, FDMA and CDMA.

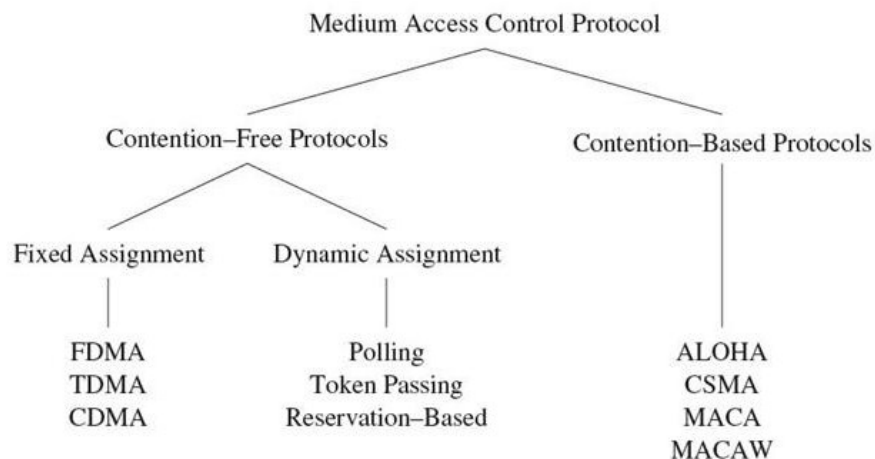


FIGURE 2.7: Categories of MAC protocols.

2.7.8 Physical layer

The physical layer defines the relationship between the device and the physical media. This layer is responsible for frequency selection and it provides an electrical interface to

the transmission medium [3]. Alternatively, all data is transmitted as bits rather than data packets between devices connected to the same physical medium.

- it uses the available IEEE 802.xx; Bluetooth, Zigbee and Wifi.
- radio frequency: ISM band 433MHz to 2.4GHz
- modulation: phase shift keying BPSK or MPSK
- data rate: 0.25Mbps to 54 Mbps
- range upto 500m [3].

2.8 Sensor nets operating systems:

2.8.1 TinyOS

TinyOS is a flexible, open-source, event based operating environment designed for low-power wireless devices such a sensor networks, ubiquitous computing, personal area networks [28].

2.8.2 Contiki

Contiki is an open source operating system for networked, memory-constrained systems with a particular focus on low-power wireless internet of things devices [29].

2.8.3 Nano-RK

Nano-RK is a real-time operating system (RTOS) from Carnegie Mellon University designed to run on micro-controllers for use in sensor networks [30].

2.9 Basics of satellite communication systems

This section takes a look at some of the available satellite constellations; we review each constellation against common criteria in order to determine the feasibility. We study Iridium, Inmarsat, Globalstar and Orbcomm communication systems.

2.10 Iridium satellite system

Iridium is a large constellation with 66 interlinked active satellites in orbit and additional 11 spare satellites. Satellites are in Low Earth Orbit (LEO) at an inclination of 86.4° and about 780Km from the earth [7]. Its signal covers the whole earth and the system enables data and voice communications all over the world. The SBD service provides a platform which quickly transfers data of a few bytes at a time (not more than 340 bytes). Typically, the received data is in the form of an attachment in a common email. The system is bidirectional and messages can be sent to the modem from the MS. The cost is currently 1.50 USD per kbyte, plus a monthly fee [36].

2.10.1 Iridium communication system.

Iridium satellite systems must have a clear view of the sky in order to have a successful SBD session or to make a call. Data transfer is accomplished by using TDMA and FDMA channel access methods in the L band spectrum between 1616MHz and 1626.5MHz. FDMA allows the Iridium system to reuse frequency bands between beams and TDMA allows users to send and receive data over the system. Modulation is Quadrature phase shift keying (QPSK) [36].

TABLE 2.4: Iridium characteristics

Frequency bands	L band – 1616MHz to 1626.5MHz Ka band – 23.18 to 23.38GHz(inter satellite links)
FDMA:	number of channels : 20 channel bandwidth : 41.6kHz
TDMA:	frame length : 90ms user channels : 8

TABLE 2.5: Iridium frequency bands

K-band down-link	18.8-20.2GHz
K-band up-link	27.5-30.0GHz
Crosslink	22.55-23.55GHz

TABLE 2.6: DC Power input specs [7]

Idle current (average)	45mA
Idle current (peak)	190mA
Transmit current (peak)	1.5mA
Transmit current (average)	195mA

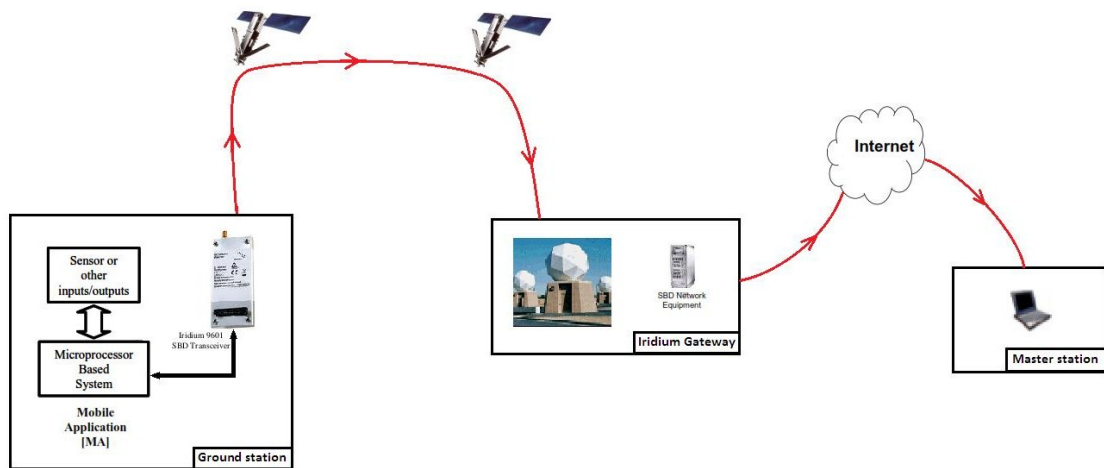


FIGURE 2.8: Iridium communication architecture [7].

2.10.2 Power supply design consideration:

[7] specifies that:

- The supply voltage drop over for a 8.3ms burst of 1.5A current should not be more than 0.2 Volts.
- The power supply should limit the in-rush current to 4 Amps maximum
- The power source shall provide for over current protection in case of device malfunction.
- The supply noise should be less than the limits in the following profile:
 - 100 mVpp from 0 to 50 kHz
 - 5 mVpp at 1 MHz measured in 50 kHz bandwidth
 - 10 mVpp at 1 MHz measured in 1MHz bandwidth
 - 5 mVpp above 5 MHz measured in 1 MHz bandwidth.

2.10.3 Hardware



FIGURE 2.9: Iridium 9602 modem.

Figure 2.9 shows an Iridium 9602 modem, they use AT commands, described in Appendix A1 for serial data communication. They operate in the L-band providing a data rate of 2.4 kbps and a link margin of 16 dB [7]. An external patch antenna is connected to the modem using a TNC connector. The modem also contains an integrated GPS port for easy interface with the GPS receiver.

2.10.4 Types of data services

The data services offered by Iridium can be grouped into two types, the Iridium-to-Iridium (or mobile-to-mobile) service and Iridium-to-PSTN service. The Iridium-to-PSTN service can again be further divided into dial-up data and direct Internet service [36]. The modem also offers SBD message services. These refer to the messages sent from the modem to the Iridium gateway and they are limited to only 340 bytes.

2.11 Globalstar

Globalstar is a satellite constellation service that utilizes low earth orbit. It has 48 active satellites in orbit. Through the GSP-1600, Globalstar offers efficient, reliable 9600 bps voice and data services [8].

2.11.1 Operating modes

- Globalstar CDMA satellite

- cellular digital CDMA (800 MHz)
- cellular analog AMPS (800 MHz)
- RF power output: 400mW(+26dBm) max
- transmit frequency: 1610.73 to 1625.49MHz

TABLE 2.7: Globalstar specs

Maximum transmit power	31dBm
Power consumption	Transmit 3.65w Standby 0.5w Shutdown mode 3.5mW
Receive frequency	2484.39 to 2499.15MHz
Cellular transmit	824.01 to 848.97MHz
Cellular receive	869.01 to 893.97MHz

A wide range of Globalstar services include:

- voice
- voice mail services
- data services
- short message services

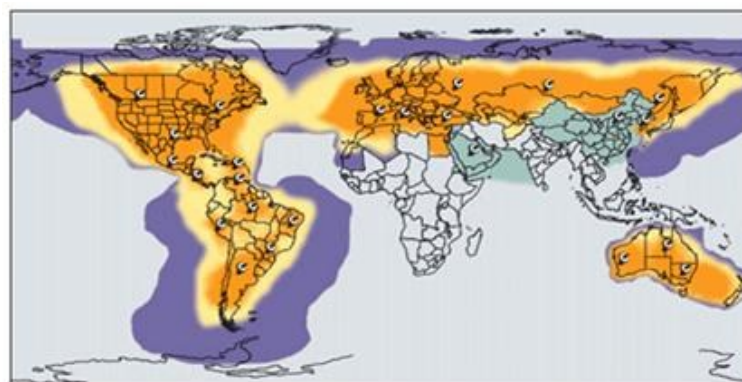


FIGURE 2.10: Globalstar coverage map.

From fig 2.10 above, it is clear from the coverage map that the system does not cover sub-Saharan Africa, nor much of South East Asia.

2.12 Orbcomm Satellite system

The Orbcomm system is a wide area, packet switched, two way data communication system [9]. Satellites are in Low Earth Orbit and they enable messages to be sent from remote units to the Orbcomm gateway.

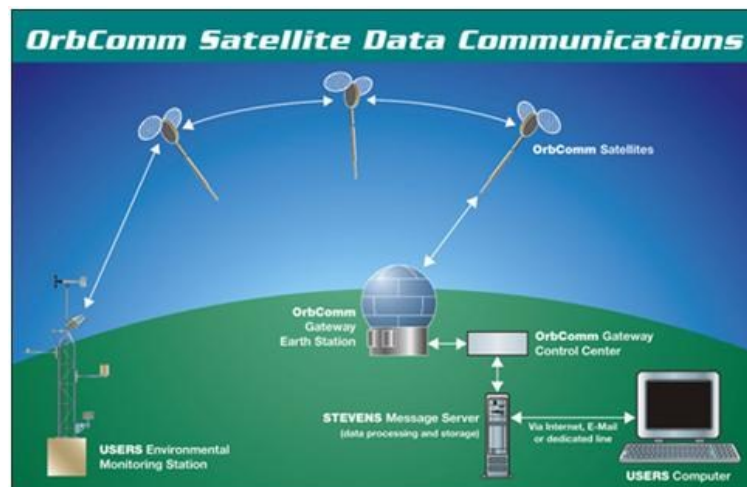


FIGURE 2.11: Orbcomm system overview.

Orbcomm operates within the 137 and 150 MHz frequency bands. The Orbcomm satellite modem can steadily transmit data packets at 4800bps, the down-link is capable of transmitting at 9600bps and the overall system is capable of providing low bit rate communications service around the world.

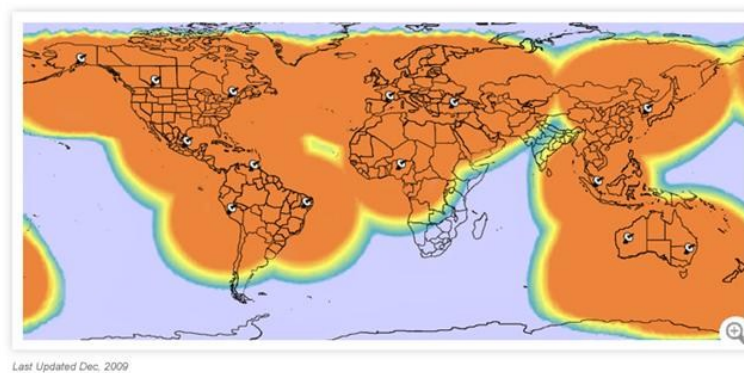


FIGURE 2.12: Orbcomm coverage map.

TABLE 2.8: Typical Orbcomm modem specs [9]

Transmit power	5W
Receive dynamic range	-116 to -80dbm
Transmit current	2A
Receive current	100mA
Sleep current	<1mA
Antenna type	wave (1m)whip
transmit frequency	148 150MHz
data rate	4800bps transmit 2400 receive

2.13 Inmarsat

International Marine Satellite constellation was meant for providing communications to ships at sea. It now also provides land based communications. It is a constellation that utilizes GEO with 10 active satellites in orbit. The Inmarsat system uses Time Division Multiplexed channels where each channel is transmitted on a unique frequency.

Thrane M4 data modem provides both packet and switched data services [9]:

Rx freq band : 1525.0 – 1559.0 MHz

TX freq band : 1626.5 – 1660.5MHz

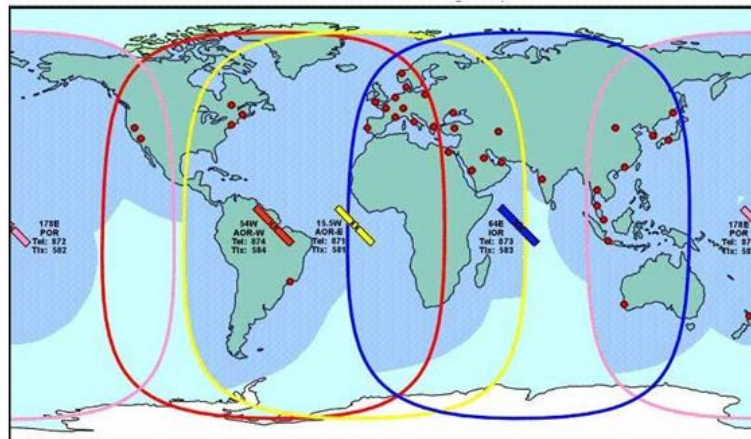


FIGURE 2.13: Inmarsat coverage map.

The study of currently available satellite systems conducted in table 2.11 and fig 2.14 concludes that for small amounts of data, Iridium's Short Burst Data (SBD) service outperforms other systems – SBD where data volume, power consumption, start-up time,

TABLE 2.9: Typical Inmarsat modem specs [9]

Frequency range	
Receive	1525 - 1559MHz
Transmit	1626.5 - 1660.5MHz
EIRP	0.9dBW
Elevation Angle Range	15.90o
Modulation	
Forward Channel	32-ary FSK, 20 bps
Reverse Channel	binary FSK, 4-128 bps
Forward Error Correction	Forward Channel Reed-Solomon
Message Length	
Forward Channel	64 bits(burst)
Reverse Channel	200 digits (max), or 133 characters (max)

TABLE 2.10: Skywave DMR-200 modem specifications

Input voltage	9 VDC to 30 VDC
Power Consumption (@ 12 VDC)	
Receive	10W
Idle	0.25W
Transmit	10W
Sleep	6W

System	Message size	Airtime cost	Monthly price (1 message per day)	Terminal power consumption (during transmission)	Two way communication	Data rate	Time to send one message	Delivery time	Monthly price (1 message per hour)
ARGOS	32 bytes	\$21/month + 3.5/6hr slot	\$124	< 1W	No	4800bps	~1s	Up to 2hrs	\$437
Iridium SBD	< 340 bytes	\$12 /month + \$0.0018 /byte	\$14.24 (30 bytes)	1.8W	Yes	2400bps	~1s	< 20s	\$31.48 (30 bytes, bulk tariff)
Orbcomm	< 200 bytes	Unlimited for \$60 / month	\$60	24W	Yes	2400bps	~1s	Up to 6hr	\$60
Globalstar Simplex	< 36 bytes	\$30 /month for 100 9-byte messages	\$30 (9 bytes)	5W	No	100bps	< 1s for 9 bytes, <3s for 36 bytes	< 30mins	\$165 (36 bytes, bulk tariff)

FIGURE 2.14: Comparison between Satellite Systems.

Its two-way, global coverage and the cost are important factors to consider. Iridium is able to offer more channels in a short space of time and it is practical to setup the Iridium hardware in areas where no telephone infrastructure exists.

Iridium goes beyond the performance of other available systems in that its message transfer is almost instant, < 20s, which makes this system favorable for real time monitoring systems and it also carries larger volumes of data compared to other systems.

As shown in Fig 2.10, Globalstar offers a wider coverage but unfortunately it fails to

TABLE 2.11: Comparison of message-based satellite systems [34].

System	Use	Type	Data size
Goes, Meteosat	Messaging	GEO	<5Kb/day
Argos	Messaging	LEO	<5Kb/day
Orbcomm	Messaging	LEO	<50Kb/day
Iridium	Voice, Data and Messaging	Big LEO	1MB/hr
Globalstar	Voice	Big LEO	1MB/hr
VSAT	Internet	Big LEO	10-30MB/hr
Inmarsat BGAN	Internet	GEO	50MB/hr

cover the bulk of Southern Africa which, unfortunately, is the study area in mind. OR-BCOMM's coverage is not continuous and also carries small volumes of data and hence for this project, Iridium is recommended.

2.14 Zigbee technology

In this section, we shall discuss four segments which cover Zigbee:

- IEEE 802.15.4 task group
- Zigbee technology
- Protocol stack
- Network topology

2.14.1 IEEE 802.15

This refers to the 15th working group of the 802 project and deals with Wireless PAN. Task group 1 is Bluetooth, group 2 defines coexistence with 802.11. Group 3 is high rate WPAN and group 4 is Low rate WPAN with long battery life. Zigbee protocols are based on 802.15.4 task group. 802.15.4 uses DSSS modulation to spread packets into symbols and reassemble them on the other end [16] and it employs CSMA/CA transmission protocol for media access.

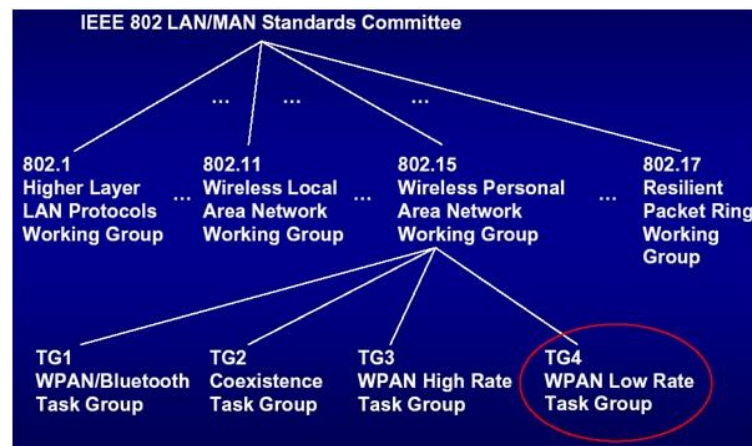


FIGURE 2.15: IEEE WPAN.

From table 2.12, we can summarise the general characteristics of 802.15.4 as a protocol designed for wireless PANs. Wireless networks built on the 802.15.4 protocol may exhibit the following characteristics:

- Data rates of 250 kbps, 20 kbps and 40kbps.

TABLE 2.12: Wireless standard comparisons [11]

	802.15.4	Bluetooth 802.15.1	Wi-Fi 802.11b 802.11b	GPRS/GSM 1xRTT/CDMA
Application focus	Many	Cable replacement	Web, video, email	WAN, voice/data
System resource	4Kb-32Kb(64KB)	250Kb+	1MB+	16MB+
Battery life (days)	100 - 1000+	1 - 7	1 - 5	1 - 7
Nodes per networks	255-65K+	7	30	1
Bandwidth	20-250	720	10000+	64 - 128+
Range (m)	1 - 250+	1 - 10+	1 - 100	1000+
Key Market attributes	Low data rates Low power, Low costs	cost Convenience, High Qos	speed, flexibility	

- Star or Peer-to-Peer operation.
- Support for low latency devices.
- Channel access is via CSMA/CA.
- Dynamic device addressing.
- Low power consumption with long battery life for remote monitoring and portable electronics
- 16 channels in the 2.4GHz band, 10 channels in the 915MHz band and one channel in the 868MHz band.

2.14.2 802.15.4 PHY

This is the first layer of the OSI reference model. This layer provides access to the physical transmission medium which is the RF device in this case. The purposes of this layer include:

- Activation and deactivation of the radio transceiver
- Energy detection within the current channel
- Link quality indication for received packets
- Clear channel assessment for CSMA/CA

- Channel frequency selection
- Data transmission and reception [16]

Channels are a portion of the RF spectrum [16]. In the 2.4GHz band, each of the channels is separated by 5MHz. The 802.15.4 radio is half duplex [16] and it can access one channel at a time, so a device communicating on a different channel will not interfere with other channels of the 2.4GHz band.

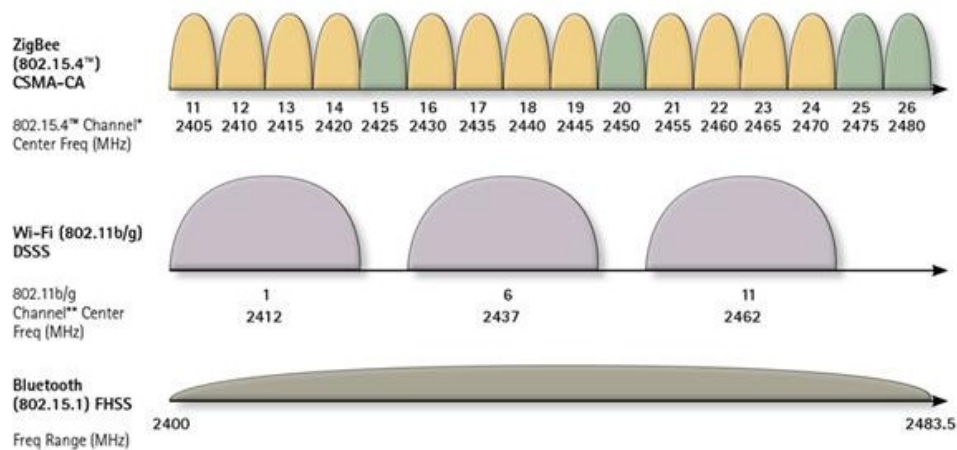


FIGURE 2.16: 802.15.4 operating frequency bands [15].

From fig 2.16 above, it is clear that the RF spectrums and available channels for 802.15.4 and 802.11b/g overlap, to avoid interference, it is always recommended to select the channels which utilise the free space between neighbouring Wi-fi channels.

2.14.3 802.15.4 Device types

An IEEE 802.15.4 WPAN is comprised of a PAN coordinator and one or more full function or reduced function devices. The coordinator is the primary controller of the network. It is responsible for setting up and maintaining the network and it should be connected to a reliable, uninterrupted power source. Routers or full function devices take part in data routing and they can be configured to allow other devices to join the network. On the other hand, end devices or reduced function devices are limited in operation in that they cannot take part in creating routes and they cannot accept join requests from other nodes, they can only send data to the devices which they connect to.

The topologies supported by this standard are peer to peer, star and mesh, fig 2.18 and described in chapter 2.16 as well.

2.14.4 802.15.4 MAC

The MAC layer is responsible for providing reliable communications between a node and its immediate neighbours. This helps to avoid collisions and improves the overall efficiency. The MAC Layer is also responsible for assembling and decomposing data packets and frames [15].

2.15 Zigbee protocol

The ZigBee protocol was developed to provide low-power, wireless connectivity for a wide range of network applications concerned with monitoring and control. ZigBee is a worldwide open standard controlled by the Zigbee Alliance. Zigbee PRO is an enhancement of the original Zigbee protocol, providing a number of extra features that are particularly useful for very large networks [11].

The ZigBee standard is built on the established 802.15.4 protocol and it greatly improves its performance by providing adaptable and versatile network topologies which are easy to setup.

Zigbee technology uses the ISM spectrum and it turns out to be simpler, cheaper and it is usually embedded in low rate, low power applications.

2.15.1 Zigbee applications:

There are many applications for the capabilities of Zigbee mesh networks (for example, low cost, and low power consumption) key ones include:

- Home automation
- Smart metering
- Internet of things
- Building automation

- Industrial automation

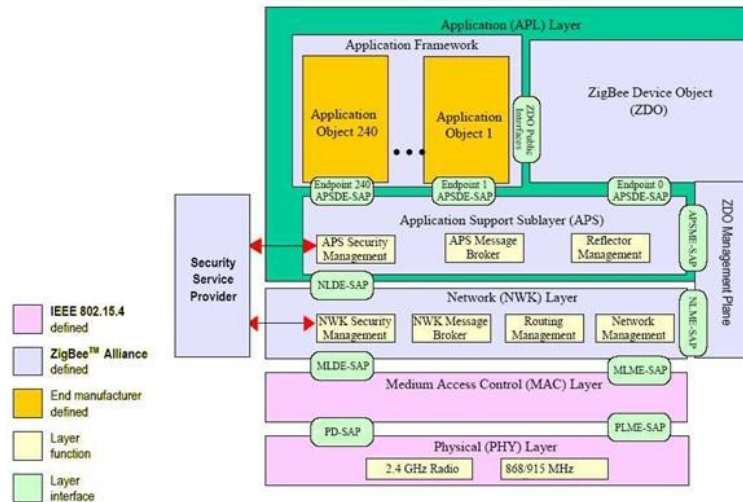


FIGURE 2.17: Zigbee stack architecture.

Zigbee architecture is displayed in fig 2.17; it consists of the physical layer, which provides access to the physical transmission media (RF transceiver) and the MAC layer which is based on IEEE 802.15.4 standard. The network layer, Application Support Sublayer (APS) and the Application layer (APL) are also added on top.

2.15.2 Zigbee Phy layer

Zigbee uses three frequency bands for transmission; the 868 MHz band has only 1 channel and has a raw data rate of 20 kbps. The 915MHz band has 10 channels and each channel's central frequency is separated from the adjacent band by 2 MHz and it has a data rate of roughly 40 kbps [14]. This layer employs the DSSS modulation method to transmit the information signals. The 2.4 GHz ISM band has 16 channels which are 5 MHz wide offers up-to 250 kbps RF data rate.

2.15.3 Zigbee Mac Layer

Channel access is primarily through Carrier Sense Multiple Access- Collision Avoidance (CSMA-CA). On a node hop to hop basis, the MAC layer can take care of transmitting data. Depending on the mode of transmission, i.e. Beacon or Non-Beacon mode, the MAC layer decides whether to use slotted or unslotted CSMA-CA. The MAC layer

takes care of scanning the channel, starting PANs, detecting and resolving PAN ID conflicts, sending beacons, performing device discovery, association and disassociation, synchronizing network device and realigning orphaned devices on the network [14]. Additionally, the MAC layer provides security features like access control and encryption of data.

2.15.4 Zigbee Network Layer

The network layer takes care of network start-up, device configuration, topology specific routing, and providing security [14]. On each device, the network layer is responsible for performing route calculations and next hop neighbor discovery among other tasks. The routing table is filled up when the devices join the network for the first time by transmitting route request packets. Adjacent devices respond to this request by issuing route response packets and the device calculates and stores the shortest route of the closet neighboring node.

2.15.5 Application Layer

It is the environment to host application objects on a Zigbee device. Up to 240 distinct application objects can be defined uniquely. It consists of the application profiles as the top layer over ZDO which provides the base functionality [14].

- Application Profiles define an accepted language for exchanging data and provide inter-operable services across different manufacturers. Zigbee Alliance has released several Standard Profiles which contain different device descriptors which have unique identifiers [14].
- ZigBee Device Objects (ZDO) provides an interface between the application objects, the device profiles, and the APS layer in Zigbee devices. It is located between the Application Profiles and the application support sub-layer. The ZDO are responsible for initializing the APS, the network layer, and the Security Service Provider, and also forming the configuration information from applications to implement discovery, security, network and binding management [14].

2.15.6 Device Types

- Coordinator
- Router
- End devices

In each PAN, the Zigbee coordinator is responsible for forming the network and it manages the network once it has been initiated. The coordinator has the ability to communicate with any device in its PAN. There is only one coordinator in each PAN and it acts as a sink and in most applications, the coordinator is the one which gets connected to an external network. Each PAN has a different PAN ID which distinguishes itself from other Zigbee networks. Being the center and the heart of the network, the Zigbee coordinator should be connected to a reliable source of power.

Zigbee Router is able to carry out multi-hop routing of messages across the network. The router can communicate with other routers and end devices to create data communication links. They also store information of other neighbouring nodes and can calculate the shortest path to the destination, in case of node failure or obstruction, they can provide backup routes. They are also important in that they can extend network coverage area by adding more routers in the network. Being one of the important nodes in the network, routers are required to be mains powered or connected to a reliable power source.

Zigbee End device or Reduced Function Device has reduced functionality. The end device can only communicate with its parent which it has joined to and cannot participate in data routing. The device can only be configured in a star topology and can go into sleep mode hence Reduced Function Devices can have shorter duty cycles.

These Zigbee devices can be connected in any of the topologies shown below:

As shown in Fig 2.18, Zigbee network devices can be configured to run in any of the three topologies: Star, Mesh or Cluster tree topology.

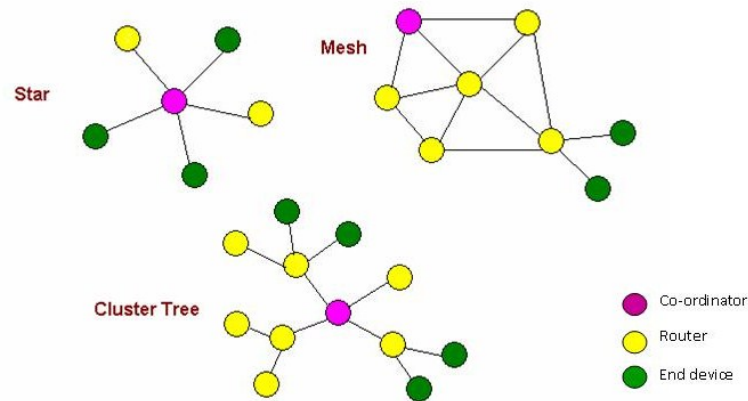


FIGURE 2.18: Zigbee network topologies.

2.15.7 Star

Each sensor node is connected to a central coordinator with a point to point connection. All data packets that traverse the network pass through the coordinator and the coordinator provides a common connection point for all sensor nodes in the network. This setup is simplest to implement and there is no multi-hopping of data as data traverses a single hop from a sensor node to the coordinator. In the event of one sensor node dying, other links in the network remain operational as each node is isolated by the link that connects it to the coordinator. The disadvantage of this topology is limited range and also, there is no alternative path to destination.

2.15.8 Cluster Tree

This topology consists of a number of star networks whose coordinators are also directly connected with a single PAN coordinator. The star networks are often called clusters where each cluster is uniquely identified by a different cluster ID.

2.15.9 Mesh

In a mesh topology, all of the devices are able to connect with one another since each node is connected through at least two pathways. Data from one node can multi-hop and be routed through other neighbouring nodes to reach its destination. More often, routers discover the shortest path to destination or the coordinator. In case of a node failure, the Zigbee mesh network demonstrates its robustness in terms of data routing.

In this case, data is relayed to the next available and closest neighbouring node and selects the path with the least communication cost until it reaches its destination.

The advantage of using this topology is that mesh networks are reliable and robust, in case of node failure, alternative routes to destination can be found.

2.15.10 Broadcast

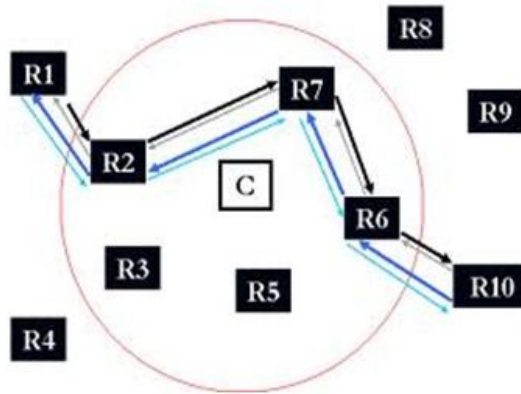


FIGURE 2.19: Propagation of a broadcast message in a Zigbee network.

In Zigbee networks, a broadcast message is a message that is conveyed to all nodes in a PAN regardless of its network address. This message is only sent once and then re-transmitted by neighbouring routers. When a router receives a broadcast message, it checks the source address of the sending node and it also checks the sequence number of the packet. After checking these details, the node decides whether it must retransmit the packet. The node retransmits the packet if other neighbouring nodes fail to respond and each node keeps a Broadcast Transaction Table which ensures routers do not repeat a message they have already repeated.

2.15.11 Zigbee addressing

Each device has a 64-bit IEEE address, this address is unique to every Zigbee device and it is assigned during manufacture.

Zigbee also consists of a 16-bit network address. This is the address which is assigned to the Zigbee device by the coordinator when it joins a PAN. This address is unique for a given network and it allows proper and easy device identification. This is the address used in routing tables and for data transmissions and it can change when the device leaves the PAN.

2.15.12 Forming a Zigbee network

When the coordinator intends to form a network, it searches for a radio channel that is unused and one that has least activity and one that is free from interference, it does this through energy scans. Channels with more energy levels are ignored and the coordinator selects a channel (frequency) with least energy to start on. The coordinator then assigns a PAN ID to the network and it assigns a network address to itself. At this stage, the network is now ready and the coordinator is ready to accept join requests from other nodes interested in joining the network.

2.15.13 Joining a Zigbee network

Once the network has been created, other devices can now join the network. Only routers and coordinators can allow other devices to join the network.

The device wishing to join the network first scans the available channels to find operational networks and identifies the one it should join depending on its PAN ID. If its PAN ID matches the one of the valid network, the join process takes place. If the requesting node identifies multiple routers and a coordinator, it selects the one that has already joined a PAN or coordinator that is allowing joining. The node then broadcasts a join request message to the relevant router or coordinator. The parent (router or coordinator) then decides whether or not to allow the device to join, this depends on the node join time or availability of address space. If these conditions are met, the parent then allows the device to join the network and a 16 bit network address is allocated to the newly joined node (daughter).

2.16 Zigbee routing

2.16.1 Mesh routing

Mesh routing is the heart of Zigbee [16] and it is based on the Distance Vector (DV) routing principle [15]. In this case, each Zigbee node in the PAN takes part in the transmission of data from source to destination and keeps a routing table entry for the route. This entry contains the path and address of the neighbouring node to the

destination. The path to destination is established on-demand using a route discovery process. The sending node broadcasts a route request packet specifying the destination 16-bit network address, the broadcast packet goes in all directions and each node that receives the broadcast relays to all nodes within range, along the way, a path cost is kept. Every hop adds a number from one to seven to indicate the strength of this hop [16]. The destination node receives multiple copies of the broadcast message through different routes and the node evaluates the cost of each message's path to determine the best route. If another route with a lower path cost comes along to any given router, it updates that route in its routing table. This process establishes a shortest path to destination and the path can be used anytime as long as routing nodes in the path do not die.

2.16.2 Proactive mesh routing

Nodes in proactive routing fully exchange neighbour and routing tables [17]. This type of routing is often difficult to implement in low cost and low memory devices owing to the amount of memory needed, hence proactive routing is especially used in industrial applications and devices with enough storage capacity [17]. However, the advantage of proactive mesh routing is that data routes and routing tables between source and destination nodes are accessible in every node and are refreshed at regular intervals. The problem, however is the issue of memory consumption which also increases as the network increases.

Examples include:

- advanced Ad-hoc On Demand Vectoring (AODV)
- Routing Protocol for Low Power and Lossy Networks (RPL), recommended for 6LoWPAN
- Dynamic Source Routing (DSR)
- Dynamic MANET On-Demand for 6LowWPAN (DYMO-low)
- 6LoWPAN Ad-hoc On-Demand Distance Vector (LOAD) [17]

2.16.3 Reactive mesh routing

Reactive routing is characterized by information that is selectively queried, and only when necessary [17]. This generates a smaller routing table but also requires a higher management effort. This means that data paths which are not yet used have to be established first. Therefore, applications which are time-crucial are not ideal to implement reactive routing, if the routes are constantly changing. Reactive routing protocols are, however, efficient for applications which are not time-crucial. Reactive routing scales vary well with the network size and some examples include:

- Graph Routing, preferred in WirelessHART
- Optimized Link State Routing (OLSR) [17].

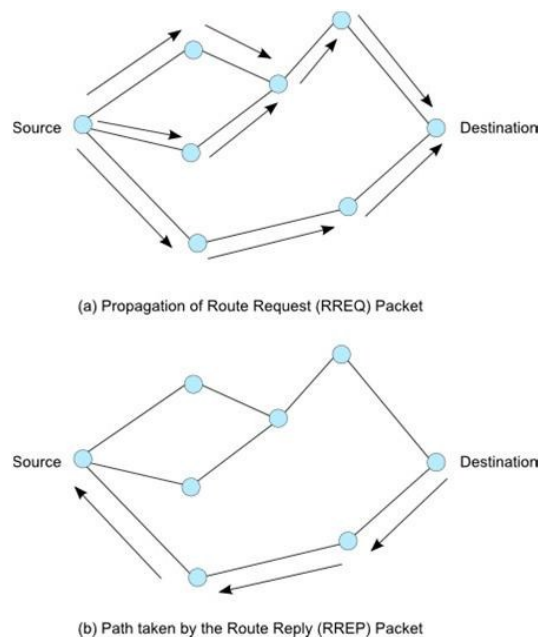


FIGURE 2.20: AODV routing example [17].

2.17 Free space propagation model

Fig 2.21 above can also be used to model the propagation of radio waves in free space. The power P_r (dB) of the signal at the receiving antennae (Rx) is given by the formula [14] which is referred to as the Friis transmission formula:

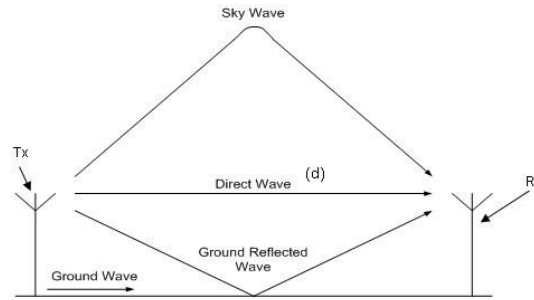


FIGURE 2.21: Wave propagation.

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2} \quad (2.1)$$

where,

P_t = Transmitted power of the signal

G_t = Transmitting antenna gain

G_r = Receiving antenna gain

d = distance between Tx and Rx antenna

λ = wavelength of the propagating signal

2.17.1 Path loss

Definition of path loss LP :

$$L_p = \frac{P_t}{P_r} \quad (2.2)$$

Path Loss in Free-space:

$$L_p f (DB) = 32.45 + 20 \log_{10} f_c (MHz) + 20 \log_{10} d (km) \quad (2.3)$$

where f_c is the carrier frequency.

This shows that the greater the f_c , more is the loss.

According to the inverse square law:

$$I_r = \frac{P_1}{A_1} = \frac{P_1}{4\pi r^2} \implies A = 4\pi r^2 \quad (2.4)$$

$$I_r \propto \frac{1}{r^2} \quad (2.5)$$

Doubling the distance:

$$I_{2r} \propto \frac{1}{(2r)^2} \longrightarrow I_{2r} \propto \frac{1}{4r^2} \quad (2.6)$$

combining 2.5 and 2.6, we get:

$$I_{2r} = \frac{1}{4}I_r \quad (2.7)$$

From equation 2.7, it is clearly noticeable that doubling the distance reduces the strength of the signal by a quarter of its original value.

2.18 Summary

The previous chapter has given a highlight on the literature available in the fields of WSN, satellite communication systems and Zigbee technology. The hardware and software components used in this project were selected based on the information provided in this chapter. The next chapter will look at the project's proposed methodology.

Chapter 3

Proposed Methodology

This chapter discusses in more detail aspects introduced in chapter 2. Some of the items include; detailed calculation of the battery charger components and calculations for battery and solar panel sizing based on the objectives and requirements set out in chapter 1.

3.1 Project development

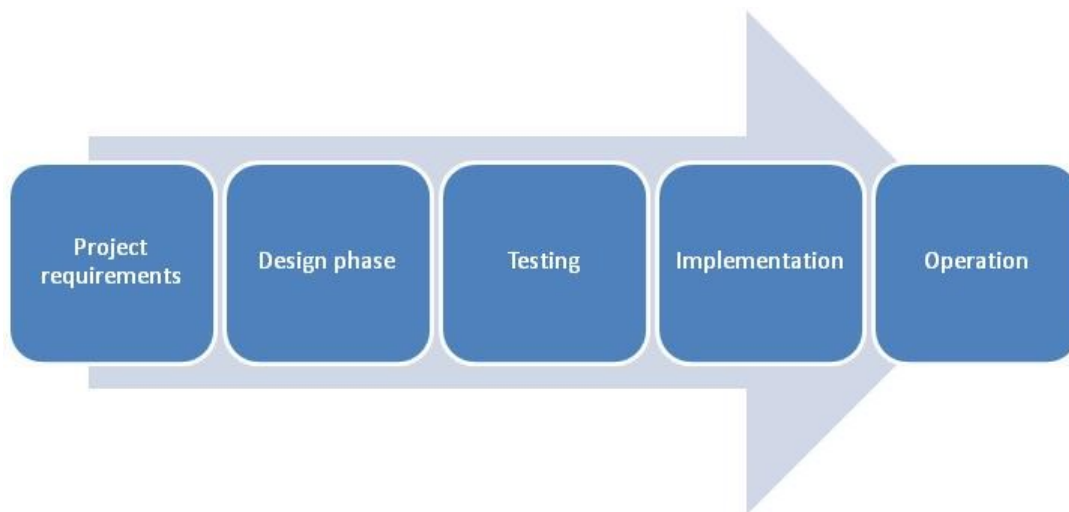


FIGURE 3.1: Stages of project development.

3.1.1 Project requirements

Development of this project begins with understanding the problem at hand and collecting the requirements of the system. Aspects to consider include:

1. Performance requirements,
2. Integration with other systems,
3. System behavior, where the system will be installed and operated,
4. Regulatory requirements.

3.1.2 Performance Requirements

This step looks at how the system will perform when deployed in the field. For WSN, number of nodes in the network, maximum range and battery capacity are important parameters to consider when designing the system. Examples of these requirements are:

- Days of autonomy
- Power consumption
- Coverage area
- Throughput
- Reliability
- Satellite coverage, costs and integration
- Range of WSN

3.1.3 Integration Requirements

For this project at hand, our WSN needs to be integrated with weather sensors and the ground station will be integrated with the Iridium satellite network. Integration requirements specify that our choice of hardware should be able to interface with different software and communication protocols, such as UART, SPI, and I2C.

3.1.4 Regulatory Requirements

ICASA, the Independent Communications Authority of South Africa is the regulator for the South African communications, broadcasting and postal services sector [18]. In terms of Notice 926 of 2008 on 29 July 2008 in Government Gazette 31290, the 2.4GHz band lies in the free unlicensed spectrum. To cut operating costs, the WSN will use a Zigbee transceiver which operates in this unlicensed free spectrum.

3.1.5 Design Phase

This section will select architecture of the WSN, layout the structure of each node and select the electronic and mechanical components.

3.1.6 Architecture Decisions

This section will decide the architecture of the communication protocol, in this case, IEEE 802.15.4/Zigbee. We will go on to select the system platform vendor (IC or radio module) with considerations to price, communication frequency band, power consumption and routing protocols. The decisions summarized above can be derived from the performance requirements described in chapter 3.2.

3.1.6.1 Star vs. Mesh Topology

The Zigbee network topology to be used is determined by the number of nodes in the network and the range of the sense area, this will influence whether a star or mesh network can be deployed. Full comparison is described in detail in chapter 2.16.

A star topology is applicable where a small number of nodes need to communicate with each other, and the sense area does not exceed the radio transmission range of approximately 100 meters indoors and 300 meters outdoors. For networks which cover a larger geographical area, a mesh topology is the appropriate choice. Mesh networks have the advantage of automatically creating data routes through intermediate hops and can therefore cover a larger geographic area.

3.1.6.2 Considerations for Battery-Powered Systems

When designing battery powered sensor networks, it is crucial to avoid potential semi to complete battery drain situations. For example, in a WSN with routers and a coordinator, it is clear that a coordinator can only receive data from its router or an end device if it does not sleep or enter idle mode to conserve battery reserves. Only an end device is allowed to enter sleep state. Routers and or the coordinator may be allowed to sleep if strict and very accurate time synchronisation protocols are employed. To avoid messages being lost, duty cycle periods, transmission-time-out periods and battery capacity need to be carefully calculated. Complete battery drain might damage the battery cells in the long run so the system must be designed with at least 20% depth of discharge. Other considerations for battery-powered systems include truncating battery status with the data so as to enable preventive maintenance.

3.1.7 Implementation

After testing nodes in a laboratory environment, this stage of development life-cycle begins after important architecture decisions have been made and the entire system deployed in the field.

3.1.7.1 Testing WSN applications

The overall development process of control algorithms running in sensor nodes and the GS is a bit complex. The problem must be split into smaller and more manageable tasks, to increase software quality, it is necessary to test the functional blocks before implementation. The following sections describe the different stages of the test procedures conducted.

3.1.7.2 Unit Test

This section focuses on the software and hardware verification procedures that tests individual different blocks isolated from each other. Typical test modules are the software blocks for sensor data acquisition, power control, GPS timing system and XBee radio communications.

3.1.7.3 Integration Test

This test examines whether the software components of a single sensor node correctly work together when combined and tested as a single unit. The purpose of this stage is to test the functionality and reliability of a sensor node. For a sensor node, integration testing checks whether the node is able to sample the environment for data, process that data and send the information to the GS through a wireless medium. In case of the GS, this stage checks whether the GS is able to store sensory information from all nodes and checks if the GS is able to send the data through the Iridium gateway interface.

3.1.7.4 Performance Test

This test checks the performance of each individual node. Tests such as power tests and throughput are conducted here. The purpose is to predict the lifetime of the nodes and to obtain the operational limits of each node.

3.1.7.5 Field Test

At this stage, the entire system is deployed in the field and its overall performance is observed. It is also important to observe the RF spectrum where the project will be deployed and note any possible sources of interference. Any adjustments or enhancements to the system are carried out at this point.

3.1.8 Operation Phase

Once the water resource monitoring system is installed, development activities have to be conducted. This stage of the project aims to educate the user on how to use the system. This part of the project also discusses how to give the user some insight into possible solutions for problems that might occur.

Referring to the characteristics of WSN explained in chapter 2.2, we hereby explain in detail the design requirements.

3.2 List of project requirements

1. Identify a suitable satellite constellation
2. Determine the exact parameters to be monitored
3. Design / obtain hardware for various system functions
4. Determine power supply requirements
5. Determine configuration of master station and set up master station
6. Develop a wireless sensor network with the following eight different weather sensors:
 - Soil temperature
 - Soil moisture
 - Air temperature
 - Relative humidity
 - Rain-gauge
 - Wind speed
 - Wind direction
 - Ambient light
7. The project shall consist of a ground station which gathers sensor data and relays the information to a master station through a satellite constellation of choice
8. The ground station shall use a Linux based SBC
9. Build prototype and deploy in the field and test
10. Develop an interactive GUI (MS) to display sensor data

3.3 Software requirements

- The sensor nodes can be programmed in any language
- The master station can be written in any user friendly language

3.4 Hardware requirements

- Sensor node processor
- Ground station SBC
- Davis anemometer and wind vane
- Davis rain gauge
- Davis soil temperature and moisture sensor
- Davis humidity station
- Satellite modem
- Wireless radios
- Solar panels
- Sealed lead acid batteries
- IP65 plastic enclosures
- GPS module

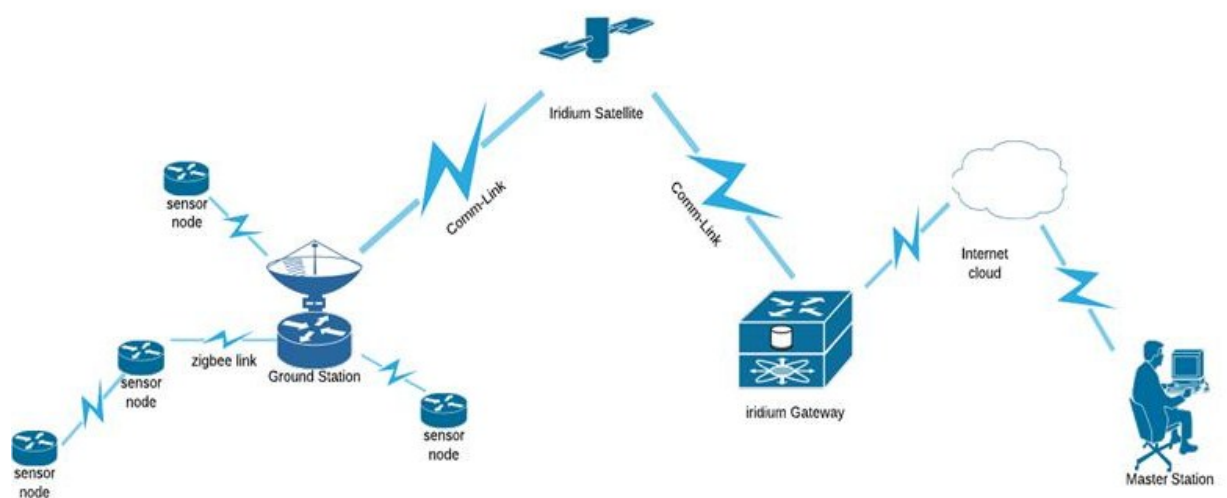


FIGURE 3.2: Project layout.

3.5 Hardware development overview

From chapter 2.2, we find that each sensor node is composed of the following:

- microcontroller
- power supply
- wireless radio
- one or more sensors

The following sections describe the components and the procedures used to select various hardware.

3.5.1 Arduino

Arduino [20] refers to a family of hardware which uses free, open source and cross platform software. The Arduino language uses an easy to learn and simplified version of C++ and it uses an interactive GUI. It is used in many educational platforms by designers, engineers and professionals across the world to build and prototype electronic systems. It is an easy to use platform which can be used by designers with little embedded programming knowledge to build applications which can sense and control the environment. The Arduino boards are also cheap to buy or one can make his/her own Arduino board since the hardware is also open source.

3.5.2 Arduino Uno

An Arduino Uno is based on the Atmega 328p, 8 bit microcontroller which clocks at 16MHz. It contains 128kB of program memory and 4kB of SRAM [22]. It has 28 pins of which 14 can be used for digital applications and 6 can be used as PWM outputs and 6 – 10 bit ADCs can be used for analog applications. It features a variety of interfaces, among them 1 USB port, 1 UART TTL (5V), I2C, power jack, ICSP header.

The following table provides a specifications summary of the Arduino Uno.

TABLE 3.1: Arduino Uno specifications

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6 – 10 bit ADCs
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB is used by the bootloader
SRAM	2KB(ATmega328)
EEPROM	1KB(ATmega328)
Clock Speed	16MHz



FIGURE 3.3: Arduino Uno.

The maximum voltage into the Arduino's ADC should not exceed 5V [20]. A voltage divider with two scaling resistors can be used to scale down the sensor voltage to a range of 0 – 5V.

$$\frac{V_{out} \times R_b}{R_b + R_a} \leq 5V \quad (3.1)$$

where

$$R_a = 450\Omega$$

$$R_b = 2250\Omega$$

$$V_{out} = 6V$$

TABLE 3.2: Key parameters for Atmega328P

Parameter	Value
Flash	32 Kbytes
Pin count	32
Max operating frequency	20MHz
CPU	8 – bit AVR
Max I/O pins	23
Ext interrupts	24
USB Interface	No
SPI	2
TWI(12C)	1
UART	1
ADC channels	8
ADC resolution (bits)	10
SRAM (Kbytes)	2
EEPROM (bytes)	1024
Timers	3
PWM Channels	6

R_a and R_b are the scaling resistors and v_{out} is the output sensor voltage. The resulting ADC reading can be converted to a voltage value using:

$$voltage = Sensor_{val} \times \frac{5.0}{1023.0} \quad (3.2)$$

3.5.3 Raspberry Pi.

The Raspberry Pi [32] is a small sized single board computer running Linux OS.

The following table summarises Raspberry Pi specs:

The following table summarises the Raspberry Pi, Beaglebone, Arduino Due and Arduino Uno.

Table 3.4 summarises the characteristics of some of the available SBCs which could be used in this project. As can be seen above, the Raspberry Pi outperforms other

TABLE 3.3: Raspberry Pi specifications

Chip	Broadcom BCM2835 SoC full HD multimedia applications processor
CPU	700 MHz Low Power ARM1176JZ-F Applications Processor
GPU	Dual Core VideoCore IV Multimedia Co-Processor
Memory	512MB SDRAM
Ethernet	onboard 10/100 Ethernet RJ45 jack
USB 2.0	Dual USB Connector
Video output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio output	3.5mm jack, HDMI
Onboard storage	SD, MMC, SDIO card slot
Operating system	Linux
Dimensions	8.6cm x 5.4cm x 1.7cm
Power rating	700mA, 3.5W



FIGURE 3.4: Raspberry Pi.

platforms on power consumption, cost and availability. Since power consumption is an important factor to consider, this makes the Raspberry Pi a suitable candidate for our project.

3.6 Power supply design

When designing a WSN, it is crucial to put more emphasis on the power supply. This system will be deployed in a remote area where battery replacement or recharging is difficult and hence it is vital to design an efficient and reliable power supply which provides at least three days of autonomy, in worst winter cases. A reliable power supply

TABLE 3.4: Beaglebone, Arduino Due and Raspberry Pi comparison

Name	Arduino Uno(R3)	Arduino Due	Raspberry Pi	Beagle bone
Price	\$29.95	\$62	\$35	\$50
Size	2.95" × 2.10"	4.09" × 2.09"	3.37" × 2.125"	3.4" × 2.1"
Processor	Atmega 328P	AT91SAM3 × 8E	ARM11	ARM Cortex - A8
Clock speed	16MHz	84MHz	700MHz	700MHz
RAM	2Kb		256Mb	256Mb
FLASH	32Kb	512Kb	(SD Card)	(SD Card)
EEPROM	1Kb			
Input voltage	7-12V	7-12V	5V	5V
Power consumption	42mA	800mA	500mA	170mA
Digital I/O pins	14	68	8	66
Analog input	6 10-bit	14		66
PWM	6	12		8
TWI/I2C	2	2	1	2
SPI	1	4	1	1
UART	1	5	1	5
Dev IDE	Arduino IDE	Arduino IDE	Linux	Linux
Ethernet	N/A	1	1 10/100	1 10/100
USB master	N/A	1	2 USB 2.0	1 USB 2.0

is required for the smooth functioning of the network and it also prolongs network lifetime. The power supply should guarantee consistent supply of the required power.

3.6.1 Battery sizing:

This section will look at steps to determine the size of batteries needed for our application. We are going to determine how much storage the batteries should provide. This is based on the amount of power needed by our application and the number of days our system will have to provide power without solar input (days of autonomy).

Incorrect battery sizing can lead to poor autonomy days, shorter network life time or damage to the battery as a result of over-charging or over-discharging.

3.6.2 Battery sizing methodology

The calculation shall be based on the IEEE standard 485 (1997, R2003) Recommended Practice for Sizing Lead-Acid Batteries for Stationary Applications [18].

1. Load classification:

The first step to determine the size of the battery required will be to determine

the loads that the battery will be supporting. Two types of loads are applicable in this project, continuous loads and momentary loads. Continuous loads are devices which operate in an always On-State and have a 100% duty cycle. Typical continuous loads are as follows:

- Arduino Uno
- GPS module
- Raspberry Pi
- Rainfall sensor
- Wind speed sensor

Momentary loads have a duty cycle which is less than 100% and they can be powered for shorter periods. Typical momentary loads include:

- XBee radios
- Weather sensors
- Iridium satellite modem

Loads applied to the battery are normally referred to as constant power, constant resistance, or constant current. However, for sizing purposes, loads are treated as constant power or constant current [18] hence each system is reviewed carefully.

3.6.2.1 Battery sizing

TABLE 3.5: Sensor node load list

For a sensor node:	Load(mA)	Duration(h) per day
Arduino uno	45	24
GPS module	40	24
XBee radio	40	0.4
sensor(s)	1.5	0.4

2. The second step will be to construct the Load Profile.

A load profile is an estimate of the variation of the energy required by the system over a specific period of time. It is constructed from a load list by stacking energy rectangles on top of each other [18].

3. The third step will be to determine the battery capacity

4. The fourth step will be to determine the size of the solar panel.

TABLE 3.6: Sensor node load profile

	Current (mA)	Number of hours	(W) at 5v	(mAh)	(Wh) per day
Sensor(s)	1.5	0.3	0.075	0.45	0.0225
Arduino Uno	45	24	0.225	1080	5.4
GPS	40	24	0.200	960	4.8
XBee	45	0.3	0.225	13.3	0.0675
Total				2054	10.29

Based on information provided in table 3.6, the minimum battery capacity required to support the system for at least three days of autonomy can be calculated as follows:

$$C_{min} = \frac{E_d(d \times k_a \times k_t \times k_c)}{V_{dc} \times (1 - k_{dod})} \quad (3.3)$$

where C_{min} is the minimum battery capacity (Ah)

E_d is the daily energy requirements (vAh)

V_{dc} is the battery voltage (v)

k_a is the battery ageing factor (%)

k_c is the capacity rating factor (%)

k_{dod} is the maximum depth of discharge (%)

k_t is the temperature correction factor (%)

d is the day(s) of autonomy

Given a depth of discharge k_{dod} of 20%, battery ageing factor k_a of 25%, temperature correction factor k_t of 0.95, k_c of 10% and a daily watt hour of 10.29 Wh, the minimum battery capacity requirement for:

- Winter scenario:

$$C_{min} = \frac{10.29(3 \times 1.25 \times 0.95 \times 1.1)}{5 \times 0.8} = 10.14Ah \quad (3.4)$$

- Summer scenario:

$$C_{min} = \frac{10.29(1.25 \times 0.95 \times 1.1)}{5 \times 0.8} = 4.02Ah \quad (3.5)$$

The same approach can be applied to determine GS battery requirements:

TABLE 3.7: GS load profile

	Current (mA)	Number of hours	(W) at 5v	(Ah)	(Wh) per day
Raspberry Pi	300	24	1.500	7.200	36
Iridium modem	45	0.5	0.225	0.023	0.11
XBee	45	24	0.225	1.08	5.4
Total				8.03	41.51

Taking equation 3.3 into consideration, we get:

- Winter scenario:

$$C_{min} = \frac{41.51(3 \times 1.125 \times 0.95 \times 1.10)}{5 \times 0.8} = 40.66Ah \quad (3.6)$$

- Summer scenario:

$$C_{min} = \frac{41.51(1.25 \times 0.95 \times 1.10)}{5 \times 0.8} = 13.56Ah \quad (3.7)$$

3.6.3 Solar panel sizing

3.6.3.1 Sensor node

From [33], Cape Town receives an average of $5400Whm^{-2}$ per day. This unit is related to the number of sun hours received in a particular area and can be represented as having units of *hours/day* of full sunlight. Since the solar panel produces 20W of power at full sunlight and at maximum insolation levels, we can determine the Wh per day that one solar panel produces and that value can be used to determine how many solar panels are required for our application.

$$\frac{5.4}{day} \times \frac{20W}{1solarpanel} = 108Wh/day \quad (3.8)$$

From 3.8, 1 20W solar panel produces 108Wh per day. To determine the number of solar panels needed to support the load, we take into consideration the system efficiency, (50%) and winter insolation level, (40% of the maximum during summer):

$$\frac{10.29Wh}{108Wh \times 0.5 \times 0.4} = 0.47 \approx 1 \times 20W solar panel \quad (3.9)$$

Equation 3.9 clearly shows that 1 20W solar panel can sufficiently supply power to the load in winter and still provide enough power to recharge the battery.

Since the system has been designed for the winter worst case, the solar panel should provide more than enough power in summer since the area receives an average of 12 hours of sunshine per day.

3.6.3.2 Ground station

The same technique conducted previously for sizing solar panel requirements for the sensor node can be applied to the ground station. From equation 3.8, a 35W solar panel produces:

$$\frac{5.4}{day} \times \frac{35W}{1solarpanel} = 189Wh/day \quad (3.10)$$

Substituting for the GS requirements in equation 3.9 yields:

$$\frac{41.51Wh}{189Wh \times 0.5 \times 0.4} = 1.098 \approx 1 \times 35W solar panel \quad (3.11)$$

Table 3.8 below gives a summary of the battery and solar panel sizes which were used. The sizes are the nearest values of the figures obtained in the previous section.

TABLE 3.8: Summary

	Battery Size (Ah)		Solar panel (W)
	Winter	Summer	Winter
Sensor Node	12	7	20
Ground Station	36	15	35

3.6.4 Charging regulator

Building an optimum lead acid battery charger is important to the performance and life of the battery cells. The UC3906 IC from LT will be used as the brains of the battery charger, in this application; the IC would be setup as a Dual Level Float Charger.

The dual level float charger has three states [19], a high current bulk charge state, an over-charge state and a float state. When power is applied to the IC, the battery charges at trickle current rate (I_t) until the voltage reaches threshold voltage (V_t) and the charger enters a high current bulk charge state thereby supplying a constant charge rate at I_{max} until the battery reaches a transition threshold, V_{12} . When the battery reaches the threshold voltage, the charger enters an over-charge cycle. In this cycle, the charger maintains the battery at a higher voltage, V_{OC} until the charge rate drops to a transition current, I_{OCT} . At this point, the capacity is nearly 100% and the charger enters in the third state where it sets the float voltage level V_F for the battery fig 3.5 and fig 3.6.

Figure 3.7 shows the circuit diagram of a dual level float charger. The charging circuit consists of the main chip, UC3906 and a general purpose power transistor. A TIP32C was selected for this application.

3.6.4.1 Design equations

$$V_{oc} = V_{ref} \left(1 + \frac{R_a}{R_b} + \frac{R_a}{R_c} \right) \quad (3.12)$$

$$V_f = V_{ref} \left(1 + \frac{R_a}{R_b} \right) \quad (3.13)$$

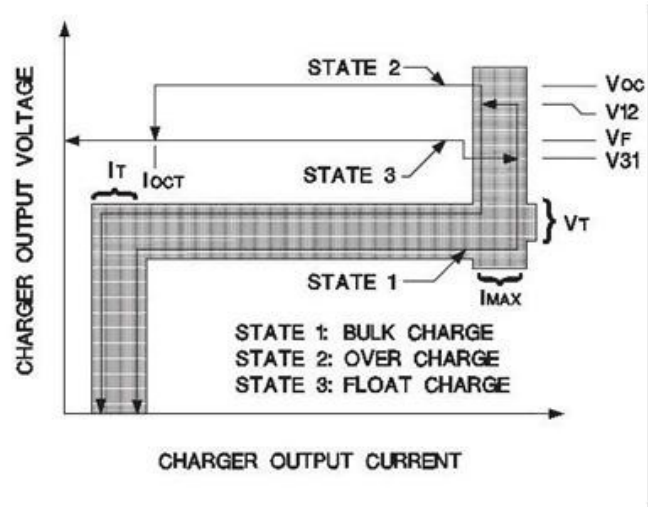


FIGURE 3.5: Dual level float charger.

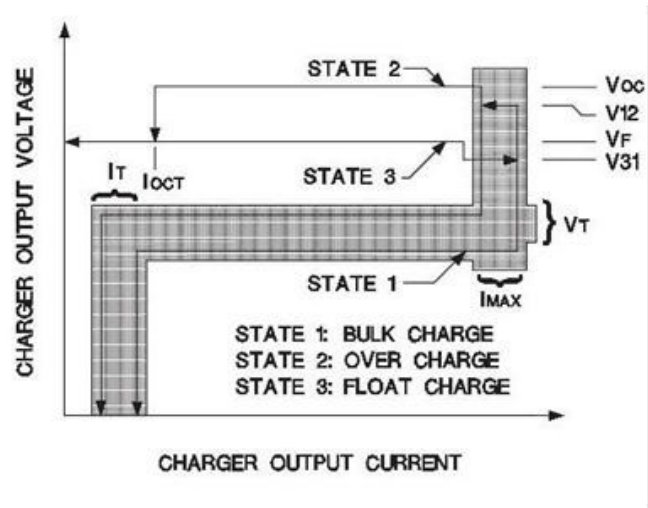


FIGURE 3.6: Dual level float charger states.

$$V_{12} = 95V_{oc} \tag{3.14}$$

$$V_{31} = 9V_f \tag{3.15}$$

$$I_{max} = \frac{0.25V}{R_s} \tag{3.16}$$

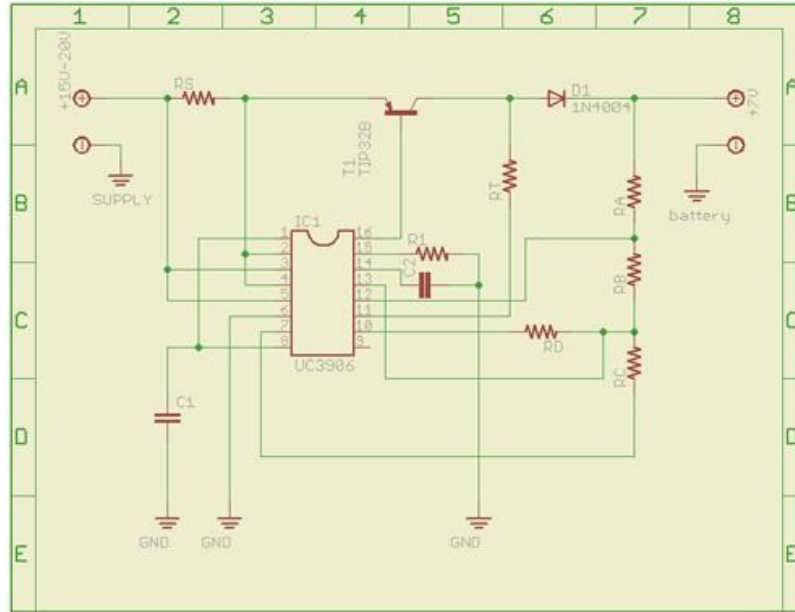


FIGURE 3.7: Dual level float charger circuit diagram.

3.6.4.2 Resistor calculations

Determining the voltage and current levels of the charger is done by the proper selection of resistors (R_a , R_b , R_c , R_b , R_t and R_s) and the equations below are used to determine these values. The first step is to pick a driver current I_d , whose value should be between $50\mu\text{A}$ to $100\mu\text{A}$. For this application, a driver current of $80\mu\text{A}$ was chosen.

$$I_d = 80\mu\text{A} \quad (3.17)$$

$$R_c = \frac{2.3V}{I_d} \implies 43k \quad (3.18)$$

$$R_a + R_b = R_{sum} = \frac{V_f - 2.3}{I_d} \quad (3.19)$$

$$R_{sum} = \frac{7 - 2.3}{53\mu\text{A}} = 88679.2 \quad (3.20)$$

$$R_d = \frac{2.3 \times R_{sum}}{V_{ov} - V_f} = 407924 \quad (3.21)$$

$$R_a = (R_{sum} + R_x) \left(1 - \frac{2.3}{V_t}\right) \quad (3.22)$$

$$R_x = \frac{R_c \times R_d}{R_c + R_d} = 38900 \quad (3.23)$$

$$R_b = R_{sum} - R_a \quad (3.24)$$

$$\implies R_a = 76.9\Omega \text{ and } R_b = 11.77\Omega \quad (3.25)$$

3.6.4.3 Charger specs

In fig 3.7, the charger is shown configured as a dual level float charger designed for a 6V 12AH lead acid battery. The specifications are listed below:

TABLE 3.9: Battery charger specifications

Input supply voltage	15 – 20V
Operating temperature range	0 – 60°C
Start up trickle current (I_t)	10mA
Start up voltage (V_t)	5.9V
Bulk charge rate (I_{max})	700mA
Bulk to over charge transition voltage (V_{12})	7.125V
Over charge voltage	7.5V
Over charge terminate current (I_{oct})	100mA
Float voltage (V_f)	7.0V
Float to bulk transition(V_{31})	6.3V

The charging regulator is powered by a solar panel which will be at 20V and it steps down the charging voltage to about 8V. This poses a serious threat in that all the excess energy will be dissipated as heat so the overall charging system would be inefficient and too wasteful.

To overcome this challenge, a highly efficient switching DC-DC converter, as per fig 3.8

is incorporated. The switching regulator steps down the voltage from 20V to 9V with minimal power losses and heat dissipation. The specifications are listed in table 3.10 below.



FIGURE 3.8: Traco Switching DC-DC converter.

TABLE 3.10: Traco Switching DC-DC converter specs

Input voltage range	12 – 36V
Output voltage	9.0V
Output current (max)	1.0A
Efficiency	95%

3.6.5 Low voltage disconnect (Battery monitor)

During worst case winter scenarios, batteries may be damaged by over-discharge when the batteries are not charging. Some components of the sensor nodes continue to draw power from the battery even if it falls below usable voltage. This may damage the battery itself or even some components as a result of low voltage operation. To protect the components from damage, a 6V battery cut-out module is connected in series with the load. This device disconnects power to the load when voltage falls below a set limit and re-energises power to the circuit when voltage returns to a pre-defined threshold after recharging.

3.7 Hardware integration

In this section, we will turn our attention to specific sensors used and how the conversion formulas were derived and calculated. These sensors are the electronic devices which measure environmental physical quantities such as rainfall, wind speed, wind direction etc. and convert them into signals for further processing, storage or display. There are many sensors used for measuring different environmental parameters, but this project only uses some of them and they are discussed in this section.

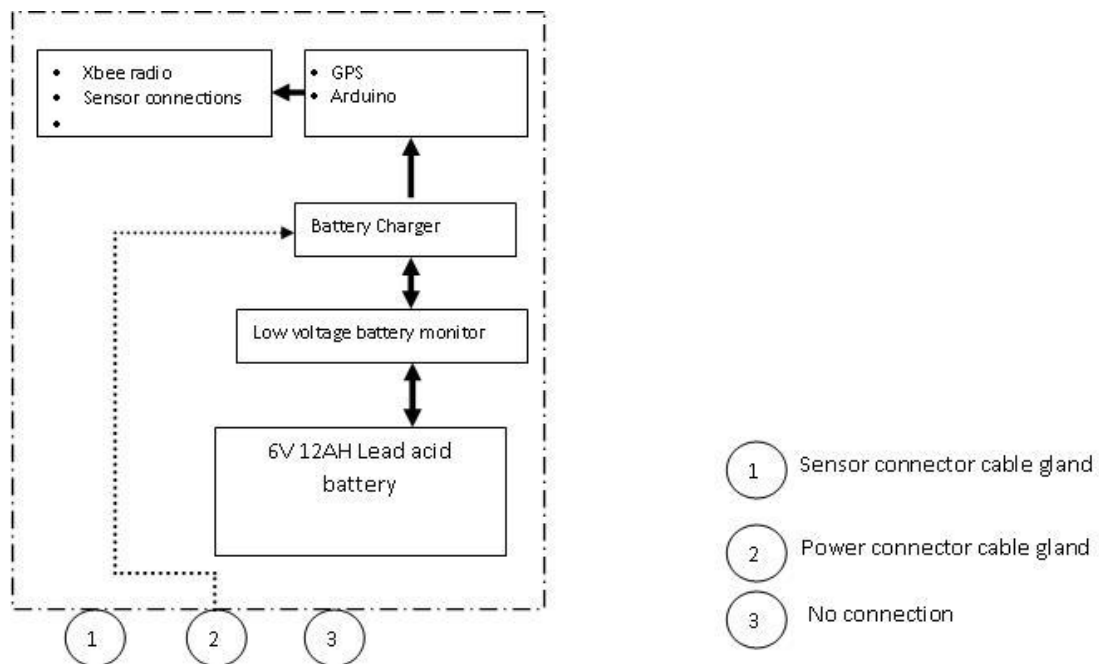


FIGURE 3.9: Sensor node block diagram

3.7.1 Soil moisture sensor

For measuring the soil moisture content, a Davis soil moisture sensor was chosen. A typical Davis sensor is shown in fig 3.11. This sensor is based on an output electrical resistance to measure the moisture. The micro-controller then converts the electrical resistance value from the sensor into a reading of cent-bars of soil water moisture. The values range from 0 to 200 cent-bars. If the soil is very wet, the value will be close to zero and if the soil is very dry, the reading will be high.

Measuring the output resistance from this sensor is a bit challenging. We cannot use a multi-meter directly to measure the resistance, but need a dedicated circuit with specific



FIGURE 3.10: Davis soil moisture sensor

timing. The reading has to be taken between 20 and 40 milliseconds after powering because DC power can only be applied for a very short period of time before the sensor polarises and gives bad readings. Explanation and implementation of the timing and switching circuitry is shown in fig 4.4. After taking the readings, the circuit has to switch off power for at least 10 minutes to allow time for the sensor to discharge all the potential before taking any readings again.

According to the datasheet, the output resistance is related to the soil water potential by the following equation:

$$kPa = \frac{4.093 \times R}{1 - 0.009733 \times R - 0.01205 \times T} \quad (3.26)$$

where:

$R = kohms$

$T = temp \text{ in Celcius}$

$kPa = soil \text{ water potential}$

Fig 3.11 shows sample results of the soil water potential against sensor resistance.

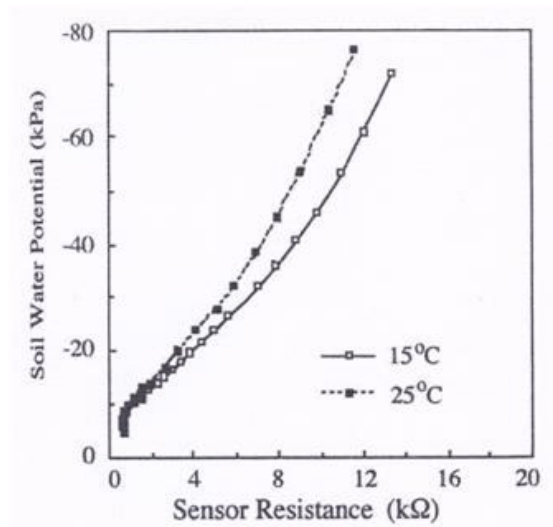


FIGURE 3.11: Davis soil moisture sensor output resistance

3.7.2 Soil temperature sensor

This sensor is based on an output resistance to measure soil temperature. It is a platinum wire thermistor whose output resistance is proportional to soil temperature. The soil temperature sensor is connected the same way as in fig 3.15.



FIGURE 3.12: Davis soil temperature sensor

The datasheet of the soil temperature sensor does not provide design equations but from fig 3.13, we can deduce an equation of resistance against temperature.

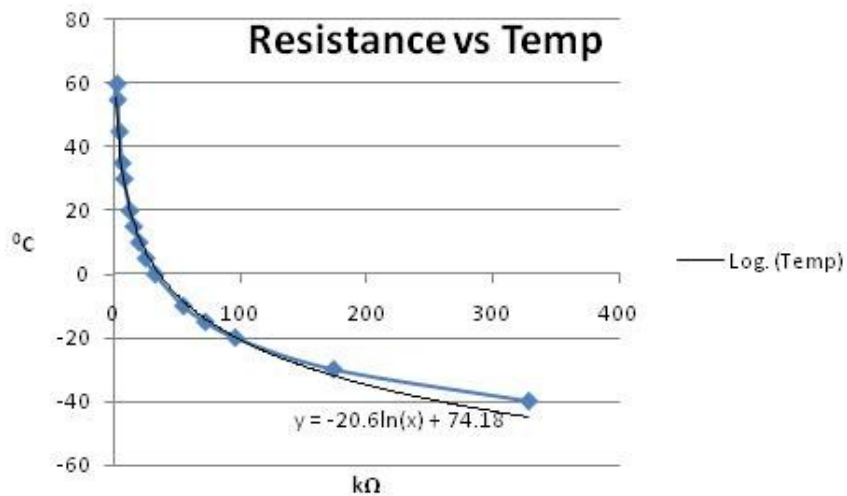


FIGURE 3.13: Graph of resistance against temperature

3.7.3 Rain gauge

A rain gauge measures the amount of rainfall that has fallen at a specific area. It consists of a collector cone and two tipping buckets. Rain enters and fills one of the two buckets, when the bucket has collected 0.2mm of rainfall, the bucket tips and causes a switch to close and this brings the second bucket into position. Rainfall can then be measured by counting the number of times the switch closes and multiplying it by a calibration constant.



FIGURE 3.14: Davis rain gauge

The rain gauge is connected directly to the digital port of the Arduino Uno since the digital ports have internal pull-up resistors. Digital channels from the Arduino have the capability to count the number of tips and the amount of rainfall can be deduced using the formula:

$$level = counts \times 0.2mm \quad (3.27)$$



FIGURE 3.15: Pull up resistor configuration.

3.7.4 Cup anemometer and wind vane

This sensor measures wind speed and direction. The cup anemometer consists of three cups which rotate proportionally with the speed of air. It consists of a solid state magnetic switch which closes with every complete rotation, the number of times the switch is closed is counted and with the correct formula, wind speed can be determined. The cup anemometer is a digital sensor which can be connected to the internal pull-up resistor of the Arduino. The wind vane uses a resistive sensor which gives different resistive values when the wind direction changes. We can directly power the circuit to get an output voltage corresponding to the wind direction, fig 3.17. The sensor has a potentiometer with a nominal value of 0 – 20k Ω which gives direction from 0 – 335 degrees.

The Arduino map function takes in the original value, *val* and values to map it to. Our value of an analog input can range from 0 – 1023 (10 - bit resolution) and we want a linear relationship between an analog input value and the angle such that, as the analog input value increases, the angle also increases.

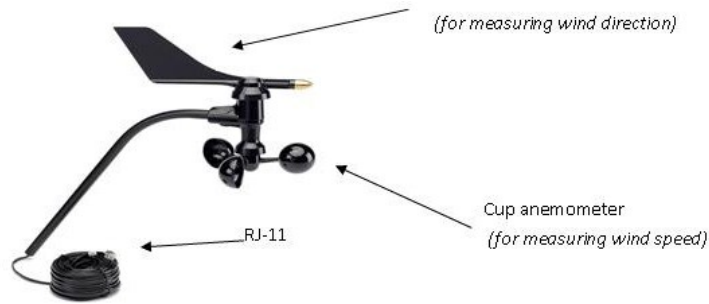


FIGURE 3.16: Davis cup anemometer

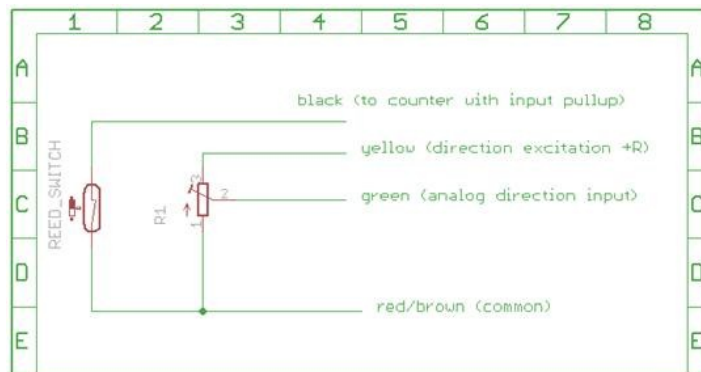


FIGURE 3.17: Internal schematic of the cup anemometer and connection to controller

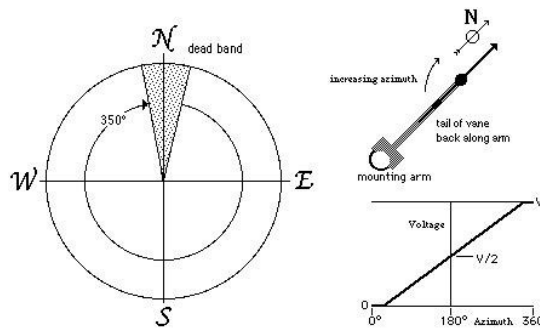


FIGURE 3.18: Graph of output voltage vs degrees

```
//arduino mapping function:
```

```
    Int range = map (drx, 0, 1023, 0, 7);
```

```
//
```

0 as the minimum and 1023 as the maximum, will map to 0 and 7 respectively, anything in between scales proportionally. If the wind vane changes direction, we get different analog values between 0 and 1023 which will be scaled to 8 different angles; 0 - 335 degrees.

For the wind speed, each time the cup anemometer completes one revolution, a magnetic reed switch closes and wind speed is determined by counting the number of times the switch closes. In a Davis 6410 cup anemometer, counting the signal in 2.25 seconds will give the result directly in units of mph.

$$Mph = 2.25 \times \frac{counts}{seconds} \quad (3.28)$$

To express wind speed in mps,

$$Mps = Mph \times 0.447 \quad (3.29)$$

3.7.5 Air temperature and air humidity sensor

The air humidity sensor uses a thin film capacitor element whose capacitance changes proportionally with changes in humidity. Inside the sensor housing, there is a dielectric polymer layer that absorbs water molecules from the air and this causes the capacitance to change. The air temperature sensor is a resistive sensor which is based on a thermistor that produces an output resistance which is proportional to temperature. Both sensors can draw a current of 0.14mA.



FIGURE 3.19: Davis air temperature sensor

The Davis humidity/temperature sensor uses the SHT1x humidity and temperature sensor from Sensirion. This is a digital sensor which uses a 2-wire protocol for communications.



FIGURE 3.20: Sensirion SHT1x temperature–humidity sensor

TABLE 3.11: SHT1x pin specifications

Input voltage: 2.4-5.5VDC
Digital 2-wire interface
Precise dew point calculation
Measurement range: 0–100%
Low power consumption: (30W)

TABLE 3.12: SHT1x pin description

1	GND	Ground
2	Data	Serial data, bidirectional
3	SCK	Serial clock, input
4	VDD	Input voltage
5	NC	No connection

3.7.6 Ambient light

The TSL257-LF sensor, fig 3.21 is used to determine the amount of available light for a particular area. Light incident on this sensor is converted to an analog voltage and the Arduino ADC scales this range of voltage to a range of ambient light (lux).



FIGURE 3.21: TSL257-LF light sensor

3.7.7 Battery backup

Battery backup is an important component of a power system in any weather monitoring system and it is indispensable when the weather station is installed in a remote area where there is no mains electricity supply. The batteries used in this project are from Yuasa. Selection criteria of this battery type and capacity have been conducted in the design requirements section (chapter 3.4.2)

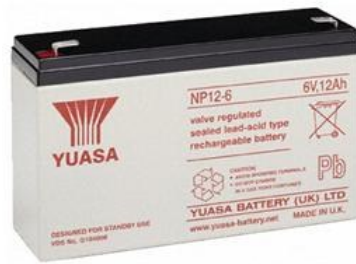


FIGURE 3.22: 12 V 12 Ah Sealed Lead Acid Rechargeable Battery

For the sensor nodes, a 6V 12 Ah sealed lead acid battery has been chosen and a 6V 30 Ah sealed lead acid rechargeable battery was chosen for the GS. The batteries should always be connected to a power source through a charging regulator so as to charge them at the correct charging voltage and to avoid overcharging the batteries. This also ensures longer battery life. In this project, the batteries are connected to a solar panel through a step down DC-DC converter in every sensor node and the GS. In day time when there is sunlight, the battery is charged by the solar panel and during the night or when the sky is overcast, the battery supplies power to the system provided the solar panel provides sufficient power to fully charge the battery.

The specifications of the battery are given in the Table 3.12 below.

TABLE 3.13: Battery specifications

Name	Yuasa NP12-6
Nominal Voltage	6V
Nominal capacity	12Ahr
Approximate weight	2.05kg
Operating temperature range	-20°C to +60°C

3.7.8 Solar Panel

Solar panel is one of the widely used and readily available charging sources for rechargeable batteries. It is an essential power source for remote standalone systems. The solar panel converts solar energy into electrical energy. M405 20W solar panels are used in this project in accordance with the analysis as previously discussed. Figure 3.23 below shows a sample solar panel used for this work.



FIGURE 3.23: Solar Panels.

The selection of this solar panel was based on calculations performed in Chapter 3. The MS405 delivers a peak power of 20 W and a maximum voltage of up to 20V and an open circuit current of up to 1.4A. The solar panel should be selected such that it can sufficiently charge the battery especially during winter or during overcast periods.

3.8 Summary

Chapter 3 has given an overview on the hardware selected. The Arduino Uno was chosen to be the brains of the sensor nodes because of its low cost and the hardware is easily available in South Africa. The Raspberry Pi was chosen to support the GS, despite its higher current consumption, the SBC has a versatile GPIO which enables us to easily integrate with our custom hardware. The following chapter will illustrate how this hardware was applied in our project.

Chapter 4

Implementation

Chapter 3 introduced various hardware selected for the sensor nodes and the GS. This chapter presents more information on how the hardware and software were implemented in this project.

4.1 Hardware design

4.1.1 Node design

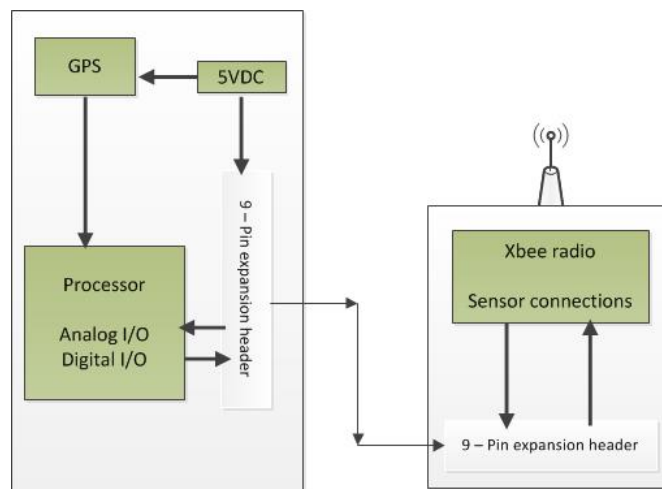


FIGURE 4.1: Sensor node architecture.

The structure of each sensor node is shown in fig 4.1. Each node basically consists of a processor, GPS module, power source, XBee radio transceiver, sensor interface and a 9 pin expansion connector. The sensor node uses an Arduino Uno, which is an 8 bit

micro-controller board with 32Kbytes Flash memory, 1024 Bytes EEPROM and a 10 bit A/D converter. For radio communication, XBee's Series 2 radios are used.

4.1.2 Radio transceiver

The sensor nodes in this network use Digi's Series 2 XBee radios which operate at 2.4GHz with up to 2mW (+3dbm) transmit output power. Transmission range is up to 300m outdoors and 30m indoors. Connectivity is through USART and consumes around 40mA of power. The radio uses DSSS modulation technique [11]. The advantage of using this technique is that the single channel is shared among multiple radios.

4.1.3 Microcontroller transceiver interface

The XBee shield [21], fig 4.2 is used to interface the Arduino board with the XBee radio. The shield is a break out board and the radio can be plugged in or out with ease, thus greatly enhancing the capabilities of the standard Arduino.



FIGURE 4.2: Microcontroller transceiver interface.

The Arduino board operates at 5V and the XBee radio operates at 3.3V so the XBee shield provides the logic level conversions from 5V to 3.3V and vice-versa.

DIN pin on the XBee represents data sent to the XBee. This is serial data to be transmitted and would be connected to Arduino's Tx pin.

DOUT pin on the XBee is data sent from the XBee. This is data which would have been received from another XBee and the pin is connected to the Rx pin on the Arduino.

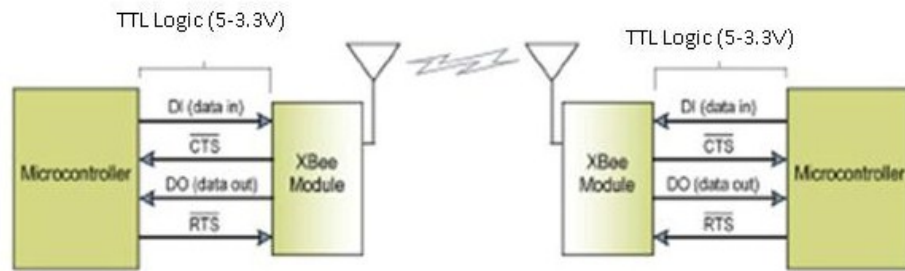


FIGURE 4.3: Microcontroller XBee UART interface.

4.1.4 XBee and sensor power control

One of the main considerations in this project was to minimise power consumption. The XBee consumes around 40mA of power, and to prolong battery life, the XBee has to be switched off when inactive. To accomplish this task, a time activated switching mechanism has to be employed. The microcontroller keeps checking time information from the GPS module and when its reached a certain time, the microcontroller energises a transistor switch which then connects power to the radio and sensors. When data transmission is completed, the microcontroller deactivates the transistor switch which then switches off power to the radio and sensors.

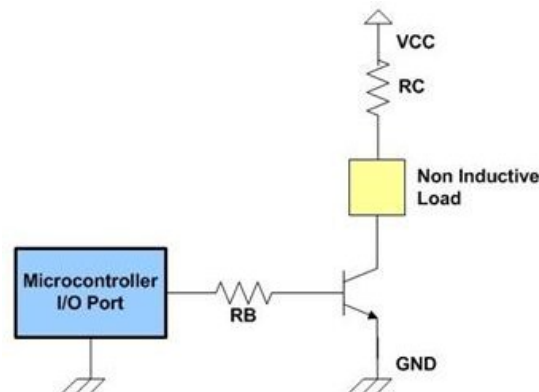


FIGURE 4.4: Sensor power control circuit.

To determine the value of R_B , we first need to determine the collector current when the switch is ON, this is the current consumed by the radio and sensors and is roughly 45mA.

$$\therefore i_c = 45mA \quad (4.1)$$

The *2N2222* has an h_{fe} of 100,

The base current should then be $I_c/h_{fe} = 0.45mA$. We will add a factor of two to make sure the transistor fully turns ON and the new I_c becomes 0.9mA. When the transistor is fully turned ON, there will be a 0.7 voltage drop across the base emitter junction.

Therefore $R_B = V/I_b = (5V - 0.7V)/0.9mA = 472\Omega$

When the microcontroller output is 0V, no base current flows and the transistor is OFF.

When the input to the base resistor is 5V, the base current is 0.45mA and turns on the transistor putting VCC across the load.

When using an inductive load such as a relay, a flyback diode might be used. This is used to protect the load by blocking sudden voltage spikes when the quickly is suddenly removed.

4.1.5 Power dissipation

The power dissipated in the transistor is the voltage drop across the collector emitter junction multiplied by the collector current. $V_{CE} = 1V \times I_C \approx 45mW$. A heat sink is rarely required as little power is rarely dissipated and full power is dissipated to the load.

4.1.6 Enclosure

A 318x238x101mm (ABS) IP65 plastic enclosure was used to house the PCBs and 6V battery.

4.2 System testing

The project consists of four major blocks which are:

1. Sensor nodes
2. Ground station

3. Iridium gateway
4. Master station

The following diagram shows the various tests performed on the four major blocks listed above.

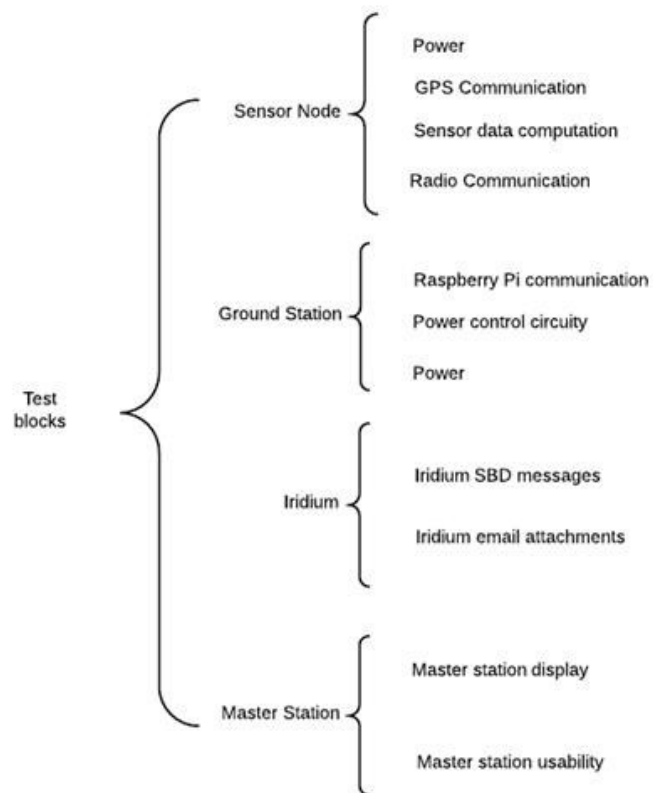


FIGURE 4.5: Project functional blocks.

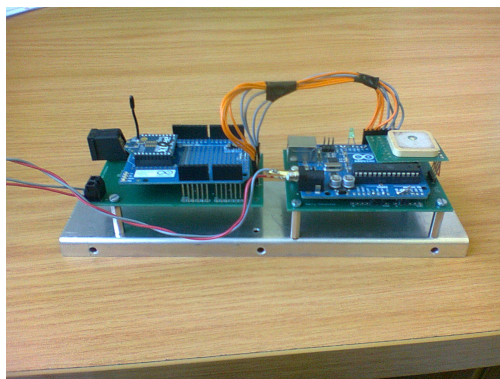


FIGURE 4.6: Sensor node circuitry.

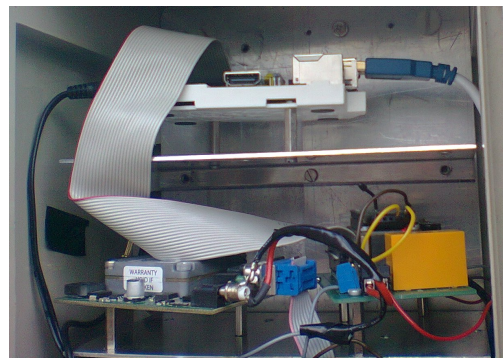


FIGURE 4.7: Ground station circuitry.

Figures 4.6 and 4.7 show the complete interconnection diagrams of the sensor node and the ground station respectively.

4.3 Sensor node development platform

Arduino is both an IDE and a hardware development platform for creating applications on 8 – bit Arduino boards which incorporate Atmel ATmega microcontrollers. Applications can be created using high level C++ [20]. The IDE is an easy to use package with an integrated AVR-GCC compiler which converts human readable code into machine readable instructions. The instructions are then linked against Arduino libraries that provide functions like *AnalogRead()* or *delay()*. This produces a single hex file which is then written to the program memory of the ATmega microcontroller on the board.



FIGURE 4.8: Arduino IDE.

The software architecture applied in the sensor nodes was written using Arduino software and its based on Round Robin architecture. This is the simplest architecture which does not use interrupts. It does not have priority or pre-emption to control events. Software which controls events is written in an infinite main loop which runs as long as power is applied to the sensor node and tasks are executed in sequence.

The advantage of this approach is its simplicity and short development cycle but the disadvantage is that this architecture cannot support complex problems

```
//  
void loop() {  
  if(task1 needs service){  
    do task 1;  
  }  
  if(task2 needs service){  
    do task 2;  
  }  
  
  if(task3 needs service){  
    do task 3;  
  }  
}  
  
//  
void loop() {  
  check_gps_time();  
  read_data();  
  perform_calculations();  
  switch_ON_radio_sensor();  
  send_data();  
  switch_OFF_radio_sensor();  
}
```

FIGURE 4.9: Round robin loop.

4.3.1 ADC Implementation

The ADC is used to convert a continuously varying analog voltage to a 10-bit digital value. The Arduino UNO has a built in 10-bit resolution ADC with 6 multiplexed ADC channels [22]. The ADC supports two operating modes: single conversion and free running mode [22]. In single conversion mode, the ADC is initiated each time for conversion and the result is moved to the data register while in free running mode, the ADC is initiated only once it continuously converts as soon as it completes converting the previous reading. We will operate the ADC in free running mode.

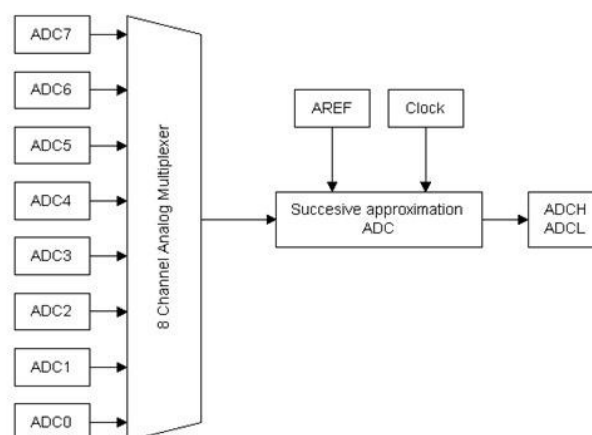


FIGURE 4.10: ADC block diagram.

The ADC contains an 8 channel multiplexer to select between the analog inputs. After each successful conversion, the result is moved to the registers ADCH and ADCL. It

takes two registers to store the result because the registers are 8 bit wide and the data is 10 bit wide. The ADC is 10 bit wide so we get 1024 (2^{10}) possible values.

After a successful conversion, the value of the ADC can be found in the registers by:

$$ADC = \frac{V_{in} \times 1024}{V_{ref}} \quad (4.2)$$

Where V_{in} is the voltage on the selected analog channel and V_{ref} is the voltage reference.

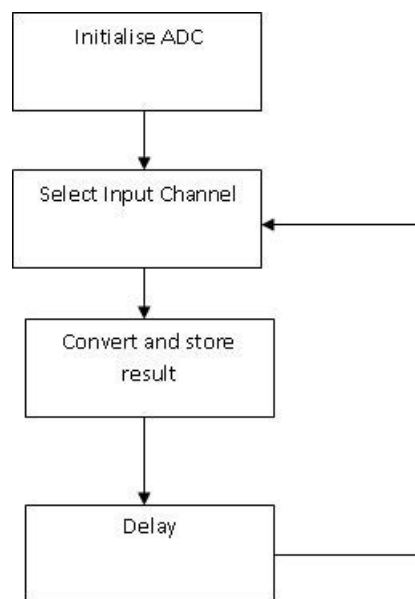


FIGURE 4.11: ADC Flowchart.

4.3.2 UART and Software Serial Implementation

The ATmega 328p has built in UASRT hardware [22]. The UASRT allows the MCU to have a serial port and this can be used to upload programs to the chip or it can be used to establish a communication link between the MCU and a computer or other devices which supports RS232 protocol. In order to use the UASRT, the user has to initialise and set the speed of the serial port and the Arduino software makes it easy to implement by using libraries for initialising and using the serial port. Examples of some functions available include:

```
Serial.begin()
```

```
Serial.available()  
Serial.read()  
Serial.print()  
Serial.write()
```

In some applications, we might have one or more devices which use the RS232 protocol. The standard Arduino Uno board has only 1 serial communication port. The Software Serial library was developed to allow the standard Arduino board to have multiple serial ports on other digital pins and in our project, we can connect a GPS modem on the Software serial port and the XBee radio can be connected to the hardware serial port. The hardware serial port can be used for programming the MCU.

4.3.3 Sensor node software Implementation:

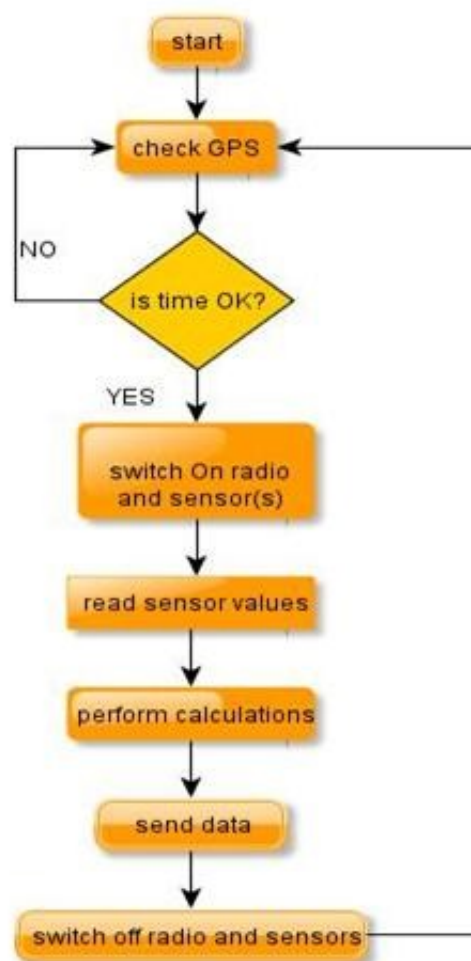


FIGURE 4.12: Sensor node flow diagram.

4.3.4 Ground station software architecture

The ground station hardware is based on a Linux CPU. Instead of writing all the code in one source file like the Round Robin architecture, Linux OS allows us to write independent scripts which are executed once the CPU reboots.

There are three scripts running, one controls the USB port and uses the Libserial library. The other script controls the UART on the Raspberry Pi GPIO and it uses ArduPi library. The final script is responsible for switching the modem and this is accomplished through the Raspberry Pi GPIO as well. All the three scripts, fig 4.13, automatically execute when the Raspberry Pi starts or reboots.

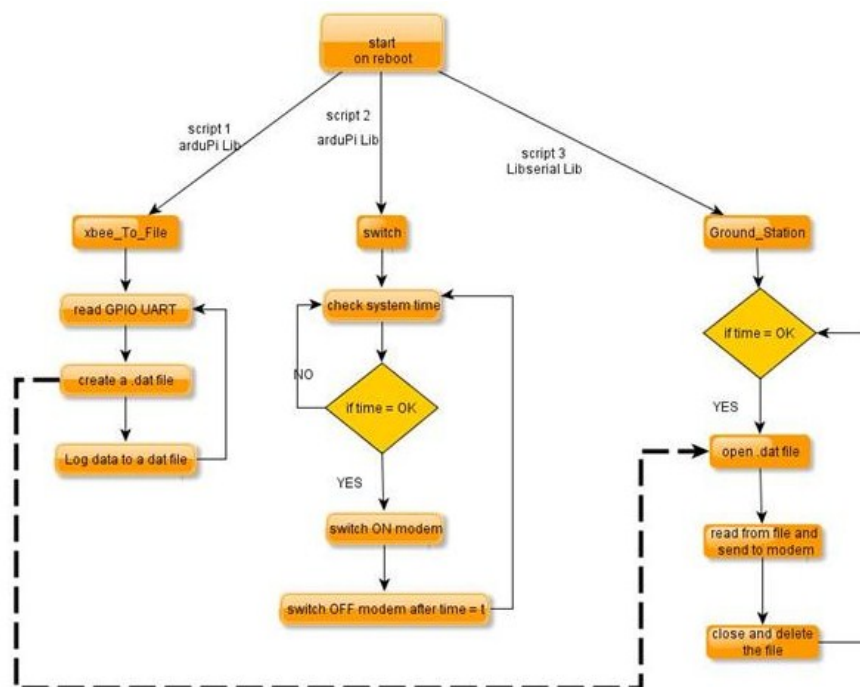


FIGURE 4.13: Ground station flow diagram.

4.3.5 GUI software

To display the sensor data, a master station GUI was developed. The GUI was written in PHP and can be used on both Windows and Linux operating systems. Explanation of GUI software development and installation is provided in Appendix D1.

4.4 Radio Communication and networking

The communication device used in this project is XBee RF radio module. The module takes in serial data from a host device using a UART and communicates in the 2.4GHz spectrum. The module can achieve 300m line of sight range and 30m non line of sight range. The devices come in 4 different packages; Series 1, Series2, Series 1 Pro and Series 2 Pro.

We chose the XBee Series 2 module due to its powerful built in mesh networking capabilities, longer range and a lower power consumption of just 40mA.

4.4.1 Hardware and electrical characteristics

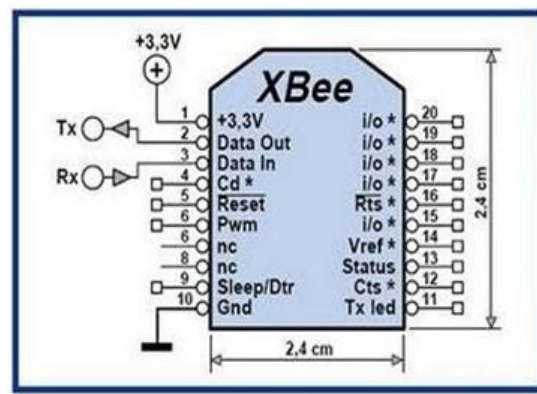


FIGURE 4.14: XBee pin-out.

As stated earlier, the RF module can communicate with any UART enabled host device or to a serial device via a level translator. Pin assignments and applications for the XBee module are shown below.

Communication begins when the module receives asynchronous serial data through the UART DIN pin. Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high) [23].

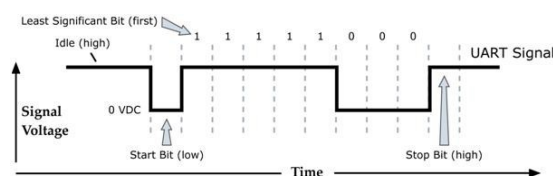


FIGURE 4.15: Example data format [23].

TABLE 4.1: XBee pin description.

Pin	Name	Direction	Description
1	Vcc	-	Power supply
2	Dout	Output	UART data out
3	Din	Input	UART data in
4	D08	Output	Digital output 8
5	RESET	Input	Module reset
6	PWM0/RSSI	Output	PWM output 0 / Rx signal strength indicator
7	PWM1	Output	PWM output 1
8	(reserved)	-	-
9	DTR/SLEEP_RQ/D18	Input	Pin sleep control line / digital input 8
10	GND	-	Ground
11	AD4/DIO4	Either	Analog input 4 or digital I/O 4
12	CTS/DIO7	Either	Clear to send flow control or digital I/O 7
13	ON/SLEEP	Output	Module status indicator
14	VREF	Input	Voltage reference for AD inputs
15	Associated/AD5/DIO5	Either	Associated indicator, analog input 5 or digital I/O 5
16	RTS/AD6/DIO6	Either	RTS flow control, analog input 6 or digital I/O 6
17	AD3/DIO3	Either	Analog input 3 or digital I/O 3
18	AD2/DIO2	Either	Analog input 2 or digital I/O 2
19	AD1/DIO1	Either	Analog input 1 or digital I/O 1
20	AD0/DIO0	Either	Analog input 0 or digital I/O 0

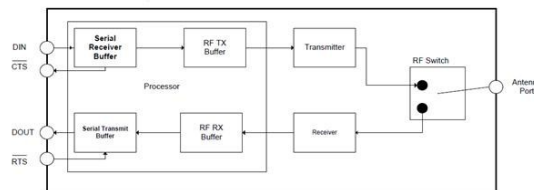


FIGURE 4.16: XBee Internal data flow.

Figure 4.15 showing transmission of a data packet.

Fig 4.16 above shows the path taken when the module receives serial data through the DIN pin from the host device to the point when the data is transmitted via the antenna port. The diagram also clearly shows the path taken when the module receives RF data through the antenna port to the point it is transmitted to the host device through DOUT pin as serial. The clear to send (CTS) pin provides flow control mechanism and

indicates to the host device whether or not to send serial data. The RTS flow control enables the host device to indicate to the module not to send out of its UART [23].

4.5 Data collection in WSN

4.5.1 Node discovery

This section discusses the different phases of data collection of the WSN. Node discovery is the first step for data collection in a WSN. As described in chapter 2.16, a Zigbee device searches for available networks when first powered up. After it has found a valid network, it joins given that the parent device is allowing joining. It is issued with a randomly generated 16bit network address which will allow it to send and or receive data from the network.

4.5.2 Data transfer

Data transfer immediately follows after discovery and it is the process where a sensor node communicates with its one-hop neighbour node. Sensory information is transferred depending on the transfer protocol being used. In some cases, the sensor nodes use routing tables to determine how data packets are routed through the network.

4.5.3 Routing to the destination

The final stage in data collection is routing the data to the destination, this can be done by another router or the coordinator. This is accomplished using different routing protocols in use. The complete processes and detailed explanation of the protocols is covered in chapter 2.16.

Fig 4.19 above shows the functional overview of the WSN and the GS. These units are powered by a sealed lead acid rechargeable battery. All the sensor nodes in the WSN are powered by a 6V 12AH battery and the GS is powered by a 6V 30AH battery. Since electricity is not available in the area, 20W solar panels are used as a charging source for the sensor nodes and the GS. A charging regulator (UC3906) is connected to both solar panel and battery as well. Different kinds of environmental sensors are connected

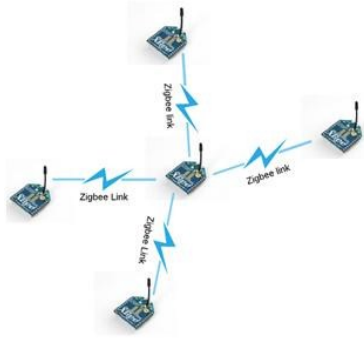


FIGURE 4.17: XBee star configuration.

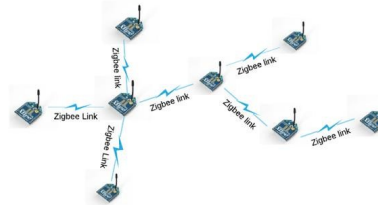


FIGURE 4.18: Mesh configuration.

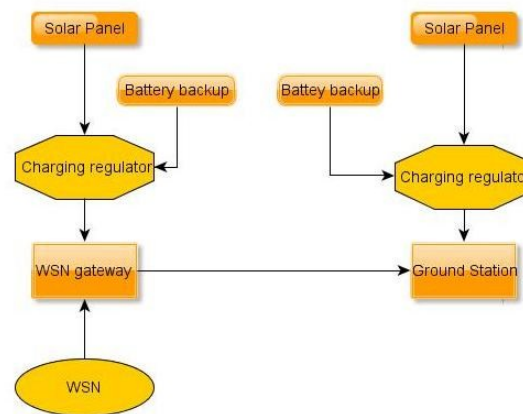


FIGURE 4.19: Power circuitry functional overview.

with the sensor nodes. There is also a Linux based Raspberry Pi SBC connected to the GS.

In the day time, the solar panel supplies power to the system and charges the battery as well. Battery gives power to the system during night and also that time whenever the solar panel is not powering the system due to insufficient sunlight. In figure 4.19 above, different colors are assigned for the lines connecting the blocks, red lines indicate the flow of power and blue lines indicate a communications link.

Every 60 minutes, data is collected by the sensor nodes. The collected data is then relayed to the ground station for storage via a Zigbee network. After every 24 hours, the ground station sends the stored data to the master station through a network of Iridium satellites. Finally, data is then relayed through the internet for display on the master station.

4.6 File types

The ground station stores sensory information in a .dat format. When it is time to send data to the Iridium modem, a software script running in the SBC opens the file, reads the file and sends data through the serial port. The Iridium mail attachment can be opened with any text editor.

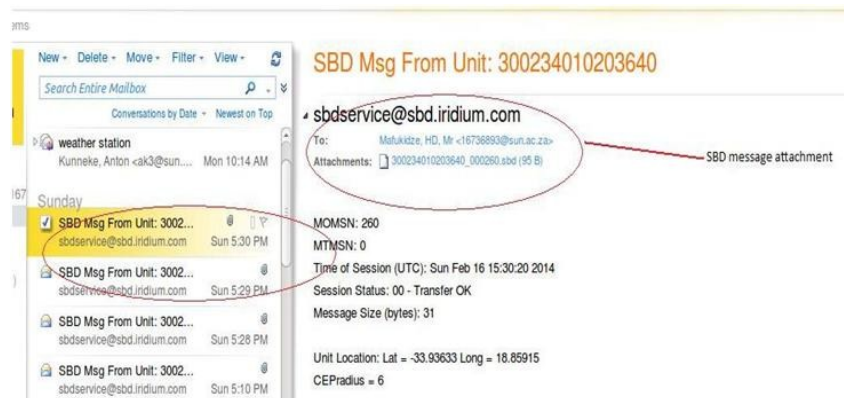


FIGURE 4.20: Example of iridium mail attachment.

```

1 005,004,0.40,22,201402111459
2 005,005,36.80,91,201402111459
3 005,007,-33.35,91,201402111459
4 005,002,0.56,90,201402111527
5 005,003,2.68,90,201402111527
6 005,004,99,90,201402111527
7 005,004,0.80,23,201402111559
8 005,005,31.28,88,201402111559
9 005,007,-44.28,88,201402111559
10 005,001,225,89,201402111631
11 005,002,0.52,89,201402111631
12 005,003,2.24,89,201402111631
13 005,004,99,89,201402111631
14 005,008,25.33,87,201402111659
15 005,009,56.37,87,201402111659
16 005,004,1.80,22,201402111659
17 005,005,22.76,87,201402111659
18 005,007,-69.88,87,201402111659
19 005,001,45,88,201402111736
20 005,002,0.38,88,201402111736
21 005,003,2.24,88,201402111736
22 005,004,99,88,201402111736
23 005,008,23.40,86,201402111759
24 005,009,63.76,86,201402111759
25 005,004,1.80,23,201402111759

```

FIGURE 4.21: Comma separated txt file.

The database only accepts comma separated .txt files

4.7 Solar panel tilt angle

To get the maximum power from a solar panel, an automated solar tracking system or a fixed solar panel can be installed. Each system has its own advantages and disadvantages, this project, however, uses fixed photo-voltaic systems. Photo-voltaic cells need maximum amount of sunshine to produce maximum amount of electric power. To achieve this, the solar panels need to be pointed in the direction of the sun to harvest the maximum amount of solar power. This angle changes throughout the year and it depends on the season. The equations below summarise the design procedure of obtaining the correct angles for aligning solar panels.

The tilt angle β of a solar panel required at any season can be expressed as a function of the latitude as follows:

$$\beta_{forwinter} = (Latitude \times 0.9) + 29degrees \quad (4.3)$$

$$\beta_{forsummer} = (Latitude \times 0.9) - 23.5degrees \quad (4.4)$$

$$\beta_{forspring} = Latitude - 2.5degrees \quad (4.5)$$

where:

β is the tilt angle of the PV system.

The equations above were used to determine the solar panel tilt angles for different seasons throughout the year. These angles enable the solar panel to produce a maximum amount of power in relation to incident solar radiation.

4.8 Measuring Arduino's power source

The challenge explained in this section is to measure the voltage bank of the sensor node. This is important in that the value indicates an approximate remaining life time and provides easy debugging in case a node dies. Installing another system might be expensive and bulky so the sensor node has to measure its own battery level. Using the conventional *analogRead* function of the Arduino does not give the actual value in that, by default, the Arduino references its voltages between 5V and GND. If the positive terminal is connected to the analog pin, we will be measuring the battery voltage against itself and irrespective of the battery level, we will always get 100% battery level recorded by the Arduino.

Other options to measure battery voltage exist, for example, incorporating op-amps but the easiest, cheapest and more accurate solution is to make use of the Arduino's inbuilt 1.1V reference voltage. The supply voltage is measured using the ADC and it is referenced against the known internal voltage reference and the result is an accurate reading of the supply voltage.

4.9 Master Station

The master station is a software interface allowing processing and display of data coming from ground stations. Each ground station is equipped with an Iridium short burst data transceiver module. The module allows sending messages to the Master Station using the Iridium Satellite System which is composed of 66 low-earth-orbit satellites.

4.10 Installation

- Install Xampp on your computer
- Navigate to C:\xampp\htdocs
- Copy the ForestryDp folder inside the htdocs directory
- Open your web browser

- Enter the URL :<http://localhost/phpmyadmin/>
- Edit dbconnection.php by inserting your own database parameters

Example:

```
$dbtype = 'mysql';  
$dbname = 'forestdp';  
$dbhost = 'localhost';  
$dbuser = 'root';  
$dbpwd = '';
```

Note:

1. The interface needs a writeable directory to export data; the default configured is “ForestryDp\exportedfiles”.
2. For mail processing, the port used by your email address has to be open (for example; Gmail uses port 993).

4.11 Interface Architecture

4.12 Connection

4.12.1 Index.php

The user is asked to enter login credentials

4.12.2 userconnection.php

This page compares login credentials provided by the user with the information stored in the database. It also configures session variables.

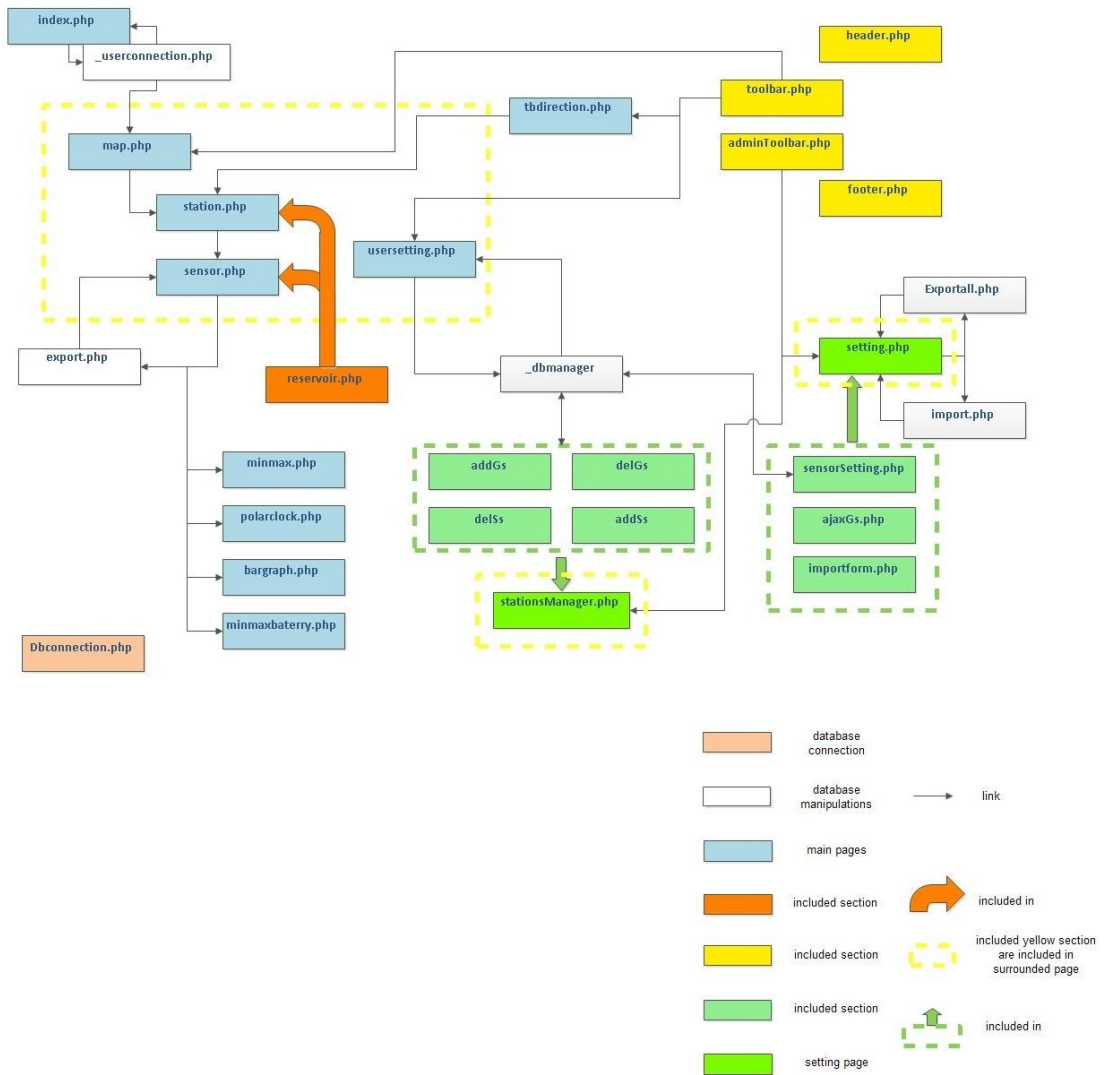


FIGURE 4.22: Interface architecture.

4.13 Include

4.13.1 Header.php

The Header.php page contains the image of the header.

4.13.2 Footer.php

This page contains the image signature of the footer

4.13.3 **Toolbar.php**

This page allows the user to access the station page or to access to user settings

4.13.4 **AdminToolBar.php**

It gives access to the same links than toolbar.php, but also to setting.php and Station-Setting.php.

4.14 **Main Pages**

All the following pages include header.php, footer.php and a toolbar which can be adminToolBar.php for administrator or toolbar.php for other users.

4.14.1 **Map.php**

On this page, a map is displayed with pointers placed on locations of ground stations. Under the map, there is a summary of all ground stations. This page makes use of the GoogleMapAPI.class.

4.14.2 **Station.php**

This page gives information about sensors belonging to a chosen ground station. It displays a summary of sensors linked to this station. For each sensor, information such as the last measurement, date and battery state are displayed.

4.14.3 **Sensor.php**

This page displays the measurements. As default, the last 100 measurements are posted, but a user can choose the period where he requires the measurements. The user can also ask for a graph display. The graph display is specific to each sensor type.

4.15 Graphs

4.15.1 Minmax.php

This page allows the display of minimum and maximum curves depending on user demand. It can be a Daily, Weekly, Monthly or Yearly display. This kind of curve is used for temperature, light and wind speed sensors.

Include:[jpgraph-3.5.0b1](#)

4.15.2 Minmaxbattery.php

This page allows the display of minimum and maximum curves depending on user request. It can be a Daily, Weekly, Monthly or Yearly display. This page is used only to sort battery measurements.

Include:[jpgraph-3.5.0b1](#)

4.15.3 Polarclock.php

This page allows the display of angular curves depending on user demand. It provides a Daily, Weekly and Monthly average. This kind of curve is used for wind direction sensors.

Include:[jpgraph-3.5.0b1](#)

4.15.4 Bargraph.php

This page allows the display of bar graphs depending on user demand. It can be a Daily, Weekly, Monthly or Yearly display. This kind of curve is used for rain fall sensors.

Include:[jpgraph-3.5.0b1](#)

4.15.5 Reservoir.php

This page allows displaying a graphic reservoir corresponding to battery value.

4.16 Settings includes

This section describes the pages included in the settings window.

4.16.1 AddGs.php

This form asks and transmits all information necessary to add a ground station.

4.16.2 DelGs.php

This form asks and transmits all information necessary to delete a ground station.

4.16.3 Addss.php

This form asks and transmits all information necessary to add a sensor.

4.16.4 Delss.php

This form asks and transmits all information necessary to delete a sensor.

4.16.5 Importform.php

This form asks and transmits all information necessary to upload data to the database.

4.16.6 AjaxGs.php

This page allows a dynamic form for the importation form. When a user chooses a ground station, sensors proposed to him are only those which belong to this ground station.

4.16.7 SensorSetting.php

This form asks and transmits all information necessary to modify or create a sensor type.

4.17 Settings

All the following pages include header.php, footer.php and a toolbar which can be adminToolBar.php for administrator or toolbar.php for other users.

4.17.1 UserSetting.php

This setting page is accessible to any user. It allows them to change their password.

4.17.2 StationSetting.php

This setting page is accessible only to administrator user. From there, the administrator can add or delete a ground station, and add or delete a sensor from a ground station.

Include: [addGs.php](#), [delGs.php](#), [adds.php](#), [delSs.php](#)

4.17.3 Setting.php

This settings page is accessible only to the administrator user. From there, the administrator can add or delete a user. He can also add or modify a sensor kind. Concerning data, he can import measurements in the database for a chosen sensor. He can also export all data from the database.

Include: [Importform.php](#), [ajaxGs.php](#), [sensorsetting.php](#)

4.18 Database actions

4.18.1 dbmanager.php

This page is used for all sql modifications in database.

4.18.2 Import.php

This page allows a text file to be saved to the database.

4.18.3 Export.php

This page allows producing a file containing data chosen by the user

4.18.4 Exportall.php

This page allows producing of folders containing files with all data

4.19 Mail

4.19.1 Mail.php

This file is automatically refreshed every 10 minutes. It allows mailbox opening and checking.

4.19.2 MailProcessing.php

This page contains the “sbdProcessing” function which allows the processing of short burst data messages sent from the ground Station.

4.20 Database Architecture

4.20.1 User

This table contains all the information concerning users.

Type value can be 0 (Administrator), 1 (Master), 2 (User).

4.20.2 Sensor

This table contains all sensor types used by the interface. Each type includes a numerical value which is used to differentiate sensors.

Four displays are available:

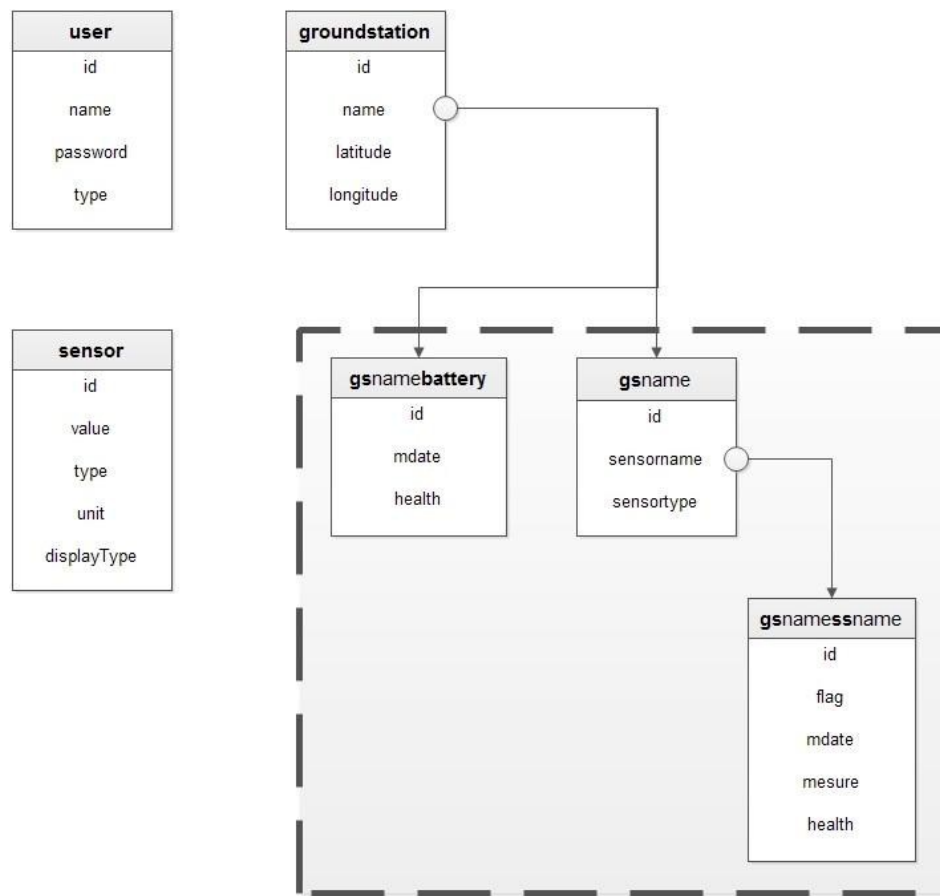


FIGURE 4.23: Database architecture.

id	name	password	type
1	admin	admin	0
2	user1	user1	1
5	user2	user2	2

FIGURE 4.24: User information.

- Min Max curve (1)
- Polar clock (2)
- Bar Graph (3)
- Reservoir (4)

id	value	type	unit	displayType
1	1	Temperature	Å° C	1
2	2	Light	lux	1
3	3	Wind Direction	Å°	2
4	4	Wind Speed	km/h	1
5	5	Rain Fall	cm	3
6	6	Battery State	%	4

FIGURE 4.25: Sensor information.

4.20.3 Ground station

This table contains name and position of all ground stations.

id	name	latitude	longitude
34	gs001	-33.9202	18.8972
36	gs002	-33.924	18.8882

FIGURE 4.26: Ground station layout.

4.20.4 Gs name

For each ground station, a table is created. The name of the table automatically begins with “gs”, followed by the name of the ground station. This table contains all sensor names and sensor types linked to this ground station.

Example:

id	sensorname	sensortype
3	gs001ss001	1
4	gs001ss002	2
5	gs001ss003	3
6	gs001ss004	4
7	gs001ss005	5

FIGURE 4.27: Sensor name layout.

4.20.5 Gsname battery

For each ground station, a table is created. The name of the table automatically starts with “gs”, and then the battery voltage follows. This table contains all battery measurements of the sensor nodes.

Example:

id	mdate	health
1	2011-01-01 10:00:00	56
2	2012-08-08 10:00:00	86
3	2012-08-06 10:00:00	96

FIGURE 4.28: Date of measurement and state of the sensor.

4.20.6 Gsname ssname

For each sensor, a table is created. The name of the table automatically starts with “gs”, followed by the name of the ground station, and then “ss”, followed by the name of the sensor. This table contains all data (dates, measurements, battery level and working flag) sent by this sensor.

Example:

id	mdate	mesure	health	flag
85	2012-11-08 10:10:00	8.4460000991821	40	0
86	2012-11-08 12:10:00	2.4460000991821	10	0

FIGURE 4.29: Value and date of the sensor.

4.21 Mail processing

Each ground station is sending data frames using the format in fig 4.30.

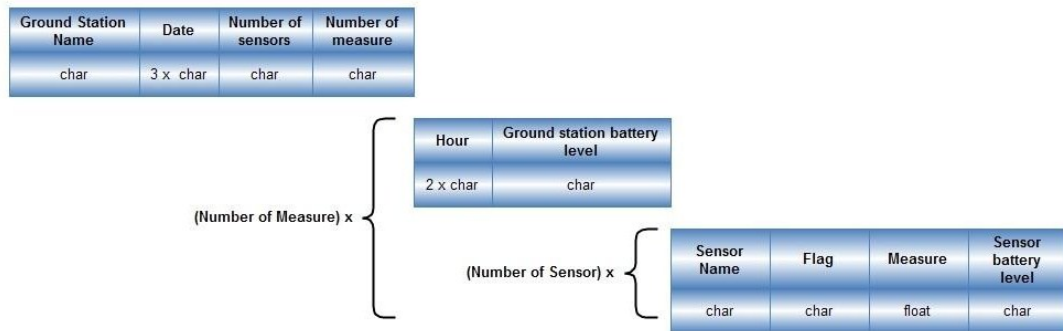


FIGURE 4.30: Mail processing architecture.

4.21.1 Mail setting

The administrator has to set in “Settings” section, an email address. This is the email address where Iridium is going to send message.

Mail Parameters

Server : Type :

Port :

Login :

Password :

FIGURE 4.31: E-mail credentials.

The reception port has to be open to be able to receive email.

4.21.2 Usage

In opening the page *ForestryDp/mail/mail.php*, the master station checks in the chosen mailbox if there are new emails. For each new email, the system checks for the Iridium mail characteristic (SBD format). Then if the mail has an attachment file, it downloads and processes it. To be completely processed, the file has to follow the specified frame format. If the message is correct, the data is added to data base.

This page has to be open every time.

Mail Reception

```
29-08-2012 13:34 Processing Start
ffffff
Ground Station gg002 Message from : 2012-11-8
Processing End

29-08-2012 13:34 Processing Start
Re: TP d'Automatique note
      Invalid Frame
Processing End

29-08-2012 13:34 Processing Start
=?ISO-8859-1?Q?Re: _C9preuve_de_rattrapage?=
      Invalid Frame
Processing End

29-08-2012 13:34 Processing Start
Re: TP d'Automatique note
      Invalid Frame
Processing End
```

FIGURE 4.32: Mail reception information.

To be recognized, a ground station or a sensor has to be added by the administrator before any data reception.

4.21.3 Error Flag

If the ground station has displayed an error flag of 255, that means there is a communication issue with the sensor.

In the database, if the flag displayed is equal to:

- 255, that means that there is a communication issue with the given sensor.
- 254, that means that the received battery value is impossible (not between 0 and 100).
- 253, that means that the received measurement is impossible.
- 252, that means that the received battery value is impossible and the received measurement is impossible.

Password :

New Password :

Confirm New Password :

FIGURE 4.33: Change password page.

4.22 User guide

4.22.1 Change password

To change the password, any user has to provide his old password and then enter a new one. To complete the change, the new password has to be confirmed.

4.22.2 Map

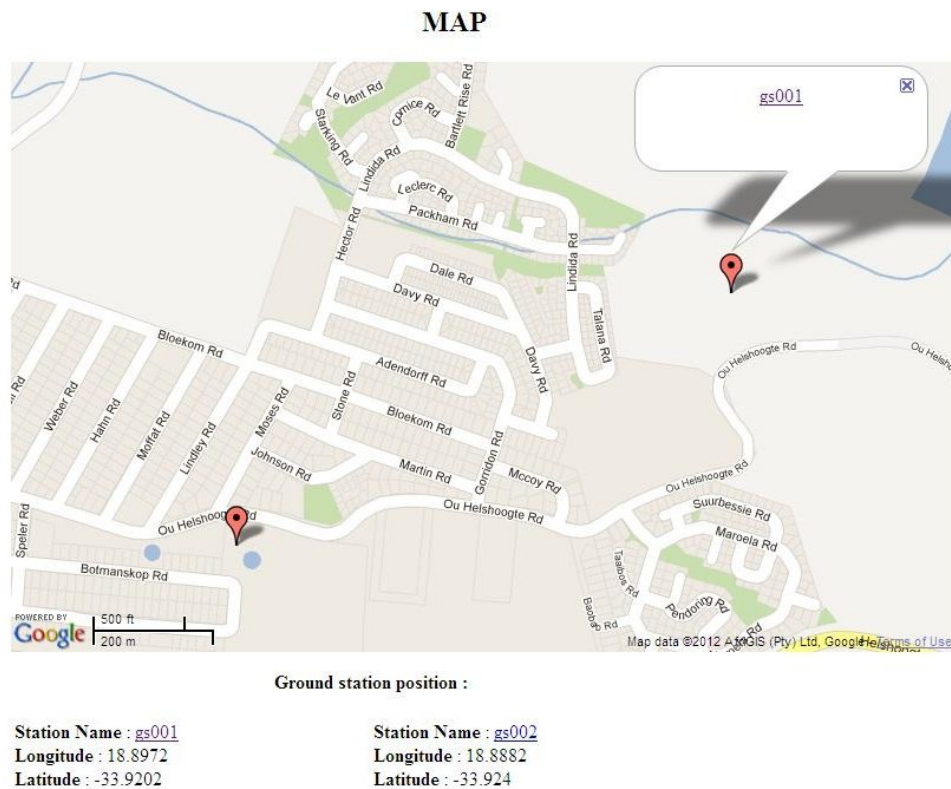


FIGURE 4.34: Main page showing map and location of the GS.

The “Map” page allows viewing the ground station locations and observing more details of their linked sensors by clicking on one of the sensor icons. The page also displays the last known state of every battery.



FIGURE 4.35: GS, battery level and warning indicators.

For each sensor, an indicator gives its current battery state with the color code as shown in fig 4.35 above. If an issue is detected, a red warning sign will be displayed:

4.22.3 Station

The “Station” page shows all sensors linked to the selected ground station. For each sensor, its type, last measurement and its battery state are given. More measurements of a sensor are available by clicking on its icon.

4.22.4 Sensor

4.22.4.1 Curve Display

Measurements can be observed in a graph. According to sensor type, the type of graph is different. Temperature, light and Wind speed are displayed with a Minimum and Maximum curve. Wind direction is displayed with a polar display. Rain fall is displayed with a bar graph. Furthermore four types of period are available:

- Daily: every measurement recorded during the day is displayed
- Weekly: measurements for each day of the week are displayed
- Monthly: measurements for each week of the month are displayed
- Yearly: measurements for each month of the year are displayed

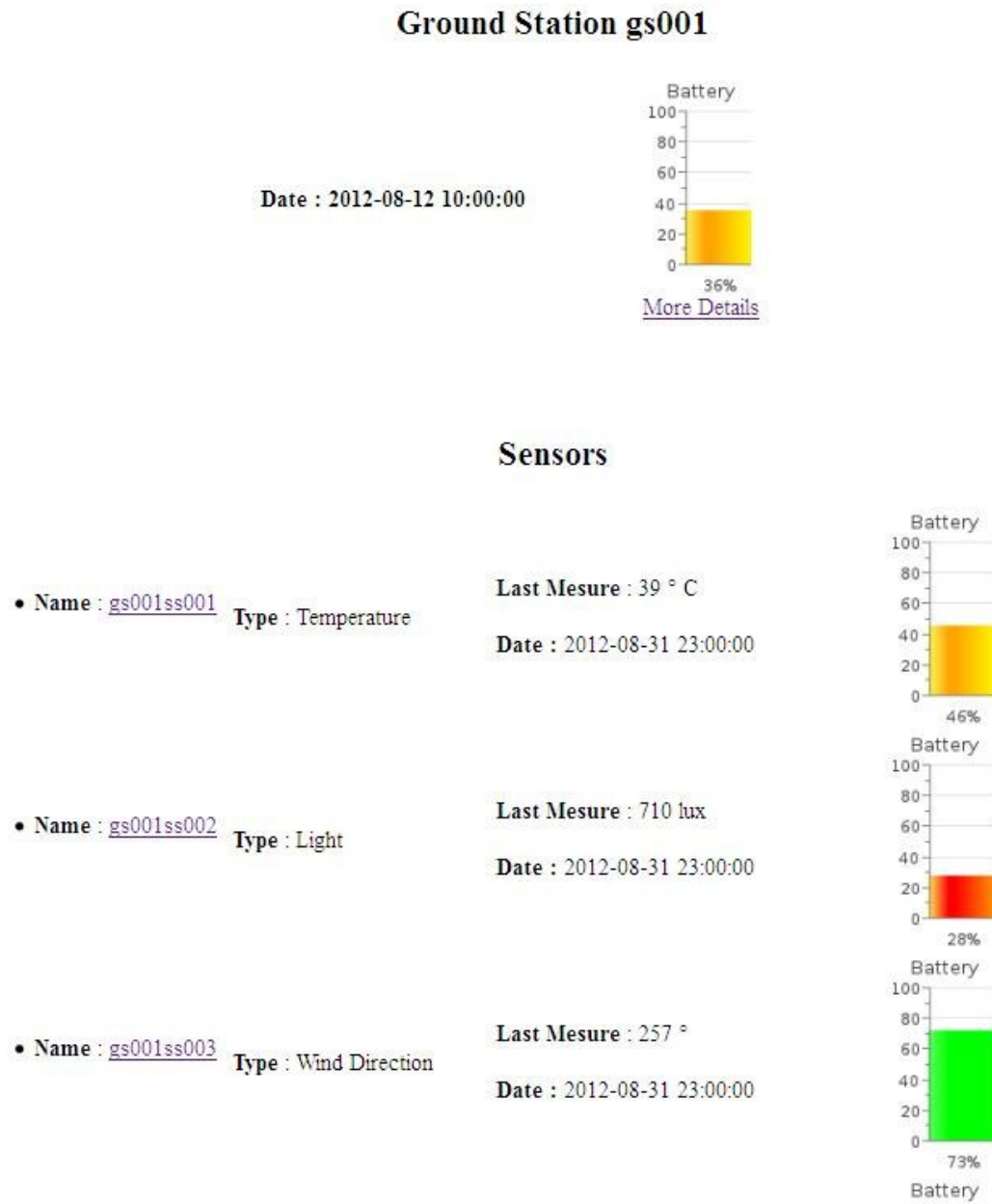


FIGURE 4.36: Page showing available sensors and their respective battery levels.

The user can choose the last day of the period, by default it is the current day.

4.22.5 Date selection

With this function, a user has got the possibility to observe data sent by a sensor between two chosen dates. A maximum number of samples has to be given.

Curve display

Daily ▾

Date :

Until Today

Until 1 ▾ - 1 ▾ - 2010 ▾

Display

FIGURE 4.37: Curve display dates selection page.

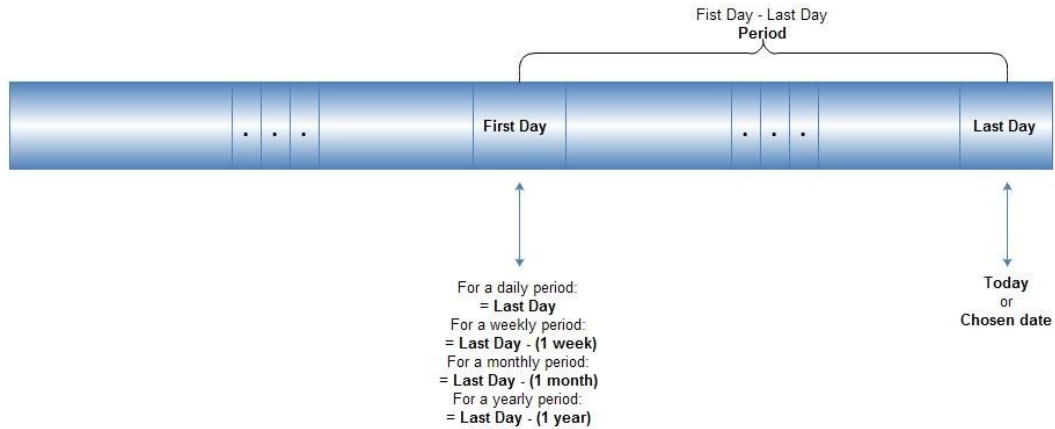


FIGURE 4.38: Dates selection layout.

Dates Selection

From 7 ▾ - 4 ▾ - 2012 ▾ To 8 ▾ - 4 ▾ - 2012 ▾
(DD-MM-YYYY)

Samples Number 25

Change

Mesures from 07-04-12 to 08-04-12

Date	Mesure	Sensor Battery Level
2012-04-08 10:00:00	26	29
2012-04-07 10:00:00	-1	70

FIGURE 4.39: Dates selection page.

The previous data table, fig 4.40, can be exported in a txt file. In each line, data is separated by tabs (“\t”) and each line is separated by a line break (“\n”). The file is

Mesures from 07-04-12 to 08-04-12

Date	Mesure	Sensor Battery Level
2012-04-08 10:00:00	26	29
2012-04-07 10:00:00	-1	70

FIGURE 4.40: Page to export data.

available in the “*exportedfiles*” folder of the *ForestryDp* folder.

Its name is *[sensorname]_ExportData_[Curent Date].txt*”

```

01/01/2011 10:00:00 20 100
02/01/2011 10:00:00 18 93
03/01/2011 10:00:00 19 88
04/01/2011 10:00:00 20 45
05/01/2011 10:00:00 19 72
06/01/2011 10:00:00 18 73
07/01/2011 10:00:00 17 63
08/01/2011 10:00:00 16 48
09/01/2011 10:00:00 14 -1
10/01/2011 10:00:00 15 57
11/01/2011 10:00:00 14 25
12/01/2011 10:00:00 16 94
13/01/2011 10:00:00 16 79
14/01/2011 10:00:00 16 47
15/01/2011 10:00:00 14 27
16/01/2011 10:00:00 13 73
17/01/2011 10:00:00 11 68
18/01/2011 10:00:00 11 70

```

FIGURE 4.41: Example of a text file to export.

This type of file can be read with a spreadsheet software.

4.23 Display

4.23.1 Min Max

This type of display shows the minimum and maximum values for a chosen period.

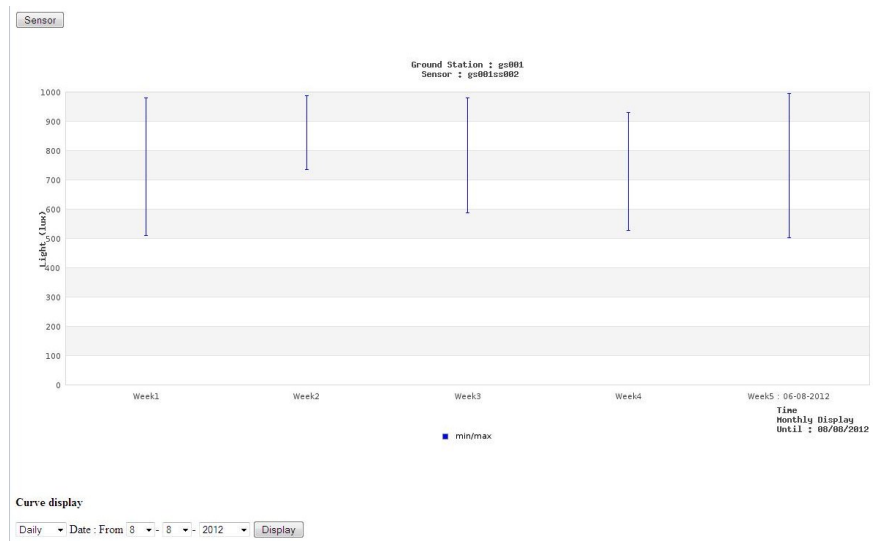


FIGURE 4.42: Min Max display.

4.23.2 Bar Graph

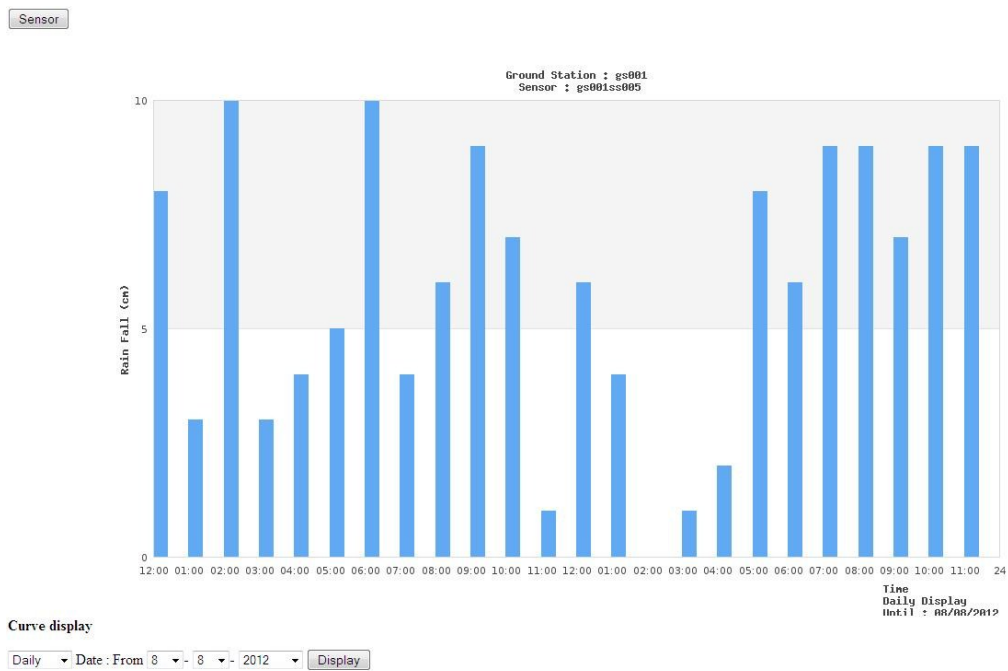


FIGURE 4.43: Bar graph display.

4.23.3 Polar clock

This kind of display displays the average value for each chosen day.

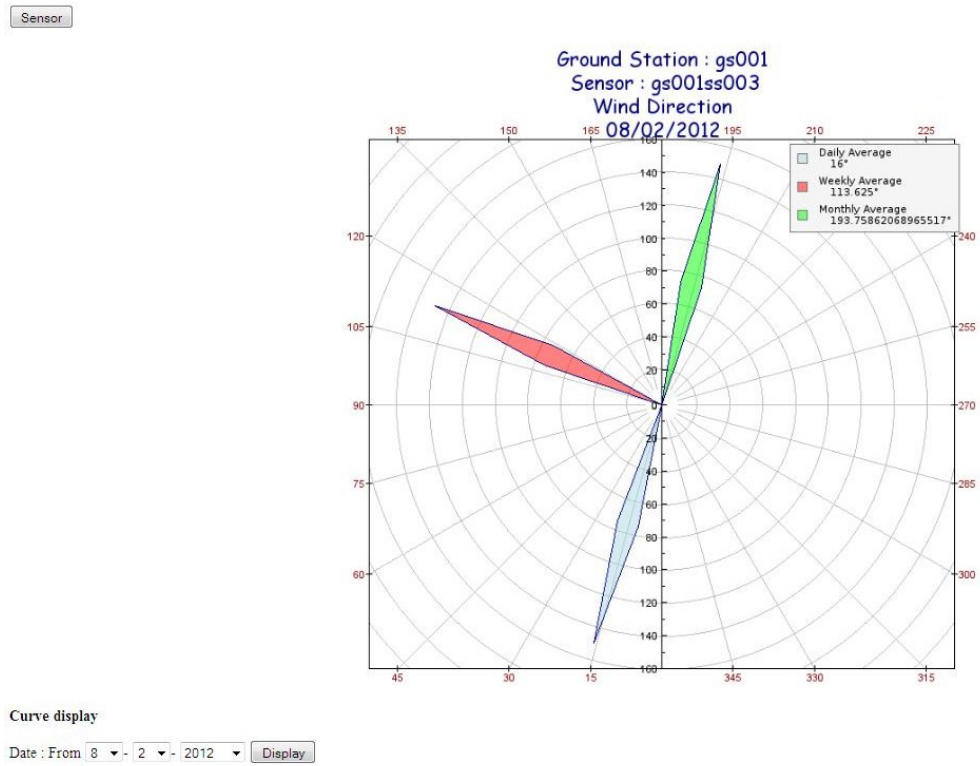


FIGURE 4.44: Polar clock display.

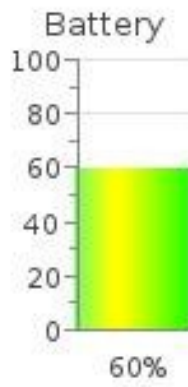


FIGURE 4.45: Battery reservoir graph.

Fig 4.45 shows a page which displays the battery level and the level can be represented as a percentage in the form of color changes.



The image shows a web application interface for station management. On the left, a sidebar menu contains three items: 'Stations Management' (highlighted), 'Setting', and 'User Setting'. The main content area is titled 'Add Ground Station'. Below the title, it says 'New ground station :'. There are three input fields: 'Station Name :', 'Longitude :', and 'Latitude :'. The 'Station Name' field contains the text 'gs'. Below the input fields is an 'Add' button.

FIGURE 4.46: Station management.

4.24 Administrator

4.24.1 Station management: Add a ground station

To add a ground station to the database, the above section has to be completed, fig 4.46. The name of this new ground station will be gs followed by the chosen name. If the station already exists, the user is warned. The name of the ground station has to be a number between 1 and 255.

To delete a ground station, the user has to click on the red cross then a confirmation message appears. If the suppression is confirmed, then the ground station and all its linked sensors and contents are erased.

4.24.2 Add a sensor

To add a sensor to a ground station, the above section has to be completed. First a station has to be chosen. Then the form in fig 4.47 has to be completed. The name of this new sensor will be gs“ground station name” ss followed by the chosen name. And a type has to be chosen. If the sensor already exists the user is warned. The name of sensor has to be a number between 1 and 255.

Add sensor

Selected Station :
gs001

Change Station :
gs001 ▾ Choose

Sensor Name :
gs001ss

Sensor Type Temperature ▾

Add

FIGURE 4.47: Add sensor.

Delete sensor

Selected Station :
gs001

Change Station :
gs001 ▾ Choose

Sensors :

name : gs001ss001
type : Temperature
X

name : gs001ss002
type : Light
X

FIGURE 4.48: Delete sensor.

4.24.3 Delete a sensor

To delete a sensor, the user has to click on the red cross then a confirmation message appears. If the suppression is confirmed, then the sensor and all its linked data are erased.

4.24.4 Add or Delete User

To add a user to the interface, the administrator user has to complete the above form. To delete a user, the administrator has to click on the red cross then a confirmation

Add User	Delete User
Login : <input type="text"/> Password : <input type="text"/> Confirm Password : <input type="text"/> Type : User <input type="text"/> <input type="button" value="Add"/>	Name : admin Password : admin X Name : user1 Password : user1 X Name : user2 Password : user2 X

FIGURE 4.49: Administrator's page to add or delete a user.

message appears. If the suppression is confirmed, then the sensor and all its linked data are erased.

4.24.5 Modify a sensor Type

Modify name and unit of a sensor type :

New name :

New unit :

Display :

FIGURE 4.50: Modify a sensor type.

Name, unit and type of display of sensor types can be modified in this section.

Add sensor type :

Name :

Unit :

Point

FIGURE 4.51: Add a sensor type.

Fig 4.51: Page to modify a sensor and to add a new sensor type. A new sensor type can be added in this section.

4.25 Import Data



Import Data
Groundstation: None
Sensor: Choose a sensor
File:

FIGURE 4.52: Import data feature.

To import data, the correct ground station has to be chosen. The administrator has to type the file path in the “Settings” window. The file has to be in same format as the export file. On each line, data has to be separated by tabs and each line by a line break. This kind of file can be edited with any spreadsheet software.

01/01/2011	10:00:00	20	100
02/01/2011	10:00:00	18	93
03/01/2011	10:00:00	19	88
04/01/2011	10:00:00	20	45
05/01/2011	10:00:00	19	72
06/01/2011	10:00:00	18	73
07/01/2011	10:00:00	17	63
08/01/2011	10:00:00	16	48
09/01/2011	10:00:00	14	-1
10/01/2011	10:00:00	15	57
11/01/2011	10:00:00	14	25
12/01/2011	10:00:00	16	94
13/01/2011	10:00:00	16	79
14/01/2011	10:00:00	16	47
15/01/2011	10:00:00	14	27
16/01/2011	10:00:00	13	73
17/01/2011	10:00:00	11	68
18/01/2011	10:00:00	11	70

FIGURE 4.53: Sample data file.

4.26 Export All



Export All Data

FIGURE 4.54: Export data feature.

An “*export all*” function is available. A folder is created, in the “*expotedfiles*” folder.

4.27 Summary

This chapter has explained the hardware used for the sensor nodes and the GS. The various configurations have been shown, and also included in this chapter are the methods of displaying and interpreting sensory information to give meaningful and useful environmental data. This chapter has gone further to explain the development and implementation of the MS.

Chapter 5

Evaluation and results

This chapter discusses results obtained from the implementation of all the components from chapter 4. In the first set of experiments conducted in the lab, power performance and battery life time estimation is conducted. The second set of experiments include determining RSSI and throughput values of the system. Lastly, results from field tests conducted at the site are given.

5.1 Power performance

5.1.1 Measurement setup

The following measurements were carried out in the laboratory; the first experiment investigates the load profile of the sensor nodes in different modes (idle, transmit and sleep modes). The second experiment analyses how the battery is discharged by the sensor nodes. The third experiment explores the effect of distance on RSSI values for LOS and NLOS conditions. In the third experiment, the distance between the transmitter and receiver was varied and then incremented by 5m at a time and noting down the resulting RSSI value.

TABLE 5.1: SLA battery specs.

Characteristics	Value
Nominal voltage	6V
Nominal capacity	12Ah
Standard load current	1A
Internal resistance	10 - 40 Ω

5.1.2 Lead acid battery considerations

5.1.3 Battery life time estimation

The battery lifetime is based on the average load current over time and is presented in the table below. This estimation method is used to highlight the difference between theoretical estimation and practical measurements. For this experiment, a sensor node is configured to send data to the ground station after every 5 seconds. In order to measure the current drawn, a shunt resistor of 10 Ω was connected between the positive supply and the sensor node, the voltage drop across the resistor was measured by a digital oscilloscope and the current drawn was determined.

The experimental set-up of this experiment is shown in fig 5.1 where the coordinator was placed at a distance of 1m from the sensor node while the voltage and timing diagram for the operation of transmitting the data to the GS is shown in fig 5.2.

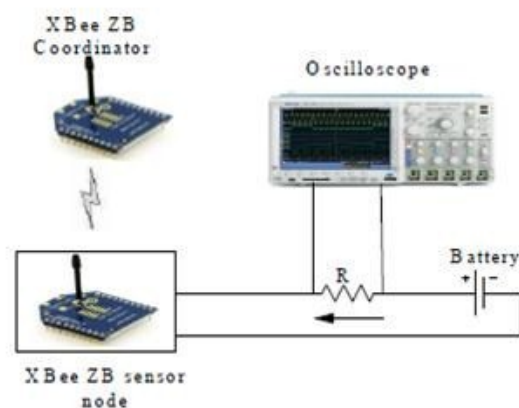


FIGURE 5.1: Experimental setup.

$$V_{in} = 6.2V$$

$$I = 0.03A \text{ and } 0.09A$$

$$R = 2.8\Omega$$

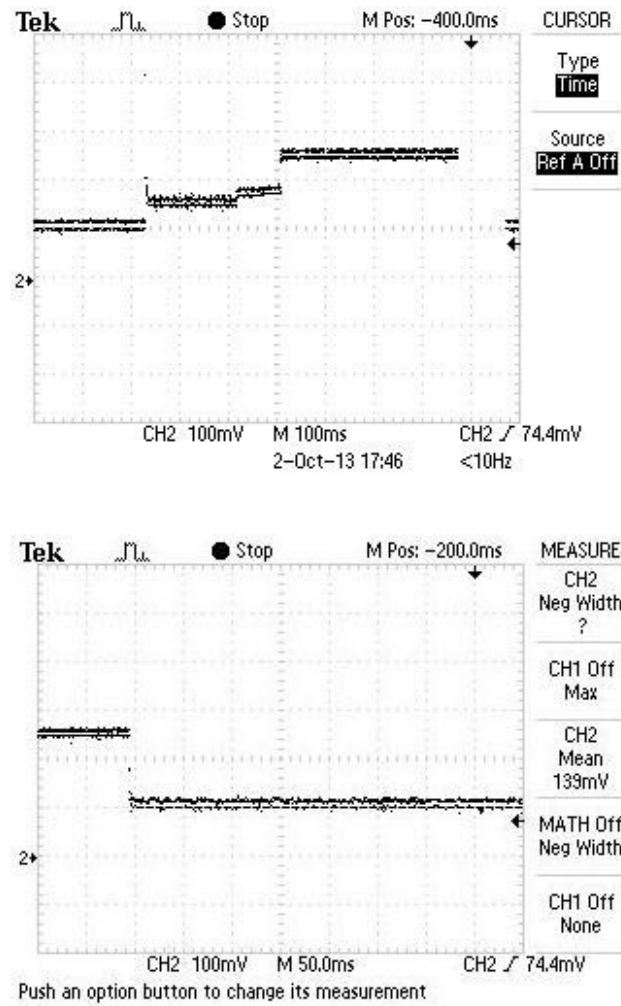


FIGURE 5.2: Snapshot of current consumption of sensor node during transmission of data.

Mean = 114mv (Arduino only)

Mean = 258mA i.e Arduino + XBee

TABLE 5.2: Node current profile.

Node	
Arduino + GPS	79mA
All ON	128mA
Tx\Rx	163mA

TABLE 5.3: GS current profile.

GS	
Rasp	300mA
Rasp + radio	420mA
Rasp + prog running	451mA
Rasp + prog run + XBee+ modem	554mA

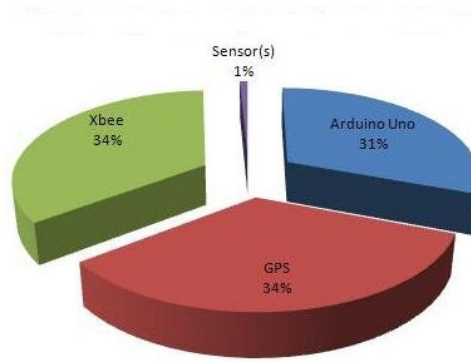


FIGURE 5.3: Sensor node power consumption breakdown.

Figure 5.3 above shows the breakdown of power consumption of a sensor node during peak transmission periods. The three main devices consume an almost equal amount of current and the sensors consume just a fraction of the total amount. This is mainly due to the fact that most sensors are resistive and contact type sensors and measurements are taken in a fraction of a second. The power consumptions of devices (Arduino, GPS, XBee) measured matches the values provided in their respective datasheets thereby validating the theoretical calculations.

The table below shows the measured current consumption and the time intervals

TABLE 5.4: current measurements.

Parameter	Stage	End device
Idle current (I_i)	1	0.04A
Activate time (t_{AX})	2	170ms
Activate current (I_{AX})	2	0.057A
Activate time (I_{AY})	3	90ms
Activate current (I_{AY})	3	90ms
Radio time (t_R)	4	15s
Radio current (I_R)	4	0.09A
Idle current (I_i)	5	0.04A
Battery capacity		12Ah
Battery voltage		6V

A sensor node should be tested for at least three days of autonomy so as to ensure reliability. To get an accurate power consumption model, it is important to actually measure the values and compare to the values given in the datasheets.

The sensor node cannot be configured to enter sleep mode and thus *Tactive* would be 24hrs.

With the average current i at which the battery is drained and considering T as the time between consecutive transmissions, we can compute the average current of each node as:

$$I_{avg} = \frac{I_{ax}T_{ax} + I_{ay}T_{ay} + I_rT_r + I_iT_i}{T} \quad (5.1)$$

where: I_{avg} is the average current

T is the active time of the sensor node

I_r is the active time of the radio

I_{ax} is the active current of period x

I_{ay} is the active current of period y and

T_{ax} , T_{ay} , T_r and T_i are the active times of the sensor node in its respective modes.

Given the above data, the node lifetime can then be estimated as:

$$nodelifetime = \frac{battcapacity}{avgcurrent \times 365 \times 24} [years] \quad (5.2)$$

Based on the analysis above, simulations were done to estimate the effect of transmission time on the overall lifetime of each sensor node.

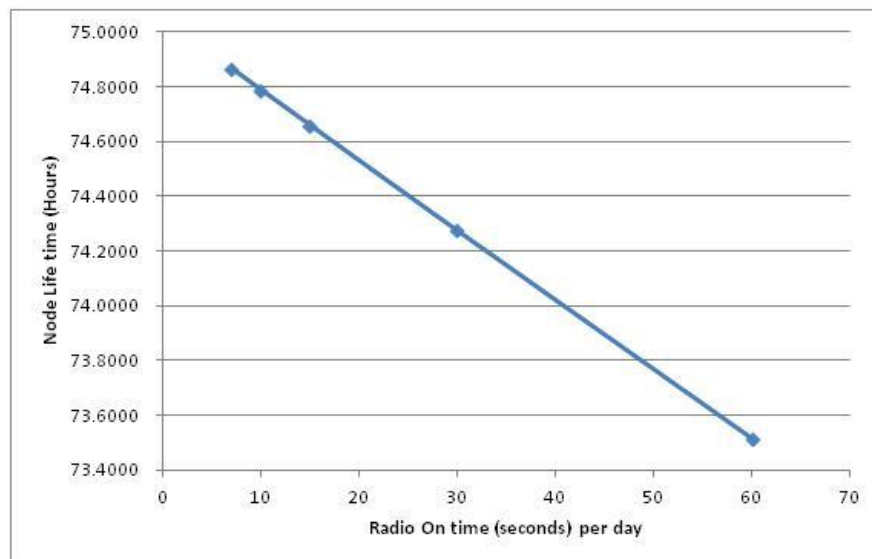


FIGURE 5.4: Node lifetime of a sensor node.

5.1.4 Interpretation

The results in Figure 5.4 above show that the lifetime of a sensor node is affected by the amount of time spent when the XBee radio is operational, which is to be expected. Having a minimum Radio ON time of 7s give the sensor node around 74.85 hours to live whereas when the radio is configured in an always ON state, gives a sensor node roughly 63.65 hours to live.

The configuration of the sensor node demands that the Arduino and GPS have to operate in an always ON state and hence no attempt was made to switch OFF either of the two so as to observe what happens when the duty cycle is altered.

The same setup can be applied to the ground station where equation 5.2 can be expanded to yield:

$$nodelifetime = \frac{battcapacity}{loadcurrent \times 365 \times 24} [years] \quad (5.3)$$

The GS is configured to send data to Iridium only once a day and taking average current data from table 3.7, we get;

$$nodelifetime = \frac{24}{0.334 \times 365 \times 24} [years] \quad (5.4)$$

≈ 71.73 hours of operation, which is close to the result from the actual experimental measurements.

5.1.5 Experimental set up

The maximum current that the board will draw will be when the radio is in transmitting mode. The maximum power dissipated would be:

$$\begin{aligned} P_{max} &= P_{ard} + P_{gps} + P_{radio} + P_{sensor} \\ &= (40mA \times 5V) + (40mA \times 5V) + (45mA \times 5V) + (5mA \times 5V) = 0.65W \end{aligned}$$

For autonomy test, a node is made to transmit a data packet once per hour for 3 consecutive days without an external power source. The voltage across the battery was

sampled and transmitted along with sensor data and a time stamp. This was used to check the amount of minimum voltage the battery would reach before the sensor would stop working.

5.1.6 Results:

The measured current was 400mA with an input voltage of 5V. Thus, the total power the node can draw is 2.0W. The node with its 30Ah battery as sized in chapter 3 lasted for almost three days without completely draining the battery.

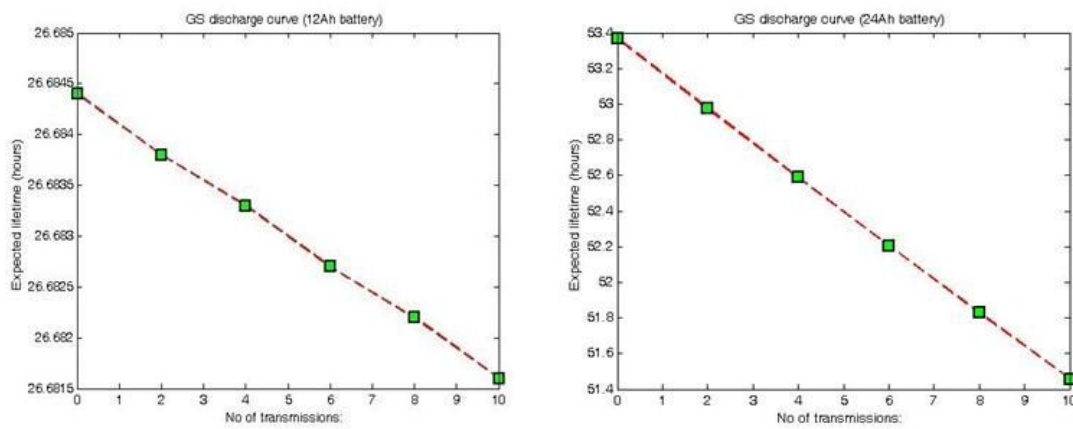


FIGURE 5.5: Matlab representation of GS discharge curves.

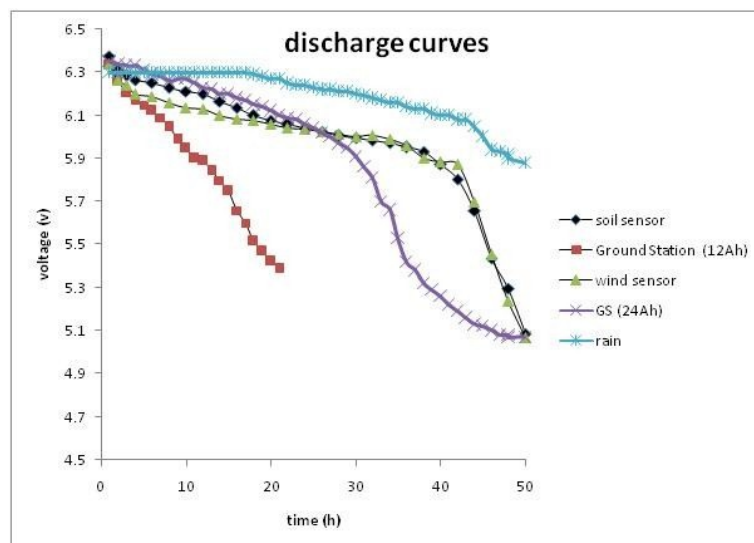


FIGURE 5.6: Discharge curves of sensor nodes and ground station.

5.1.7 Interpretation

The ground station connected to a 12Ah and 24Ah source have a lower lifetime compared to the sensor nodes. This is because of a large current drawn by the Raspberry Pi and the Iridium modem during peak transmission periods. The sensor nodes have a discharge curve which is almost similar to one another since they have the same controllers, radio but different sensors.

As a note, the batteries should be fully charged prior to starting the experiment.

5.2 RSSI

RSSI is the indication of the signal strength level of the last received packet. The detected signal strength decays with distance. The relationship between the received power and the transmitted power can be determined using the Friis transmission equation 5.6

$$Pr = Pt \times Gt \times Gr \left(\frac{\lambda}{4\pi d} \right)^2 \quad (5.5)$$

Where Pt is the transmission power, Gt and Gr are gain of transmitter and receiver respectively. λ is the wavelength and d is the distance between sending and receiving radios. From equation above, it is quite clear that the power at the receiver is mainly determined by the distance traveled and its power decays in an inverse proportional relationship with distance.

RSSI can be defined in terms of the ratio of the received power Pr to the reference power Pr_{ref}

$$RSSI = 10 \log \frac{Pr}{Pr_{ref}} \quad (5.6)$$

The experimental setup will be connected in a simple point to point link with the coordinator and router/end device. The router will be sending data to the coordinator and the RSSI value would be taken from XCTU software. The distance between the coordinator and router was varied so as to determine the relationship between RSSI and distance traveled. Fig 5.7 shows the measured RSSI values as a function of distance between the two nodes.

5.2.1 Experimental setting.

To ensure accurate results, a large open area that provided LOS conditions was required. The roof on the fifth floor of the Engineering building was chosen. This was perfect in that the area is isolated and it has few obstructions which would cause interference

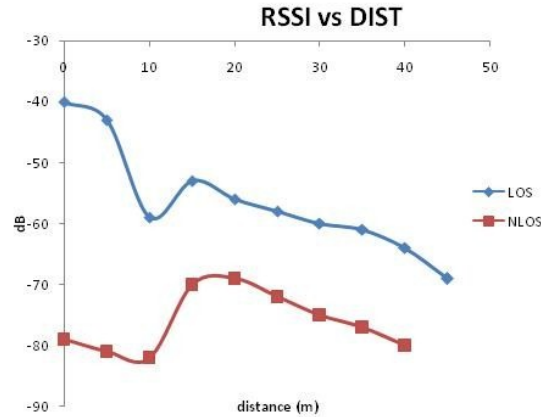


FIGURE 5.7: Representation of RSSI against distance for LOS and NLOS conditions.

5.3 Throughput analysis

Throughput refers to the amount of data that can be transferred from the sender to the receiver in a given amount of time. Throughput of the network is affected by factors such as network protocols used but in this case, we consider the influence of speed of transmission and the packet size on throughput. This parameter is measured in kbps, Mbps or Gbps. In this network, all the sensor nodes will be transmitting packets to the ground station hence it is important to know the expected performance of this system. We are mainly interested in the maximum sustained throughput, which is the value of the throughput averaged over a long period of time.

$$throughput = \frac{8 \times \text{numberofbytes}}{\text{time}} \quad (5.7)$$

5.3.1 Experimental setup

The aim of this experiment is to determine the throughput of the radios as a function of speed and the packet size. For this experiment, the star topology of fig 2.18 is chosen. The network will be configured with one coordinator and the rest of the radios are configured as routers.

To measure the transmission time, we program an Arduino Uno to send packets varying from 10 bytes to 80 bytes maximum. The packet is sent at a 1 second interval to a remote XBee configured as a router with hardware loopback enabled. In this scenario, the XBee's TX is connected to its RX pin to echo back every packet it receives. This configuration is advantageous to the software loopback that might include some processing time and hence influence the total latency. Transmission time is then noted from the time just before the packet is sent and the time when the entire packet has been received, the difference of the times gives us the total transmission time.

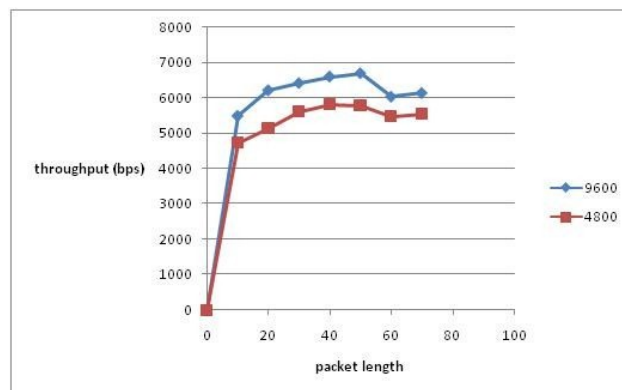


FIGURE 5.8: Throughput measurements results for 9600bps for different packet lengths.

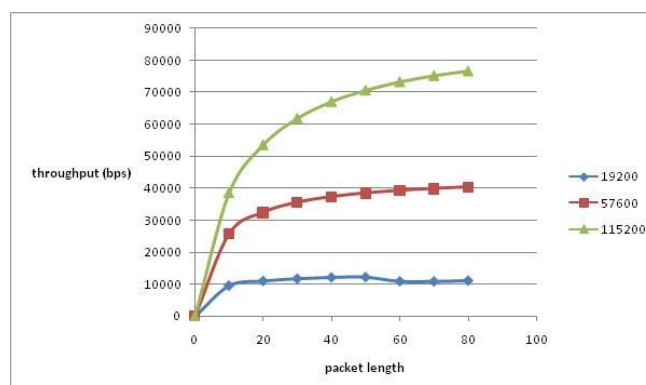


FIGURE 5.9: Throughput measurements results for 19200bps for different packet lengths.

Fig 5.8 and fig 5.9 above show the throughput as a function of baud rates and packet size as derived from equation 5.7. It is seen that the throughput increases as the baud rates increases. In fig 6.0, 4800bps and 9600bps speeds yield a lower maximum throughput of 6kbps and 7kbps respectively whereas 19200bps, 57600bps and 115200bps yield a much higher throughput. Maximum throughput of 76kbps can be achieved at 115200 baud rate. According to the XBee datasheet, XBee series 2 modules can achieve a throughput of 85kbps, but the results obtained from this experiment show lower performance characteristics for lower baud rates. This result can be attributed to a number of variables pertaining to the radios used. These include; security enabled or disabled and route discovery attempts among others. For our experiment, a one hop configuration, router to router and security disabled configuration was chosen.

Distance was varied from 0 – 50m for 9600bps baud rate and a 30 bit packet and the results show that distance has an insignificant effect on throughput. The throughput when the transmitting node is 50m from the receiving node is almost the same as the throughput when the sender is very close to the receiver. As equation 5.7 illustrates, throughput is greatly determined by the size of the packets being transmitted and the rate at which the packets are being transmitted.

5.4 Summary

Chapter 5 has given the results of the experiments conducted in this work. Power tests were conducted in the laboratory and the nodes achieved more than three days of autonomy as set out in the project requirements. Range tests were conducted at the Coetzenburg sports field and the nodes attained more than 200m range. The project's deployment field was tested for RF activity and the channel was found to be clean of RF interference. All tests verified that the project is suitable for deployment in terms of the intended application.

The next chapter concludes the project and gives an overview on the work that might be done in the future to improve its performance.

Chapter 6

Conclusion

6.1 Introduction

This chapter provides a summary of the work done, the methodology, project implementation, results obtained and the main outcomes of the work. The future work section looks at how this work may be extended or improved.

This thesis has described the design and development of a weather environmental monitoring system designed to operate in any remote area without the need for any telecommunication infrastructure. The main objective is to develop an automatic system which could be deployed in remote areas to provide weather data in near real time. Other objectives include designing a system which is supposed to withstand three days of autonomy in harsh winter periods and the master station is supposed to display the information in an informative and user friendly way. Other important parameters such as cost, size and available RF devices were also considered, but they were less critical. The work covered in this project spans a variety of fields of study. A detailed literature study was conducted on hardware/software development, RF devices and the satellite constellation used.

For the methodology followed, we started the development of the project by identifying a suitable satellite network. The hardware platform was subsequently designed. On this part, we integrated off the shelf Arduino commercial microcontroller boards with custom

PC boards to enable RF radio communication, GPS and sensor hardware integration. After hardware implementation, the developed nodes were programmed using Arduino software and finally, the complete system was tested and verified.

The sensor node hardware platform comprises of an Arduino UNO board, GPS module, Digi XBee radio, Davis weather sensor and a 9-pin connector. The GS consists of a Raspberry Pi Linux SBC, XBee radio, Iridium modem and a custom PCB to control power to the radio and sensors. The software for the sensor nodes was written using the Arduino programming language which incorporates a GCC compiler, which is an extension to the common C++.

Several tests were conducted at various levels to verify the functionality of the system.

1. In the laboratory, power tests were conducted so as to determine and characterize the battery lifetime.
2. RSSI tests were conducted on the roof of the Engineering building. This test examined the behaviour of the power of the received signal as a ratio of the distance between the transmitter and receiver.
3. Outdoor range tests were conducted at the Coetzenburg sports field. This test was carried out to observe the maximum line of sight transmission range of the radio modem. The achieved range was 280m, which is perfectly acceptable for the intended purpose.
4. A spectrum analyzer was used to examine the 2.4GHz spectrum at the test site, as a check for possible interference.

In summary, the developed water resource monitoring system serves as a reliable and feasible choice. The system also provides as an alternative to the currently used system where data is downloaded from the data logger. Power factors and coverage area limitations have been adequately addressed and successfully tested.

Overall, the completed system performed satisfactorily as intended and proved its suitability for the particular application.

6.2 Future work

This section briefly reviews options that could possibly improve the performance and functionality of the developed system.

- **Low power consumption and energy harvesting:** The current system is powered by sealed lead acid batteries and solar panels. However, this concept is limited in case of dense vegetation and tall trees which reduce solar energy. The system can be improved by employing protocols which reduce power consumption or by scavenging energy such as kinetic or thermal energy from the environment. The latter solution also gives another advantage of reducing the overall size of the nodes. These energy sources are typically constrained and their use will depend on the application.
- **Long range operation:** There might be need to increase the sense area of the wireless network, in this regard, the range of the WSN can be increased by replacing the current XBee series 2 radios by more powerful and longer range Digi 9-Xtend radios.
- **Data frequency:** The current system records environmental data at an hourly period. The system can be adapted to monitor data minute by minute, at the cost of higher energy usage, of course.
- **User interface:** The user receives email attachment of the data, manually sorts the data and uploads on the MS. A software script can be written so as to achieve a fully automatic environmental monitoring system.

Appendix A

Project images



FIGURE A.1: Sensor node layout.



FIGURE A.2: Ground station layout.

Figure [A.1](#) shows the final design of the sensor node. In total, four sensor nodes were built. Fig [A.2](#) shows the final design of the GS, only one GS was built and they were deployed in an actual testing environment.

Appendix B

XBee RF modules

B.1 XBee ZB module configuration

XBee radios are used to transmit data wirelessly by setting up a Zigbee network which is based on the IEEE 802.15.4 standard. This protocol enables the radios to be configured to operate in 3 different configurations which are: coordinator, router or end device (as discussed in chapter 2).

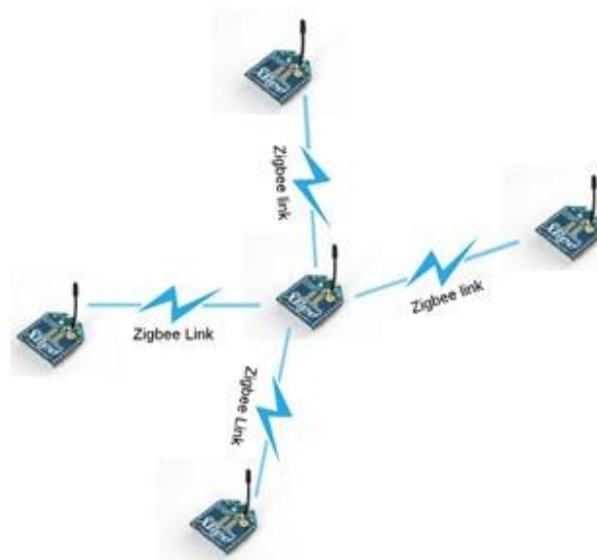


FIGURE B.1: XBee star configuration.

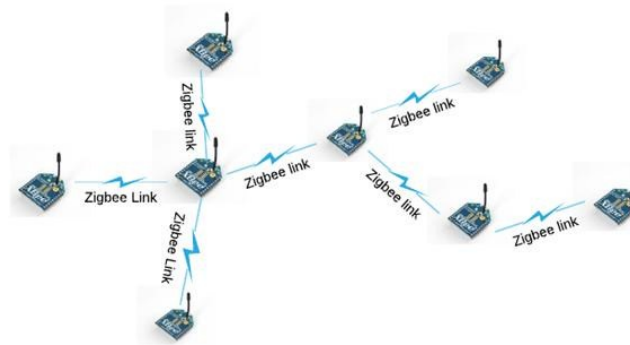


FIGURE B.2: XBee mesh configuration.

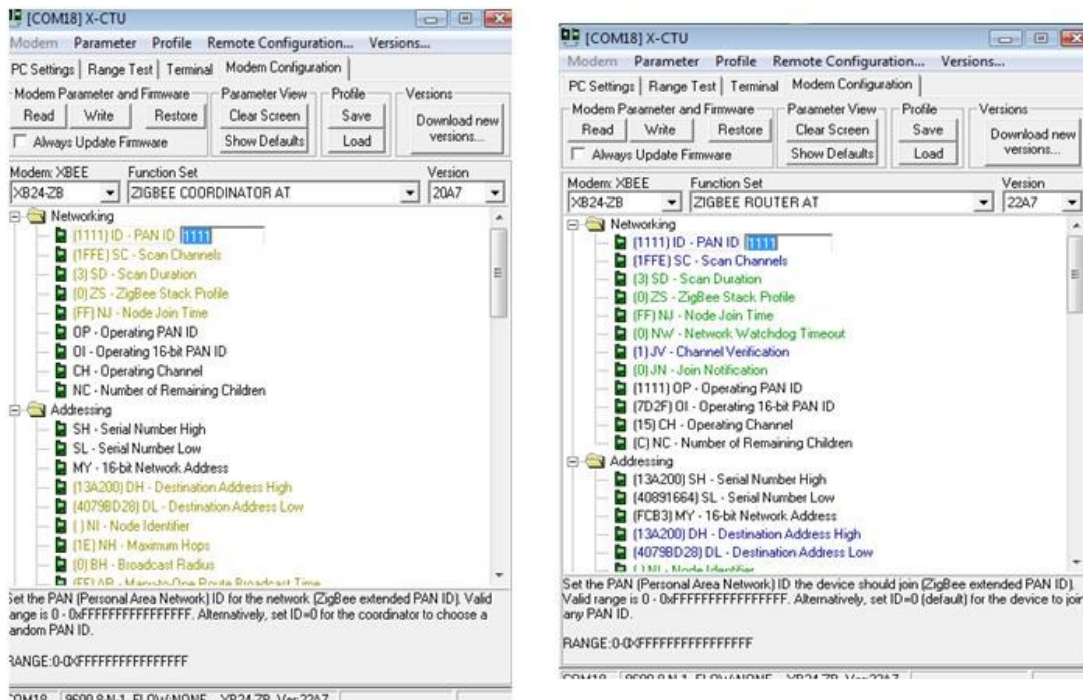


FIGURE B.3: XCTU screenshots.

B.1.1 Setting up the XBee coordinator

1. Install the XBee module to the interface board and open the XCTU software
2. Under the PC settings tab, choose the COM Port associated with the module on the interface board.
3. Verify that the serial port settings match the internal settings of the radio. Default settings are:
 - Baud rate: 9600
 - Flow control: None

- Data Bits : 8
 - Parity : None
 - Stop bits : 1
4. To verify communications with the module, click the TEST/QUERRY button. A pop up message that shows the status will be displayed.
 5. In the XCTU Modem configuration tab, click Read. A snapshot as shown in fig [B.3](#) will be displayed. For the coordinator, parameters in orange may be changed but parameters in black are read only.
 6. Select the ZB24-ZB modem and in the Function set option, select ZIGBEE COORDINATOR AT and choose the most recent version.
 7. Enter the desired PAN ID: this should be the same in all modules of one network. Other parameters are optional but the most important are the Destination Address High and Destination Address Low parameters, (DH and DL).
 8. For the coordinator, set DH to 0000 and DL to FFFF (FFFF is the broadcast address). In order to address a specific router, DL of the coordinator is matched to SL of the router and DH of the coordinator is matched to SH of the router.
 9. You can set multiple parameters at once and then click write button to change the settings.
 10. Configure the router modules using the same procedure above but select ZIGBEE ROUTER ATi n the function set option (the RF module will be configured to function as a router in AT mode).
 11. Enter the same PAN ID of the coordinator.
 12. DH and DL can both be set to 0000 (destination addresses of the Coordinator) , click write to save the changes to memory.
 13. In the terminal tab of the router, text entered will show up in the terminal window of the coordinator. Text sent by the router is shown in blue while text received by the coordinator is shown in red.

Appendix C

Project cost breakdown

TABLE C.1: Project cost breakdown

Component	Description	Price (R)
Sensor Nodes	Arduino Uno	Arduino MCU Board
GPS module	LEA-4P-T-SMRT	195.00
Arduino Uno	Arduino UNO ATmega328 MCU Board Rev R3	269.78
XBee shield	Digi International XBee Zigbee Shield	250.75
XBee radio	XBee RF Module Whip Antennae	350.00
Enclosure	ENCL PLA 318 × 238×101	169.99
6V lead acid battery	Bat SLA 6V 12AH 151×50×101 6120	147.34
Miscellaneous	Resistors, PCB headers, PCB spacers, cable connectors, transistors	200.00
Cup anemometer and wind vane	Davis Anemometer: Wind direction/speed	1949.00
Raingauge	Davis Rain gauge 0.2mm, tipping bucket, magnetic reed switch	1094.00
Soil moisture sensor	Temp/ air sensor multi purpose	570.67
Soil temperature sensor	Soil sensor	729.45
Humidity station	Humidity sensor with radiation shield	3277.34
Solar panels	20W 16V	289.32×5
2m galvanised steel poles miscellaneous	D shucle 8mm, Turn buckle 6mm, Crosby clamp 5mm, strain cable 3mm, tent peg 300×8mm	579.00
Sensor node PCBs	Arduino Uno and XBee sensor boards	418.24×8
Battery charger PCBs	Battery charger pcbs	423.49×5
Ground Station		
GS PCB	Switching control PCB	456.24
Raspberry Pi	Raspberry Pi type B 512Mb	399.56
Iridium modem	Iridium 9602 SBD modem	4231.58
Total Price		R22230

Appendix D

Iridium SBD modem

D.1 9602 structure

There are several Iridium transceivers that implement the short burst data (SBD) communication system, namely:

- Iridium 9522B modem
- Iridium 9601 modem
- Iridium 9602 modem

The transceiver chosen for this project is the Iridium 9602 modem. The 9602 is compact, low cost, small size and its availability makes it preferable over other modems. The 9602 is capable of transmitting messages of 340 bytes and can only receive messages of 270 bytes.

Access to the 9602 is made possible through the user connector which provides the following connections:

- DC power supply input
- Power on/off control
- Serial data interface



FIGURE D.1: 9602 Iridium modem.

- Network available output
- Supply power indicator output

The 2 row 20 way header user connector provides power and communication to and from the modem.

TABLE D.1: 9602 power profile [7]

Parameter	Value
Supply voltage	5.0V DC +/-0.5V
Idle current	45mA
Transmit current (peak)	1.5A
Transmit current (average)	190mA
Receive current (peak)	195mA
Receive current (average)	45mA
SBD message transfer average current	190mA
SBD message transfer average power	<=1.0W

RF and antenna characteristics parameters

- Frequency range 1616MHz to 1626.5MHz
- Duplexing method TDD (Time Domain Duplex)
- Input/Output impedance - 50Ω

- Multiplexing method TDMA/FDMA
- Polarisation RHCP
- Connector SMA (M) plug
- Bandwidth 15MHz
- VSWR 1.5 Max (at centre frequency)

Communication to and from the modem is done via the serial port and is controlled using specific AT commands. A 9 wire serial port connection is implemented to control communication between the modem and the field application (GS) where the digital signal levels are at 3.3V. A 3 wire serial port connection can also be selected where the GND, TX and RX pins are used. The 9 wire connection, however, offers a better control of the communication. The baud rate of the modem can be set through AT commands by the GS.

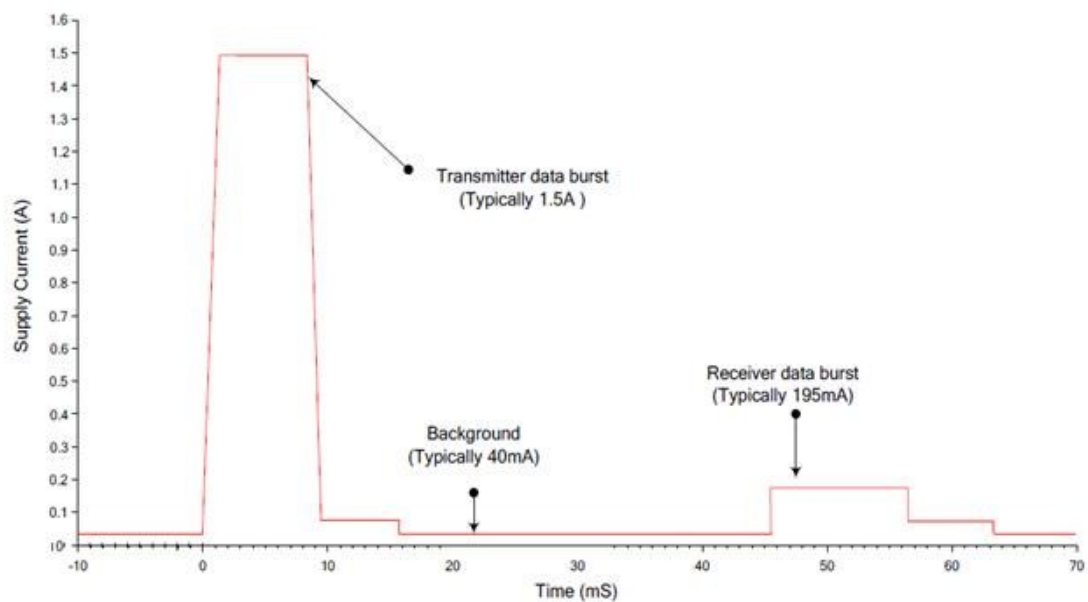


FIGURE D.2: 9602 typical power profile [7].

Figure D.2 shows a graphical representation of a typical supply current profile of a 9602 SBD transceiver sending and receiving SBD messages. The actual usage profile can vary as a result of:

1. View of the sky – if the modem is in poor visibility of the sky where a clear line of sight is not available between the transceiver and the satellite.
2. How often the 9602 module is activated or deactivated by the remote unit.
3. Manufacturing variation from transceiver to transceiver.

D.2 RF interface

The 9602 uses an external antenna for communication with the Iridium GS. Section A.1.1 above shows the specifications of the RF interface, the selected Iridium-band antenna should be able to meet the characters above.

D.3 9602 AT commands

The modem is configured and operated through the use of AT commands. AT (Attention Code) is the prefix for all commands.

There are two options to transfer data from the DTE to the single mobile originated buffer in the 9602 which are:

- SBDWB (write binary data to the ISU), 340bytes maximum
- SBDWT(write text message to the ISU), 340bytes maximum

D.3.1 AT+SBDWB

This command is used to transfer binary data from the DTE to the mobile originated (MO) buffer in the 9602, this command should also be accompanied by the message length. The operation of this command is shown in the table below.

The DTE or GS initiates binary data transfer to the modem by issuing the AT+SBDWB=*n*;*message length*; command, upon receiving this command, the modem responds by transmitting a READY message or ERROR. If the modem sends a READY message, the FA should begin sending the binary data followed by a carriage return within 60seconds, or a time-out error message will be issued. For error checking and control, the FA should also

TABLE D.2: Binary data commands

To modem (From RU)	To GS (from modem)	Description
AT + SBDWB = 340		Transfer binary data to modem
	READY	
<binary transfer>		
	0 , OK	
AT + SBDI		Perform SBD session
	+SBDI: 0,23,0,-1,0,0	SBD status result

truncate a 2-byte checksum to the data.

{*binary message*}+{*2-byte checksum*} , this checksum is the least significant 2bytes of the summation of the binary message. This format is verified by the modem and it issues a response. If everything is correct, the FA initiates the SBD session by issuing the AT+SBDI command, this command instructs the modem to transfer any data stored in the MO buffer to the Iridium GSS.

D.3.2 AT+SBDWT

Alternatively, the FA can also transfer data as text, as shown in table...

TABLE D.3: Text message commands

To modem (From RU)	To GS (from modem)	Description
AT + SBDWT		Transfer text data to modem
	READY	
<message transfer>		
	0 , OK	
AT + SBDI		Perform SBD session
	+SBDI: 0,23,0,-1,0,0	SBD status result

The FA initiates text message transfer by issuing the AT+SBDWT command, the modem responds by sending a READY message once it is prepared to receive. Once the FA receives the READY message, data is sent from the FA to the MO buffer in the 9602 and the modem acknowledges that data has been successfully stored in the MO buffer

by sending 0. The FA will issue the AT+SBDI command to perform an SBD session between the modem and the Iridium GSS.

Since either mode of data transfer accepts 340bytes at a time, it is less costly for the FA to transfer the message as text.

Bibliography

- [1] Alkhatib, Ahmad Abed Alhameed, Baicher and Gurvinder Singh, Wireless Sensor Network Architecture, *International Conference of Computer Networks and Communication Systems*, 35(3):11–15, April 2012. URL <http://connection.ebscohost.com/c/articles/83362722/wireless-sensor-network-architecture>.
- [2] Ian F. Akyildiz and Mehmet Can Vuran, Wireless Sensor Networks, *Wiley*, 1236–1239, June 2010. URL <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-047003601X.html>.
- [3] Walteneagus Dargie and Christian Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice, *Wiley*, 72ISBN: 978-0-470-99765-9:1–20, July 2010. URL <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470997656.html>.
- [4] Alessio Vecchio, Adaptability in wireless sensor networks, *IEEE Electronics, Circuits and Systems*, ISBN: 978-1-4244-2181-7:1261–1264, August 2008. URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4675089>.
- [5] Millennial Net, Maximising Data Reliability in WSN, *A Millennial Net White Paper*, January 2009. URL <http://archives.sensorsmag.com/articles/0705/23/main.shtml>.
- [6] Adrian Perrig, John Stankovic and David Wagner, Security in wireless sensor networks, *Communications of the ACM Wireless Sensor Networks*, 47(12):53–57, June 2004. URL <http://dl.acm.org/citation.cfm?id=990707>.
- [7] Iridium, Iridium 9602 SBD transceiver developer’s guide, *iridium Developer’s Guide*, January 2002. URL <http://iridium.com/products/Iridium9602.aspx>.

-
- [8] F. Dietrich, P. Metzen and P. Monte, The Globalstar cellular satellite system, *IEEE Antennas and Propagation*, 46(12):935–942, June 1998. URL <http://link.aip.org/link/?RSI/72/4477/1>.
- [9] John Tandler, Automating the Operations of the ORBCOMM Constellation, *Small Satellite Conference*, 1–9, January 1996. URL <http://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=2449&context=smallsat>.
- [10] G. Berzins, R. Phillips, J. Singh and P. Wood, Inmarsat Worldwide satellite services on seas, in air and on land, *IAF International Astronautical Congress*, 40(12):1–11, October 1989. URL <http://adsabs.harvard.edu/abs/1989mala.iafcQV...B>.
- [11] Digi International Inc, XBee ZNet2.5/XBee-PRO ZNet2.5 OEM RF Modules, Product Manual v1.x.4x, *Digi International Inc*, 1January 2005. URL <http://www.digi.com>
- [12] Laura Vanzago, Overview on 802.15.4/Zigbee, *Advanced System Technology R&D, DICO*, 621–20, May 2006.
- [13] Patrick Kinney, Kinney Consulting LLC, Wireless Control that Simply Works, *Communication Design Conference*, 621–20, October 2003. URL <http://link.aip.org/link/?RSI/62/1/1>.
- [14] Anshul Thakur, Zigbee Technology, *Review of Scientific Instruments*, October 2008. URL <http://www.engineersgarage.com/articles/what-is-zigbee-technology?page=3>.
- [15] Sikora A and V. F Groza, Coexistence of IEEE802.15.4 with other Systems in the 2.4 GHz-ISM-Band, *Instrumentation and Measurement Technology Conference*, 61786–1791, May 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1604479.
- [16] Drew Gislason, Zigbee Wireless Networking, *Elsevier Inc*, 1, ISBN: 978-0-7506-8597-9, August 2004. URL <http://www.amazon.com/Zigbee-Wireless-Networking-Drew-Gislason/dp/0750685972>.
- [17] Manuel Pietschmann, Routing and Network Topologies, *802.15.4 Standard for Low Power Devices*, 72(12):, 1–20, July 2012. URL <http://>

- [//www.dresden-elektronik.de/fileadmin/Downloads/Dokumente/white_paper/routing_and_network_topologies-WP-en.pdf](http://www.dresden-elektronik.de/fileadmin/Downloads/Dokumente/white_paper/routing_and_network_topologies-WP-en.pdf).
- [18] 485-2010, IEEE Recommended Practice for Sizing Lead-Acid Batteries, IEEE Recommended Practice for Sizing Lead-Acid Batteries. *IEEE*, 2ISBN: 978-0-7381-6507-3, 1–90, April 2011. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5751584&sortType%3Dasc_p_Sequence%26filter%3DAND%28p_IS_Number%3A5751583%29.
- [19] R.A. Mammano and Unitrode IC Corp, Charging the new batteries-IC controllers track new technologies. *Annual Battery Conference on Applications and Advances*, 10ISBN: 0-7803-2459-5, 171–176, January 1995. URL <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=3177>.
- [20] S. Barret, Arduino Microcontroller Processing. *Morgan & Claypool*, 3 ISBN: 1627052542, January 2013. URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6812716&queryText%3Darduino>.
- [21] John Mcneill, C .W. Michael, W. Coln and J. Brian, Split ADC Architecture for deterministic Digital Background Calibration of 16-Bit 1-MS/s ADC, *IEEE Solid-State Circuits*, 40ISBN: 0018-9200, 2437–2445, December 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1546220.
- [22] Atmel, Littrow configuration tunable external cavity diode laser with fixed direction output beam, *Atmel Atmega*, January 2000, URL www.atmel.com.
- [23] Digi International Inc, XBee/Xbee-PRO RF Modules, Product Manual, *Digi International Inc*, URL www.digi.com.
- [24] Kevin Fall and Kannan Varadhan, Littrow configuration tunable external cavity diode laser with fixed direction output beam, *NS Manual*, 62, November 2011. URL <http://www.isi.edu/nsnam/ns/doc/>.
- [25] Neil Madden, Arjen, David .N and Welton, Tcl Tutorial, *TCL*, URL www.tcl.tk/man/tcl8.5/tutorial/Tcl0a.html.
- [26] D. Estrin, M. Handley, J. Heidemann and S. McCanne, Littrow configuration tunable external cavity diode laser with fixed direction output beam, *IEEE Computer*,

- 33ISBN: 0018-9162, 63–68, November 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=881696.
- [27] Digi Inc, Digi Inc, *Digi Inc*, 1, 1–20, January 1991, URL <http://digi.com>.
- [28] P. Levi, S.Madden, J. Polastre, R.Szewczyk and K. Whitehouse, Tiny-OS: An Operating System for Sensor Networks. *Ambient Intelligence*, 72(12):, 115–148, January 2005. URL <http://www.cs.berkeley.edu/~culler/papers/ai-tinyos.pdf>.
- [29] A. Dunkels, B. Gronvall and T. Voigt, Contiki a lightweight and flexible operating system for tiny networked sensors, *IEEE Local Computer Networks*, ISBN: 0742-1303, 455–462, November 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1367266.
- [30] A. Eswaran, A. Rowe and R. Rajkumar, Nano-RK: an energy aware resource centric RTOS for sensor networks, *IEEE Real-Time Systems Symposium*, 26ISBN: 1052-8725, 10–265, December 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1563113.
- [31] I. F. Akyildiz, Y. Sankarasubramaniam and E. Cayirci, State of the art in sensor networks research, *Elsevier Journal of computer networks*, March 2002. URL <http://www.ece.gatech.edu/research/labs/bwn/>.
- [32] Raspberry Pi, Raspberry Pi, *Raspberry Pi*, 1–20, January 2012. URL raspberrypi.org.
- [33] Austin Whiller, The determination of hourly values of total solar radiation from daily summations, *Archiv fr Meteorologie, Geophysik und Bioklimatologie, Serie B*, 7(2), ISBN: 0066-6424 197–204, March 1996. URL <http://link.springer.com/article/10.1007/BF02243322>.
- [34] Meldrum, DEVELOPMENTS IN SATELLITE COMMUNICATION SYSTEMS USEFUL SATELLITE SYSTEMS FOR DATA BUOY OPERATORS, *Data Buoy Cooperation Panel*, March 2008. URL <http://www.jcommops.org/doc/satcom/satcom.pdf>.
- [35] Michel Mouly, Marie-Bernadette Pautet, The GSM System for Mobile Communications, *Telecom Publishing*, March 1992. URL <http://dl.acm.org/citation.cfm?id=573838>.

-
- [36] K. Maine, C. Devieux, P Swan, Overview of Iridium Satellite Network, *Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies*, ISBN: 0-7803-2636-9, 7 – 9 November 1995. URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=485428>.