

SHARC Buoy

Robust firmware design for a novel, low-cost autonomous platform for the Antarctic Marginal Ice Zone in the Southern Ocean



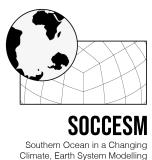
Jamie Nicholas Jacobson

Department of Electrical Engineering
University of Cape Town
Rondebosch, Cape Town
South Africa

March 2021

MSc.(Eng.) thesis submitted in partial fulfilment of the requirements for the degree of
MSc. in Electrical Engineering in the Department of Electrical Engineering at the
University of Cape Town

Keywords: IoT; Firmware; Southern Ocean; Sea ice; Marginal Ice Zone; Autonomous platform, Firmware design



Declaration

I, Jamie Nicholas Jacobson, hereby:

1. grant the University of Cape Town free licence to reproduce the above thesis in whole or in part, for the purpose of research;
2. declare that:
 - (a) this thesis is my own unaided work, both in concept and execution, and apart from the normal guidance from my supervisors, I have received no assistance except as stated in my bibliography and acknowledgements.
 - (b) neither the substance nor any part of the above thesis has been submitted in the past, or is being, or is to be submitted for a degree at this University or at any other university.

Jamie Nicholas Jacobson
Department of Electrical Engineering
University of Cape Town
Friday 12th March, 2021

Abstract

SHARC Buoy

Jamie Nicholas Jacobson

Friday 12th March, 2021

Sea ice in the Antarctic Marginal Ice Zone (MIZ) plays a pivotal role in regulating heat and energy exchange between oceanic and atmospheric systems, which drive global climate. Current understanding of Southern Ocean sea ice dynamics is poor with temporal and spatial gaps in critical seasonal data-sets. The lack of in situ environmental and wave data from the MIZ in the Antarctic region drove the development of UCT’s first generation of in situ ice-tethered measurement platform as part of a larger UCT and NRF SANAP project on realistic modelling of the Marginal Ice Zone in the changing Southern Ocean (MISO). This thesis focuses on the firmware development for the device and the design process taken to obtain key measurements for understanding sea ice dynamics and increasing sensing capabilities in the Southern Ocean.

The buoy was required to survive the Antarctic climate and contained a global positioning system, temperature sensor, digital barometer and inertial measurement unit to measure waves-in-ice. Power was supplied to the device by a power supply unit consisting of commercial-grade batteries in series with a temperature-resistant low dropout regulator, and a power sensor to monitor the module. A satellite modem transmitted data through the Iridium satellite network. Finally, Flash chips provided permanent data storage. Firmware and peripheral driver files were written in C for an STMicroelectronics STM32L4 Arm-based microcontroller. To optimise the firmware for low power consumption, inactive sensors were placed in power-saving mode and the processor was put to sleep during periods of no sampling activity.

The first device deployment took place during the SCALE winter expedition in July 2019. Two devices were deployed on ice floes to test their performance in remote conditions. However, due to mechanical and power errors, the devices failed shortly after deployment. A third device was placed on the deck of SA Aghulas II during the expedition and successfully survived for one week while continuously transmitting GPS coordinates and ambient temperature. The second generation featured subsequent improvements to the mechanical robustness and sensing capabilities of the device. However, due to the 2020 COVID-19 pandemic, subsequent Antarctic expeditions were cancelled resulting in the final platform evaluation taking place on land. The device demonstrates a proof of concept for a low-cost, ice-tethered autonomous sensing device. However, additional improvements are required to overcome severe bandwidth and power constraints.

Acknowledgements

The creation of this work would not be possible without the contributions of the following people:

Keith Machuchon for designing the buoy stand and for offering insight into past deployments with the Trident buoy.

Do Yeou Ku for developing the circular buffer algorithm and for all her advice which helped guide the development of the firmware into its current form. Additionally, thank you to Tilen Majarle of <https://stm32f4-discovery.net/> whose tutorials and guides helped with the initial stages of firmware development. I would also like to acknowledge the driver code for the MPU6050 IMU written by Adafruit in C++ for the Arduino which provided the basis for the STM32L4 driver code in C.

Yusuf Heylen for contributing towards development and integration of the IMU.

Nicholas Bowden for creating the C library for the Flash chips.

Jason Cloete for developing the circuit board for the first prototype of the device.

Ben Adey for his work on developing a power budget for the system.

Anran Chen for his contribution towards developing a compression algorithm for raw inertial time-series data.

James Irvine and Hannah Harrison-Saayman for their contribution towards testing the current monitor and Flash chips.

Michael Noyce for his work on developing an integrated PCB for future versions of the project.

Robyn Verrinder and Justin Pead for designing the prototype circuit boards for the first version of the device.

Thank you to my supervisors Robyn Verrinder, Amit Mishra and Marcello Vichi whose insight and wisdom help guide this project to fruition.

To Jarryd Son and Nadir Vorajee who initiated the project and helped guide me through the initial phases.

Thank you to the UCT-UDE sea ice team for teaching me about the importance of sea ice in an ever changing climate. Also special mention to Ehlke Hepworth and Ashley

Womack for their role in deploying the buoys. Also to Sebastian Skatulla for leading the sea ice team and helping coordinate the SHARC Buoy activities. I'd also like to thank Marc de Vos from the SAWS for his insight and input that guided the project.

Thank you to the UCT mechnatronics lab for giving me the support, assistance and friendships that got me through the long development process.

To Sasha Gold for always supporting me and helping me when I need it. To Joseph Miller; your friendship helped lift me up when I was down.

To my family for supporting me and putting up with rooms and tables filled with cluttered hardware. You all gave me the strength to undertake this project.

Finally, a project would not be complete without coffee and people to drink it with. For that I'd like to thank Charlie Lin for sharing in this journey from start to finish.

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Dedicated to the captain and crew of the SA Aghulas II whose bravery and wit helped bring science to the edge of the earth and back.

Contents

Declaration	i
Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 Background to investigation	1
1.1.1 Antarctic sea ice and the Marginal Ice Zone	1
1.1.2 Observations of the Marginal Ice Zone	4
1.1.3 Overview of the existing technology and devices	6
1.2 Problem statement	9
1.2.1 Project initiation	10
1.2.2 Scope	11
1.2.3 Limitations and constraints	12
1.2.4 Assumptions	13
1.3 Plan of development	13
1.4 Report structure	13
2 Literature review	16
2.1 In situ climate sensing technologies	17
2.2 System level overview	20
2.2.1 Remote communication	20
2.2.2 Power supply	22

2.2.3	Polar performance	25
2.2.4	Overall cost	30
2.3	Subsystem overview	31
2.3.1	Processing capabilities	32
2.3.2	Measurement of wave data using inertial measurement systems . .	34
2.3.3	Measurement of ice drift using GPS	38
2.3.4	Temperature sensing and measurement	44
2.3.5	Atmospheric pressure sensing and sensors	46
2.4	Conclusion	47
3	Design methodology	50
3.1	Design overview	50
3.2	User requirements	50
3.2.1	Analysis of UR001	51
3.2.2	Analysis of UR002	52
3.2.3	Analysis of UR003	52
3.2.4	Analysis of UR004	53
3.2.5	Analysis of UR005	55
3.2.6	Analysis of UR006	55
3.2.7	Analysis of UR007	56
3.3	Functional requirements	57
3.3.1	Operational requirements	57
3.3.2	Electronic requirements	58
3.3.3	Software requirements	58
3.3.4	Other requirements	58
3.4	System overview	59
3.4.1	Technical specifications	61
3.4.2	Acceptance test protocols	62
3.5	Conclusion	63

4 Platform design	64
4.1 Mechanical features	64
4.1.1 Buoy stand	64
4.1.2 Enclosure	65
4.2 Electronics	67
4.2.1 GPS	67
4.2.2 Iridium	69
4.2.3 Environmental sensors	71
4.2.4 Power monitoring sensors	73
4.2.5 Inertial measurement unit	74
4.2.6 Memory	76
4.2.7 Processor	77
4.2.8 Power electronics	80
4.3 Final assembly	82
4.4 Conclusion	86
5 Software Design	87
5.1 Software architecture	88
5.1.1 Hardware abstraction layer	89
5.1.2 Subsystem application peripheral interface (API) layer	90
5.1.3 Main program layer	90
5.2 Project structure	90
5.2.1 Project tools and files	90
5.2.2 Program startup and entry	91
5.2.3 Input clock selection	93
5.2.4 Power mode selection	94
5.3 Main program design	95
5.3.1 Execution	96
5.3.2 Asynchronous behaviour	99

5.3.3	Subsystem execution	105
5.4	Data management	106
5.4.1	Debugging	106
5.4.2	Drift data acquisition	108
5.4.3	Environmental data acquisition	111
5.4.4	Power data acquisition	112
5.4.5	Total ice drift data	113
5.4.6	Wave measurement data	113
5.4.7	Timing requirements	115
5.5	Data flow	116
5.6	Conclusion	121
6	Testing	122
6.1	Subsystem tests	122
6.1.1	Unit tests	123
6.1.2	GPS	123
6.1.3	Iridium modem	125
6.1.4	Environmental sensor	127
6.1.5	Power monitor	129
6.1.6	Flash chips	130
6.1.7	IMU	132
6.2	System tests	134
6.2.1	SYS001 accelerated system test	134
6.2.2	SYS002 power test	137
6.2.3	SYS003 low temperature test	140
6.2.4	Results	141
6.2.5	SYS004 field test	141
6.3	Field testing	142
6.3.1	Configuration	145

6.4	Final evaluation	148
6.4.1	Functional requirement validation	149
6.5	Discussion	151
6.5.1	Power requirements	151
6.5.2	System performance	152
6.5.3	Mechanical features	152
6.5.4	Power system	153
6.5.5	Future work on wave measurements	153
6.5.6	Short-burst data modems VS. Telephone modems	154
6.5.7	Evaluation against the State of the Art	154
6.6	Conclusion	154
7	Conclusion	155
7.1	Key contributions to firmware and validation	155
7.2	User requirement verification	156
7.3	Full system testing	156
7.4	Verification's against the state of the art	156
7.5	Final Remarks	157
8	Recommendations	158
8.1	Improvements to the power system	158
8.2	Improvements to hardware	158
8.3	Improvements to the communication modules	158
8.4	Firmware improvements	159
8.5	Expansion of nodes into a network	160
8.6	Future deployments	160
A	Numerical modelling	161
A.1	Modelling of polar stochastic processes	161
A.1.1	Numerical modelling of sea ice	161

A.2 Modeling of GPS dilution of precision	164
A.3 Numerical techniques for modelling ocean waves	165
A.3.1 Kuik method	165
A.3.2 Welch-Earle method	166
A.4 Temperature sensing measurement techniques	168
A.5 Pressure sensing techniques	169
B Project design	171
B.1 Stakeholder analysis	171
B.2 Acceptance test protocols	172
B.3 Schematics and renders	180
C Software design	183
C.1 Events and interrupt handling protocols	183
C.2 Initialization Routines	185
C.3 Code	189
C.3.1 BMP280 temperature compensation formula	189
C.3.2 INA219 calibration algorithm	190
C.3.3 Data structs	192
D Supplementary tables	194
E Test protocols	196
E.1 Unit tests	196
E.2 System test results	199

List of Figures

1.1	Diagram showing the lifecycle of Antarctic sea ice as observed using remote sensing based on barber2005microwave . The cycle begins with the congealing of the ocean surface known as Freeze Up with peak ice formation during Winter, the second half of the cycle is characterised by various stages of melt with complete sea ice retreat occurring in the Summer.	2
1.2	(A) Diagram showing different types of ice present in the Marginal Ice Zone including (a) pancake ice floes (b) frazil ice and (c) brash ice. Some ice floes exhibit snow growth along the ridges while other floes exhibit flooding. (B) Sea ice in the Southern Ocean MIZ during July 2019, where pancake ice is the predominant concentration while brash ice is the smallest. Swell waves can also be observed propagating through the region. Photos taken during the early sea ice formation period during the SCALE winter cruise July 2019 by the author.	3
1.3	Diagram showing the expanse of sea ice in the Southern Ocean. Packed ice forms closer to the continent in calmer conditions while strong oceanic currents, winds, wave action and extreme temperatures result in the formation of semi-consolidated ice in the Marginal Ice Zone (1 to 5). Here, ice formation is highly seasonal expanding to a maximum in winter and retreating to a minimum in summer. Sea ice acts as a boundary layer influencing heat and gaseous exchange between the atmosphere and ocean. Figure taken from Antseaice .	4
1.4	Photo taken in the Marginal Ice Zone from onboard the SA Aghulus II during the SCALE winter expedition in 2019 by the author. The vessel is anchored in consolidated ice with the UCT ¹ -UDE ² sea ice team performing ice coring activities on the surface of the ice.	5
1.5	Diagram showing the current state of Antarctic sea ice measurement technologies for each level of observation as well as the estimated deployment location. Diagram taken when sea ice extent is at a maximum. This diagram is derived from the technology implementation strategy identified from the 2016 SCAR roadmap (kennicutt2019sustained) ³ and has been adapted to show sea ice observational techniques.	6

1.6	Practical examples of instruments used to collect in situ measurements in the sea ice region. These comprise: (A) the Surface Wave Instrument Float Tracking (SWIFT) buoy developed by the University of Washington (thomson2012wave); (B) the Ice Mass Balance (IMB) buoy developed by Dartmouth College [(PLANCK2019102792) image source: (simbfact)]; and (C) the Southern Hemisphere Antarctic Research Collaborative (SHARC) buoy developed by the University of Cape Town (photo courtesy of R. Verrinder).	7
1.7	Novel ice drift, environmental monitoring and wave measurement autonomous platform: the Southern Hemisphere Antarctic Research Collaboration (SHARC) Buoy. Developed by the University of Cape Town. Photo by R. Verrinder.	9
2.1	Map of the Southern Ocean surrounding the Antarctic continent. Image created by Hogweed2015Ocean and licensed by CC BY-SA 3.0.	16
2.2	A block diagram of a typical device for in situ polar sea ice measurements. Each device contains modules for environmental sensing and drift sensing connected to a processor. This can be a microcontroller or microprocessor. Additionally, a data storage module (such as an SD card) is included to store data during the operation. Some devices include modules for local communication or wave sensing units. Finally, a remote communication module is included to transfer the data to a research centre or a user. The electronics are placed inside a physical enclosure for protection against the polar climate while a portable power module supplies the device during its operation.	18
2.3	Devices used for the comparison of autonomous instruments deployed in the sea ice region. Each device has been selected for its notability in published work as well as prevalence in sea ice and wave interactions in the Marginal Ice Zones. These devices are: (A) Wave in Ice Buoy developed by rabault2017measurements , (B) Wave in Ice Observational System by kohout2015device , (C) Novel Wave Directional buoys by doble2017robust , (D) Surface Kinematic buoy by guimaraes2018surface , (E) Surface Wave Instrument Float Tracking buoy by thomson2012wave , (F) Seasonal Ice Mass Balance buoy by polashenski2011seasonal ; meng2014optimal , (G) Polar ISVP by MetOcean (uptempo), (H) Trident buoy by Trident (trident).	19
2.4	Examples of popular Iridium modems selected for remote communications. (A) The 9522B modem (image source: (9522B))), (B) the 9602 modem (image source: (9602))) and (C) the 9603 modem (image source: (9603)))	21

2.5 A block diagram of a typical power module for a remote sensing device based off the information from rabault2019open ; doble2017robust ; vidal2019xev . Batteries are used as a primary source of energy which is connected to a battery management system to control and regulate the power supply to the device. Optionally, solar panels are implemented with a rechargeable battery as a secondary power supply. Shown in the figure: flow of power (red), control lines (blue) and ground (black).	23
2.6 Examples of different deployment protocols for ice tethered devices. (A) In regions of consolidated ice in favourable conditions, manned crews will step foot on the ice to deploy the device (image source: (kohout2020observation)). (B) In unfavourable conditions, devices may be deployed from a personnel basket attached to a crane or a manned crew will lower the buoy onto a suitable ice floe from the safety of the personnel basket (image source: N. Taylor).	27
2.7 During the 2019 SCALE winter expedition, a SWIFT device was retrieved early due to lost communication with the host. This failure was attributed to a build up of ice along the rim from ocean spray. Photo taken by Author.	29
2.8 Diagram of a typical microprocessor (yellow) on a development board. This device is an Intel Edison which was used as the main processor by kohout2015device for the WIIOS. The development board allows for fast prototyping and integration into projects. The processor interfaces with external peripherals through physical input/output pins (red) and contains standard serial communication ports such as USB (orange), micro USB (teal) and a connector for external voltage (purple). Image source: (edison).	32
2.9 Diagram showing the characterization of sea state through statistical methods described by kuik1988method and earle1996nondirectional ; welch1967use . A wave consists of a maximum vertical point (crest) and a minimal vertical point (trough) with the distance between the two points referred to as the wave height (H_0). The period between consecutive crests of a similar frequency is referred to as the wave length and the point where a wave crosses the mean sea state level is the zero crossing. From a statistical analysis, the significant wave height is the average of the top third of observed wave heights in a given time period. (kuik1988method). This diagram is based off the figure by (seastatediag).	34
2.10 Diagram showing examples of the different types of inertial measurement systems (IMUs) available for commercial use such as (A) an Attitude Heading Reference System (AHRS) VN100 (vn-100) used by (rabault2019open), (B) an analog capacitive accelerometer 8330B3 Servo-KBeam accelerometer used by (kohout2015device) (kistlerIMU) and (C) a 9 degree of freedom MPU9520 (mpu9520) used by (kohout2015device).	35

2.11 Diagram showing examples of the different types of global positioning devices available for commercial use such as (A) MTK3339 (image source: (mtk3339)) , (B) SkyTraq S1351R (image source: (s1315F)) (C) Qstarz BT-Q1000eX (image source: (BTQ1000EX))	39
2.12 Diagram showing how satellite distribution affects the positional estimation. Global navigation satellite system (GNSS) satellites have an associated inaccuracy which is represented by the circles. Locking on to more than one satellite reduces these inaccuracies by interpolating the position from the phase differences between satellites. This value is greatly affected by the spread of satellites. (A) a larger spread results in a more accurate positional estimate (B) while a shorter satellite spread results in a less accurate positional estimate. A measure of this inaccuracy is called the dilution of procession (DOP). Figure was adapted from images by (GISGeo2020DOP) for A and B.	41
2.13 Examples of temperature sensors employed by remote sensing devices. These types of devices include (A) thermistors such as the PR303J2 (image source: (pr303j2)), (B) digital temperature sensors such as the DS18B20 (image source: (ds18b20)) and (C) digital temperature chains (DTC) such as the Bruncin DTC (image source: (bruncindtc)).	44
2.14 Examples of pressure sensing technologies using a diaphragm based MEMS with different methods of measuring the strain. These are piezoresitive sensors: (A) the BME280 (image source: (BME280)), (B) the MPX-5100-AP (image source: (MPX5100AP)) or capacitive sensors: (C) the PTB100 (image source: (PTB110))	47
3.1 Block diagram of the proposed autonomous system showing subsystem arrangement, data flow and interfaces with the environment.	60
3.2 Breakdown of subsystems from Table 3.7 into usable components.	61
4.1 Diagram of the buoy stand for the SHARC buoy. The stand consists of stainless steel and painted mild steel to withstand the climate of the Southern Ocean and stop the stand from rusting. A cradle at the top of the stand houses the buoy and secures it to the stand via a screw. The buoy stand elevates the electronics 1 m above the ice to overcome the effects of snow growth and flooding. Finally, spikes at the bottom of the base will secure the stand to the ice floe. Drawn by R. Verrinder.	65
4.2 2-D exploded view of the buoy enclosure showing (A) the bottom enclosure for the power module, (B) the connector block which also acts as a base for the electronics and (C) the top enclosure which covers the electronics. Drawn by R. Verrinder.	66
4.3 The u-blox GPS Modules selected for this project which are: (A) the u-blox NEO-7M on a Waveshare breakout board (waveshare) and (B) a u-blox NEO-M9N on a Sparkfun breakout board (UBLOX_M9N_DATA).	68

4.4	A Rock7 RockBLOCK 9603 satellite module containing an Iridium 9603 modem (9603) and a coaxial Sub-miniature version A (SMA) connector for an external antenna. Image by rockblock2019image	69
4.5	Environmental sensing devices selected for the buoy. (A) The Bosch Sensortech BMP280 digital pressure and temperature sensor on an Adafruit sensor breakout board (BMP280breakout) and (B) a Maxim Integrated DS18B20 in a through hole TO-226-3 package (DS18B20manual).	72
4.6	3-D render ⁴ of the Texas Instruments INA219 current/power monitor I ² C chip in an 8-pin small outline integrated circuit (SOIC) package (INA219).	73
4.7	A TDK Invensense MPU6050 6 degrees of freedom inertial measurement unit on a Sparkfun development board. The board allows for fast prototyping with communication performed via I ² C. Image source: (mpu6050breakout).	75
4.8	The Adesto Technologies SPI serial flash chips in a surface mount package. Image source: (flashchipimage).	77
4.9	The 32-bit microcontrollers used for the different versions of SHARC buoy. (A) is the STMicroelectronics STM32F407VTG6 (stmdisc) on a 100-pin Discovery development board which was used for version one. (B) is the 64-pin STMicroelectronics Nucleo development board which initially contained an STMicroelectronics STM32F446RE (stm32nucleo) for version 2. However, it was replaced with an STMicroelectronics STM32L476RG (stm32l4) for the final version.	78
4.10	3-D render ⁵ of the Texas Instruments LP3876 linear low drop out regulator in a 5-Pin DDPAK/TO-263 surface mount package (LP3876).	81
4.11	Simplified schematic of the final version of SHARC buoy showing (dash) power supply, (solid) communication and digital (arrows) connections. Also shown in the figure are lines indicating: (black) ground, (red) 5 V power, (pink) 3.3 V power, (Purple) analog inputs, (orange) I ² C, (yellow) UART, (green) SPI and (blue) digital input/output.	83
4.12	Power module for the SHARC BUOY. A custom PCB with a low dropout regulator and current sensor connected to a battery pack.	84
4.13	Electronic PCB stack for the top module consisting of connector board, microcontroller board and sensor board attached to the connector block.	85
5.1	Block diagram of the Advanced RISC Machines (Arm) Cortex-M4 microprocessor. This device features a Harvard bus matrix and Armv7E-M central processing unit. The processor also contains a hardware floating point unit (FPU), nested vector interrupt controller(NVIC) for controlling interrupts and a wakeup interrupt controller (WIC) for low power operational control. This processor can be programmed through serial wire via STLink or via JTAG. Image source: (ARMprocessor).	87

5.2	Diagram showing the decomposition of the overall firmware into distinct layers and the relationship between each part.	89
5.3	Diagram showing the firmware execution from (green) start. On start up, the <i>startup_stm32l476xx.s</i> file executes and loads the main program. This program runs in a continuous loop until the power supply to the device is turned off. This diagram shows the interaction of the firmware layers (see Figure 5.2): (blue) subsystem API, (red) HAL and (grey) the main program. Also shown are arrows indicating (black) the main flow of the program, (green) references to functions, (red) calls to function prototypes and (orange) interrupts.	92
5.4	Diagram showing a general low power operation profile for the microcontroller with two distinct phases: process and Inactive occurring over a period T.	94
5.5	Diagram showing a typical run of the main program. The program is designed to run sequentially based on the previous completed state. The program begins from power on/wake up and reads the last known state from the RTC back up registers. Based on this state, the program will execute the corresponding routine. Finally, the program will determine the next state and write this to the back up register. If no activity is required, the device will enter low power mode.	97
5.6	ASM chart for the proposed state machine to run on the processor showing entry/exit conditions and functions to be run during states.	98
5.7	Application diagram with event and interrupt sequencing.	100
5.8	Flow chart for the state-check algorithm	101
5.9	Diagram showing the algorithm for brown out event recovery and handling.	103
5.10	Diagram showing the algorithm for handling an external interrupt from a Wake Up pin connected to one of the modules	104
5.11	Example of an enumeration of possible error sources for a peripheral library function. Each error code is a numeric value which is returned when a specific error condition is encountered thereby providing feedback to higher levels.	107
5.12	Timing diagram for the SHARC buoy showing the sequence and states as well as the expected current draw. The cycle begins with (purple) an initialisation phase where the sensors are verified and configured into the correct modes. Then (blue) the sample periods where the sensors are sampled followed by (orange) a period of inactivity. After four sample sessions, (green) the transmission state is entered where data are condensed into packets and transmitted. Diagram is a representation of the estimated current VS timing and is not to scale.	116

5.13 Flow chart showing the ice drift sampling routine called during the sample state. This diagram contains arrows showing: (blue) the main flow and sequence of steps in the routine, (black) the flow of steps for sampling data from sensors. (white) the flow of data from temporary storage (as an ice drift measurement) to permanent storage. Also shown are (red) GPS subsystem, (blue) environmental sensor and (purple) power sensor.	118
5.14 Data struct for storing drift data collected from the sensors during a sample period where Coord_t and Diagnostic_t are shown in Appendix C.10 - C.11	119
5.15 Diagram showing the structure of a drift data packet including byte position, size and data being collected	119
5.16 Diagram showing the flow of data during a cycle of the buoy. The data are sampled by the sensors and converted into packet form where it is stored until it is ready to be transmitted. The transmitted data arrives at a server and is sent via email to the user.	120
6.1 Diagram showing the configuration for the power test. All subsystems were connected and activated with version 2 sensors and firmware. Power was supplied by an Agilent xxxx bench top power supply set to a constant 7.2 V output. A data logger was connected to the INA219 sensor to sample the power information at a rate of 1 Hz	138
6.2 Graph showing a typical current cycle of the buoy during the various phases. (Cyan) initialisation phase, (Black) sample phase, (Red) sleep phase and (Green) transmit phase. Data were sampled at 1 Hz with all modules connected, sample intervals set to 30 minutes the INA219 sensor connected to an external data logger and the device placed in a partially obstructed environment.	139
6.3 Average current consumption at each phase in the operational cycle of the buoy. Ordered chronologically.	140
6.4 Diagram showing the trajectory of the SA Aghulas II for the 2019 SCALE winter cruise from Cape Town to the Southern Ocean Marginal Ice Zone along the Good Hope line. Diagram created by SCALE planning team. .	143
6.5 SHARC buoy version 1 used for deployment testing during the 2019 SCALE winter cruise in July. The device contains an Iridium modem, GPS and temperature sensor for sea ice drift and ambient temperature measurements. Photo by R. Verrinder.	144
6.6 SB01 and SB02 buoy deployment locations. Ice edge on 26 to 28 July 2019. Ice edge data provided by E. Hepworth from https://www.natice.noaa.gov/products/kml_daily.html	145

6.7 Photos taken during field testing of the SHARC buoy system showing (A) the device fully assembled and secured to the stand (photo: R. Verrinder), (B) the device deployed onto an ice floe via a basket and crane deployment procedure, (C) SHARC buoy on an ice floe following a successful deployment.	146
6.8 The GPS trajectory of the SA Aghulas II ship from the Marginal Ice Zone to East London. The plot shows the estimated position (magenta) taken from the buoy samples (red) compared to the actual trajectory (cyan). The positional error (PDOP) of each measurement is shown as an exaggerated area around the measured position.	147
6.9 Air temperature recorded by the buoy (yellow) over 11 days compared to the air temperature recorded by the ship (blue).	148
B.1 Schematic of top enclosure which protects the electronics.	180
B.2 Schematic of bottom enclosure for the batteries and power system.	181
B.3 Schematic of the connector block which provides a base for the electronics to be mounted on.	181
B.4 Full enclosure schematic.	182
C.1 u-blox NEO-7M and u-blox NEO-M9N initialisation routine.	186
C.2 Rock Seven RockBLOCK 9603 initialisation routine.	186
C.3 Adesto Technologies AT45DB641E initialisation routine.	187
C.4 Bosch Sensortech BMP280 initialisation routine.	187
C.5 Texas Instruments INA219 initialisation routine.	188
C.6 TDK Invensense MPU6050 initialization routine.	188
C.7 Function written to compensate a 32 bit Temperature reading for sensor irregularities using the 32 bit version of the recommended compensation formula from the datasheet (BMP280_Datasheet). The formula uses the compensation parameters stored on the sensor	189
C.8 Function written to compensate a 32-bit pressure reading for sensor irregularities using the 32-bit version of the recommended compensation formula from (BMP280_Datasheet). The formula uses the factory trim compensation parameters stored on the sensor.	190
C.9 Calibration routine for INA219 current sensor for a maximum current of 1.2 A, maximum bus voltage of 16 V and maximum shunt voltage of 160 mV.	192
C.10 Coord_t data structure to store incoming GPS coordinates as IEEE754 32-bit floats.	192

C.11 Data structure for storing GPS signal diagnostic information.	193
--	-----

List of Tables

1.1	Legend showing the key stakeholders in the initiation of the project as discussed in the phases below. Legend includes name, reference number and department/institution.	10
1.2	Description of report structure including key phases of the project and significance	13
2.1	Devices used for the comparison including the device name, lead developer and the institution. These consist of both commercial and institutional devices for in situ sea ice and wave measurements.	20
2.2	The following Iridium modems are compared in their key specifications. Devices in the table were suitable for IoT applications based on prevalence in literature and recommendations from the manufacturer. Key parameters include weight, power consumption and transmission latency. Information is taken from (iridium_product) with prices as of February 2021.	21
2.3	A comparison of power supply strategies of the different devices showing the the power source, topology of the power supply module as well as the voltage supplied at the output of the module. Information that was unavailable at the time of research has been labeled as "Not Reported".	24
2.4	Comparison between the functionality and purpose of the buoy showing the critical measurements as well as the significant deployment locations either in the polar ice zones or in a location critical to the validation of the device.	26
2.5	Comparison of price and weight of each device according to the published literature or commercial listing. Weight provides an indicator of the ease of handling whereas price provides an indicator of affordability. Prices have been converted to South African Rand (R) online (usdcoversation)where applicable while weight has been converted to kg. "Not Reported" or "N/R" is given where a value could not be obtained.	30
2.6	Comparison of the processing strategy implemented by each device. Multiple processors have been used in a selection of devices, hence included in the comparison is the number of processors used, the type of processor and the function of each processor.	33

2.7 Comparison of the inertial measurement systems selected for each device showing the sensors included as well as the degrees of freedom.	36
2.8 Comparison of sampling strategies implemented in each device. This includes the desired measurands, sample rate and sample period of each IMU session.	37
2.9 Table showing the ratings for a dilution of precision (DOP) measurement with a numerical range corresponding to a rating as shown by tahsin2015analysis . These ranges provide context for positional accuracies associated with GPS measurements with a DOP of 1 being an ideal reliable measurement with an associated small margin of uncertainty whereas a DOP of 20 is completely unreliable with a large uncertainty.	42
2.10 Comparison between different GPS devices, sampled data and periods between samples implemented by each device.	43
2.11 Comparison of the different components used by each device for temperature measurement as well as the sensor type and the measurement objectives of the device. Each device measures temperature of a different process. For example: the WIIB measure air temperature whereas a device like the SIMB measures temperature profiles of ice floes in addition to air temperature.	45
2.12 Comparison of the different pressure sensing components used by each device as well as the measurement objective guiding the selection of the component. "Not Reported" is given where a device includes a pressure sensor but provides no information to its use. "-" is given where a column does not apply to the device. "None" is given where a device does not contain a pressure sensor.	46
3.1 User requirements obtained by meeting with the principal stakeholders. These will be used to determine the desired functionality of the buoy.	51
3.2 Comparison of standard measurements for meteorological data including temporal resolution, measurement unit and accuracy from: (worldmeteorologicalorganization)	
3.3 Requirements addressing the mechanical needs for the system during operation.	57
3.4 Requirements addressing the electronic needs for the system including the modules, components and sensors that satisfy the user requirements.	58
3.5 Software functional requirements for the system addressing the system function, performance, operation and control during the lifetime of the device.	58
3.6 Other system requirements being addressed.	58
3.7 Table showing the subsystems that are critical to the functionality of the buoy and the level of importance indicated by rank	59

3.8	Technical specifications for the overall system	62
3.9	A summary of acceptance test protocols from Appendix B.2 showing the target and purpose of the test.	63
4.1	Primary measurements of the buoy enclosure taken from the schematic in Appendix B.3.	66
4.2	Comparison of key parameters between the initial u-blox NEO-7M GNSS module and the updated u-blox NEO-N9M module.	68
4.3	Pinout for the Rock7 RockBLOCK 9603 Iridium modem.	70
4.4	Table showing the mechanical feature parameters of the Rock7 RockBLOCK 9603 module (9603).	70
4.5	Table showing the power consumption parameters of the Rock7 RockBLOCK 9603 module (9603).	71
4.6	Table showing the communication parameters of the Rock7 RockBLOCK 9603 module (9603).	71
4.7	Comparison of performance between the Bosch Sensortech BMP280 and Maxim Integrated DS18B20 environmental sensors. The term "Not applicable" or N/A is given where a parameter does not apply to a device. . .	72
4.8	Comparison between supply voltage and current draw of the BMP280 and DS18B20.	73
4.9	Performance specifications for the INA219 current monitor chip.	74
4.10	Power specifications for the INA219 current monitor chip.	74
4.11	Performance characteristics of the TDK Invensense MPU6050 6-axis inertial measurement unit (IMU) showing the accelerometer performance.	75
4.12	Performance characteristics of the TDK Invensense MPU6050 6-axis inertial measurement unit (IMU) showing the gyroscope performance.	75
4.13	Performance characteristics of the TDK Invensense MPU6050 6-axis inertial measurement unit (IMU) showing the device temperature rating, power consumption and price.	76
4.14	Key performance characteristics for the AT45DB641E flash chips (AT45DB641E) including temperature rating, power consumption, storage capacity and price.	77
4.15	Type and number of communication ports to facilitate communication among all the external modules.	78
4.16	Electrical parameters and power consumption characteristics for the STM32L476RG microcontroller.	79

4.17 Processing characteristics and parameters for the STM32L476RG microcontroller.	79
4.18 Communication ports and specifications for the STM32L476RG microcontroller.	80
4.19 Current consumption of various components as well as the estimated maximum possible current draw.	80
4.20 Key performance characteristics for the Texas Instruments LP3876 linear low dropout regulator (LP3876).	82
4.21 Approximate procurement cost for a single SHARC buoy node.	84
5.1 Configuration parameters for the system clock and real time clock (RTC) including sources and frequencies.	94
5.2 Power mode selection for each phase of the buoy's operational cycle. Power modes are preset by STMicroelectronics and are based on the device's architecture. A full explanation of each power mode can be found in (stm32l4ref).	95
5.3 Numerical and 8-bit unsigned integer representation of the synchronous states on the STM32L4 microcontroller.	100
5.4 Table showing the types of states that the system checks for ordered by priority with 1 being the highest priority and 3 being the lowest.	102
5.5 Table Showing the device behavior in case of a critical failure in one or more of the subsystems. Critical failures are defined in AT006 (Table B.7) testing protocol.	106
5.6 Breakdown of a typical NMEA message string with fields indicating start/stop sequences and character information.	108
5.7 Description of ZDA message string showing variables, description and how the example datum 5th September 2002 08:27:10 am is stored.	109
5.8 Description of GSA message string showing variables, description of parameters and how the variables are stored.	109
5.9 Description of GLL message string showing variables, description and how a set of coordinates e.g. (47°17.11364'N, 8° 33.91565') is stored.	110
5.10 Data collected from the GPS in a single sample session. Square brackets denote a numbered sequence of like variable types.	111
5.11 Description of output values from BMP280 post processing.	112
5.12 Description of parameters used to calibrate the INA219 current sensor.	112
5.13 Description of output values from INA219 current sensor.	113

5.14 Total drift data collected during a single sample point.	113
5.15 Description of output values from the MPU6050 IMU showing variable name, size and significance.	114
5.16 Parameters of the IMU and their configured value for this application.	114
5.17 Breakdown of data accumulated from the IMU with the sample parameters mentioned in Table 5.16.	115
5.18 Timing requirements for each sensor showing the interval between samples and the number of samples collected during each session.	115
6.1 Objectives of the unit tests defined in Appendix E.1 showing how the test protocols help validate the firmware in the device peripheral layer.	123
6.2 Baseline functionality of the GPS UART communication module of the firmware and the unit test used to verify this functionality.	124
6.3 Baseline functionality of the GPS UART communication module of the firmware and the test used to verify subsystem functionality.	125
6.4 Baseline functionality of the Iridium UART communication peripheral of the firmware and the test used to verify unit functionality.	126
6.5 Baseline functionality of the Iridium UART communication peripheral of the firmware and the test used to verify unit functionality.	127
6.6 Baseline functionality of the BMP280 SPI communication peripheral of the firmware and the tests used to verify unit functionality.	127
6.7 Baseline functionality of the BMP280 SPI communication peripheral of the firmware and the tests used to verify unit functionality.	128
6.8 Table showing the configuration parameters for the BMP280 environmental sensor for the final version of the buoy firmware.	128
6.9 Baseline functionality of the INA219 I ² C communication peripheral of the firmware and the tests used to verify unit functionality.	129
6.10 Subsystem functionality of the INA219 I ² C communication peripheral of the firmware and the tests used to verify unit functionality.	130
6.11 Baseline function of the AT45DB641E Flash chips SPI communication peripheral of the firmware and the tests used to verify unit functionality.	131
6.12 Subsystem functionality of the AT45DB641E Flash chips SPI communication peripheral of the firmware and the tests used to verify unit functionality.	132
6.13 Baseline functionality of the Iridium UART communication peripheral of the firmware and the test used to verify unit functionality.	132

6.14 Subsystem functionality of the MPU6050 six DoF IMU and the tests used to verify subsystem functionality.	133
6.15 System output during an accelerated test as described in SYS001. Data output occurs over the period from initialiation to first sample and sleep with a failed GPS signal acquisition.	136
6.16 System output during an accelerated test as described in SYS001. The data output occurs over the period from first wake up to the end of second sample with successful GPS signal acquisition.	137
6.17 Average current draw (mA) and cycle duration over the full sample period.	139
6.18 Data transmitted from the SHARC buoy as compressed binary messages in ASCII hexadecimal format. This table contains ice drift data shown in packets begining with 01 ("1") and IMU data in packets beginning with 57 or "W". Also shown are diagnostic data including the transmitting device, payload size and number of credits used to transmit the message.	142
6.19 Hardware components used during field testing for version 1 of SHARC buoy deployed during the 2019 SCALE winter cruise to the Southern Ocean Marginal Ice Zone.	143
6.20 Table showing the parameters the GPS was configured with before deployment.	145
6.21 Deployment conditions for buoy 1 (2019-WC-SB01) and buoy 2 (2019-WC-SB02) including deployment coordinates, time and environmental conditions	146
6.22 Results of the full system acceptance tests indicated by a tick in the appropriate column.	149
6.23 Results of the platform evaluation and how each functional requirement was addressed.	150
 B.1 Table showing the key stakeholders in the project, their level of involvement as well as their interests in the project.	171
B.2 Acceptance test protocol for subsystem connectivity testing.	172
B.3 Acceptance test protocol for fault testing.	173
B.4 Acceptance test protocol for component selection.	174
B.5 Acceptance test protocol for Subsystem Robustness Testing	174
B.6 Acceptance test protocol for accelerated system testing.	175
B.7 Acceptance test protocol for subsystem calibration testing.	176
B.8 Acceptance test protcol for power test.	177
B.9 Acceptance test protocol protocol for low temperature test.	178

B.10 Acceptance test protocol for final system deployment test.	179
C.1 Description of interrupt generated by the Iridium module on an external digital input line.	183
C.2 Description of routine for interrupts generated by the IMU on an external digital input line.	184
C.3 Description of event handling routine for a brown out recovery event. . .	184
C.4 Description of routine for handling low power events.	185
C.5 Description of routine for handling a software reset event.	185
C.6 Colour guide for the initialization routine flow diagrams.	185
D.1 List of data services provided by Iridium for transmission of data over the satellite network including the bandwidth and purpose of the service taken from (iridium_mobile).	194
D.2 Strategies used by the devices to transfer data from remote locations. Table includes transmission technologies and services used as well and transmission strategies and transmission intervals where given. Prices are converted to Rands (R) via (usdcoversion).	195
E.1 Protocol for firmware initialisation/deinitialisation functional test.	196
E.2 Protocol for the signal acquisition firmware test, This test is specific to receivers with wireless capabilities.	196
E.3 protocol for testing the peripheral communication functional unit. . . .	197
E.4 Protocol for testing a software function for validating sensor data.	197
E.5 Protocol for testing data streams and direct memory access (DMA) functions.	197
E.6 Protocol to test functions that interface with sensors by configuring settings in the register or reading data from a register.	198
E.7 Protocols for fault testing of software subsystem functions.	198
E.8 Protocols for testing digital input/output peripheral functional units. . .	198
E.9 Results of subsystem acceptance tests for each of the identified modules. Modules that were successfully validated were marked with a ✓, failed tests were marked by an X and tests that could not be applied to a subsystem were marked by an N/A.	199

Chapter 1

Introduction

1.1 Background to investigation

1.1.1 Antarctic sea ice and the Marginal Ice Zone

Antarctica plays host to ocean-atmospheric processes, which are significant drivers of global climate. These interactions are strongly influenced by sea ice coverage and extent, which acts as a boundary layer between the atmosphere and ocean (**parkinson2004southern**). Sea ice reflects incoming solar radiation which, otherwise, would be absorbed by the ocean influencing ocean temperature and salinity (**parkinson2004southern**). Additionally, sea ice expanse has been shown to insulate the ocean and reduce evaporation while regulating heat flux, and gaseous exchange (**deconto2003rapid**) providing a stable habitat for the diverse ecosystem that inhabits the region (**arrigo2004large**) (see Figure 1.3). Sea ice provides a damping effect on oceanic kinetic energy as it regulates the mass and movement of waves around the Southern Ocean (**parkinson2004southern**; **roach2020antarctic**). High winds, cyclone frequency and prominent wave activity perturbs the ice preventing it from congealing at the ocean edge. This result in a region of semi-consolidated ice masses known as the Marginal Ice Zone (MIZ) (**wadhams1987ice**).

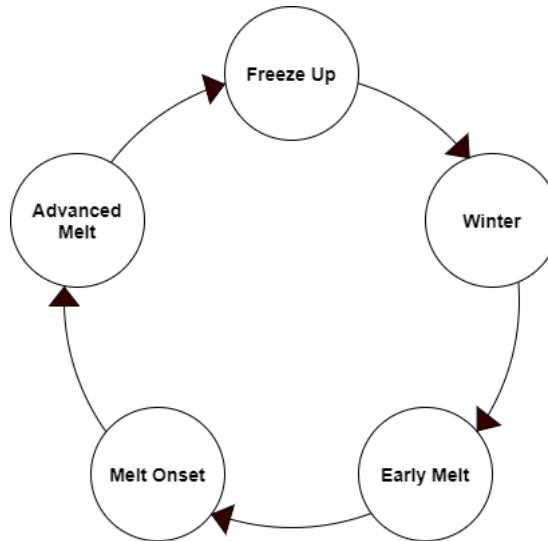


Figure 1.1: Diagram showing the lifecycle of Antarctic sea ice as observed using remote sensing based on **barber2005microwave**. The cycle begins with the congealing of the ocean surface known as Freeze Up with peak ice formation during Winter, the second half of the cycle is characterised by various stages of melt with complete sea ice retreat occurring in the Summer.

Sea ice formation in the Southern Ocean begins when the surface layer of the ocean congeals causing ice crystals to form. These crystals combine to form frazil ice; the concentration of which increases as the heat from the ocean is removed by the atmosphere (**arrigo2004large**). As the wind and wave activity begin to calm, frazil ice combines into grease ice, which grows into pancake ice floes (**arrigo2004large**). This growth period is known as "freeze-up" and forms the first stages of the sea ice life cycle (**barber2005microwave**). Additionally, strong winds in the Marginal Ice Zones cause the floes to drift over long distances (**alberello2019drift**). These winds, coupled with high ice floe density result in collisions between floes causing them to break apart (**STEER2008933**) resulting in brash ice (**icedefinition1992**). During the winter months, newly formed ice grows a layer of snow (**barber2005microwave**) and is termed "first year ice"¹. Brine is present between the ice fragments and the layer of snow is affected by the level of precipitation (**barber2005microwave**).

¹First year ice is newly consolidated ice that has been growing for less than one winter's growth period (**icedefinition1992**).

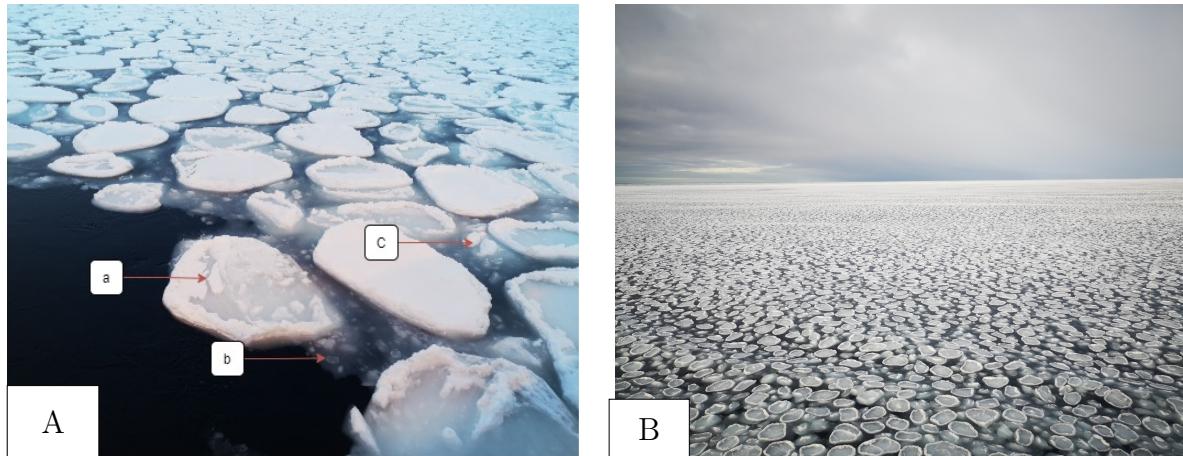


Figure 1.2: (A) Diagram showing different types of ice present in the Marginal Ice Zone including (a) pancake ice floes (b) frazil ice and (c) brash ice. Some ice floes exhibit snow growth along the ridges while other floes exhibit flooding. (B) Sea ice in the Southern Ocean MIZ during July 2019, where pancake ice is the predominant concentration while brash ice is the smallest. Swell waves can also be observed propagating through the region. Photos taken during the early sea ice formation period during the SCALE winter cruise July 2019 by the author.

Finally, the seasonal cycle closes with three phases of melting. Early melt marks the transitional period where moisture is continuously present in snow cover on the sea ice (**barber2005microwave**). The level of moisture in the snow increases through "melt-onset" brought on by the transition from summer to winter and increasing ocean/atmospheric temperatures (**barber2005microwave**). Eventually, the snow and ice surface begin to melt rapidly during the "advanced melt" phase resulting in complete desalination of the ice floes followed by the breaking up of the sea ice sheets (**barber2005microwave**).

The result of these processes is a region of semi-consolidated ice, which extends an estimated 19 million km² from the Antarctic continent (**MAKSYM2012seaiceextent**). These ice floes increase in size through gradual heating, fusing the floes into packed ice (**arrigo2004large**). Sea ice concentration in the MIZ covers an observed range of 100% of total ice concentration from summer to winter (**alberello2019drift**). However, **alberello2019drift** further show that this cover is an unconsolidated mixture of different ice types rather than a single ice cover therefore showing that this metric may not represent the true concentration of sea ice types on a high resolution scale. The variability of the MIZ, coupled with the strong storms and weather patterns, drive the strongest atmospheric-ocean-sea ice interactions in the region. **alberello2019drift** further highlight that knowledge of MIZ sea ice dynamics is required to model the response to storms as well as predict the regional response to changes in climate.

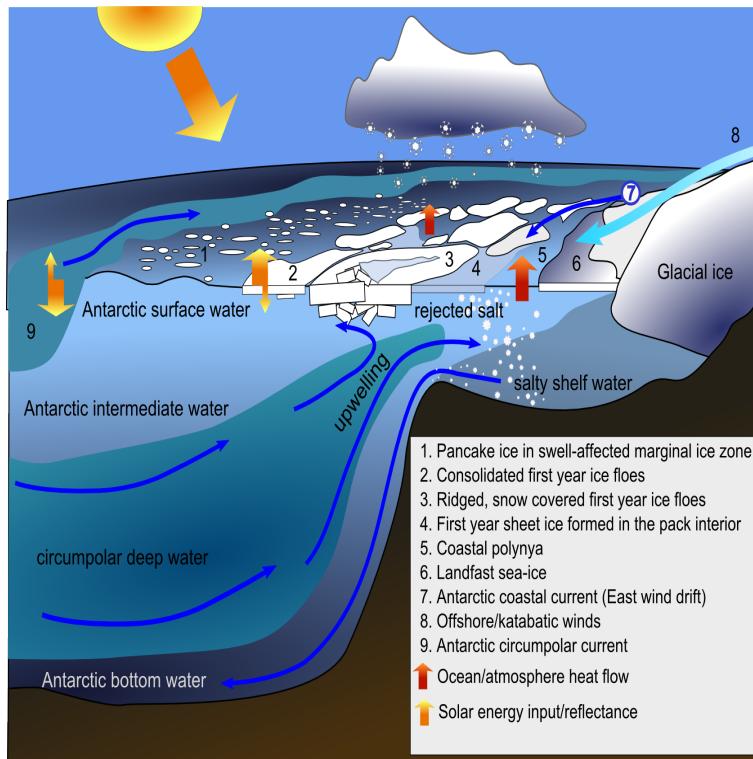


Figure 1.3: Diagram showing the expanse of sea ice in the Southern Ocean. Packed ice forms closer to the continent in calmer conditions while strong oceanic currents, winds, wave action and extreme temperatures result in the formation of semi-consolidated ice in the Marginal Ice Zone (1 to 5). Here, ice formation is highly seasonal expanding to a maximum in winter and retreating to a minimum in summer. Sea ice acts as a boundary layer influencing heat and gaseous exchange between the atmosphere and ocean. Figure taken from [Antseaice](#).

1.1.2 Observations of the Marginal Ice Zone

Antarctic sea ice has gained recognition for playing a critical role in global climate systems ([kennicutt2016delivering](#)). There has been growing interest by the global scientific community in Antarctic Research since the first International Polar Years ([kennicutt2016delivering](#)). International collaborations have sought to formalise Antarctic Research and unite efforts under common goals ([kennicutt2016delivering](#)) with the formation of the Scientific Committee on Antarctic Research (SCAR), which has resulted in increased sampling in the MIZ. However, data from the marginal and packed ice zones are under-sampled and poorly represented ([vichi2019effects](#)). Very little in situ data exist to fully understand the environmental conditions surrounding the key metamorphic phases of sea ice in the Antarctic MIZ. Current climate models and observations exist based on data sets from the Arctic ([vichi2019effects](#)), which, when applied to Antarctic sea ice, fail to accurately capture the dynamics of the region on a high-resolution scale. Sea ice expeditions have traditionally taken place during the summer months when the sea ice extent is at a minimum ([kennicutt2016delivering](#)). This results in temporal and spatial gaps in seasonal measurements, which fail to characterise sea ice expanse during the fundamental formation periods ([MAKSYM2012seacieextent](#)). Additionally, these data are criti-

cal for understanding Southern Ocean sea ice dynamics and observed phenomenon in the Marginal Ice Zone such as waves-in-ice (**kohout2014storm**). Furthermore, polar oceanic and atmospheric measurements are critical for understanding the local climates since the high cyclonic activity in the region affects the heat and moisture delivery to higher latitudes (**vichi2019effects**).

Almost all data collected from in situ measurements in the region are taken during seasonal manned expeditions. Only 22 countries have access and shipping capabilities to initiate expeditions to the region. Additionally, these expeditions require vast resources and complex logistical operations. Furthermore these missions are time sensitive and cancelled expeditions create gaps in seasonal data sets. The harsh seasonal climate causes certain, vital areas of the MIZ and packed ice zones to become inaccessible during winter months. As a result, missions only occur during certain seasons resulting in temporal gaps. Attempts have been made to fill in these gaps using data from Arctic climate programs e.g. **lee2012program**. However, these attempts fail to characterise the region and accurately capture seasonal variability. For example, in 2016, an anomaly in the sea ice extent was detected where the ice retreated 48% faster than the mean rate (**turner2017unprecedented**). **vichi2019effects** have shown the region to be a hot-spot for cyclonic activity, which regularly impacts ice formation within the marginal ice zone. However these interactions are not captured by current climate models (**vichi2019effects**).



Figure 1.4: Photo taken in the Marginal Ice Zone from onboard the SA Aghulas II during the SCALE winter expedition in 2019 by the author. The vessel is anchored in consolidated ice with the UCT²-UDE³sea ice team performing ice coring activities on the surface of the ice.

Therefore, a call to increase sensing in the region has arisen to fill in the gaps of these temporal data sets (**kennicutt2019sustained**). A review by **kennicutt2016delivering** highlights a need to revolutionise Antarctic science to overcome these challenges. As part of the plan, SCAR identified technology as playing a pivotal role in Antarctic research.

²University of Cape Town

³University of Duisburg-Essen

Air, sea and space-borne technologies can replace manned-expeditions which can provide in situ monitoring on a macro and micro scale (**kennicutt2016delivering**). Technology can provide a potential solution to long-term monitoring. Robust, power-efficient solutions that are capable of performing long-term functions in a non-invasive manner are required to reduce the need for implementing new infrastructure.

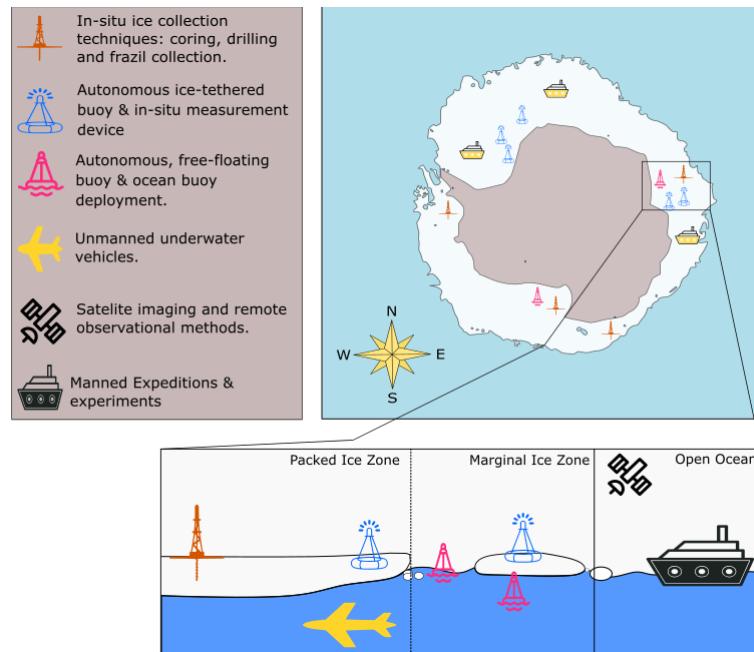


Figure 1.5: Diagram showing the current state of Antarctic sea ice measurement technologies for each level of observation as well as the estimated deployment location. Diagram taken when sea ice extent is at a maximum. This diagram is derived from the technology implementation strategy identified from the 2016 SCAR roadmap (**kennicutt2019sustained**)⁴ and has been adapted to show sea ice observational techniques.

1.1.3 Overview of the existing technology and devices

Modern technology has seen an increased use in remote monitoring of the continent (**kennicutt2016delivering**). For example, satellite imaging is the most prevalent technology for monitoring sea ice in both the Arctic and Antarctic regions. It provides large-scale mapping of sea ice extent, thickness, snow cover at the cost of a low spatial resolution (**turner2017unprecedented**; **galin2011validation**; **alberello2019drift**). These techniques allowed for the detection of the rapid sea ice retreat recorded in 2016 (**turner2017unprecedented**) and are very useful for large-scale representation. However, satellite imaging is severely constrained by its resolution (**emery1997satellite**). Pixel sizes of synthetic aperture radar (SAR) images are in the order of 10 to 100 m (**galin2011validation**) where, for example, snow thickness can vary down to the cm. Furthermore, cloud cover can compromise the measurements resulting in missing data.

⁴Figure made using icons from Flaticon.com, Buoy by hunotika from the Noun Project, oil rig by Mourad Mokrane from the Noun Project, Plane by jokokerto from the Noun Project.

Finally, these measurements require validation against data models, which tend to under-represent climate in the region (**galin2011validation**; **emery1997satellite**). Hence, a need arises for the development of in situ technology that can provide accurate, detailed information down to the highest possible resolution and allow for long term, large scale monitoring of ocean-ice-atmosphere processes. Hence we turn to autonomous platforms as a solution.

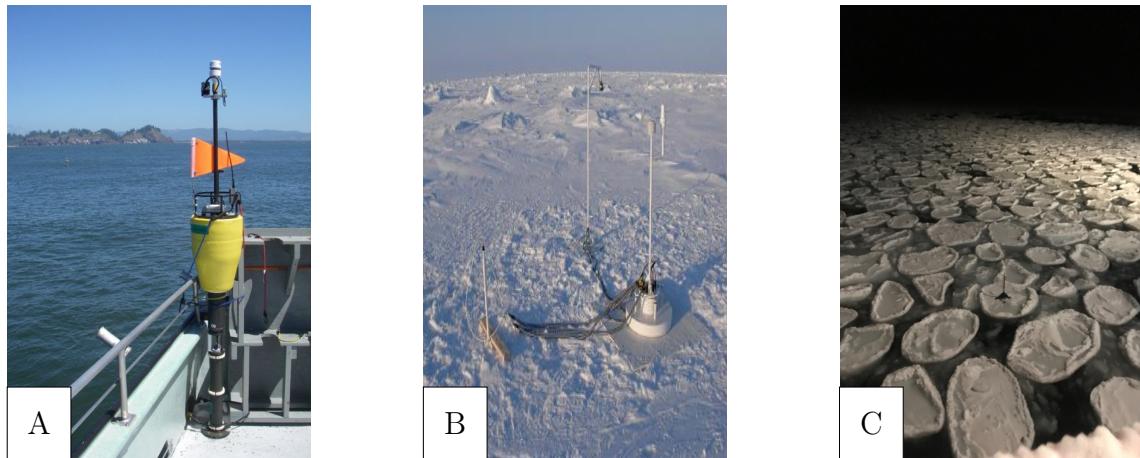


Figure 1.6: Practical examples of instruments used to collect in situ measurements in the sea ice region. These comprise: (A) the Surface Wave Instrument Float Tracking (SWIFT) buoy developed by the University of Washington (**thomson2012wave**); (B) the Ice Mass Balance (IMB) buoy developed by Dartmouth College [(**PLANCK2019102792**) image source: (**simbfact**)]; and (C) the Southern Hemisphere Antarctic Research Collaborative (SHARC) buoy developed by the University of Cape Town (photo courtesy of R. Verrinder).

Delivering a remote system capable of long-term functionality is a high priority for Antarctic science (**kennicutt2016delivering**) and will accrue robust, reliable, time-sensitive data-sets to populate these models. This will bring climate models in line with current observations and will allow for a quantified, thorough description of a local phenomenon, for example, the role of ocean swells in ice formation in the Marginal Ice Zone (**doble2013wave**; **doble2017robust**). A successful system should have the following characteristics⁵:

1. Autonomous or sustained deployment capabilities
2. Adequate remote sensing capabilities
3. Improved, robust power supplies
4. Real-time data collection, transfer and analysis
5. Survivable under extreme weather conditions

⁵Taken from a collective survey distributed during the Antarctic Roadmap Challenge in 2016 (**kennicutt2016delivering**).

However, the current state of modern Antarctic observational technology is underdeveloped; current prototypes are too expensive and difficult to obtain by the scientific community (**kennicutt2016delivering**). Many institutions have initiated projects to develop autonomous systems such as buoys (examples shown in Figure 1.6) and unmanned surface vehicles (USVs) such as Wave Gliders (**liquidrobot2016wave**), which have been utilised successfully in Antarctic and Arctic oceanographic studies (**swart2020submesoscale**). For example, SWIFT buoys deployed in the Arctic Marginal Ice Zone were used to validate Marine Radar measurements of sea ice drift in the Beaufort Sea (**lund2018Arctic**). However, these systems are niche and require a technical crew to deploy and retrieve the buoys. The devices are generally proprietary with fixed sets of sensors and fewer sensing capabilities rendering the device inflexible to the needs of scientists (**rabault2017measurements**). **rabault2019open** note the growing use of open source hardware and off-the-shelf technology in modern systems. Off-the-shelf components have reached a state where the components are well documented and specified to withstand the requirements for polar systems (**rabault2019open**). Open-source hardware has allowed for the free exchanging of designs allowing scientists to build their own devices without needing to design and test prototypes (**rabault2019open**). As a result, there has been a growth in literature on open access devices with designs and code bases readily available on code sharing sites such as GitHub⁶. Hence, inclusion of cost-effective technology as a solution is a growing trend.

Additionally, these devices have shown promise in the field. **rabault2019open** developed an open-source multi-sensing autonomous system and **kohout2015device** developed a multi-sensing system with off-the components. The devices were deployed in the Arctic and Antarctic Marginal Ice Zones. The buoys developed by **kohout2015device** encountered technical issues resulting in a total of 39 days survivability with two buoys lost immediately after deployment, one buoy surviving for nine days and another for 17.5 days (**kohout2015device**). The devices were deployed again in the Ross Sea by **kohout_smith_roach_williams_montiel_williams_2020** in April 2020 where a buoy lasted for 66 days (**kohout_smith_roach_williams_montiel_williams_2020**). The buoys developed by **rabault2019open** included solar panels to recharge the batteries. However these systems survived for only 12 days during the spring. Other systems deployed in the region are the MetOcean buoys (**uptempo**), Surface Wave Instrument Float Tracking (SWIFT) (**thomson2012wave**) buoys and Trident buoys (**trident**). These systems, however, are expensive and do not have the sensing capabilities specifically for sea ice dynamics. Full details of these remote sensing technologies are discussed in Chapter 2. Consequently, this presents a problem for in situ sea ice observations in that multiple systems are required to collect desired data for models with a need for back up systems in case of failure.

Therefore we are presented with a unique opportunity to design a series of novel ice-tethered autonomous systems to increase remote sensing at an affordable rate. The goal of the project is to design a proof of concept system with expandable, modular capabilities capable of running off a single power module for seasonal periods as part of the South African National Antarctic Programme (SANAP), and it is led by the Marine and Antarctic Research centre for Innovation and Sustainability (MARIS) at the

⁶For example: see **rabault2019open** Github repository at https://github.com/jerabaul29/LoggerWavesInIce_InSituWithIridium.

University of Cape Town (UCT), in collaboration with the South African Weather Service (SAWS). The focus of this collaboration is to better understand sea ice lifecycle in the Southern Ocean. Hence, the proposed system aims to provide a low-cost, easy to deploy environmental data measurement system that can be expandable to operate in a network allowing for a single-system deployment strategy. In this thesis, the firmware design of a novel ice-tethered buoy for the Antarctic Marginal Ice Zone is presented. The goal is to develop a robust software system for a platform built using off-the-shelf components to monitor ice drift, atmospheric conditions and waves-in-ice measurements over a full seasonal cycle.



Figure 1.7: Novel ice drift, environmental monitoring and wave measurement autonomous platform: the Southern Hemisphere Antarctic Research Collaboration (SHARC) Buoy. Developed by the University of Cape Town. Photo by R. Verrinder.

1.2 Problem statement

This project aims to design a prototype system to monitor environmental conditions that lead to the formation of ice floes in the Marginal Ice Zone. The goal of the project is to increase in situ sensing capability in an affordable manner while allowing for easy access to the technology and data. Conclusion of this project will result in a fully automated system that can be deployed on an ice floe in various locations in the Marginal Ice Zone up to the packed ice boundary layer. Transfer of data from the system will occur using the available infrastructure in the region to reduce costs and make the system as non-invasive as possible. Furthermore, the resulting system will run off a portable power source with limited charging capability to survive for at least one month. By identifying and engaging with key stakeholders in the project, we aim to design a system using off-the-shelf components and synthesise these components into a low-cost, high-performance solution with the final deliverable being a deployable system for the next South African Antarctic expeditions to the Marginal Ice Zone. Thereby creating the following project

objectives:

1. Perform a literature review of the current state of remote sensing in the Southern Ocean, analysing the techniques and strategies implemented by each system.
2. Engage with key stakeholders to create a set of user requirements for the system, translate these requirements into system specifications, identifying critical subsystems and synthesising them into a high-level system.
3. Identify a suitable network for remote communication as well as the corresponding communication module.
4. Select suitable hardware components and develop a robust set of libraries and unit tests for each system.
5. Identify the energy requirements and select a suitable power source.
6. Identify a processor/set of processors that meet the requirements for the system and develop firmware in C using a Hardware Abstract Layer (HAL).
7. Connect the subsystems to a motherboard and place the system in an enclosure capable of protecting the system from the harsh climate.
8. Evaluate the platform against a series of unit tests, robustness tests and hardware tests.
9. Deploy the system in the Marginal Ice Zone and monitor performance.

1.2.1 Project initiation

The project was initiated in 2018 with input from the following stakeholders.

Table 1.1: Legend showing the key stakeholders in the initiation of the project as discussed in the phases below. Legend includes name, reference number and department/institution.

Reference:	Name:	Institution	Department
1	Jarryd Son	University of Cape Town	Electrical Engineering
2	Nadir Vorajee	University of Cape Town	Electrical Engineering
3	Prof Amit Mishra	University of Cape Town	Electrical Engineering
4	A/Prof Marcello Vichi	University of Cape Town	Oceanography
5	A/Prof Sebastian Skatulla	University of Cape Town	Civil Engineering
6	Marc de Vos	South African Weather Service	N/A

⁶Image source:<https://www.digikey.co.za/short/d95btr8d>

Concept phase

The concept design was performed by Son (1) and Vorajee (2) under the supervision of Prof Mishra (3). The design was presented at the first meeting on the 11 September 2018 to the principle stakeholders A/Prof Vichi (4) and A/Prof Sebastian Skatulla (5). The proposed device was presented as an upgraded version of the Trident sensing buoy with expanded sensing capabilities. During the meeting, it was agreed to deploy and test the system during the Winter and Spring SCALE expeditions in 2019. The provisional dates for these cruises were: July 2019 and November 2019 respectively. A follow up meeting occurred on the 4th October 2018 where de Vos (6), (South African Weather Service (SAWS)) expressed interest in the research and the development of the project. de Vos (6) provided additional context for the project and presented current work conducted by SAWS.

Procurement phase

A preliminary user requirement was conducted by Son and Vorajee (1, 2). From there, suitable components for each of the subsystems were selected with orders being placed on the 19th December 2018. This order included sensors, a u-blox GPS receiver, Zigbee short range communication modules, Rock Seven satellite communication modules, STMicroelectronics STM32F1 microcontroller and inertial measurement units (IMUs).

Handover phase

The project was handed over to Jamie Jacobson under the supervision of Robyn Verrinder on the 8th February 2019, officially commencing the prototyping phase of the project.

1.2.2 Scope

This thesis focuses on the firmware design for the buoys. The outcome of this study will be a robust set of firmware and unit tests to validate the firmware against. The project timeline takes place from February 2019 to December 2020 with key deliverable dates being:

1. 18 June 2019 - SCALE Winter Cruise (Version 1.0 Complete)
2. 12 October 2019 - SCALE Spring Cruise (Version 1.0 Revisions)

As this thesis focuses on software development of the project, hardware design is not included in this study and future designs are discussed as recommendations. This includes the mechanical structure and the electronic modules. However, this study includes the selection of devices, which are evaluated against the technical specifications. Additionally, the hardware modules are described.

System testing is conducted on a subsystem software level and the full system. The testing procedures are described in Tables B.2 to B.10.

Large scale calibrations are not included in the project scope due to tight timeline constraints. Finally, the design, implementation and calibration of an IMU-based wave measurement algorithm are not explored in this project. The IMU however, will be validated and verified by sampling enough data to fit into a single Iridium transmission packet.

1.2.3 Limitations and constraints

The largest limitation to the project is time constraints. The project timeline coincides with the SCALE research cruise using the winter and spring expeditions for buoy deployment thereby limiting the time frame for development. Additionally, the firmware development is limited to the capabilities of the selected processor.

The firmware development is heavily constrained by the hardware selected for the platform. Peripheral drivers were written for modules that were agreed upon by the project members. Additionally, the IMU, processor, environmental sensor and satellite modem components were pre-selected in 2018. The firmware is thus constrained by design choices originally made in 2018 and early 2019 for the first version of the system. However, these designs were revised for subsequent versions of the buoy from September 2019. As a result of these constraints, the devices were designed without previous knowledge of the environment and with a limited number of sensors. Finally, the selected processor has a limited number of communication peripherals which influenced the types of sensors that were selected.

The communications network in Antarctica is severely limited and the most reliable form of remote communications is the Iridium satellite network. The amount of data that can be transmitted is limited in terms of bandwidth, data costs, packets structure and reliability of transmission. Testing for Antarctic conditions is restricted by available testing facilities, therefore, rigorous environmental tests may only be conducted during the expeditions.

The first prototype was deployed with a limited number of sensors due to development constraints. Mechanical failures resulted in the buoys ceasing operation within an hour of deployment during the 2019 SCALE Winter cruise. Further development occurred in 2020 to increase the sensing capability of the buoy. However due to the 2020 COVID-19 outbreak, all expeditions were cancelled for the year and therefore final system testing in the Antarctic MIZ was not possible. Attempts were made to deploy the devices on other expeditions from other countries. However, shipping delays were encountered preventing the device from reaching the expedition team on time. Currently, a prototype version has been sent with the SA Agulhas II to the SANAE IV base on the Antarctic continent where testing is expected to take place in late February 2021. This falls outside the time frame of this dissertation. Instead, the buoy will be tested on the home continent with low temperature tests being conducted in a commercial -20°C freezer. Currently, two buoys are onboard the RV Polarstern and should be deployed in mid-March 2021 alongside with the Alfred Wegener Institute (AWI) buoys.

1.2.4 Assumptions

The following assumptions were made during the development of the firmware for the buoy. The device has sufficient power to access any of the sub modules if required. Devices that pass a connection test, are considered "online" and capable of producing reliable data. The processors are the Arm-based STMicroelectronics STM32L4 and STM32F4 microcontrollers and do not come preloaded with any real time/operating system. Development will take place using the Hardware Abstract Layer (HAL) driver files and all hardware that has been selected is rated for the environmental conditions described in Section 1.1. A system/subsystem is considered valid if it passes a suite of acceptance tests and verified if it meets the functional requirements. Devices that are not active need to be placed in power down mode. Finally, the system is considered complete if it can complete a single measurement cycle from power on without the assistance of any auxiliary equipment.

1.3 Plan of development

The plan of development describes how the project was conducted through the various stages. A literature review was conducted to analyse the current state of in situ monitoring technology in the region. Then, a problem statement was defined by engaging with project stakeholders and developing a set of user requirements. The user requirements were used to formulate acceptance tests and technical specifications which were used to guide subsystem design and selection. Then, the firmware stage was initiated with the development of Application Programming Interface (API) libraries for each module of the device. These were then synthesised and sequenced into a software system defined by short periods of activity and long periods of inactivity. This was used to optimise the device for low power consumption. The system was tested by performing a power consumption test, which was used to evaluate the power characteristics. The device was setup outside to run a full transmission cycle. Finally the results were analysed and used to validate the buoy as a viable tool for remote Antarctic monitoring.

1.4 Report structure

Table 1.2: Description of report structure including key phases of the project and significance

Chapter	Phase	Description
Chapter 2	Literature review	A description of current sampling techniques using un-manned instrumentation in Polar sea ice research is presented. From this review, the key measurement objectives of each system are identified and an analysis of the state of the art will be used to identify the usefulness and areas where SHARC Buoy can provide a solution.

Chapter 3.1 to 3.3	System development	An analysis of project stakeholders is provided as well as an assessment of their needs. Then, a set of user requirements is developed and ranked in order of importance. The functional requirements selected will guide the device selection and, ultimately, be used to evaluate the performance of the final system. This lead to the identification of the critical subsystems shown in Table 3.7 and a final system topology choice. A set of technical specifications were derived for subsystem hardware selection and a suite of acceptance tests were written to ensure the components conformed to the desired specifications.
Chapter 3.4 to 4.3	Platform overview	A description of the mechanical and electronic components that were selected for the device. The specifications of each component and price are given. The final system consists of a u-blox NEO-7M GPS, Rock SEven RockBLOCK 9603 Iridium modem, Bosch Sensortech BMP28 environmental sensor, TDK InvenSense MPU6050 Inertial Measurement Unit and Texas Instruments INA219 power monitor. Flash chips were selected as a permanent storage solution for data during phases of inactivity. The components were synthesised on a stack of three PCBs shown in Figure 4.13. A separate power module is shown in Figure 4.12. An overview on the assembly of the project is given to close the section.
Chapter 5	Software overview	A complete overview of the software design process is provided. The key features and focus of the software are outlined along with the firmware structure as shown in Figure 5.2. The project structure includes a breakdown of files, structure and driver files. The configuration of the processor for this application is shown as well as various peripherals and configurations used to set the device up for execution. Then a brief discussion about power mode and system selection ensued to provide clarity on the power-consumption optimisation process. Then a description of the firmware is given. A decision was made to implement a state machine. Here, a finite set of routines were defined and a description of the sequencing was given. Finally an over view of the flow of data from the device to the user was provided.
Chapter 6.2 to 6.3	Testing	In this section, the tests conducted on the platform and the system are given. These include subsystem acceptance test, full system tests, power tests and preliminary deployment test results from the 2019 SCALE winter expedition.
Chapter 6.4	Final evaluation	The results of subsystem acceptance tests are used to validate the system. The outcome of the project is compared to the functional requirements to determine the system's performance and verify that the project goals have been met.
Chapter 6.5	Discussion	This section provides a discussion of the results and key findings. The discussion focuses on the limitations of the power module and the outcome of the power test. Additionally, the performance of the device during the deployment test is discussed. An in-depth analysis of additional subsystem limitations is provided along with the performance of the firmware in spite of these limitations. The section concludes with a comparison of the buoy against other devices in the field.
Chapter 7	Conclusion	The outcome of the project is presented and a conclusion is made about the project outcomes, goals and whether the firmware was able to achieve them.

Chapter 8	Recommendations	Improvements and recommendations are provided for future work on the project. These include tests that could not be conducted, research and focus areas as well as hardware/software improvements.
-----------	-----------------	--

Chapter 2

Literature review

As highlighted in Section 1.1, there is a great desire by research scientists for in situ instruments to provide observation data which fills spatial and temporal measurement gaps in Antarctic sea ice monitoring in the Southern Ocean as shown in Figure 2.1. This chapter investigates the current use of autonomous monitoring technology used by research scientists in this region focusing on progress made to improving in situ observations and highlighting areas for improvement.

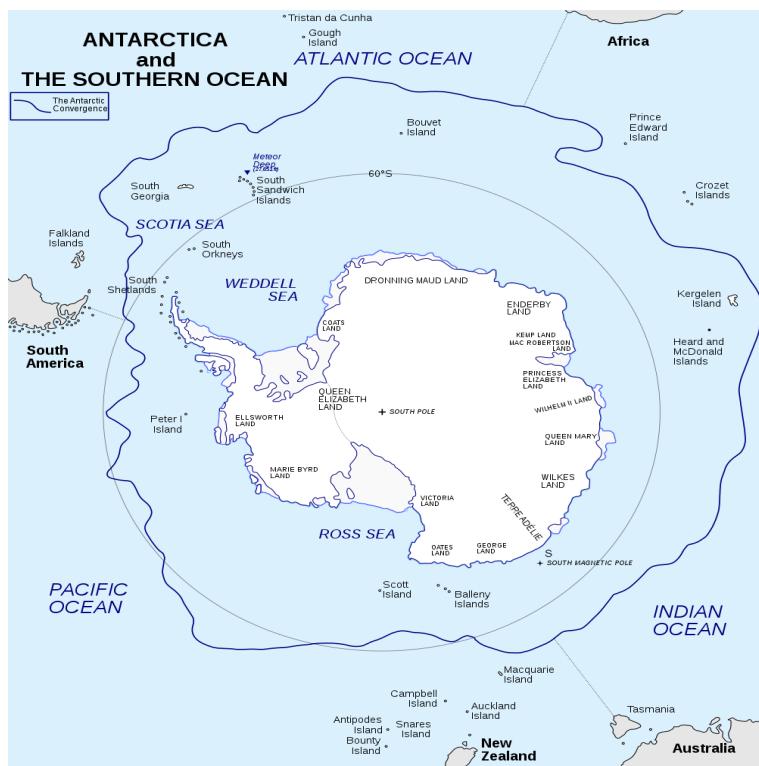


Figure 2.1: Map of the Southern Ocean surrounding the Antarctic continent. Image created by **Hogweed2015Ocean** and licensed by CC BY-SA 3.0.

In this review, a system-level analysis is provided where the objectives of each buoy are described in the contexts of the requirements for new remote sensing technology

as highlighted by **kennicutt2016delivering**. This includes an analysis on how these devices operated in remote conditions and the communication techniques employed in spite of a lack of infrastructure. The system-level analysis concludes with a discussion of the device measurement objectives, their performance in the Antarctic sea ice region and how this performance compares to their deployment in the Arctic region.

Next, a subsystem level analysis is given discussing how each device met their measurement objectives, what technology was used and how this technology works. Additionally, a comparison of different techniques for each subsystem is provided amongst devices with similar components and, where possible, a discussion of their accuracy is given. This section also highlights the critical measurement objectives of each device to understand which sub-modules are the most important for remote sensing. Finally, the processing architecture is discussed in the context of how it was implemented in the devices, what purpose it serves and how the data requirements for each device's sensors were met.

2.1 In situ climate sensing technologies

Autonomous instrumentation has seen increased use for in situ observations in the polar sea ice regions (**kennicutt2016delivering**). These devices have typically been developed by the commercial sector (**rabault2017measurements**) from companies such as Trident (**trident**), MetOcean (**uptempo**), Seabird (**seabird2021website**) and Sea Technology Services (STS) (**sts2021website**). Additionally, academic institutions have also developed in situ measurement devices such as the University of Washington's SWIFT buoy (**thomson2012wave**) or the University of Dartmouth's Seasonal Ice Mass Balance (SIMB) buoy (**polashenski2011seasonal**). While these technologies have the benefit of reliability, they are often expensive (**rabault2017measurements**) and inflexible to the specific needs of polar scientists. Technology has reached a point where low-cost and open-source alternatives are well documented and reliable enough to be integrated into customised solutions (**rabault2019open**).

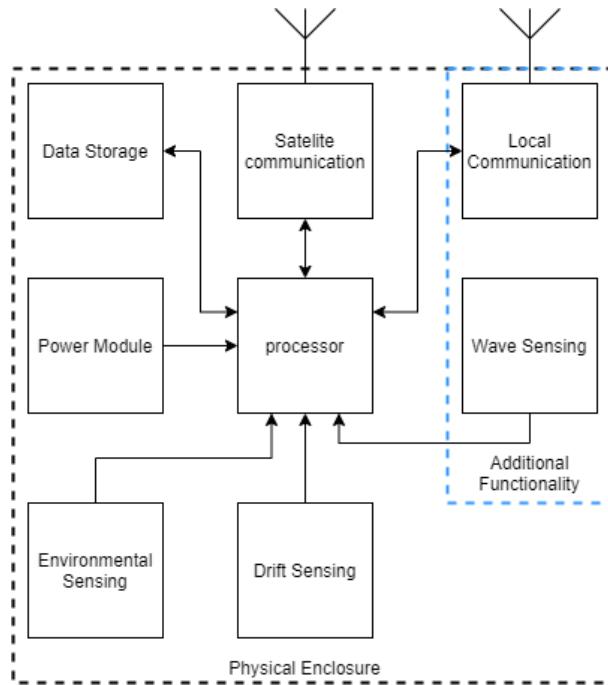


Figure 2.2: A block diagram of a typical device for in situ polar sea ice measurements. Each device contains modules for environmental sensing and drift sensing connected to a processor. This can be a microcontroller or microprocessor. Additionally, a data storage module (such as an SD card) is included to store data during the operation. Some devices include modules for local communication or wave sensing units. Finally, a remote communication module is included to transfer the data to a research centre or a user. The electronics are placed inside a physical enclosure for protection against the polar climate while a portable power module supplies the device during its operation.

In this section, a comparison of data collection devices used in the polar regions is presented as well as description of their design, operation and deployment. Where possible, certain specifications have been converted into standardised formats. To ensure a fair evaluation, data were collected from the latest technical publication of each platform, where possible. These publications may not contain all relevant data. In these cases, the data entries have been marked with a "Not Reported" or "NR".

Eight platforms were selected for the comparison and are shown in Figure 2.3 with each device designed by a private company or an institution. The key collaborators as well as the name of the institution are provided in Table 2.1. Where a buoy name is not given, the device will be named after the key contributor to the project. These systems have been selected due to their prevalence in global polar/oceanographic science as well as notability in publications. Device performance is evaluated in the context of the requirements set out by **kennicutt2016delivering** for an autonomous in situ measurement device shown in Section 1.1.

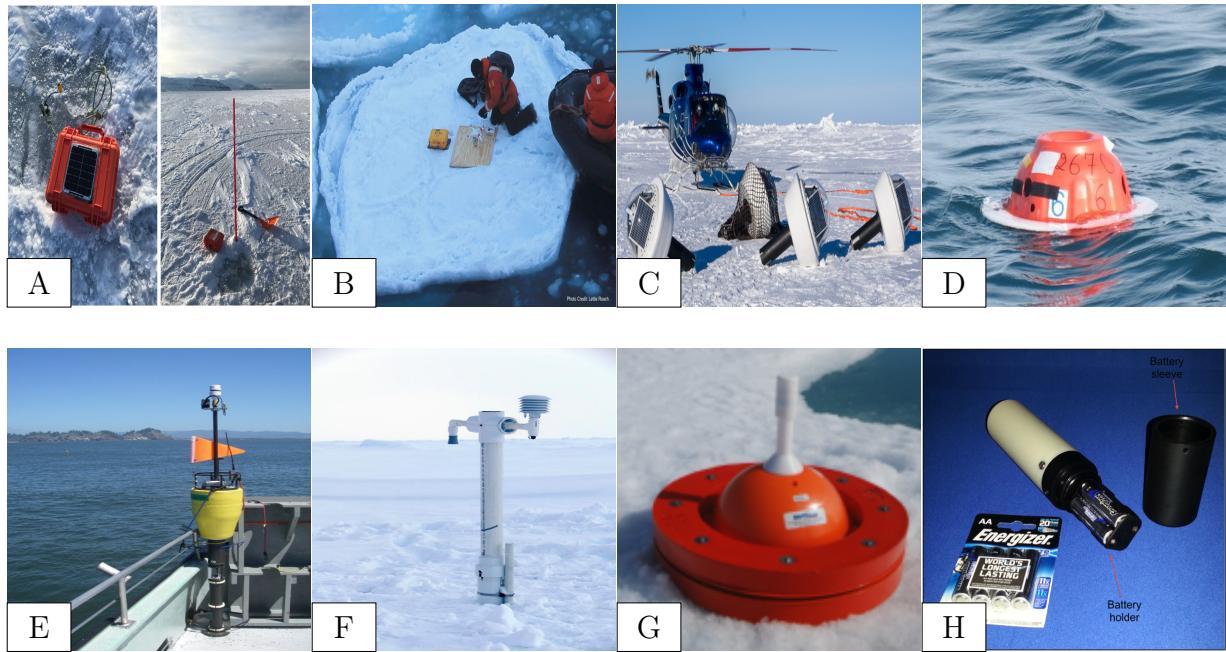


Figure 2.3: Devices used for the comparison of autonomous instruments deployed in the sea ice region. Each device has been selected for its notability in published work as well as prevalence in sea ice and wave interactions in the Marginal Ice Zones. These devices are: (A) Wave in Ice Buoy developed by [rabault2017measurements](#), (B) Wave in Ice Observational System by [kohout2015device](#), (C) Novel Wave Directional buoys by [doble2017robust](#), (D) Surface Kinematic buoy by [guimaraes2018surface](#), (E) Surface Wave Instrument Float Tracking buoy by [thomson2012wave](#), (F) Seasonal Ice Mass Balance buoy by [polashenski2011seasonal; meng2014optimal](#), (G) Polar ISVP by MetOcean ([uptempo](#)), (H) Trident buoy by Trident ([trident](#)).

A device that has sufficient autonomy and sustained capabilities will operate remotely with no human intervention. Therefore, the operating period of each device is compared. This is the period between deployment and final transmission where the buoy is active. Additionally, the techniques for remote communication for each buoy are examined in terms of data rates, coverage and transmission strategy to determine the techniques used to achieve remote communication effectively including a brief overview of available satellite networks. Then, to measure the sensing capabilities of each device, the measurement objectives are discussed as well as the hardware modules and software used to determine information. To compare the real time data collection, transfer and analysis, the device's processing strategy and storage strategies are considered. Data transfer techniques are analysed with the remote communication section. Finally, device performance in the polar regions is evaluated through the success of each deployment, device deployment time, data integrity as well as devices that failed and the causes of those failures.

Table 2.1: Devices used for the comparison including the device name, lead developer and the institution. These consist of both commercial and institutional devices for in situ sea ice and wave measurements.

Device Name	Developed By	Institution
Waves in Ice Buoy (WIIB)	Jean Rabault	University of Oslo, Norway (rabault2019open)
Waves in Ice Observational System (WIIOS)	Alison Kohout	National Institute of Water and Atmospheric Research (kohout2015device), New Zealand
Novel Directional Wave Buoys (NDWB)	Martin J. Doble	Polar Scientific (Ltd.), United Kingdom (doble2017robust)
Surface Kinematic Buoy (SKIB)	Pedro Veras Guimarães	Université de Bretagne Occidentale, France (guimaraes2018surface)
Surface Wave Instrument Float with Tracking (SWIFT) Buoy	Jim Thompson	University of Washington Applied Physics Laboratory, United States of America (thomson2012wave)
Seasonal Ice Mass Balance Buoy (SIMB)	Donald K. Perovich	Dartmouth College (PLANCK2019102792)
Polar ISVP	MetOcean	MetOcean
Trident Buoy	Trident Sensor	Trident Sensor

2.2 System level overview

2.2.1 Remote communication

On the Antarctic continent, remote communication is critical for ongoing scientific activities allowing for data to be transmitted from instruments to research stations and camps ([Sanghyun2016satellite](#)). These activities are further supported by high speed, high bandwidth communication networks such as fibre links ([jabbar2001multi](#)). However, these networks have been implemented on a small scale to support permanent field camps ([Sanghyun2016satellite](#)) on the continent. [Sanghyun2016satellite](#) show that communication from polar stations and field sensors to the rest of the world occurs using satellite constellation networks. These constellations are classified as geostationary earth orbit (GEO) ([jabbar2001multi](#)) such as Inmarsat ([inmarsat2021website](#)) and Intelsat ([intelsat2021website](#)) or low earth orbit (LEO) such as ORBCOMM ([orbcomm2021website](#)), Iridium ([iridium2019website](#)), Globalstar ([globalstar2021website](#)). GEO satellites consist of 2 to 8 satellites orbiting the equator ([jabbar2001multi](#)). As a result, the network coverage is strong for mid-latitudes and weak for low latitudes. However, these satellites cover large areas providing longer connectivity (of up to 6.5 hours) ([Sanghyun2016satellite](#)). LEO satellites cover less surface area and have a smaller connectivity window (10 to 30 minutes). However, these constellations consist of more satellites. Additionally, the Iridium satellite network is the only LEO network that reaches the polar region ([jabbar2001multi](#)) and allows for longer network connectivity. The constellation consists of 66 satellites ([Sanghyun2016satellite](#)) and is well opti-

mised for marine applications making it suitable for Southern Ocean sea ice activities. Iridium is a satellite network with global coverage and a variety of modems for various IoT uses. The company offers four main data services. Each service places constraints on the data transmission rates, bandwidth and modem selection. Each modem runs a data service that dictates the transmission rates, bandwidth and protocols. Table D.1 shows the available network services. Furthermore, a full description of these modems is shown in Table 2.2.



Figure 2.4: Examples of popular Iridium modems selected for remote communications. (A) The 9522B modem (image source: (9522B)), (B) the 9602 modem (image source: (9602)) and (C) the 9603 modem (image source: (9603))

Table D.2 shows the satellite network used by each device. All devices use the Iridium satellite constellation as the primary remote communication method. Other short range wireless systems such as Zigbee ([guimaraes2018surface](#)) are alluded to, however these systems are only used when the device is close by. Notably, the SIMB buoy details consideration for remote communication using the ARGOS satellite network. However, the unreliability of the network resulted in irregular timestamped data ([PLANCK2019102792](#)). The network service, modem, price¹ and transmission strategy of each device is shown in Table D.2.

Table 2.2: The following Iridium modems are compared in their key specifications. Devices in the table were suitable for IoT applications based on prevalence in literature and recommendations from the manufacturer. Key parameters include weight, power consumption and transmission latency. Information is taken from ([iridium_product](#)) with prices as of February 2021.

Device Name:	9602	9603	9522B ²	9523	Edge
Weight [g]	30	11.4	420	32	330
Input voltage [VDC]	5	5	4 to 32	3.2 to 6	9 to 32
Idle current [mA]	35	34	250	70	300
Transmit current [mA]	140	145	2500	500	300
Receive current [mA]	40	39	2500	110	300
Packet latency [s]	20	20	N/A	45	20
Price	R2,526 ³	2,526 ⁴	R23,317 ⁵	R13,123 ⁶	R5,694 ⁷

¹GBP 1 = ZAR 20.41; USD 1 = ZAR 15.1

Unanimously, all devices use the Iridium satellite network for remote communication with the Iridium 9602/3 Short Burst Data (SBD) modem being used the most. This choice is justified for its small form factor, low power and easy interfacing as shown in Table 2.2. However, it suffers greatly from limited bandwidth having a maximum transmission size of 340 bytes. Systems that use these modems for transmission of wave data rely on complex data processing algorithms and therefore do not transmit the raw time series. The only notable exception to this is the wave buoy developed by **doble2017robust**, which continuously transmitted data from an attitude and heading reference system (AHRS) as well as IMU time series data once every minute. For this purpose, they used the 9522B modem which allowed for continuous transmission using the router-based unrestricted digital internet working connectivity solutions (RUDICS) data service. This modem, along with the SBD modem used for the SWIFT Buoy also has a much larger SBD data buffer (1.92 KB). However this device draws the most current during idle, transmit and receive states. Additionally this modem is expensive costing R23,317 compared to the 9523 (R13,123) and the 9602/3 (R2,526).

2.2.2 Power supply

A robust power supply is critical to support the functionality of a remotely deployed device. A successful power supply can extend the deployment range, duration, processing capabilities and functionality of sensors (**kennicutt2016delivering**). Device operation in the Southern Ocean MIZ presents a challenge where the constant freezing/refreezing of the ocean surface layer prevents long-term infrastructure from being implemented. Hence, a remote device for sea ice monitoring requires a portable power source. Figure 2.5 shows a diagram of a typical power module for a remote sensing device.

²source: <https://www.rock7.com/shop-product-detail?productId=49>

³source: <https://www.rock7.com/shop-product-detail?productId=50>

⁴source: <https://satellitephonestore.com/catalog/sale/details/iridium-9522b-transceiver-496>

⁵source: <https://www.rock7.com/shop-product-detail?productId=56>

⁶source: <https://www.rock7.com/shop-product-detail?productId=56>

⁷source: <https://www.rock7.com/shop-product-detail?productId=56>

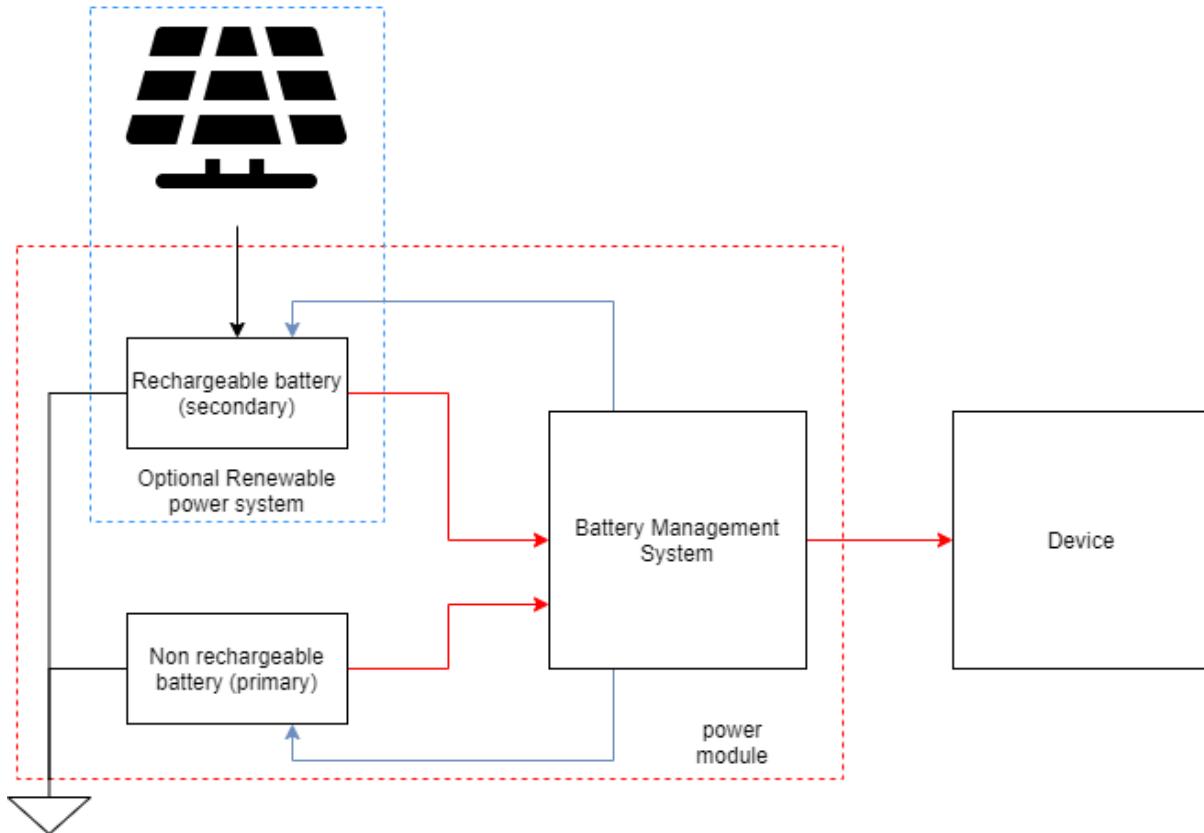


Figure 2.5: A block diagram of a typical power module for a remote sensing device based off the information from **rabault2019open**; **doble2017robust**; **vidal2019xev**. Batteries are used as a primary source of energy which is connected to a battery management system to control and regulate the power supply to the device. Optionally, solar panels are implemented with a rechargeable battery as a secondary power supply. Shown in the figure: flow of power (red), control lines (blue) and ground (black).

Batteries such as lithium-ion are commonly used as sources of power for in situ devices in low temperatures. Batteries typically fall into one of two categories:

1. Primary: Cells that cannot be recharged once they are depleted. These batteries have a high energy density and can store charge for long periods (**besenhard2008handbook**).
2. Secondary: Cells that can be recharged. These cells are more cost effective with longer usage cycles (**besenhard2008handbook**).

Lithium-ion and alkaline batteries are examples of primary cells that have commonly been used as they provide a cost effective solution due to their high energy density and long-term consistent life cycles (**zareer2018review**). Secondary cells such as lead-acid, nickel-metal hydride and lithium polymer are commonly used coupled with a secondary charging circuit such as a solar panel (**manimekalai2013overview**) to provide a more constant source of power. While batteries are cost effective solutions, the climate of the Southern Ocean can affect the battery performance. Freezing air temperatures can reduce the capacity of a standard lithium cell by up to 50% for temperatures below -10° C (**doble2017robust**; **ZHANG2003137**). Additionally, frozen

batteries or ice formation on batteries can stop them from working (**doble2017robust**; **manimekalai2013overview**). A solution to compensate for this reduction in capacity is to use rechargeable batteries coupled with a renewable power sources such as solar (**doble2017robust**; **rabault2019open**), wind and geothermal energy (**manimekalai2013overview**). Solar photovoltaic (PV) cells are the fastest growing renewable energy source with the highest energy density (**jordehi2016parameter**). PVs have numerous advantages such as low maintenance and operational costs, wide temperature operations and long lifecycles (**jordehi2016parameter**) which makes them ideal for long term operation in remote environments. However, the power output of PV cells is significantly affected by weather conditions (**sharma2015solar**). Therefore, areas with poor sunlight coverage will not benefit from solar panels. Therefore, making this energy source not suitable for winter Antarctic expeditions due to the lack of sunlight (**lever2006solar**). Additionally, power generation from solar panels is inconsistent and requires an additional storage bank capable of frequent charging and discharging. Wind energy can provide a viable alternative to solar energy. **vichi2019effects** show that the Southern Ocean hosts strong, consistent winds. However, the design and implementations have not been discussed in any of the literature. Therefore this options is provided as an area for future research. Finally, a critical component of the power system is a battery management system. This allows the power supply to operate under safe conditions while meeting performance requirements (**vidal2019xev**). This module includes power monitoring, power control and energy cycle optimisation. Table 2.3 below shows the power sources used by each device and the strategy used to manage it.

Table 2.3: A comparison of power supply strategies of the different devices showing the the power source, topology of the power supply module as well as the voltage supplied at the output of the module. Information that was unavailable at the time of research has been labeled as "Not Reported".

Device Name	Primary power source	Secondary power source	Battery management system	Output regulation strategy	Output voltage [V]
WIIB	Lithium Iron Phosphate (LiFePO ₄) battery	None	ATMEL ATMega328P for Power control	Boost converter	5
WIIOS	Alkaline battery	None	Integrated into firmware	8-cell series configuration, no regulator	12
NDWB	Alkaline battery	Solar panel and lead-acid battery	Not Reported	Not Reported	12
SKIB	Lithium thionyl chloride (LiSOCl ₂) battery	None	Not Reported	Not Reported	3.6
SWIFT	Alkaline or lithium battery	None	Not Reported	Not Reported	14
SIMB	Alkaline battery	None	Not Reported	Texas Instruments LMZ12003 Step down converter (5 V, 3.3 V) Microchip MIC29201-12W Low dropout regulator (12 V)	3.3 5 12
Polar ISVP	LiSOCl ₂ battery	None	Not Reported	Not Reported	12
Trident	Lithium cell battery	None	Integrated into firmware	Low dropout regulator	5

Table 2.3 shows the power supply strategies of each device. All systems use primary batteries as a source of power with the most common choice being alkaline or lithium-based batteries. However, **doble2017robust** are the only exception where a secondary power source was added consisting of a solar panel and lead-acid batteries. As a consequence of the added power system the NDWB had an increased weight which affected its portability and ease of handling. This is discussed further in Sub-section 2.2.4. Furthermore, systems deployed in the Arctic Marginal Ice Zone have been designed with a recharging system such as a solar panel in the case of WII Buoy and NDWB. However, most long-range deployment buoys have opted for non-rechargeable systems composed of lithium thionyl chloride (LiSOCL_2) or alkaline batteries. In the case of the high-power buoys (SIMB, WIIOS, NDWB, Polar ISVP) an array of 3.3 V to 3.7 V cells are connected to provide a nominal voltage in series with a regulator to provide a stable output. The strategy for each system is to pack as many batteries in as possible to satisfy the long-term energy requirements (**doble2017robust**; **rabault2019open**). Finally, few devices have reported their battery management strategies. **rabault2019open** used an ATMEL ATmega328P microcontroller as a power controller for their device which monitored the status of the battery. **trident** and **kohout2015device**, however, integrated power control into their main firmware allowing for them to control and monitor their power source off a single processor.

2.2.3 Polar performance

This section outlines the deployment of the systems in the Arctic/Antarctic Marginal Ice Zones and compares the survivability and performance of each system. The focus of this section is predominantly on devices deployed in the Marginal Ice Zone. Table 2.4 shows the significant deployment locations in the Arctic and Antarctic sea ice zones as well as the deployment objectives of each device.

Table 2.4: Comparison between the functionality and purpose of the buoy showing the critical measurements as well as the significant deployment locations either in the polar ice zones or in a location critical to the validation of the device.

Device name	Deployment objectives	Antarctic deployment locations	Arctic deployment locations
WIIB	Wave energy attenuation	Ross Sea landfast ice	Templefjord (Svalbard) landfast ice
	Significant wave height	(rabault2020development)	(rabault2019open)
	Data quality		Northeast Barents Sea (rabault2019open)
WIIOS	Ice drift	Ross Sea Marginal Ice Zone	Not Reported
	Waves in ice	(kohout_smith_roach_williams_montiel_williams_2020)	
	Ambient temperature	Ross Sea packed ice zone	
	Atmospheric pressure	(kohout2015device) Weddel Sea Marginal Ice Zone (vichi2019effects; albarello2020drift)	
NDWB	Ice drift	Not Reported	Beaufort Sea
	Wave induced breaking		(doble2017robust)
	Ambient temperature		
	Atmospheric pressure		
SKIB	Ice drift	Not Reported	North Atlantic Ocean (France)
	Surface waves		(guimaraes2018surface)
SWIFT	Ocean surface	Ross Sea	Chukchi Sea
	Ocean waves	(ackley_2020_seoice)	(hosekova2020Attenuation)
	Turbulence profiles	Weddel Sea	Beaufort Sea
	Ocean current profiles	(DeSanti2018OceanWave)	(lund2018Arctic)
	Conductivity		
SIMB	Wind speed and direction		
	Surface and bottom ice position	Weddel Sea (hoppmann2015fmot)	Beaufort Sea Marginal Ice Zone (PLANCK2019102792)
	Snow depth		
	Ambient temperature		
Polar ISVP	Atmospheric Pressure		
	Vertical ice temperature profile		
	GPS location		
	Sea ice drift	Weddel Sea Marginal Ice Zone	Western Arctic Ocean
Trident	Ambient temperature	(grosfeld2016online; ipab)	(lei2020comparisons)
	Atmospheric pressure		
Trident	Sea ice drift	Weddel Sea	Not Reported
	Ambient temperature	(vichi2019effects; alberello2019drift)	
	Battery voltage		

kohout2015device deployed five Wave in Ice Observational Systems (WIIOS) in the East Antarctic Marginal Ice Zone during the Sea Ice Physics and Ecosystem Experiment (SIPEX) mission⁸ with the goal of capturing wave-in-ice events with measurement goals shown in Table 2.4. Three devices were deployed using helicopters on ice floes while two devices were deployed from the ships crane. **kohout2015device** note that deployment via crane was successful in spite of 7 m swell and 25 ms⁻¹ winds. The device was fitted inside a Pelican case with a sealed membrane surrounded by a tyre for protection and flotation in case of melting (**kohout2015device**). Consequently, this places the buoy

⁸First deployment occurred September 2012 (**kohout2015device**)

directly on the surface of the floe rendering it susceptible to snow build up and flooding as mentioned in the previous sections. After deployment, the crew received 600 samples of data over 39 days in total. However, the first device failed 20 hours after deployment coinciding with the first large wave event captured the buoys (**kohout2015device**). The second large wave event resulted in the failure of two more systems just 9 days after deployment (**kohout2015device**). The fourth buoy lasted for 17.5 days. The final buoy survived the longest at 39 days. As a result only one device lasted for the expected time with the majority of data captured during calm events.

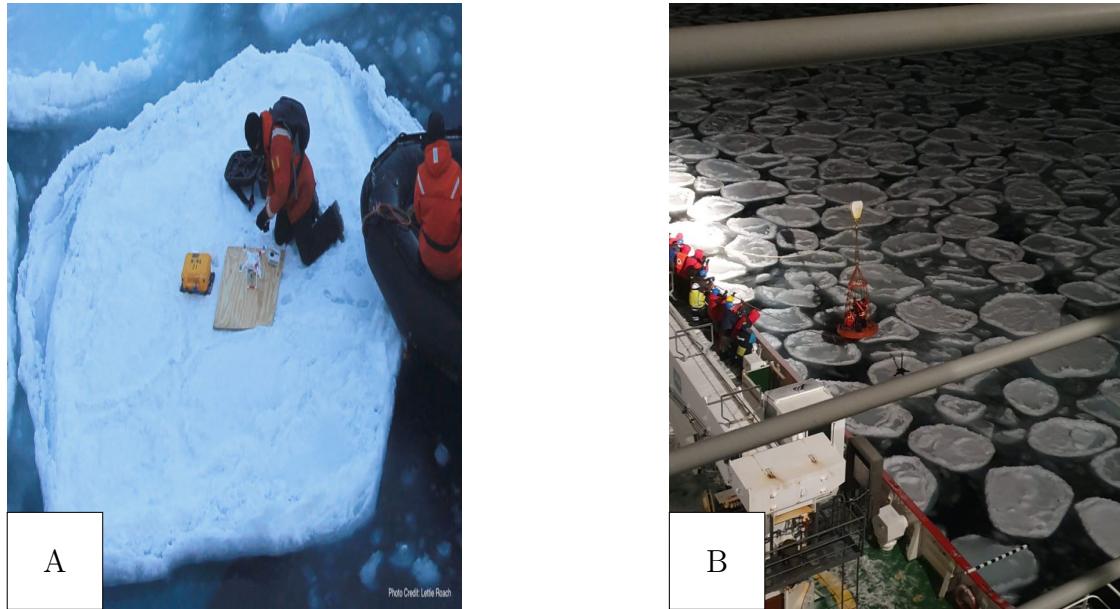


Figure 2.6: Examples of different deployment protocols for ice tethered devices. (A) In regions of consolidated ice in favourable conditions, manned crews will step foot on the ice to deploy the device (image source: (**kohout2020observation**)). (B) In unfavourable conditions, devices may be deployed from a personnel basket attached to a crane or a manned crew will lower the buoy onto a suitable ice floe from the safety of the personnel basket (image source: N. Taylor).

Two WIIOS buoys were deployed by NYU Abu Dhabi (**vichi2019effects; alberello2019drift**) during the winter of 2017⁹ similar to the ones by (**kohout2015device**). Two devices were deployed on two separate ice floes 3 m in diameter and 100 km from the ice edge (**vichi2019effects; albarello2020drift**). One system survived for 8 days and 18 hours while sampling every 15 minutes before transmission ended (**vichi2019effects; albarello2020drift**). The second buoy, however, survived for 6 days sampling every 15 minutes until it switched to power saving mode surviving for a total time frame of 3 weeks. **vichi2019effects; alberello2019drift** deployed a second pair of WIIOS buoys. However, the first buoy stopped responding after three days while the second buoy survived for 16 days (**vichi2019effects**). While the buoys' survival is largely attributed to power optimisation, the lifespan could be influenced by the selection of the ice floe. Ice floe size and proximity to the ice edge affect the exposure of the floe to open-ocean processes and storms (**vichi2019effects**). This could result in failure due to ice mechanics

⁹First deployment occurred in July 2017 (**vichi2019effects; alberello2019drift**)

which is discussed in Section 2.2.3.

rabault2019open deployed the Waves In Ice Buoy (WIIB) on landfast ice in the Ross Sea (**rabault2020development**) to test the device's performance in the Southern Ocean. In a similar fashion to the WIIOS buoy, the device was placed in a Pelican case and attached to a flotation device. However, expected survival time for this device was significantly lower compared to the WIIOS devices: a maximum of 8 days (**rabault2019open**) of continuous operation. The buoys by **kohout2015device** were designed to be expendable whereas the buoys by **rabault2019open** were designed to be retrievable. Additionally, the WIIB devices were deployed in the summer¹⁰. Two devices were deployed in close proximity to each other. However an ice break event resulted in the separation of the devices. The devices survived for 2.5 weeks (**rabault2019open**) which **rabault2019open** attribute the failure to the devices having been crushed by ice and wave activity. Despite this, the devices were able to record significant wave events and maintain a fully charged battery throughout the deployment which **rabault2019open** attributes to the solar panel.

doble2017robust alluded to a series of environmental considerations when designing the NDWB systems. One such consideration is the frosting over/rimming of the device due to freezing ocean spray. Additionally long periods of heavy cloud cover and no sunlight can affect the performance of the solar powered battery systems (**doble2017robust**; **lever2006solar**). Since the buoys were deployed by a manned crew, the design also had to account for ease of handling by the crew and not be too heavy (**doble2017robust**). The mechanical enclosure consisted of a float and a keel with the electronics contained above the surface in a dome. Twenty buoys were deployed in the Arctic Marginal Ice Zone with each device anchored by drilling a hole in the ice and placing the keel inside. Nineteen buoys survived the deployment with one system failing to boot. The buoys survived for extremely long periods with twelve systems surviving for 200 days off a single alkaline battery pack (**doble2017robust**). A significantly longer period than both the WIIOS and WIIB systems. Seven systems ran for 70 days on alkaline batteries before switching over to the solar powered lead-acid batteries. During this period, devices transmitted continuously over the Iridium network and were able to interpolate sea ice phases (see Section 1.1) from the tilt of the buoy (**doble2017robust**). However, unlike the design phase for the NDWB, was unconstrained by costs unlike the design phases of devices by **rabault2019open**; **PLANCK2019102792**; **kohout2015device**.

Reasons for failure

Eventually, the systems by **doble2017robust** lost transmission 300 days after their deployment. This can be attributed to the depletion of the alkaline battery packs. The solar powered lead acid battery voltage eventually dropped below the alkaline battery voltage due to the lack of consistent solar coverage (**doble2017robust**). Additionally, sub zero temperatures have a tendency to reduce battery capacities by up to 50% (**doble2017robust**). However, **doble2017robust** found this estimate to be over conservative. Systems by **kohout2015device** and **doble2017robust** encountered similar failures with devices eventually depleting the onboard batteries. Additionally,

¹⁰First deployment date: December 2019 (**rabault2019open**)

vichi2019effects and **alberello2019drift** attributed failure of the first WIIOS system to the battery being depleted.

Additional sources of failure experienced by (**doble2017robust**) include ice convergence. The systems were subject to ice-mechanics and as a result, ended up crushed by the floes due to rafting or buried under ice **doble2017robust**. These failures were identified when more than one system suddenly went offline. Devices also experienced freeze-over or were buried under snow which resulted in the devices going offline for temporary periods (**doble2017robust**). Additional evidence of rafting and ridging was captured by webcams on the buoy shortly before transmission ended (**doble2017robust**). Buoys that survived the spring melt refroze during the gradual refreezing of the ice. During the second cycle, none of the buoys rebooted when the ice melted in the spring (**doble2017robust**). Finally, the buoys developed by **kohout2015device** and **rabault2017measurements** sit in close proximity to the ice floes. As discussed previously, during the winter cycles, snow accumulates on the surface that can reach up to 1 m in height. This snow formation can result in flooding where the floe becomes submerged. Prolonged burying under snow may have resulted in the device freezing over thereby losing contact while prolonged contact with the seawater may have resulted in the buoys failing on several occasions ((**kohout2015device**; **vichi2019effects**; **albarello2020drift**; **rabault2019open**)).



Figure 2.7: During the 2019 SCALE winter expedition, a SWIFT device was retrieved early due to lost communication with the host. This failure was attributed to a build up of ice along the rim from ocean spray. Photo taken by Author.

Finally, **vichi2019effects** discuss the findings surrounding the failure of the first WIIOS system. **vichi2019effects** observed a major cyclonic event. The cyclone formed on 2 July 2017 and achieved lysis on 5 July 2017 which coincided with the buoy deployment. Following the event, four more cyclonic events were recorded with three explosive cyclones

characterising a change of pressure over 24 hours (**vichi2019effects**). During this time, **vichi2019effects** observed winds speeds of up to 33 ms^{-1} while noting that the air temperatures had increased to values "close to melting" (**vichi2019effects**). Additional observations found an increase in significant wave height in the activity. These conditions indicate deformation (**vichi2019effects**) which may have subjected the buoys to forces experienced by **doble2017robust** during their Arctic deployment which were verified against the temperature and pressure readings of the second WIIOS during the cyclonic event. The buoys were deployed close to the ice edge exposed to greater open ocean processes and cyclonic activities than other semi-consolidated and consolidated regions (**vichi2019effects**). As a result, air advection, storms and large wave movement delay the consolidation of sea ice considerably (**vichi2019effects**). Hence, the ice floes were more likely to experience rafting, ridging (**icedefinition1992**), extended flooding, and freezing over which may have caused the failures of the WIIOS buoys.

2.2.4 Overall cost

A key factor of buoy development is the cost and weight of the overall device. Since these devices are still deployed manually, such deployment techniques involve the use of on-ship cranes (**alberello2019drift**; **kohout2015device**), or by setting foot on the deployment site (**PLANCK2019102792**; **rabault2019open**). Additionally, **doble2017robust** deployed buoys by transporting them on a twin Otter aircraft and installing them on arrival. This shows that the weight and form factor of the device can impact the ease of deployment as well as where the device can be deployed. Table 2.5 provides insight into the quoted price¹ and development cost of each device to understand how expensive the current state of the art is.

Table 2.5: Comparison of price and weight of each device according to the published literature or commercial listing. Weight provides an indicator of the ease of handling whereas price provides an indicator of affordability. Prices have been converted to South African Rand (R) online (**usdcoversion**)where applicable while weight has been converted to kg. "Not Reported" or "N/R" is given where a value could not be obtained.

Device name	Weight [kg]	Price
WIIB	4.5	R30,200 ¹¹
WIIOS	N/R	N/R
NDWB	150	N/R
SKIB	N/R	R39,784
SWIFT	30	N/R
SIMB	34	R58,909
Polar ISVP	11.34	R52,119
UptempO	105	R863,686
Trident	0.42	R30,525 ¹²

¹Price as of February 2021.

Table 2.5 shows the current cost of in situ sensing devices. The cheapest wave sensing device is the WIIB at R30,200 showing that **rabault2019open** succeeded in developing a feature-rich device for a relatively lower cost than the current state. Also included in the price comparison is a version of the Polar ISVP for deployment on ice floes called the UpTempo. This device is the most expensive at R875,800 with only a GPS, pressure sensor and temperature sensor. This shows that to provide a cost effective solution, a new device will have to cost less than R30,200. However, this can be possible through the use of open-source and off-the-shelf solutions (**bonvoisin2017source**). As discussed in Section 1.1 the goal of remote sensing technology is to increase sensing in the region. To achieve this, more devices spread over a larger region are required. Finally, the cost efficiency of stand alone devices impacts the temporal and spatial scalability that can be achieved as it may not be possible to afford the required number and type of device to deploy during an expedition. This results in insufficient data to cover the Antarctic Marginal Ice Zone over a seasonal period.

2.3 Subsystem overview

This section focuses on the subsystem analysis and component selection for each system. As stated previously, each buoy was created with a unique objective shown in Table 2.4. These objectives have influenced the sensor selection, topology and layout of the overall subsystems. In addition, the device designs choices have influenced objectives for developing new in situ technologies by **kennicutt2016delivering** as the device developers have factored in power consumption (**kohout2015device**), ease of use and deployment (**rabault2019open**), long-term operation (**doble2017robust**), cost (**PLANCK2019102792; rabault2019open**) and availability of infrastructure (**doble2017robust**) over and above functionality. For example WIIB was developed using open-source hardware (**rabault2019open**) to increase access to readily available technology while WIIOS was developed using off-the-shelf components (**kohout2015device**) to create a cost effective device. From Table 2.4 we saw that the principle measurements of each device share the following measurement objectives:

1. Ice drift
2. Wave data/Waves-in-ice data
3. Ambient temperature
4. Atmospheric pressure

The following sections discuss how these objectives have influenced sensor selection for each of these measurements as well as the hardware, and software strategies implemented to achieve these objectives.

¹¹USD 1 = ZAR 15.1

¹²GBP 1 = ZAR 20.41

2.3.1 Processing capabilities

The processing capabilities of each device facilitate the functionality of each buoy. A processor allows for the implementation of in situ data processing and wave analysis algorithms (**kohout2015device**; **rabault2019open**). Additionally, due to the bandwidth constraints mentioned in Section 2.2.1, some devices require data compression algorithms to allow the data to fit in the transmission buffers. The functionality of these devices has been made possible through the use of microprocessors.

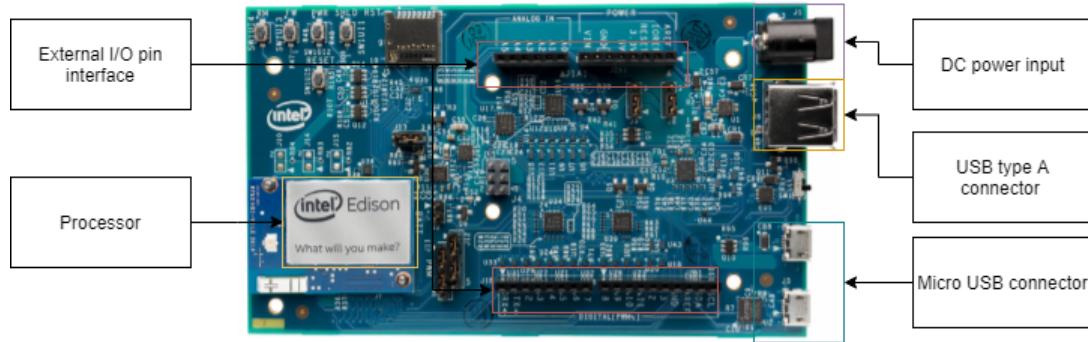


Figure 2.8: Diagram of a typical microprocessor (yellow) on a development board. This device is an Intel Edison which was used as the main processor by **kohout2015device** for the WIIOS. The development board allows for fast prototyping and integration into projects. The processor interfaces with external peripherals through physical input/output pins (red) and contains standard serial communication ports such as USB (orange), micro USB (teal) and a connector for external voltage (purple). Image source: ([edison](#)).

These are programmable integrated circuits that contain processing elements (**subham2018micro**) and form the basis of microcontrollers and microcomputers (**crisp2003introduction**). These components are small and are increasingly becoming integrated into affordable, widely available components such as Raspberry Pis and Arduinos (**rabault2019open**). Table 2.6 shows a comparison of processors and topologies implemented in each design.

Table 2.6: Comparison of the processing strategy implemented by each device. Multiple processors have been used in a selection of devices, hence included in the comparison is the number of processors used, the type of processor and the function of each processor.

Device name	Number of processors	Processor name	Function
WIIB	3	ATMEL ATmega 328P	Low power unit
		Arduino Mega 2560	Data logger
		Raspberry Pi Zero	Wave processing
WIIOS	2	Intel Edison dual core	Wave processing
		ATMEL ATmega 328P	Low power unit
NDWB	1	ACME Systems Fox G20	Power control
SKIB	2	Silicon Labs EFM32-M3	Wave spectral processing
		Unspecified processor	Power control
SWIFT	1	Sutron Xpert	Data processing
SIMB	1	ATMEL ATSAMD21G18	Data processing control
Polar ISVP	1	Global Platform Transceiver Controller (GPTII) ¹³	Data processing and control
		Unnamed microprocessor	Data processing
Trident	2	Unnamed low power unit	Power control

From Table 2.6, we can see that each device has been developed either with a single, powerful processor or with multiple, low powered processors. NDWB for instance builds its system around the dominant sensor i.e. the AHRS IMU with a single processor controlling all the peripherals as well as allowing for data processing. Drift loggers such as Trident, and Polar ISVP feature sparser sets of electronics with smaller, lower-powered processors for power control and peripheral control. In contrast, WIIOS and WIIB compartmentalise subsystems with a cluster of processors handling different aspects from the buoy. This shows a focus on computation rather than sensing as multiple controllers are used to free the main processor for implementing advanced digital signal processing. SWIFT buoy appears as the outlier as the system is built around a dedicated data logger i.e. the Sutron Xpert with an integrated processor and satellite communication link abstracting data processing strategies on the buoy side. The SIMB buoy has the most advanced and largest number of sensors of all the buoys. A commonality among the buoys is the use of off-the-shelf components and processors. Another predominant feature, the GPS is an Adafruit MTK339 as well as ATMEL SAMD chips, Raspberry Pis and Arduino boards whereas, for Trident and MetOcean, more expensive solutions are used. This shows that developers have opted for ready-made components that are auxiliary to the main measurements. This should explain why some components on a system are more advanced than others. Finally, Trident and MetOcean are commercial systems and have therefore developed their own integrated solutions for sensors, processors and circuitry

¹³Developed by **uptempo**™

while researchers have opted for off-the-shelf components and development boards.

2.3.2 Measurement of wave data using inertial measurement systems

Typical wave state estimation is derived from calculating wave parameters such as significant wave height and dominant wave frequency ([williams2013wave](#)). Additionally, wave data can be analysed in terms of its power spectral density. The two main methods for wave data analysis presented in this Section are the Kuik Method and the Welch-Earle Method. Further explanations can be found in Appendix A.3.1. Both approaches rely on a discrete time series of inertial data from a device with 3 axes of acceleration of 3 axes of rotation ([kuik1988method](#); [earle1996nondirectional](#)).

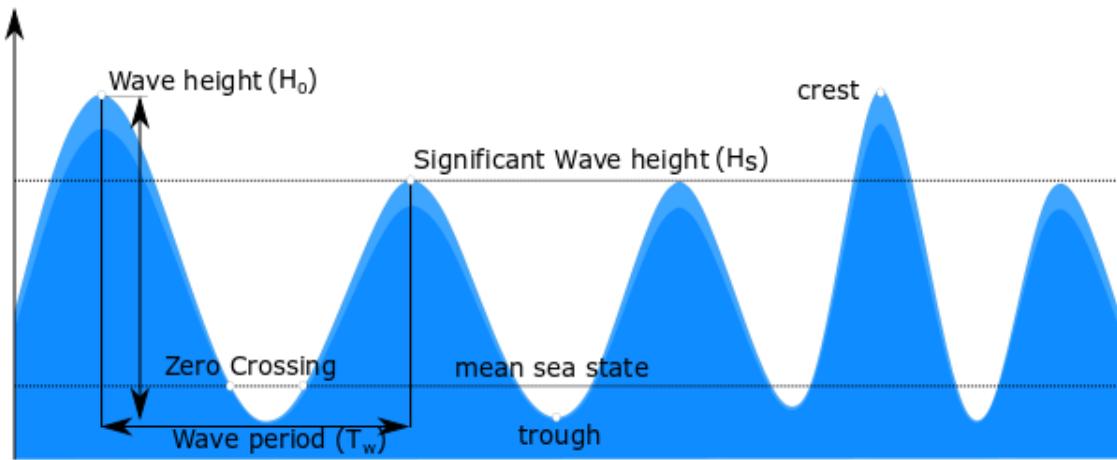


Figure 2.9: Diagram showing the characterization of sea state through statistical methods described by [kuik1988method](#) and [earle1996nondirectional](#); [welch1967use](#). A wave consists of a maximum vertical point (crest) and a minimal vertical point (trough) with the distance between the two points referred to as the wave height (H_0). The period between consecutive crests of a similar frequency is referred to as the wave length and the point where a wave crosses the mean sea state level is the zero crossing. From a statistical analysis, the significant wave height is the average of the top third of observed wave heights in a given time period. ([kuik1988method](#)). This diagram is based off the figure by ([seastatediag](#)).

Devices that measure wave parameters such as significant wave height, wave spectra or ocean states, incorporate a sensor that measures the vertical acceleration and roll, pitch, and yaw in discretised space ([earle1996nondirectional](#)). These parameters can be measured using an accelerometer for axial acceleration and a gyroscope for rotational velocity ([fong2008methods](#)). These devices are typically integrated circuits manufac-

tured using micro electro-mechanical systems (MEMS) and are often combined to form an inertial measurement unit (IMU) allowing for 6 degrees of freedom from a single device (**fong2008methods**). IMUs can also be expanded to include a magnetometer which measures magnetic bearing in 3 axes (**ahmad2013reviews**). IMU selection is dependent on the following factors¹⁴:

1. Package size
2. Accuracy
3. Response rate
4. Degrees of freedom

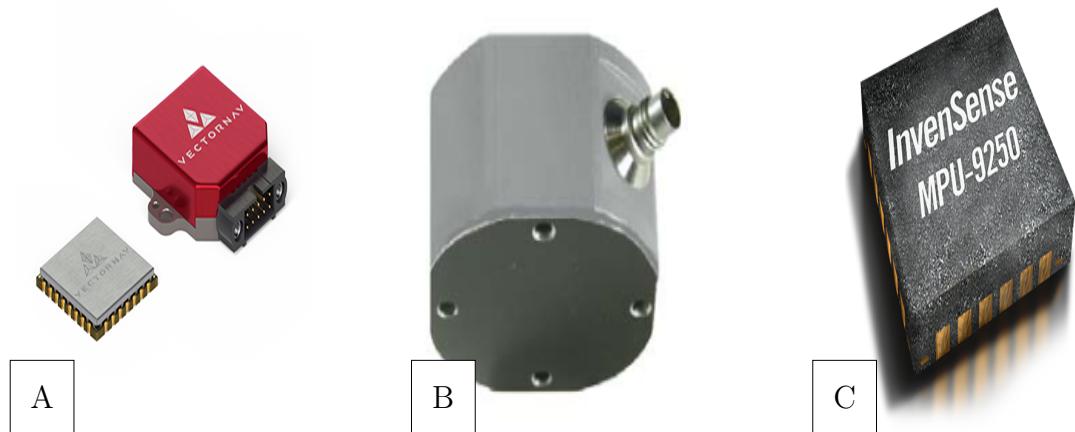


Figure 2.10: Diagram showing examples of the different types of inertial measurement systems (IMUs) available for commercial use such as (A) an Attitude Heading Reference System (AHRS) VN100 (**vn-100**) used by (**rabault2019open**), (B) an analog capacitive accelerometer 8330B3 Servo-KBeam accelerometer used by (**kohout2015device**) (**kistlerIMU**) and (C) a 9 degree of freedom MPU9520 (**mpu9520**) used by (**kohout2015device**).

ahmad2013reviews show that package size can limit the applications of the IMU. Additionally, IMUs require careful calibration and filtering to reduce the effects of bias offsets as well as low and high frequency drift (**fong2008methods**). More advanced filtering methods have been developed to improve the accuracy of IMUs such as Kalman filters (**simon2001kalman**) for positional estimates or Real-Time Kinetic Fusion (RTK) for velocity (**meng2014optimal**). While important to the integrity of IMU data, these methods are usually implemented in software which is discussed in the upcoming sections. Finally, the degrees of freedom (DOF) influences the application of the IMU. The magnetometer is used to improve the accuracy of the gyroscope measurement and account for low frequency noise (drift) (**ahmad2013reviews**). However, the magnetometer is sensitive to magnetic distortion which can affect the measurements. **kohout2015device** encountered magnetic distortion during the SIPEX II deployments in September 2012 which rendered the magnetometer readings useless. In this section, the IMU is integral

¹⁴**ahmad2013reviews**.

to deriving a time-series representation of inertia to calculate wave data as described above. However, another application relevant to the literature is using IMUs to improve navigation (**ahmad2013reviews**). An IMU can be coupled with a Global Positioning System (GPS) device to determine position in areas with poor signal which can assist greatly with determine ice drift (discussed in Section 2.3.3). Based on these considerations, Table 2.7 shows the IMU component selection for each device.

;

Table 2.7: Comparison of the inertial measurement systems selected for each device showing the sensors included as well as the degrees of freedom.

Device name	Inertial Measurement Unit	Sensors	Degrees of freedom
WIIB	VectorNav VN-100	Accelerometer	
		Gyroscope	9
		Magnetometer	
WIIOS	Kistler 8330B3 Servo-KBeam TDK InvenSense MPU9250	Vertical acceleration	1
		Accelerometer	
		Gyroscope	9
NDWB	SBG IG-500N	Magnetometer	
		Accelerometer	
		Gyroscope	9
SKIB	LIS3DH	Magnetometer	
		Accelerometer	3
SIMB	Bosch Sensortec BNO055	Accelerometer	
		Gyroscope	9
		Magnetometer	
Polar ISVP	None	-	-
Trident	None	-	-

As mentioned previously, systems such as WIIOS and WIIB have built their purpose around wave measurements and therefore have specified high powered, high accuracy IMUs for wave measurements. However, the WIIOS buoy separates itself from WIIB by having a cheaper 9 DOF IMU to complement the measurements (**kohout2015device**). SWIFT buoy and the NDWB buoy use an integrated system known as an Inertial Navigation System (INS) with the former containing an SBG Elipse AHRS (**thomson2012wave**) and the latter containing a SBG IG-500-A1G2 (**doble2017robust**). This device contains a GPS and an onboard processor for RTK fusion and Kalman filtering whereas other devices use an external processor for filtering. The SIMB Buoy is the only buoy on the list that has an IMU for non-wave related measurements. It uses a cheaper Bosch BNO055 which is used solely for measuring the orientation of the device.

Software processing

An IMU is a powerful device however extensive software processing is required to extract key parameters (**ahmad2013reviews**). Examples of software processing algorithms are shown in **kuik1988method** and **earle1996nondirectional**¹⁵ for wave processing algorithms. Additionally, as mentioned previously, advanced filtering techniques are required to reduce the effects of low and high frequency noise. In this subsection the software processing strategy for each device given for extracting the desired parameters shown in Table 2.4.

Table 2.8: Comparison of sampling strategies implemented in each device. This includes the desired measurands, sample rate and sample period of each IMU session.

Device name	Degree of freedom used	Sample rate [Hz]	Sample period [minutes]
WIIB	Vertical acceleration		
	Pitch	10	25
WIIOS	Roll		
	3-axis acceleration		
	3-axis gyroscope	2	11
	3-axis magnetometer		
NDWB	Vertical analog acceleration		
	3-axis acceleration		
	3-axis magnetometer		
SKIB	Heave	1	Continuous
	Roll		
	Pitch		
SWIFT	3-axis acceleration	25	10
	3-axis acceleration		
SIMB	Tilt	5	9
	Horizontal rotation		
SIMB	Tilt	-	-
	Orientation		

rabault2019open extract wave parameters by passing the raw time series data through an Extended Kalman filter running at 800 Hz then through a low pass filter. Wave spectral data were calculated using the method by **earle1996nondirectional** where co-spectra were calculated using the method by **kuik1988method**. Significant wave height was calculated by double integration using a Fast Fourier Transform (FFT). Sampling is performed at 10 Hz to satisfy the Nyquist sampling criteria for open ocean waves as described in (**rabault2019open**; **earle1996nondirectional**).

kohout2015device, however, opted for a reduced sampling rate of 2 Hz over a shorter

¹⁵See Appendix A.3.2.

sample time. This was achieved by oversampling IMU data at 640 Hz then down-sampling the data through a multistage decimator (**kohout2015device**). Before each decimation, data were filtered using a Butterworth filter at 8 Hz with a cut-off frequency of 2 Hz, then once again at 40 Hz. Significant wave height is calculated by double integration using the Welch-Earle method (**earle1996nondirectional**; **welch1967use**) multiplying the transformed data set by a response weighted function to remove low frequency drift. Finally, the Longuet-Higgins parameters are calculated thereby characterising wave-in-ice activity.

doble2017robust did not apply any data processing algorithm to the raw time series as it is transmitted directly over Iridium. However, the raw time series is filtered using an Extended Kalman Filter running at 10 Hz.

The SKIB collects data from a sample window which is processed using a classical RC filter to attenuate frequencies below 0.04 Hz. Thereafter, **earle1996nondirectional** spectra and co-spectra calculation are then applied.

The SWIFT buoy is the only device that used multiple sensor types for sea state calculation. Additionally, data was collected more frequently in short intervals (9 minute sample periods every 12 minutes) which include Doppler profiles, camera images and IMU data. The inertial navigation system (INS) outputted a real-time kinematic (RTK) fusion data set where IMU data was passed through a Coning & Sculling Extended Kalman Filter running at 1 kHz (**thomson2012wave**) while the Doppler profiler was sampled at 8 Hz. Turbulence profile was calculated through time-averaging and data fitting of the Doppler profile. Then, the ocean current state was calculated using the Stokes drift equation over a time-averaged velocity series. Finally, wave data were calculated through image processing from a still of the sea state taken from an onboard camera.

The SIMB uses the IMU in a non-critical manner. The device transmits tilt and orientation metadata describing the current status of the SIMB (**PLANCK2019102792**). While no method has been described by **PLANCK2019102792** to calculate tilt, it can be achieved by measuring the orthogonal acceleration values over an angle of rotation and taking the inverse tangent of the resulting value for each axis (**tuck2007tilt**).

Therefore IMUs are integral for performing sea state calculations and measuring wave spectra, a sufficient processor is required to filter and process the data. Therefore, significant consideration should be given to coupling a powerful processor with a sufficiently powerful inertial measurement unit.

2.3.3 Measurement of ice drift using GPS

This section discusses the technology and techniques used for determining ice drift. The predominant approach towards understanding modeling ice drift is by using the techniques presented by **hibler1979dynamic**¹⁶ where kinematic data are used to study ice drift dynamics and calibrate the ice drift model. Additionally, **lepparanta2001sea** present two methods for collecting ice drift data. The first method utilises measurement beacons attached to the ice floes to track trajectories. The second method uses imaging

¹⁶See Numerical Modeling in Appendix A.1

devices such as radar, and satellites to determine ice displacement ([lepparanta2001sea](#)).

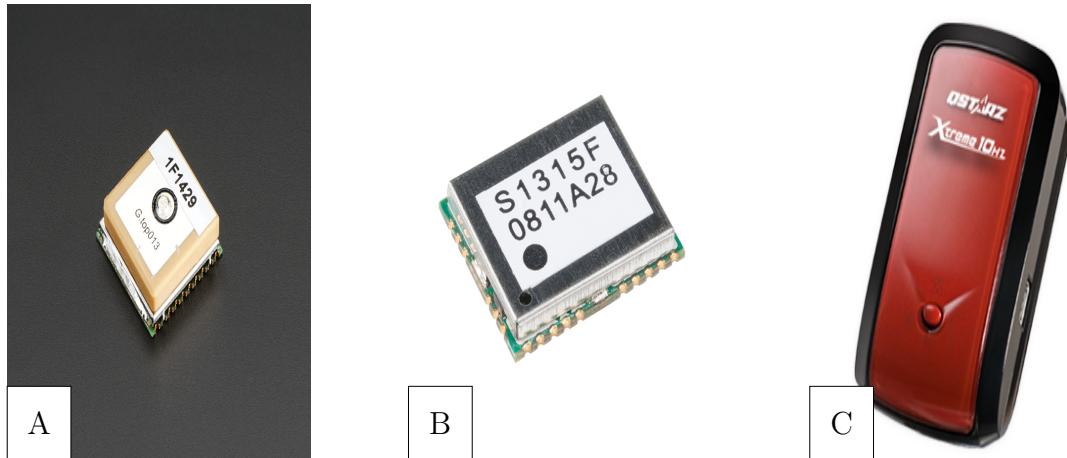


Figure 2.11: Diagram showing examples of the different types of global positioning devices available for commercial use such as (A) MTK3339 (image source: ([mtk3339](#))) , (B) SkyTraq S1351R (image source: ([s1315F](#))) (C) Qstarz BT-Q1000eX (image source: ([BTQ1000EX](#)))

The discussion of modern technology for ice observations from Chapter 1.1 showed that observations from satellites such as OSI-SAF and METSAT are unsuitable for ice drift measurements ([lepparanta2001sea](#); [galin2011validation](#)) due to their low spatial resolutions. Hence, a need arises for in situ drift measurement devices. Global positioning systems (GPS) have been the standard for ice drift measurements ([lepparanta2001sea](#)) as they are capable of measuring data at relatively high temporal resolutions ranging from 15 minutes ([alberello2019drift](#)) to 25 minutes ([rabault2019open](#)) as opposed to the limit of 3 hours ([alberello2019drift](#)). This Section provides an overview of GPS technology and how the devices have been used for measuring positional or drift data.

Overview of GPS

The principles that govern GPS have remained unchanged since its inception in 1973 ([spilker1996global](#)). The system consists of a satellite constellation that constantly broadcast their estimated position. A GPS device determines its position by matching a user-generated signal to that of four received satellites and comparing the phase difference to an onboard crystal oscillator ([spilker1996global](#)). This technique is called ranging and four satellites spread in a uniform geometry will allow for a device to calculate latitude, longitude, altitude and time to a relative degree of accuracy. The number of unknown signals correlates to the number of satellites required. Generally, a GPS device will have a lesser degree of accuracy than the satellites. Hence, an incoming signal can be used to correct the device's clock ¹⁷. To accurately predict the satellite's trajectory, satellite ranging is performed by a network of global monitoring stations which calculate the future position and send it back to the satellite. GPS signals are transmitted on two frequencies: 1575.42 MHz and 1227.46 MHz ([spilker1996global](#)). These are

¹⁷provided altitude or time are already known ([spilker1996global](#))

synchronously generated signals and allow a device to correct for ionospheric distortion. These bands carry modulated signals which are as follows: (**spilker1996global**)

1. Clear Acquisition (CA) Code: This is a short code transmitted at 1.023 MHz and is used to request the Standard Positioning Service or SPS.
2. Precise (P) Code: This is a much longer acquisition code. This signal is transmitted at 10.23 MHz which is 10 times the rate of a CA code. This results in a much more accurate signal with less noise. This signal allows for the acquisition of Precise Positioning Service. However, this service is not available to unauthorised users and cannot be spoofed. As a result, this signal requires additional decryption.

spilker1996global also mention that military operators can degrade GPS signals which result in decreased accuracy from 20 m up to 100 m. The reduction of these accuracies requires differential GPS techniques. However, due to the scope of this projects, this Section will not be explored any further. Finally, once the acquisition signal is transmitted, the GPS device begins modulating at 50 bits/s allowing the satellite to transmit its position as well as clock correction information to the device.

The GPS constellation consists of 24 GPS satellites. These are configured into three rings of eight satellites orbiting at different latitudes. These orbital altitudes were selected as 10.98 Nautical Miles (**spilker1996global**). This altitude was chosen to optimise user visibility with the number of crossings over United States ground stations, and cost of launching the satellites (**spilker1996global**). These satellites carry onboard atomic clocks with a stability of 1×10^{-13} ppm . This allows for extremely accurate signaling as well as allowing for much more predictable time and position signals (**spilker1996global**). To achieve this, these atomic clocks are made out of either Cesium or Rubidium. Also, a frequency correction at 4.5×10^{-10} Hz is provided to correct for relative shifts.

GPS error modeling

As mentioned in the previous section, GPS signal accuracy is greatly affected by Earth effects and satellite distribution. The main source of distortion is attributed to the Earth's ionosphere (**spilker1996global**). The ionospheric free electrons cause a delay in the modulated signal which is proportional to the sum of electrons along the signal's trajectory and inversely proportional to the signal's frequency squared. This delay is modeled as the product of a theoretical 90° delay (Zenith delay) and a function of the elevation angle (obliquity factor). This results in a ratio of between 1.0 to 3.0 at small elevation angles (**spilker1996global**). This results in delays of 3 m (often at night) to 20 m (after midday). These delays result in errors in positional accuracy as shown in figure 2.12 which can increase the measurement confidence interval resulting in an unreliable positional reading (**spilker1996global**). Fortunately, these delays are usually resolved by satellite correlated positions.

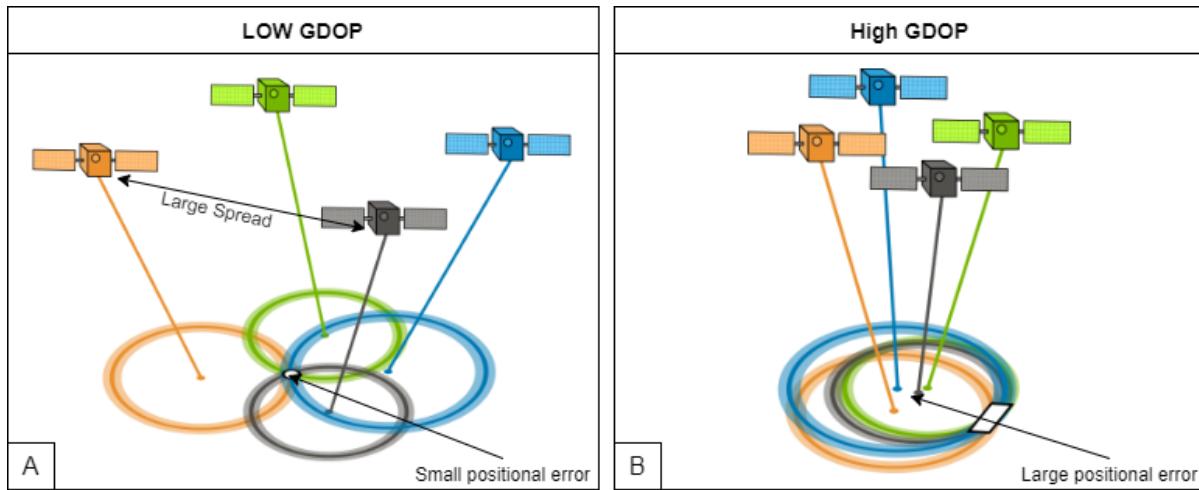


Figure 2.12: Diagram showing how satellite distribution affects the positional estimation. Global navigation satellite system (GNSS) satellites have an associated inaccuracy which is represented by the circles. Locking on to more than one satellite reduces these inaccuracies by interpolating the position from the phase differences between satellites. This value is greatly affected by the spread of satellites. (A) a larger spread results in a more accurate positional estimate (B) while a shorter satellite spread results in a less accurate positional estimate. A measure of this inaccuracy is called the dilution of processsion (DOP). Figure was adapted from images by ([GISGeo2020DOP](#)) for A and B.

Navigation errors are characteristic of GPS performances. These errors are affected significantly by satellite spread and ranging errors. Assuming the incoming signal is uncorrelated with a mean of zero, the RMS positional error is calculated as:

$$RMS_{error} = (\text{Geometric Dilution})(\text{RMS Ranging errors}) \quad (2.1)$$

The geometric dilution of precision (GDOP) is the error in the precisional spatial and temporal measurements, which is inversely proportional to the volume of the shape formed by four satellite ([jwo2001efficient](#)). [jwo2001efficient](#) outlines the procedure for the calculation of this value which is shown in Appendix A.2. A GPS position is calculated from the incoming signal of three or more satellites. This is because at least three satellites are required for a valid three-dimensional positional fix and four satellites for an accurate time fix ([jwo2001efficient](#); [spilker1996global](#)). However, if the internal satellite clocks are not synchronized, a clock offset appears resulting in a dilution in the temporal accuracy of the GPS signal. This value is called the time dilution of precision (TDOP). Signal noise can also contribute to spatial inaccuracies in estimating the one-dimensional vertical position (VDOP), the two-dimensional horizontal position (HDOP) or the three-dimensional position (PDOP). DOP values represent the ratio of measurement errors to pseudo-range measurement error¹⁸ where the higher the DOP number, the larger the measurement error and hence, the more inaccurate the positional fix will be ([tahsin2015analysis](#)). A description of DOP accuracies is provided in Table A.2.

¹⁸See Section A.2

Table 2.9: Table showing the ratings for a dilution of precision (DOP) measurement with a numerical range corresponding to a rating as shown by **tahsin2015analysis**. These ranges provide context for positional accuracies associated with GPS measurements with a DOP of 1 being an ideal reliable measurement with an associated small margin of uncertainty whereas a DOP of 20 is completely unreliable with a large uncertainty.

DOP Value	Rating
1	Ideal
2 to 4	Excellent
4 to 6	Good
6 to 8	Moderate
8 to 20	Fair
20 to 50	Poor

Finally ranging errors are shown to come from 6 sources (**spilker1996global**):

1. Satellite ephemeris
2. Satellite clock
3. Ionospheric group delay
4. Tropospheric group delay
5. Multipath scattering
6. Hardware/software errors

These errors become much more prominent in the polar region due to higher Ionospheric total electron current (TEC) than mid-latitude regions (**bishop1990ranging**). Furthermore, **bishop1990ranging** show that these errors can be corrected by coupling statistical models with satellite readings to reduce these errors. However, no adequate models exist for the polar region. Furthermore, the discussion in Section 2.2.3 shows that freezing temperatures and ice dynamics can be significant sources of failure for GPS signal acquisition (**doble2017robust**). These sources of error need to be accounted for when incorporating a GPS device into an in situ remote sensing device for the Southern Ocean.

GPS component selection

The component selection for each device is shown in Table 2.10.

Table 2.10: Comparison between different GPS devices, sampled data and periods between samples implemented by each device.

Device Name	GNSS device	Data sampled	Period between samples [minutes]
WIIB	Adafruit MTK3339	Geographical position	25
WIIOS	Adafruit MTK3339	Latitude (decimal degrees)	180
		Longitude (decimal degrees)	
NDWB	SkyTraq S1351R	Geographical position	60
SKIB	Adafruit MTK3339	Geographical postion	10
SWIFT	Qstarz BT-Q1000eX	Geographical position	0.003
		Horizontal velocity	
SIMB	Adafruit MTK3339	Geographical position	60
Polar ISVP	Jupiter F2 Module	Geographical position	180
Trident	Unspecified GPS	Geographical position	0 to 5
			5 to 60
			60 to 1440

Table 2.10 above shows the GPS receivers implemented in each system as well as measurements of interest. The most common GPS receiver is the Adafruit MTK3339 (**mtk3339**). Additionally, all devices have fixed transmission periods between samples except the Trident buoy which has user configurable periods that can be set from a couple of minutes to hours. Most devices use the GPS for positional location however, SWIFT buoy measures horizontal velocity in addition to vertical position (**thomson2012wave**). **thomson2012wave** justify this by stating that the accuracy of horizontal velocity measured by a GPS is accurate enough ($\pm 0.5 \text{ ms}^{-1}$) to be used for determining the orbital motions of waves. **kohout2015device** and **rabault2019open** provide the measurements recorded by the GPS. However, very little discussion is given regarding the performance and accuracy of the device. **doble2017robust** showed that during their year long deployment, 14 buoys maintained a near perfect satellite signal acquisition however, software issues with the GNSS receiver resulted in a failure to obtain measurements in five systems. However, **doble2017robust** had implemented two-way Iridium communication and were able to partially overcome this set back by transmitting commands that performed a software reset on the GPS receiver. Additionally, two devices lost signal for 6 days while another two devices lost signal for 62 days and 41 days respectively (**doble2013wave**). This shows that acquisition failures can cause temporal gaps in data and therefore, need to be reduced to provide robust data sets.

Each ice floe follows a unique trajectory (**lepparanta2001sea**) and individual trajectories combine to form a continuum. It has previously been believed that sea ice drift has been linked strongly to significant wave events (**alberello2019drift**). An experiment was conducted by **alberello2019drift** to measure the drift of sea ice during a cyclone event. Here it was found that wind velocity is the dominant driver of sea ice drift (**alberello2019drift**) causing ice drifts of up to 0.75 ms^{-1} (**alberello2019drift**). Sea ice drift speed is extremely sensitive to sampling rates. (**alberello2019drift**) where sampling rates of 6 hours can underestimate the ice velocity by 5% (**alberello2019drift**) and up to 20% for 12-hour sample rates. The consequence is reduced, near unusable, estimates of sea ice velocity components as well as drag coefficient and wind factor estimates.

High temporal resolutions are capable of capturing important, inter-daily activities such as ice oscillations. **alberello2019drift** state that to accurately capture sea ice dynamics, a temporal resolution of at least 3 hours is required (**alberello2019drift**). Table 2.10 shows that all devices sample GPS at a rate of 3 hours or less thereby fulfilling this criteria. This not only allows for the capture of accurate drift speeds but provides an accurate characterisation of instantaneous velocities and Coriolis forces (**alberello2019drift**).

2.3.4 Temperature sensing and measurement

As discussed in Chapter 1.1, Antarctic sea ice follows a unique formation cycle with different phases marked by changes in season. Sea ice formation additionally is affected by environmental factors which can arise from one of two sources: Long-period seasonal cycles (**barber2005microwave**) which affect snow growth and sea ice formation, and short term extreme weather events (**vichi2019effects**; **albarello2020drift**) which affect the thermodynamics of the formed ice and results in ice drift and collisions (**arrigo2004large**). One key measurement for correlating environmental effects to sea ice formation is air temperature. **barber2005microwave** show that the melting and freezing phases in the sea ice life cycle is preceded by increases in air temperature above freezing and decreases in temperature below freezing respectively. In addition, **vichi2019effects** show that warmer air temperatures were observed during polar events. This shows that temperature is critical measurement for understanding sea ice lifecycles and the effects of ocean processes. Additionally **kohout2015device**; **doble2017robust**; **doble2017robust** have discussed the significance of sea ice melting. This was considered a significant phase of sea ice and was classified as a phase in the development of their buoy. Hence, we turn to temperature sensing as a solution to understanding the effects of temperature on Antarctic sea ice.

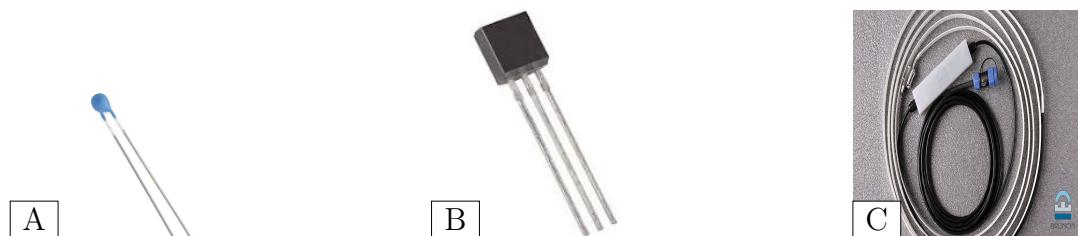


Figure 2.13: Examples of temperature sensors employed by remote sensing devices. These types of devices include (A) thermistors such as the PR303J2 (image source: ([pr303j2](#))), (B) digital temperature sensors such as the DS18B20 (image source: ([ds18b20](#))) and (C) digital temperature chains (DTC) such as the Bruncin DTC (image source: ([bruncindtc](#))).

Temperature sensing technology is commonly used for thermal compensation as measurements such as pressure and humidity are dependent on environmental temperature (**mansoor2015silicon**). Thermal sensing technology exists in a variety of forms. However, choice of sensor is heavily dependent on the application i.e. the object to be measured, the material state and the type of contact with the sensor (**mansoor2015silicon**; **childs2000review**). In addition, sensor selection is based on the range, accuracy, resolution and precision of the device to ensure correct use. An overview of different electric

sensors is given below (**child2000review**). A full discussion of the types of sensors is provided in Appendix 2.11 where the types of technology, advantages and disadvantages are provided. Additionally, the technology discussed has been selected for its implementation in the remote sensing devices shown in Table 2.11 below.

Table 2.11: Comparison of the different components used by each device for temperature measurement as well as the sensor type and the measurement objectives of the device. Each device measures temperature of a different process. For example: the WIIB measure air temperature whereas a device like the SIMB measures temperature profiles of ice floes in addition to air temperature.

Device name	Temperature sensor	Sensor type	Measurement objective
WIIB	VectorNav VN-100	Digital temperature sensor	Air temperature
WIIOS	Maxim Integrated DS18B20	Digital temperature sensor	Air temperature
NDWB	SBG IG-500	Digital temperature sensor	Internal ambient temperature
SKIB	None	none	None
SWIFT	Aquadop profiler	Thermistor	Water temperature
SIMB	DS18B20	Digital temperature sensor	Air temperature
	Bruncin DTC	Digital temperature chain	Ice temperature profile
Polar ISVP	Littelfuse PR303J2	thermistor	Sea surface temperature
Trident	unspecified	Not Reported	Air temperature

As shown in Table 2.11, digital temperatures sensors are the most common with thermistors being the second most prevalent. Thermistors are semi-conductor based temperature sensors whose resistance changes as a function of temperature which can be described with a negative temperature coefficient. These devices have a typical accuracy of 0.05° C to 0.01° C with a temperature range of –100° to 300° C (**child2000review**). These devices have a fast response and are cheaper than their counterpart: the resistance temperature device (RTD) (**child2000review**). However, they are not strong enough to reach the desired operating ranges as standalone devices and require additional circuits (**tong2001improving**). Additionally, these devices have a larger measurement uncertainty as they are more susceptible to noise (**child2000review**).

Another common technology basis for temperature sensing is using a semi conductor temperature device (STD). These consist of diode and transistor based circuits whose voltage changes with temperature. These devices have measurement range from -55 °C to 150 °C with an accuracy of 0/08°C. Additionally, these devices are suitable for MEMS-based circuits and high powered electronics (**willander2006silicon**). These devices are readily available with a reliable accuracy however, the performance of the device is heavily dependent on the material used. For example, silicon has a low temperature stability but a lower voltage output (**child2000review**).

Table 2.11 shows that most devices use digital temperature sensors for ambient air temperature measurement with thermistors begin the second temperature sensors type as used by **thomson2012wave**. Most device use temperature sensors integrated into high powered IMU/AHRS systems, as is the case with WIIB and NDWB. While the SWIFT buoy and the Polar ISVP both utilise the same sensing technology. However the Polar

ISVP used a standalone Littelfuse PR303J2 whereas the SWIFT used the temperature sensor onboard the Nortek Aquadopp Profiler which is, primarily, a current profiler. This shows that temperature sensing was not a primary focus for this device design as there is a lack of discussion regarding the performance of the temperature sensor in the literature.

2.3.5 Atmospheric pressure sensing and sensors

There has been an increased demand for in situ environmental monitoring as mentioned by **vichi2019effects; kennicutt2014polar; kennicutt2016delivering; kennicutt2019sustained; alberello2019drift**. It can provide insight into wind currents and storm events as well as predict trajectories of these storms. In addition, atmospheric pressure is crucial for characterising atmospheric-ocean process such as tropical storms which create areas of low pressure during their trajectory (**vichi2019effects**). Current devices have employed pressure sensing techniques which have been critical for determining accurate AHRS measurements (**vn-100**) in addition to environmental monitoring. Examples of pressure sensing devices are shown in Table 2.12.

Table 2.12: Comparison of the different pressure sensing components used by each device as well as the measurement objective guiding the selection of the component. "Not Reported" is given where a device includes a pressure sensor but provides no information to its use. "-" is given where a column does not apply to the device. "None" is given where a device does not contain a pressure sensor.

Device name	Pressure sensor	Sensor type	Measurement objective
WIIB	VectorNav VN-100	Digital barometric pressure sensor	Atmospheric pressure
WIIOS	None	-	-
NDWB	NXP MXP-5100-AP	Piezoresistive silicon pressure sensor	Atmospheric pressure
SKIB	None	-	-
SWIFT	SBG Eclipse-N	Not Reported	Not Reported
SIMB	Bosch Sensortec BME280	Digital barometric pressure sensor	Atmospheric pressure Humidity
Polar ISVP	Vaisala PTB100	Silicon capacitive barometric sensor	Atmospheric pressure
Trident	None	-	-

Table 2.12 above shows the common pressure sensing types used in remote sensing. While different devices are reported, **eaton1997micromachined** shows that the basis of modern sensors are built around MEMs with diaphragm-based sensors being the most common system. These devices consist of a miniature diaphragm mounted in a specific way. Pressure is determined by measuring the deflection of the diaphragm against a reference pressure and converting the result into an electrical signal. While these devices are common for sensing pressure (**eaton1997micromachined**), the diaphragm is susceptible to plastic deformation which can affect the long term reliability of the device (**eaton1997micromachined**). Furthermore, mechanically stronger diaphragms have a non-linear relationship with current which can result in inaccuracies from calculation errors.

Another pressure sensor type is a piezoresistive sensor which is a semiconductor based device. These devices are diaphragm-based MEMS sensors whose resistance changes with a change in stress ([eaton1997micromachined](#)). A major advantage of these devices is the linear relationship between resistance and pressure. They are robust with reduced hysteresis and measurement drift effects and exhibit elastic properties at low temperatures and are stronger than strain gauges ([eaton1997micromachined](#)). However, device accuracy is significantly affected by thermal expansion which causes thermal drift. These devices are also susceptible to resistor noise therefore requiring resistors with identical temperature characteristics. Furthermore these devices require extensive compensation techniques which can increase the complexity of the device.

While pressure is important for understanding atmospheric processes, Table 2.12 shows that it is the least critical component for in situ remote sensing. The WIIOS, SKIB and Trident sensors do not include a pressure sensor at all. The consequence is that these devices are not suitable for characterising the effects of climate on sea ice as well as capturing the/predicting severe weather events such as the cyclones discussed in [vichi2019effects](#). The SWIFT buoy includes the capability of pressure sensing. However, do not sample the sensor for standalone pressure measurements. The devices that include pressure measurements focus on measuring atmospheric pressure through digital barometric pressure sensors ([rabault2019open](#); [PLANCK2019102792](#)), piezoresistive silicon sensors ([doble2017robust](#)) or silicon capacitive sensors ([uptempo](#)). Examples of these sensors are shown in Figure 2.14.

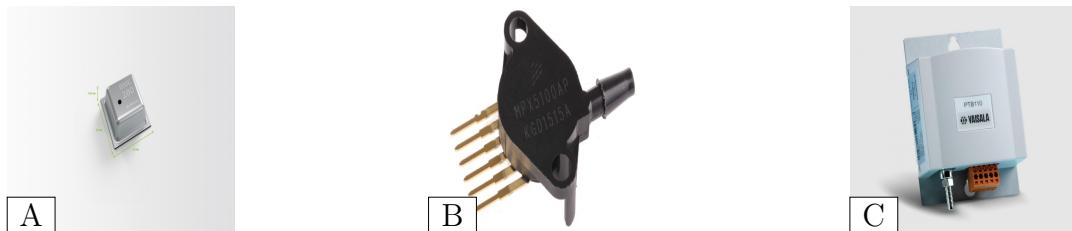


Figure 2.14: Examples of pressure sensing technologies using a diaphragm based MEMS with different methods of measuring the strain. These are piezoresistive sensors: (A) the BME280 (image source: ([BME280](#))), (B) the MPX-5100-AP (image source: ([MPX5100AP](#))) or capacitive sensors: (C) the PTB100 (image source: ([PTB110](#)))

This shows that silicon-based pressure sensors are the most commonly used for remote sensing. [eaton1997micromachined](#) discuss the importance of micro-machined pressure sensors and provide an overview of various sensors that have been used in remote sensing devices as shown in Table 2.12.

2.4 Conclusion

From the system analysis, we can see that all devices were successful in communicating remotely with a user through the Iridium satellite network due to its high area coverage and reliability. However, due to network and bandwidth constraints, most devices compensated for this by implementing computationally intensive algorithms to compress data

into the required bandwidth, the popular choice for network modems is the Iridium 9603 which is ideal for its low power consumption, small form factor and relatively low cost. However it only allows for short burst data transmissions. For continuous transmissions, the 9523 modem is ideal.

Systems were able to operate remotely for long periods of time using batteries as a power source. Lithium batteries and alkaline batteries were the most prevalent as primary power sources allowing devices such as the WIIOS to survive for a maximum of 39 days. **doble2017robust** showed that in the Arctic Marginal Ice Zone by coupling a solar power to a secondary lead acid battery, the device could survive for up to a year without human intervention. However, **rabault2020development** achieved lower survivability even with an onboard solar panel. Additional energy harvesting techniques are required for secondary power source to be viable.

In spite of advances in power optimisation, the Marginal Ice Zone is a dangerous region. A significant number of devices were reported by **doble2017robust** to have failed when devices froze over and failed to recover during sea ice melt. Additionally, polar storms and large swells caused the WIIOS buoys to be offline during their deployments (**kohout2015device**; **alberello2019drift**). Additional sources of failure have been attributed to freezing over **rabault2018investigation**. However, this was due to the device being placed in the ground allowing it to be covered by snow growth and flooding (**barber2005microwave**). Additionally, **rabault2019open** reported failures due to rafting and ridging of ice floes which critically damaged devices.

From the system analysis, we can determine that the most important measurement objectives are:

- Ice drift
- Wave-in-ice measurements
- Ambient temperature
- Atmospheric pressure

Wave analysis is the most computationally expensive measurement objective as Table 2.8 show that raw IMU time series measured by each wave sensing device was sampled above 2Hz (**kohout2015device**). Advanced filtering techniques were required to compensate for the effects of low drift and biases. Some devices achieved this by using components with integrated filters such as the VN-100 AHRS sensor, However, these components are expensive and resulted in a higher overall cost for the device (**guimaraes2018surface**). Additionally, devices such as WIIOS and SWIFT required more expensive and powerful processors to compensate for higher data rates and advanced processing algorithms.

Additionally, all remote sensing devices in this review used a GPS to measure sea ice drift showing that these devices are reliable enough to perform these measurements (**doble2017robust**). A sample time of at least 3 hours between GPS measurements is required to accurately capture sea ice dynamics such as oscillations and collisions. **thomson2012wave** show that GPS velocity measurements are suitable for wave measurements in addition to telemetry and used these measurements to determine the orbital

motion of waves. However, the positional accuracy of GPS is not suitable for measuring significant wave height as the uncertainty in positional measurements is too high (± 5 m). GPS readings are affected by ionospheric interference which causes a phase delay resulting in a dilution of precision (DOP), this can also be caused by the spread of satellites, azimuth and number of satellites used to take measurements. Hence, these measurements form a critical part for statistical analysis of GPS signals as this section was under-reported in the current literature.

Environmental sensing is critical to determining the conditions around sea ice formation. Temperature and pressure changes are key indicators of short term weather events such as polar cyclones and as such, all remote sensing devices in this review employ some form of temperature measurement either actively through components such as thermistors, and silicon temperature devices or passively embedded in higher powered sensors such as an AHRS. Additionally, the effects of temperature sensing on atmospheric temperature measurements were significantly under-reported in the literature as was the effect of pressure sensing. Pressure sensing was also the least critical objective for the devices as this device was excluded from three out of eight devices while the SWIFT buoy had pressure sensing capabilities, this feature was not used. Hence, more research is required to fill in this knowledge gap to understand the effects of environmental sampling and their impact on climate measurements.

In conclusion, while the literature shows breakthroughs in low cost in situ remote sensing in the Marginal Ice Zone, some of the effects remain unknown. Devices are still extremely specialised. However, open-source, off the shelf devices can pave the way for more accessible, more affordable devices thereby increasing access to science and increasing our understanding of the Marginal Ice Zone.

Chapter 3

Design methodology

3.1 Design overview

In Chapter 2, the critical areas, challenges and techniques used by remote sensing devices to meet the objectives outlined in Chapter 1.1 were discussed. Remote sensing devices aim to increase the temporal and spatial resolution of Southern Ocean data sets by increasing survivability and concentration of remote sensing platforms. The challenges identified in Chapter 2 include the sea ice dynamics which can damage the device. Furthermore, strong wave activity and low temperatures can freeze the device and degrade the battery life. A key difference between each device was the measurement objectives as shown in Table 2.4 which dictated the sensors, processing strategy and functionality of each system.

This section outlines the steps taken to identify the user requirements and translate them into hardware subsystems. The design phase began with a user requirements analysis. These requirements resulted in well-defined measurement objectives for the device and the selection of subsystems critical to the device functions. Then, a high-level system diagram was created to show the interaction of the subsystems within the system. This allowed for the selection of hardware components to satisfy the requirements for each subsystem. Finally, to ensure that the subsystems met the user requirements, a set of acceptance tests were derived.

3.2 User requirements

The user requirements analysis began with an identification of the project stakeholders shown in Appendix B.1. Through constant engagement with the primary stakeholders, a set of user requirements was generated which identified the objectives the device would need to meet. The formatting, presentation and selection of these user requirements was done in accordance with IEEE Standards¹.

¹IEEE_STD_UREQ.

Table 3.1: User requirements obtained by meeting with the principal stakeholders. These will be used to determine the desired functionality of the buoy.

User Requirement ID	Description
UR001	System must be able to withstand the Southern Ocean climate.
UR002	System must be able to transmit data remotely without additional infrastructure or user input.
UR003	System shall provide higher temporal and spatial resolution data.
UR004	System must be capable of measuring ice floe dynamics and environmental conditions surrounding sea ice formation.
UR005	System must be user-friendly and easy to deploy.
UR006	System Must be cost effective.
UR007	System must be able to store and process data in an organised manner.

3.2.1 Analysis of UR001

System must be able to withstand the Southern Ocean climate.

kohout2015device encountered 7 m swell and 25 ms^{-1} winds during their deployment in the Marginal Ice Zone which led to the immediate failure of one of their systems (**kohout2015device**). Strong wind and waves were also observed by **vichi2019effects**; **alberello2019drift** during their deployment which has been attributed to multiple cyclonic events coinciding with the deployment of the WIIOS buoys. The effects of wind and waves result in the delay of consolidation of sea ice at the edge of the Marginal Ice Zone which created survivability challenges for the devices. Additionally, **albarello2020drift** showed that wind speed resulted in ice floe drifting at 0.35 ms^{-1} . Placing the buoy close to the surface of the ice subjects it to collisions, breaking and rafting which resulted in the failures of devices during the deployments by **doble2017robust**; **rabault2019open**. Snow and ice formation on the surface of the floe can bury the device (**doble2017robust**) and constant wave energy on unconsolidated ice cause flooding can further damage the devices. A device that is elevated above the surface while tethered to the floe can reduce exposure to these events. However the buoy elevation must be at least 1 m to compensate for snow growth (**barber2005microwave**).

Furthermore, for a device to survive in the Southern Ocean, the software must contain a robust set of error checking for each module. The software should ensure that critical subsystems are functional and responsive while constantly providing feedback on statuses. Should a module be damaged or go offline, the software should flag this issue, identify the failure and attempt to rectify it. Alternatively, if the failure is critical, the device should flag the module as malfunctioning and continue to operate without it.

3.2.2 Analysis of UR002

System must be able to transmit data remotely without additional infrastructure or user input.

kennicutt2016delivering shows a fundamental lack of infrastructure on the Antarctic continent including data networks. Operations taking place on sea ice are isolated from any resources that exist via Antarctic bases. Sea ice in the Marginal Ice Zone is subject to conditions inhospitable to humans (**kennicutt2016delivering**). Therefore access to the buoys is limited once the device is deployed often making it difficult to retrieve. The life cycle of sea ice presents an additional access challenge through the consolidation of sea ice during the freezing periods and melting of the ice floes during the warming periods (**womack_2020**). Additionally, manned expeditions are typically inflexible (**kennicutt2016delivering**) resulting in additional challenges in retrieving the buoy and hence the data. Therefore, most devices deployed in the region are designed to be expendable ((**kohout2015device**; **rabault2019open**; **trident**)). These device circumvent this by using satellite networks with global coverage such as Iridium² which allows the users to receive data and status updates from the buoy without needing to retrieve the devices.

The lack of user input implies that device routines and sub routines must be performed automatically. The device should control and sample sensors in a fixed, predictable manner ensuring that the correct data are sampled during the correct periods. Furthermore, the device needs to transmit data over the Iridium network. Therefore the software is required to control the interactions with the mode ensuring that it is functional. The software should condense data to fit the bandwidth of the device and ensure it is successfully uploaded to the transmission buffer. Finally, the software should be able to check for sufficient network availability and initiate a transmission. Should an error occur, the software needs to respond and handle it efficiently.

3.2.3 Analysis of UR003

System shall provide higher temporal and spatial resolution data.

Phases of the sea ice life cycle are defined by periods of freezing followed by periods of melting (**barber2005microwave**) with maximum extents occurring in winter (freezing) and summer (melting) respectively (**barber2005microwave**). The formation and consolidation of sea ice floes are influenced by atmospheric and oceanic processes resulting in the delay of sea ice consolidation (**vichi2019effects**; **albarello2020drift**). Each period coincides with a seasonal cycle typical lasting a few months (**vichi2019effects**; **albarello2019drift**; **barber1948generation**) which is the length of time a buoy needs to survive to provide sufficient data on a temporal scale.

Increasing remote sensing in the region is also required to provide spatial coverage (**albarello2020drift**). Certain observational methods such as satellite observation are performed on a 10 m scale (**galin2011validation**) where sea ice variability can scale

²See Chapter 2.1

down to centimetres (**vichi2019effects**). **doble2017robust** achieved large spatial coverage by deploying the devices in clusters of five every 5 km. Additional deployments from **vichi2019effects**; **kohout2015device**; **albarello2020drift** also achieved this by deploying multiple systems spaced 3 to 4 m apart (**vichi2019effects**). Therefore increasing the spatial resolution can be achieved by increasing the concentration of devices in an area spaced apart.

For a remote sensing device to survive the required period, a built-in power source is required to maintain functionality without constant maintenance. This power supply primarily needs to come from a replenishable source. **doble2017robust** and **rabault2019open** coupled battery arrays to a rechargeable power system which showed promise however, insufficient cloud cover (**doble2017robust**) resulted in the solar panel being under utilised. As discussed in Chapter 2, commercial batteries are readily available and well-specified to supply power for long periods (**rabault2017measurements**). However, the system must conserve energy. Otherwise, this could result in low survival rates (**kohout2015device**). Additionally, batteries in sub-zero temperatures have a significantly reduced capacity of up to 50% (**doble2017robust**).

Therefore, the software needs to contain power monitoring capabilities. A sensor should monitor the current and power usage of the device and provide feedback to the system. The device software should be optimised to conserve power which means turning off sensors that are not in use and entering low power mode during periods of inactivity. Finally, the software needs to handle power events that could disrupt the functionality of the buoy such as brownouts and power resets. The software should be able to recognise when the power is too low to operate and respond accordingly.

3.2.4 Analysis of UR004

System must be capable of measuring ice floe dynamics and environmental conditions surrounding sea ice formation.

As discussed in Chapter 2.1, in situ, remote sensing devices have been deployed in the Southern Ocean with specific measurement objectives. Table 2.4 shows that the following measurements were common to one or more devices:

- Ice drift
- Collisions between ice floes
- Waves-in-ice
- Ambient temperature
- Atmospheric pressure

Meteorological sensing requirements

Temporal resolutions and measurement standards need to be taken in accordance with the World Meteorological Organisation to ensure effective communication and standardisation of the data sets ([worldmeteorologicalorganization_2010](#)). For environmental data and wave spectra, the data are provided in Table 3.2.

Table 3.2: Comparison of standard measurements for meteorological data including temporal resolution, measurement unit and accuracy from: ([worldmeteorologicalorganization_2010](#))

Variable name	Resolution	Accuracy
Temperature	1 hour	± 0.5 K
Pressure	1 hour	± 0.5 hPa
Wind speed	1 hour	± 2 ms ⁻¹

Meteorological data should be sampled from a height of 1 to 40 m ([worldmeteorologicalorganization_2010](#)). Therefore, all meteorological sensors should be mounted 1 m above the ground to meet these standards. Additionally, the software should be capable of retrieving data from a temperature and pressure sensor. The software should also contain error checking to ensure robust sampling and reliable data from the sensors.

Ice drift sensing requirements

trident; uptempo; kohout2015device and **albarello2020drift** show that ice drift measurements can be taken using a GNSS tracker can be used to monitor by recording the global coordinates against an accurate time reference. **alberello2019drift** show that temporal resolutions affect the behaviour of drift data. Devices that went into low power mode during deployment ((**vichi2019effects**; **alberello2019drift**)) had decreased sampling rates which failed to capture ice drift oscillations. **albarello2020drift** further show that a temporal resolution of 3 hours is required to capture ice drift oscillations. Additionally, the GPS reading can be affected by the number of satellites picked up by the receiver (dependent on the receiver antenna gain) (**spilker1996global**), the spread of satellites and the angle of elevation above the horizon. A characteristic of this error is called the dilution of precision (DOP). This value details the positional or temporal measurement inaccuracy. Moreover, the software needs to accommodate a GPS module and interface with it through a suitable communication port. The software needs to sample the GPS at a frequency of once every 3 hours or less. Finally, the software should configure the GPS module to transmit diagnostic information, positional error and temporal error information in addition to time and date information.

Waves-in-ice sensing requirements

Waves-in-ice measurements play a critical role in understanding the formation of sea ice in the Marginal Ice Zone (**alberello2019drift**). Low frequency, high energy waves propagate through the area and displace the ice floes (**womack_2020**). Current devices such as those developed by **rabault2017measurements**; **thomson2012wave**; **kohout2015device** utilise either a statistical (**kuik1988method**) or spectral (**earle1996nondirectional**; **welch1967use**) approach which allows for measurements of waves independent of the dynamics of the ice floe. These methods require roll, heave, pitch and vertical acceleration measurements for 1000 s at sample rates above 0.5 Hz, which corresponds to the upper band of ocean waves (**earle1996nondirectional**). The software should be capable of interfacing with an IMU and should ensure that data are sampled at a fixed sample rate for the required period. The software should also store the data in a memory location with sufficient space. Finally, the software should contain error checking algorithms to ensure robust communication with this device.

3.2.5 Analysis of UR005

System must be user-friendly and easy to deploy

Table 2.5 shows the weight of each system. Notably, the lightest device being the Trident buoy at 0.42 kg and the heaviest being the buoys by **doble2017robust**. The weight of the device significantly affects the deployment protocol. Heavy devices will require a large team to deploy the device and specialised courier methods to transport it such as those described in **doble2017robust**. Devices that weighed 4.5 kg to 30 kg were light enough to be carried by one person and deployed much faster.

The deployment is also affected by the set up of the device. The SIMB Buoy, while being relative light, requires a team to assemble the components on the ice and drill a hole to place it in (**PLANCK2019102792**). **kennicutt2016delivering** shows that certain polar regions are too dangerous for manned missions, which can limit the deployment location and time frame the crew can spend setting up the buoy. However, systems that are relatively easy to deploy (Trident, WIIOS, SWIFT) allow for sensing near dangerous environments such as the ice edge of the Marginal Ice Zone . These devices are preassembled and set up leading to a "drop and go" style deployment from ship cranes (**vichi2019effects**; **alberello2019drift**) or boats (**rabault2019open**; **kohout2015device**). The software should be configured to begin execution as soon as the device is operational. Software settings and parameters should be loaded into permanent storage and the sensors should be calibrated before being sent on the expeditions.

3.2.6 Analysis of UR006

System must be cost effective

PLANCK2019102792; **guimaraes2018surface**; **rabault2019open** consider cost to be a significant constraint in designing a system. Additionally, some devices such as SIMB

buoy, gradually factored in price after two iterations (**planck2019evolution**). This shows that optimising device performance for cost is critical for increasing the affordability and availability of devices. Table 2.5 showed a comparison of reported costs and weights for each system where applicable.

From Table 2.5 we can see from the cost breakdown that **rabault2019open** succeeded in creating a low cost buoy through the use of open-source hardware and off-the-shelf components resulting in this device having the lowest reported price out of all the system. The next cheapest device is the Trident buoy which is only R300.00 more expensive than the WIIB. However, this device contains fewer sensors and is only capable sea ice drift and temperature measurements. On average, commercial systems (UptempO, Polar ISVP, Trident) proved to be significantly more expensive than research devices with similar attributes. However, due to the absence of prices for SWIFT, WIIOS and Trident it is difficult to draw conclusions from this comparison. Furthermore, commercial systems had design capabilities to design and print custom circuit boards and chip sets while research systems (SWIFT, WIIB, WIIOS) did not have such capabilities. Therefore, the developers optimised for procurement time by using off the shelf components (**kohout2015device**; **rabault2019open**). A novel sensing device that is cost optimised should result in an overall cost cheaper than the devices in Table 2.5 with comparable performance, this will also allow for quicker and cheaper device procurement allowing for more devices to be produced for deployment thereby allowing for a larger spatial area in the Marginal Ice Zone to be covered.

3.2.7 Analysis of UR007

System must be able to store and process data in an organised manner

The proposed system will require multiple subsystems to satisfy the user requirement UR004. These subsystems will generate large volumes of data that needs to be stored and organised efficiently. As shown in the discussions for user requirements UR001 to UR005, the system will require multiple sensors and modules. Each module has different operational and communication requirements. Therefore, a suitable processor needs to be selected to provide sufficient ports to interface with each sensor. This processor should accommodate sufficiently high volumes of data with a wide enough byte size to accept sensor data without calculation errors. The software will need to meet the requirements of each sensor and sequence these functions into a routine. This can be achieved by developing sensor libraries for each device. Included in these libraries will be initialisation routines, peripheral communication drivers and sensor sampling routines to meet the requirements for each sensor. Furthermore, as shown in the discussion above, each sensing activity is time-critical. This places a strict timing requirement on the device to ensure the integrity of the data set on a temporal scale. Therefore, an accurate timing reference is required in the form of a real-time clock (RTC).

Additionally, each sensor has a different data rate requirement. Some sensors will need to stream large volumes of data of an unknown length. To deal with this, efficient data acquisition techniques need to be implemented by the software to accommodate these volumes. Such techniques include setting up direct memory access (DMA) channel in a circular buffer to stream data to a memory location with an input capture timer on slave

reset mode to close the channel when the data stream is completed.

Finally, long periods of inactivity between measurements can result in wasted power. The discussion of user requirement UR003 showed that the software should turn the device off or into low power mode during these periods. If the data are in the processor memory when the device is off, it could get lost. Therefore, a permanent storage device is required to store data during these periods of inactivity. The software should include functions to read from the storage space and write to it efficiently. Additional error checks should be written to ensure the storage device is online, operational and the data have not been corrupted.

Finally, the aforementioned hardware and software modules need to be verified with a suite of acceptance tests to ensure that all the user requirements have been met.

3.3 Functional requirements

Analysis of the aforementioned user requirements resulted in the procurement of a set of functional requirements that dictate how the buoy will function.

3.3.1 Operational requirements

Table 3.3: Requirements addressing the mechanical needs for the system during operation.

Requirement ID	Description	User requirements addressed
FR001	The system shall have a protective enclosure against precipitation and frost.	UR001 UR005
FR002	The enclosure shall be constructed from strong, corrosion resistant materials with strong thermal characteristics.	UR001 UR005
FR003	The device will protect electronics from internal humidity.	UR001
FR004	The electronics will be elevated above the ground by 1 m.	UR001 UR005
FR005	All subsystems shall be rated for extreme temperatures.	UR001 UR003

3.3.2 Electronic requirements

Table 3.4: Requirements addressing the electronic needs for the system including the modules, components and sensors that satisfy the user requirements.

Requirement ID	Description	User requirements addressed
FR006	System will transmit data via Iridium satellite network.	UR002
FR007	Device shall be battery powered.	UR001
		UR003
FR008	System shall measure ice drift using a global positioning system (GPS).	UR004
FR009	Device shall measure ambient temperature.	UR004
FR010	Device shall measure atmospheric pressure	UR004
FR011	Device shall contain an inertial measurement unit (IMU) to record acceleration (3-axes) and rotation (3-axes) of an ice floe.	UR004

3.3.3 Software requirements

Table 3.5: Software functional requirements for the system addressing the system function, performance, operation and control during the lifetime of the device.

Requirement ID	Description	User requirements addressed
FR012	Device to contain sufficient memory for data storage.	UR006 UR007
FR013	Device to contain a processing unit to control sensors and process data.	UR006 UR007
FR014	Device to be optimised for low-power consumption and power event handling.	UR003 UR006 UR007

3.3.4 Other requirements

Table 3.6: Other system requirements being addressed.

Requirement ID	Description	User requirements addressed
FR015	The device shall be calibrated prior to shipping and delivered in a state where it can be deployed at a moment's notice	UR005
FR016	The device will cost less than currently available systems.	UR006

3.4 System overview

To meet the functional requirements, the overall system was designed using a top down approach. The requirements highlighted in Tables 3.3 to 3.4 will be used to identify key subsystems to achieve the required functionality.

Table 3.7: Table showing the subsystems that are critical to the functionality of the buoy and the level of importance indicated by rank

Name	Rank
Firmware	1
Power system	2
Communication module	3
Processor	4
Sensors	5
Permanent storage	6
Mechanical features	7

The firmware is the most critical subsystem in the device and ranks the highest in terms of priority. The firmware is crucial for the operation of the device and to control all the modules. The next most critical component is the power system. Any failures in the power system will cause the device to stop functioning. All device components need to be rated for subzero operation to ensure robust operation. A voltage regulator will be included to ensure a constant voltage. Finally, a power sensor will be included to monitor the batteries and warn the system when the batteries are almost depleted.

After the power system is the communication subsystem as Iridium will be used to transmit the data obtained by the system. Should this device fail, the device will be unable to transmit data unless it is retrieved. Satellite communication for Iridium is performed using a satellite mode while GPS communication is performed using a GPS receiver. They require, clear, unobstructed views of the satellites which can be achieved with high-gain antennas. These devices need to be mounted as close to the sky as possible.

The sensors are the primary interface between the system and the environment. The electronics need to be as close to the exterior of the system as possible to measure ambient temperature and pressure. The IMU can be mounted anywhere inside the buoy however, the device needs to be calibrated for its position on the buoy as well as its orientation inside.

The sensors will interface with a central processing system which will control the sensors and sample data from them. Data coming from the sensor will be processed and stored in packets in a permanent memory storage system. Finally, a metal stand will be created to anchor the device to an ice floe and suspend the electronics well above the sea ice surface to prevent it from being covered in snow. The electronics will be placed in a thermal-resistant and waterproof enclosure to protect the system with desiccant placed inside to prevent moisture from interfering with the device. A block diagram of the proposed

system is shown in Figure 3.1.

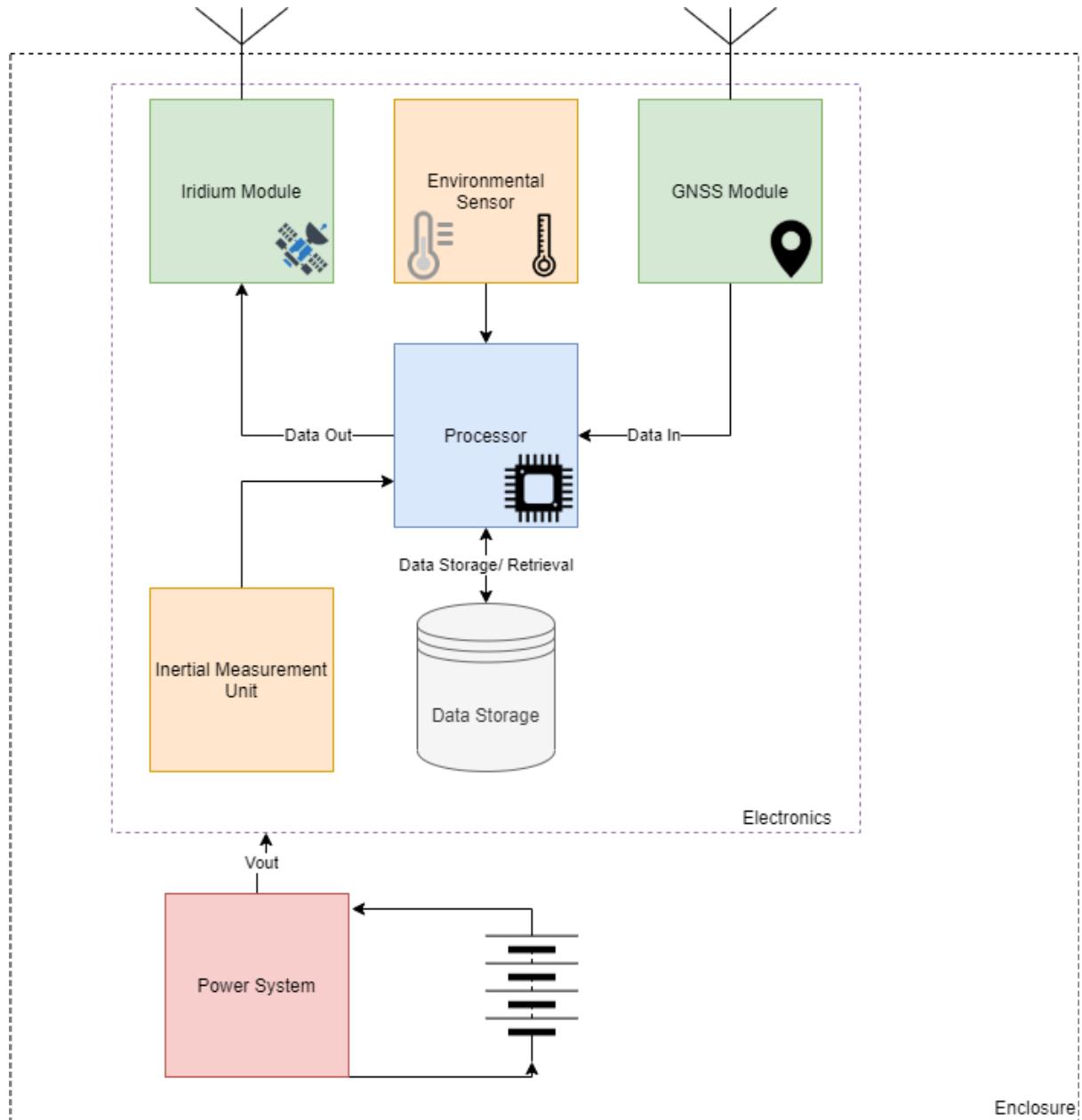


Figure 3.1: Block diagram of the proposed autonomous system showing subsystem arrangement, data flow and interfaces with the environment.

These subsystems can be further broken down into components requirements as shown in Figure 3.2.

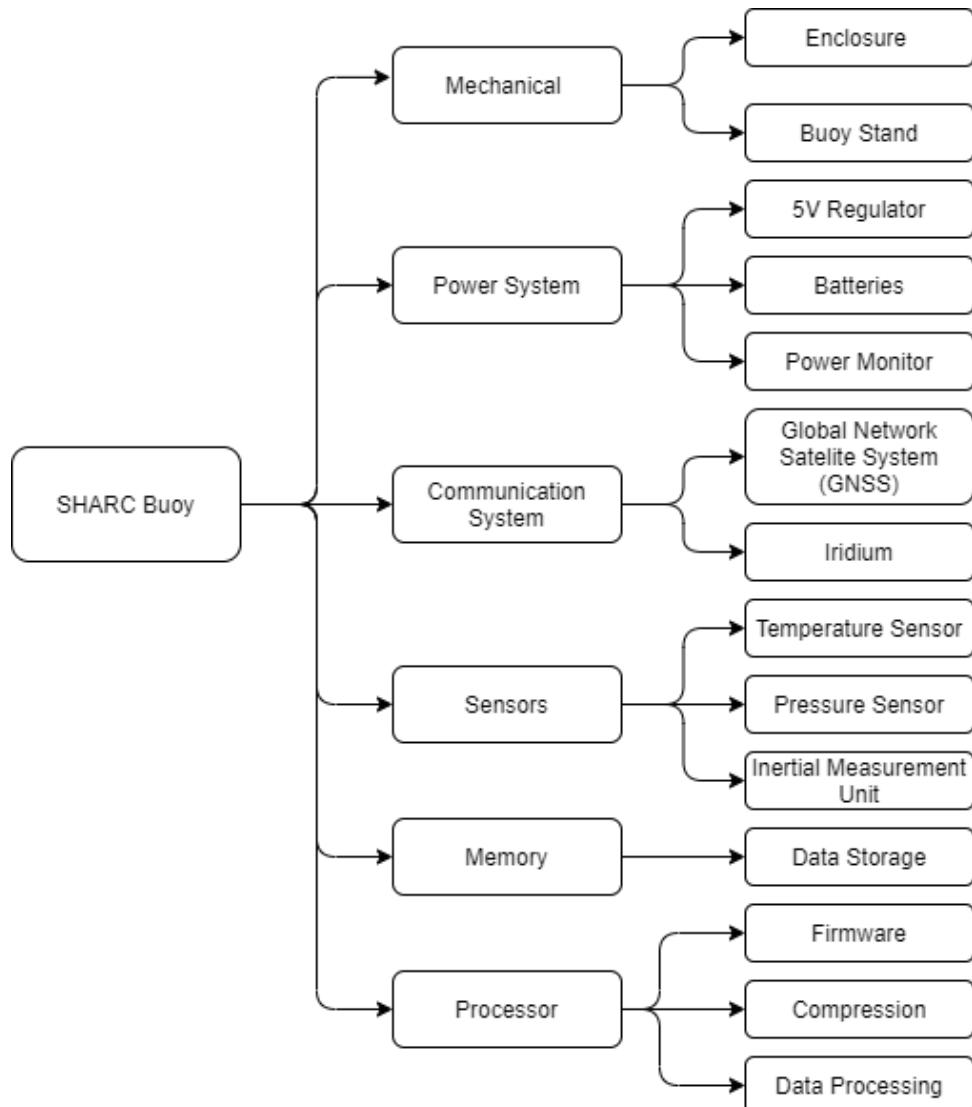


Figure 3.2: Breakdown of subsystems from Table 3.7 into usable components.

3.4.1 Technical specifications

These technical specifications will be used to determine what hardware is required to construct the subsystems as shown in Figure 3.2.

Table 3.8: Technical specifications for the overall system

Specification ID	Description	Functional requirement addressed
SP001	Enclosure built using thermal resistant plastic.	FR001 FR002
SP002	Electronics to be mounted on a 1.5 m stand constructed from non-corrosive metal.	FR002 FR003 FR004
SP003	System to include desiccant packets inside the enclosure.	FR003
SP004	Device to have a temperature operating range of -40° C to 20° C with 1° C uncertainty.	FR009
SP005	Subsystems to be rated for 3.3 V to 5 V power.	FR008
SP006	Device shall survive for 1 month on a single set of cells	FR007
SP007	The device should cost less than R10,000	FR016
SP008	System will contain flash chips for permanent storage.	FR012
SP009	System will use the STMicroelectronics STM32 series of microcontroller.	FR013
SP010	The system shall be supplied by a regulated 5 V supply.	FR014
SP011	The low power threshold occurs for voltages $< 5\text{ V}$	FR014
SP012	Maximum current operations: 500 mA maximum start-up current.	FR0014
	100 mA maximum active current.	
	10 mA sleep current.	
	Device to be powered off or placed in sleep mode when inactive.	FR014

3.4.2 Acceptance test protocols

These tests are designed to ensure that the devices meet the functional requirements outlined in Table 3.3 to 3.6. A full description of the acceptance test protocols can be found in Appendix B.2. The goal of the acceptance criteria of each test as well as the targeted module is given in table 3.9.

Table 3.9: A summary of acceptance test protocols from Appendix B.2 showing the target and purpose of the test.

Acceptance test	Target	Purpose
AT001	Sensor modules.	Ensure module is online and functional
AT002	All hardware modules.	Test for faults and errors.
AT003	Device components.	Ensure selected components are rated for this application.
AT004	Sensor peripheral libraries.	Verify software correctly interfaces with subsystem modules.
AT005	Full system.	Ensuring an accelerated functional cycle meets the timing and sensing requirements.
AT006	Sensor modules.	Calibrate the sensors against a known reference.
AT007	Power subsystem.	Verify the power system meets the user requirements.
AT008	Full system.	Ensure the device can operate in low temperatures.
AT009	Full system.	Ensure device functions in a remote environment.

3.5 Conclusion

To summarise, this chapter outlines the design procedure for identifying critical subsystems and technical specifications to meet the user requirements. This will provide the basis for component and module selection which will be discussed in the next chapter.

Chapter 4

Platform design

In this chapter, the hardware design processes are outlined based on the user requirements and technical specifications given in Chapter 3. This chapter begins with a description of the mechanical features showing how the design accommodates the electronic subsystems while meeting the functional requirements outlined in Table 3.3. This chapter then shows the electronic components selected for each subsystem and how they affect the overall system. Finally, this chapter will conclude with the final design considerations for the buoy.

Son and Vorajee in 2018 performed initial concept work for the buoy, which strongly influenced the current design choices for this device. Furthermore, MacHutchon designed the original buoy stand with modifications by Verrinder to accommodate the buoy. Verrinder designed the physical enclosure and electronic hardware subsystems for the first (V1) and second (V2) versions of the buoy with further design contributions from Jacobson (V1, V2), Cloete (V1) and Pead (V2).

4.1 Mechanical features

The mechanical design for the system falls outside the scope of this project. However, it forms an integral part in protecting the electronics against sea ice dynamics, strong wave activity and freezing. This subsystem consists of two parts:

1. Buoy stand
2. Enclosure

4.1.1 Buoy stand

The principle goal of the stand is to anchor the device to the ice floe and protect it from the harsh environment. A buoy stand was designed by K. MacHutchon with modification from R. Verrinder and was constructed by the University of Cape Town's Mechanical

Engineering Workshop to satisfy this requirement. The stand is shown in Figure 4.1 and is 1.2 m tall with a width of 0.71 m. The stand has a cylindrical cradle at the top where the device will be placed. A screw hole in the side of the cradle allows the buoy to be fastened to the stand to prevent it from falling out during deployment. The base of the stand is pyramid shaped with metal spikes to anchor the system to an ice floe. Due to the height of the stand, the system may be susceptible to tipping. This has been overcome by constructing the base to be heavier than the top thereby lowering the centre of gravity. The stand was originally designed for the Trident buoys. However, this design has been modified by increasing the radius of the housing.

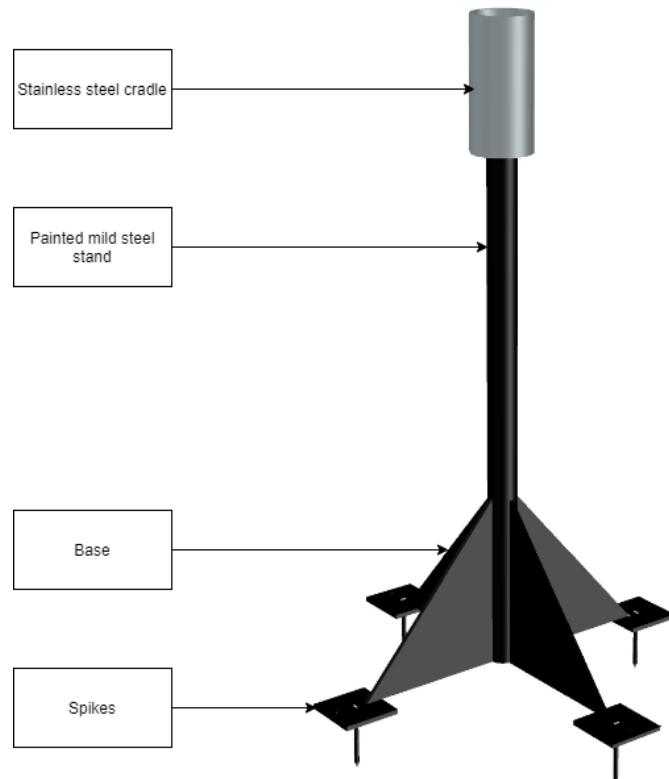


Figure 4.1: Diagram of the buoy stand for the SHARC buoy. The stand consists of stainless steel and painted mild steel to withstand the climate of the Southern Ocean and stop the stand from rusting. A cradle at the top of the stand houses the buoy and secures it to the stand via a screw. The buoy stand elevates the electronics 1 m above the ice to overcome the effects of snow growth and flooding. Finally, spikes at the bottom of the base will secure the stand to the ice floe. Drawn by R. Verrinder.

4.1.2 Enclosure

The second part of the mechanical subsystem is the physical buoy enclosure, which was designed by R. Verrinder. The greatest challenge for designing this system was selecting

a material that was both lightweight and low-temperature resistant. A decision was made to use High-Density Polyethylene (HDPE) which can survive temperatures up to -120° C before becoming too brittle ([drnovska2003surface](#)). The enclosure was designed to fit the housing on the buoy stand while providing ample room for the antennas of the various communication modules. It was split into three parts: A top enclosure, a bottom enclosure and a connector block. A schematic of the enclosure is shown in Figure 4.2

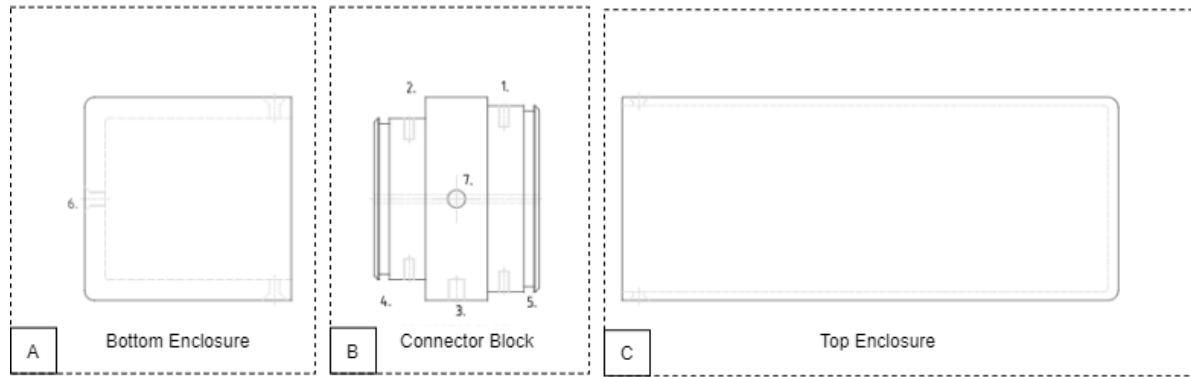


Figure 4.2: 2-D exploded view of the buoy enclosure showing (A) the bottom enclosure for the power module, (B) the connector block which also acts as a base for the electronics and (C) the top enclosure which covers the electronics. Drawn by R. Verrinder.

Table 4.1: Primary measurements of the buoy enclosure taken from the schematic in Appendix B.3.

Component	Dimension	Size [mm]
Top enclosure	Height	240
	Outer diameter	98
	Wall thickness	4
	Base thickness	5
Bottom enclosure	Height	100
	Outer diameter	98
	Wall and base thickness	10
Connector block	Height	80
	Outer diameter	98
	Top inner diameter	89.94
	Bottom inner diameter	77.85
	O-ring thickness	2.6

This design allows for easy access to the electronics as well as separation between the various subsystems. The connector block acts as a connection point for the electronics in version 1 of the buoy. This point of contact was a three-dimensional printed connector

for a vertically mounted printed circuit board (PCB). In version 2 of the buoy, this was replaced by a row of screw holes around the connector block to connect a horizontal stack of customised PCBs. This was found to greatly improve the robustness of the system and prevented components from breaking during transport and deployment. The communication modules, microcontroller and sensors were mounted in the top enclosure while the batteries and power system were placed in the bottom enclosure. The power system was connected to the top enclosure through a drill hole in the connector block. The system was waterproofed by placing two o-rings on either side of the connector block. The top and bottom enclosure are fastened to the connector block using a flat head counter-sunk hex screw. Finally a drill hole in the connector block allowed the system to be secured to the buoy stand preventing it from falling out during deployment.

4.2 Electronics

The electronics for the system refer to the communication subsystems, power electronics, sensors and processors. Due to project time constraints, the approach to developing the platform was to select off-the-shelf components that satisfied the subsystem specifications shown in Table 3.8. Further consideration was given to components that were low power (SP011, SP012) and cost effective (SP013). Additionally, devices with intelligent operations were selected as this would allow us to effectively control the current consumption and operations of the device. These consisted of components with programmable settings such as digital sensors and modems. The following section gives an overview of the selection consideration for each subsystem.

4.2.1 GPS

A u-blox NEO-7M¹ GPS receiver was initially selected during the 2018 design concept phase as it was easy to procure and has a small form factor. The positioning module was designed around a Waveshare² development board which significantly decreases the development time. The board comes with an active patch antenna which has a gain of about 30 dB (**waveshare**). In addition, the component is low power with a relatively fast acquisition time and accuracy. The device also can be configured to output diagnostic information such as dilution of procession with the associated measurement which can provide a greater understanding of satellite connectivity in the region. However, this product has been depreciated by the time the latest buoy was developed. To overcome this, we opted for the u-blox NEO-M9N³ module which had improved performance at a higher cost. Both GPS modules are shown in Figure 4.3.

¹UBLOX_M7N_DATA.

²waveshare.

³UBLOX_M9N_DATA.

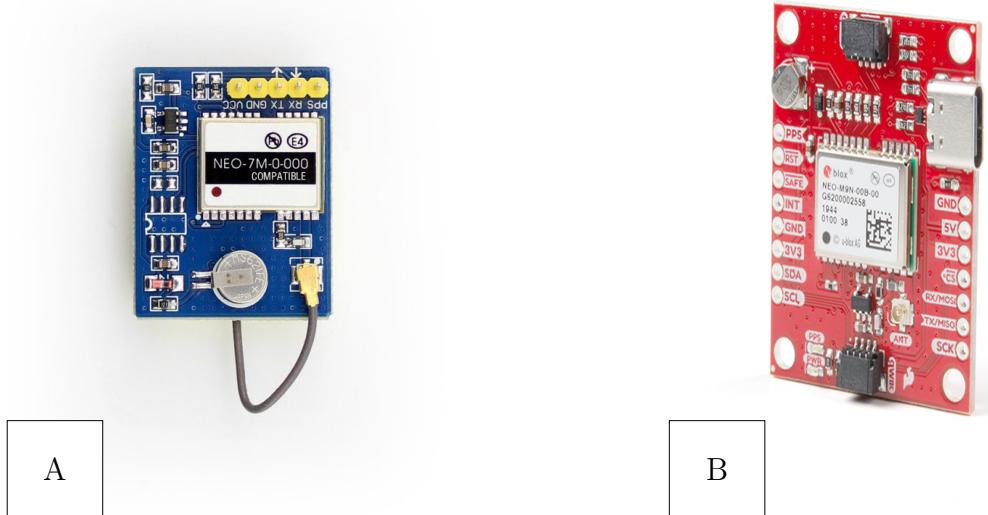


Figure 4.3: The u-blox GPS Modules selected for this project which are: (A) the u-blox NEO-7M on a Waveshare breakout board ([waveshare](#)) and (B) a u-blox NEO-M9N on a Sparkfun breakout board ([UBLOX_M9N_DATA](#)).

Table 4.2 below shows a comparison of the two modules and their key performance parameters.

Table 4.2: Comparison of key parameters between the initial u-blox NEO-7M GNSS module and the updated u-blox NEO-N9M module.

Specification	NEO-7M	EO-M9N
Positional accuracy [m]	2.5	2.0
Communication type	UART I ² C SPI	UART I ² C SPI
Cold-start time [s]	30	26
Supply voltage [V]	1.65 to 3.6	2.7 to 3.6
Active current draw [mA]	32	42
Price ⁴	R269 ⁵	R1,195 ⁶

Shows the u-blox NEO-M9N offers improved performance over the u-blox NEO-7M at a higher cost and higher power consumption. This module comes on a Sparkfun GPS Breakout⁷ - NEO-M9N U.FL (Qwiic) board which optionally comes with an integrated

⁴Price as of March 2021

⁵Source: <https://www.digikey.co.za/>

⁶Source: <https://www.robotics.org.za/>

⁷Available at: <https://learn.sparkfun.com/tutorials/sparkfun-gps-neo-m9n-hookup-guide>

chip antenna. The chip antenna however has a very small gain making it unsuitable to be used for this application. Therefore, an additional antenna was bought.

4.2.2 Iridium

As discussed in Section 3.2, the Iridium modem is critical for ensuring data can be transmitted from remote locations where other forms of communication are unavailable. When selecting a modem, key considerations were given to the physical size, bandwidth as well as coverage. In addition, we require a module that is low powered and cost effective. For this reason, we have selected the Rock7 RockBLOCK 9603([rockblock2019image](#)) which is shown in Figure 4.4 below.



Figure 4.4: A Rock7 RockBLOCK 9603 satellite module containing an Iridium 9603 modem (**9603**) and a coaxial Sub-miniature version A (SMA) connector for an external antenna. Image by [rockblock2019image](#).

This module contains an Iridium 9603 modem on a specially designed power board. The device communicates via Universal Asynchronous Receiver/Transmitter (UART) with the option for flow control. Communication is performed through a ten-pin Molex PicoBlade connector. The module contains four communications pins, one digital input pin and two digital output pins. Power is supplied either through a 5 V pin or 3.3V pin in addition to the ground pin. A brief description of the pinout is given in Table 4.3.

Table 4.3: Pinout for the Rock7 RockBLOCK 9603 Iridium modem.

Pin number	Label	Pin description
1	RXD	UART Output Pin
2	CTS	Flow control clear to send
3	RTS	Flow control request to send
4	NetAv	Network available
5	RI	Ring inidicator
6	TXD	UART input pin
7	OnOff	Sleep control
8	5V	5 V max supply pin
9	Li-Ion	3.7 V max supply pin
10	GND	Ground

The device communicates via UART through the RXD and TXD pins. Additional flow control pins Clear To Send (CTS) and Request To Send (RTS) pins are available in the interface. However, these are optional. The OnOff pin can be used to put the device to sleep which significantly improves power performance. Finally, the NetAv and Ring Indicator are digital output pins that can be used to indicate whether there is sufficient signal to transmit as well as to notify when a message is waiting to be downloaded respectively. Finally, the key characteristics for the device are shown in Table 4.4 to 4.6.

Table 4.4: Table showing the mechanical feature parameters of the Rock7 RockBLOCK 9603 module (**9603**).

Mechanical features	
Antenna	Patch or external SMA
Temperature rating [° C]	-40 to 85
Dimensions [mm]	45.0 × 45.0 × 15.0
Price ⁸	R3,571 ⁹

⁸Cost as of March 2021 from ([rockblock2019image](#))

⁹GBP 1 = ZAR 20.41

Table 4.5: Table showing the power consumption parameters of the Rock7 RockBLOCK 9603 module (**9603**).

Power characteristics	
Supply Voltage [V]	5
	3.3
Start-up current [mA]	450
Active current [mA]	100
Sleep current [μ A]	200

Table 4.6: Table showing the communication parameters of the Rock7 RockBLOCK 9603 module (**9603**).

Communication	
Baud rate [bits/s]	19200
Data bits	8
Stop bits	1
Parity	none
Maximum upload size [bytes]	340
Maximum download size [bytes]	270

4.2.3 Environmental sensors

Two versions of the buoy were developed from 2019 to 2020 with different sensing capabilities. The first version consisted of a Maxim Integrated DS18B20 temperature sensor (**DS18B20manual**). This is a low cost digital silicon temperature sensor with a small form factor that uses one-wire interface. However, in version two, this device was replaced with the Bosch Sensortech BMP280 sensor (**BMP280_Datasheet**). The BMP280 sensor featured pressure sensing as well as temperature sensing as well as a programmable interface. Furthermore, the BMP280 can optionally be ordered on an Adafruit BMP breakout board (**BMP280breakout**) whereas the DS18B20 arrives as a stand-alone through-hole component that requires an additional circuit. The devices are shown in Figure 4.5 while a comparison of the features of each device is given in Table 4.7.

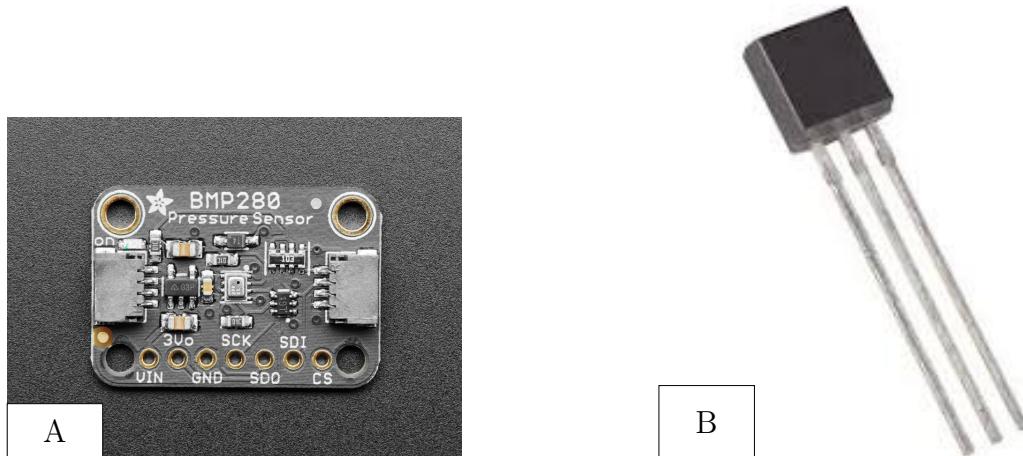


Figure 4.5: Environmental sensing devices selected for the buoy. (A) The Bosch Sensortech BMP280 digital pressure and temperature sensor on an Adafruit sensor breakout board (**BMP280breakout**) and (B) a Maxim Integrated DS18B20 in a through hole TO-226-3 package (**DS18B20manual**).

Table 4.7: Comparison of performance between the Bosch Sensortech BMP280 and Maxim Integrated DS18B20 environmental sensors. The term "Not applicable" or N/A is given where a parameter does not apply to a device.

	BMP280	DS18B20
Temperature range [° C]	-40 to 85	-55 to 125
Temperature sensor accuracy ¹⁰ [° C]	±1	± 1
Pressure range [hPa]	300 to 1100	N/A
Pressure sensor accuracy ¹¹ [hPa]	±1.7	N/A
Price ¹²	R47 ¹³	R77 ¹⁴

Table 4.7 shows a comparison between the Bosch Sensortech BMP280 and Maxim DS18B20 performance characteristics. The temperature ranges of both sensors are capable of measuring subzero temperatures at an accuracy of ±1° C which, is required to survive in the Southern Ocean climate as discussed in Chapter 2. Only the BMP280 can measure temperature and pressure, whereas the DS18B20 can only measure temperature. Furthermore, the BMP280 cost R47 cheaper than the DS18B20 at R77. However, this is for the standalone BMP280. This device can be, optionally, ordered on an Adafruit breakout board. However, this increases the price to R218¹⁵. Therefore, the BMP280 was selected for version 2 as it contains more environmental sensing features for a lower price. The power consumption characteristics of each device are shown in Table 4.8.

¹⁰At temperatures below 0°

¹¹At temperatures below 0°

¹²As of March 2021

¹³Source: <https://www.digikey.co.za/short/80dv2nhm>

¹⁴Source: <https://www.digikey.co.za/short/mqb0qm4j>

¹⁵Source: <https://www.digikey.co.za/short/qbn473mr>

Table 4.8: Comparison between supply voltage and current draw of the BMP280 and DS18B20.

	BMP280	DS18B20
Supply voltage [V]	3.0 to 5.5	1.7 to 3.6
Sleep current [μ A]	0.75	0.3
Active current [μ A]	4.2	1500

Table 4.8 shows the power performance characteristics of the BMP280 and the DS18B20. The DS18B20 has a lower operating voltage requirement with a smaller sleep current draw than the BMP280. However, the DS18B20 has a much higher active current. Furthermore, both devices can operate at 3.3 V. Therefore, the BMP280 will have the lowest active power consumption during operation.

In conclusion, the BMP280 offers both temperature and pressure sensing at a lower price than the DS18B20 that can only perform temperature measurements. Therefore the BMP280 was selected for version 2 of the buoy to increase environmental sensing capabilities.

4.2.4 Power monitoring sensors

Finally, a digital sensor for power monitoring was selected to provide constant feedback on the status of the power system. This device is used to monitor the battery voltage and current to keep track of the buoys power consumption. To achieve this, the Texas Instruments INA219A I²C power monitor (**INA219**) was selected. This device has a high reported accuracy of 1% over a full temperature range and is fully programmable. The device communicates via I²C with 16-bit registers storing values for current (mA), voltage (V) and power (mW). The device is extremely low power with a high voltage measurement range and on-board calibration features. The integrated circuit (IC) is shown in Figure 4.6 while the key performance characteristics are shown in Table 4.9.

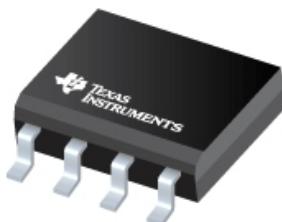


Figure 4.6: 3-D render¹⁶ of the Texas Instruments INA219 current/power monitor I²C chip in an 8-pin small outline integrated circuit (SOIC) package (**INA219**).

¹⁶Source: <https://www.ti.com/product/INA219>

Table 4.9: Performance specifications for the INA219 current monitor chip.

Performance characteristics	
Operating temperature [° C]	-40 to 125
V _{shunt} range [mV]	40 to 320
V _{bus} range [V]	0 to 16 0 to 32
ADC resolution	12-bits
Measurement error	±1%
Price ¹⁷	R16 ¹⁸

Table 4.10: Power specifications for the INA219 current monitor chip.

Power characteristics	
Supply voltage [V]	3.3 V to 5 V
Quiescent current [mA]	0.7 mA to 1mA
Standby current [μ A]	6 to 15

Table 4.9 above shows that the device is capable of measuring voltage, current and power at low temperatures up to -40° C thereby making it a suitable choice for active power monitoring of the device in the polar climate.

4.2.5 Inertial measurement unit

The TDK InvenSense MPU6050(**mpu6050**) is a 6-axis inertial measurement unit IMU that measures the acceleration and rotational velocity of 3 axes respectively. This component has a small form factor, low power and is fully programmable allowing the device to operate in different modes thereby optimising the data flow to and from the device. While the device does not contain a magnetometer, this is not an issue since the region suffers greatly from magnetic distortion (**kohout2015device**) thereby rendering all magnetic readings to be unreliable. In addition, the acceleration of waves can be defined by Stoke's upper limit as 0.5 g for a non breaking wave (**kohout2015device**). The device has a programmable full scale range for both the accelerometer and gyroscope. It contains an infinite impulse response (IIR) filter and on-board self testing for added robustness and data integrity thereby making it the ideal device for this application. The device can be, optionally, ordered on a SparkFun Triple Axis Accelerometer and Gyro Breakout - MPU-6050 which is shown in Figure 4.7. The key parameters for the device are shown in Table 4.11 to 4.13.

¹⁷Price as of March 2021

¹⁸Source: <https://www.digikey.co.za/short/255dtrfm>

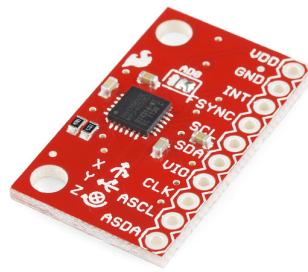


Figure 4.7: A TDK Invensense MPU6050 6 degrees of freedom inertial measurement unit on a Sparkfun development board. The board allows for fast prototyping with communication performed via I²C. Image source: ([mpu6050breakout](#)).

Table 4.11: Performance characteristics of the TDK Invensense MPU6050 6-axis inertial measurement unit (IMU) showing the accelerometer performance.

Accelerometer	
Full scale resolution [g]	±2 to ± 8
Sensitivity [$\mu\text{g}/\text{LSB}$]	61.17 to 488.281
Sample rate [Hz]	4 to 1000
Noise performance [$\mu\text{g}/\sqrt{\text{Hz}}$]	400

Table 4.12: Performance characteristics of the TDK Invensense MPU6050 6-axis inertial measurement unit (IMU) showing the gyroscope performance.

Gyroscope	
Full scale resolution [$^{\circ}/\text{s}$]	±250 to ± 2000
Sensitivity [$(\mu^{\circ}/\text{s})/\text{LSB}^{-1}$]	7.63 to 60.98
Sample rate [Hz]	4 to 8000
Noise performance [$(\mu^{\circ}/\text{s})/\sqrt{\text{Hz}}$]	0.005

Table 4.13: Performance characteristics of the TDK Invensense MPU6050 6-axis inertial measurement unit (IMU) showing the device temperature rating, power consumption and price.

Device characteristics	
Temperature range [° C]	-40 to 85
Low pass filter range [Hz]	5 to 256
Supply voltage [V]	2.4 to 3.5
Maximum active current [mA]	3.9
Low power mode current ¹⁹ [μ A]:	20
Price ²⁰ :	R110 ²¹

Table 4.13 shows that the TDK InvenSense MPU6050 IMU has the capabilities to measure acceleration and angular velocity necessary to perform wave calculations as described in Chapter 2.3.2. The device can operate with a voltage supply of 2.4 V to 3.5 V. Furthermore, the lowest sample rate of the IMU is 4 Hz which is well above the Nyquist sampling criteria for open ocean waves as discussed by **earle1996nondirectional** in Subsection 2.3.2. This device is also capable of operating in low temperatures up to -40° C which, is suitable for the Southern Ocean climate. The chip can, optionally, arrive on a SparkFun Triple Axis Accelerometer and Gyro Breakout - MPU-6050. However, this board has an increased cost of R437²² compared to R110 for a standalone chip. For this reason, the Sparkfun board was used to develop and test the firmware for the device. Verrinder and Pead designed the circuit for the MPU6050 for the final version of the buoy.

4.2.6 Memory

Physical memory is an important feature in the device as it allows for permanent storage of data during the life cycle of buoy. Having the device in various sleep modes may result in lost data if the data are stored in non-volatile memory.

Flash chips were selected as a permanent solution. A total of four Adesto Technologies AT45DB641E SPI serial flash chips (**AT45DB641E**) were selected and mounted on a PCB directly interfacing with the system. Each chip can hold up to 64 megabits of data. Data can be read/written at speeds of up to 85 MHz or 15 MHz in low power mode. These chips are low power with high data retention requiring a supply voltage of 1.7 V to 3.6 V and draws a maximum of 11 mA in active read mode thereby making it one of the lowest power consumption components in the system. In addition, the device comes with two 256 byte buffers that can store data while a read/write operation is taking place. Memory is Organised into sectors (2 to 256 KB long), blocks (2 KB long) and pages (256

¹⁹for output data rate (ODR) < 5 Hz

²⁰Price as of March 2021

²¹Source: <https://www.digikey.co.za/short/rh95nm3v>

²²Source: <https://www.digikey.co.za/short/bpqqdjdv>

bytes) with write, read and erase options at each level. The device comes in a surface mount package shown in Figure 4.8. Table 4.14 shows key performance characteristics.



Figure 4.8: The Adesto Technologies SPI serial flash chips in a surface mount package. Image source: ([flashchipimage](#)).

Table 4.14: Key performance characteristics for the AT45DB641E flash chips (**AT45DB641E**) including temperature rating, power consumption, storage capacity and price.

Performance characteristics	
Operating temperature [° C]	-40 to 85
Storage capacity [Mbit]	64
Supply voltage [V]	1.7 to 3.6
Standby current [μ A]	45
Active current [mA]	22
Price	R65 ²³

4.2.7 Processor

For the processor, a single processing unit was selected to reduce complexity of the system. However, to satisfy the requirements for the buoy, a processor must be selected with sufficient peripheral ports to handle communication from all sensors, communication modules and memory banks. In addition, there should be sufficient digital input and output pins to control the sensors and provide feedback. The communication peripheral requirements are condensed into Table 4.15:

²³Source: <https://za.rs-online.com/>

Table 4.15: Type and number of communication ports to facilitate communication among all the external modules.

Peripheral name	Qty
UART	2
I ² C	2
SPI	2
Digital pins	11

From Subsections 4.2.1 to 4.2.6, the modules given have a data requirement and a communication requirement. Some modules, such as the RockBLOCK 9603 modem, require additional digital pins. Furthermore, all sensors output data at different resolutions. The BMP280 and MPU6050 have an 8-bit resolution, while the INA219 has a 16-bit resolution. Therefore, the selected processor needs to have sufficient width to accommodate incoming data of varying sizes while containing sufficient peripherals to interface with and control the subsystems. For this reason, a 32-bit microcontroller was selected for the processing system. During the prototyping period, three processors were selected from the STMicroelectronics STM32 range of microcontrollers for different versions of the buoy. These microcontrollers are shown in Figure 4.9.

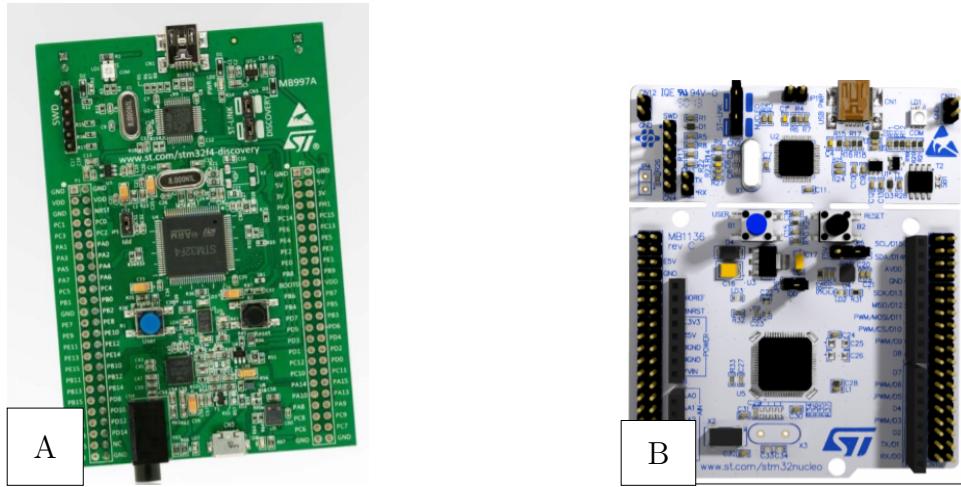


Figure 4.9: The 32-bit microcontrollers used for the different versions of SHARC buoy. (A) is the STMicroelectronics STM32F407VTG6 (**stm32disc**) on a 100-pin Discovery development board which was used for version one. (B) is the 64-pin STMicroelectronics Nucleo development board which initially contained an STMicroelectronics STM32F446RE (**stm32nucleo**) for version 2. However, it was replaced with an STMicroelectronics STM32L476RG (**stm32l4**) for the final version.

The first version of the buoy contained the STMicroelectronics STM32F407VGT6 (**stm32disc**) which is available on a 100-pin Discovery development board. However, this device was found to have more peripherals than necessary and had a large power requirement. Therefore the device was replaced by the STMicroelectronics STM32F446RE (**stm32nucleo**).

This device came on the STMicroelectronics Nucleo development board and had reduced peripherals and more optimal performance. The final processor selected was the STMicroelectronics STM32L476RG (**stm32l4**). Development boards were selected for the prototyping phase as they were quick to set up and program. In Figure 4.9, both the Discovery and the Nucleo have pinheaders allowing for quick interfacing with the peripherals of the microcontroller. Which matched the STM32F446RE in pinout and number of peripherals. However, it was more optimised for low power operation. This microcontroller had additional wake-up pins and lower current consumption. Furthermore, the Nucleo boards have an on-board STLink debugger to program the microcontroller. However, this can be removed to reduce the physical size of the module. The STM32L476RG can be configured to detect brownout events and low power events. This provides critical feedback regarding the device's performance. Some key performance parameters of the STM32L476RE are shown in Table 4.16 to 4.18 .

Table 4.16: Electrical parameters and power consumption characteristics for the STM32L476RG microcontroller.

Electrical characteristics	
Input voltage [V]	1.7 to 3.6
Active current [$\mu\text{A}/\text{Hz}$]	100
Shutdown mode current [nA]	30
Standby mode current [nA]	420
V_{brownout} threshold [V]	1.7 to 2.9

Table 4.17: Processing characteristics and parameters for the STM32L476RG microcontroller.

Processor characteristics	
Processor	Arm Cortex-M4
Processor size	32-bit
Float representation	Hardware FPU
Flash memory size [MB]	1
Volatile memory (RAM) size [KB]	128
	Low speed external (LSE)
	High speed external (HSE)
Input clock sources	Low speed internal (LSI)
	Mixed speed internal (MSI)
	High speed internal (HSI)
Input clock frequency [MHz]	4 to 80
Dhrystone Benchmark:	1.25 DMIPS/Hz

Table 4.18: Communication ports and specifications for the STM32L476RG microcontroller.

Communication ports	
Total communication ports	20
UART ports	5
I ² C ports	3
SPI Ports	3

Tables 4.16 to 4.18 show the performance characteristics of the STM32L476RG. It features seven general purpose timers as well as two advanced timers and two low power timers. In addition, the device has five wake up pins which allow the device to be woken up from deep sleep (shutdown) via an external source. The device is capable of digital signal processing (DSP) using external libraries provided by the manufacturer. This allows the device to perform wave spectra calculations as described by **kuik1988method** and **earle1996nondirectional**. Furthermore, the STM32L4 Nucleo board contains an onboard real-time clock (RTC) with a 32.768 kHz LSE crystal to provide an accurate reference for time-based applications thereby making it the ideal component to be a processor for the buoy.

4.2.8 Power electronics

Based on the aforementioned hardware selection, the following power requirements are outlined in Table 4.19.

Table 4.19: Current consumption of various components as well as the estimated maximum possible current draw.

Device name	Qty	Supply voltage [V]	Opperating current [mA]
u-blox NEO-M9N	1	3.3	42
Rock7 RockBLOCK 9603	1	5	450
Bosch Sensortech BMP280	1	3.3	0.0042
Texas Instruments INA219A	1	V_{Bat} ²⁴	1
TDK Invensense MPU6050	1	3.3	3.9
Adesto Technologies AT45DB641E	4	3.3	88
STMicroelectronics STM32L476RG	1	5	2.6
		total [mA]	587.50

From Table 4.19, a maximum current draw of 587.50 mA can be expected if all the sensors are running concurrently. The largest consumer of power is the RockBLOCK 9603 which

²⁴INA219A supply voltage is fixed to the reference power supply (**INA219**)

can draw up to 450 mA when charging. Therefore, the power supply needs to be to supply at least 500 mA during startup. Current can be conserved by placing the devices into sleep mode which further reduces the current consumption from the batteries. These techniques will be discussed in Chapter 5. Finally, by only turning the components on when required, more power can be conserved.

Therefore, a regulator is required to supply the required current at a constant voltage while being able to stand the drastic changes in current consumption. A decision was made to use a 5 V low dropout regulator (LDO) to supply the 5 V components directly. The 3.3 V components are powered through the STMicroelectronics Nucleo-L476RG on-board 3.3 V regulator. The 5 V LDO is a Texas Instruments LP3876 7 V LDO (**LP3876**) which is shown in Figure 4.10.

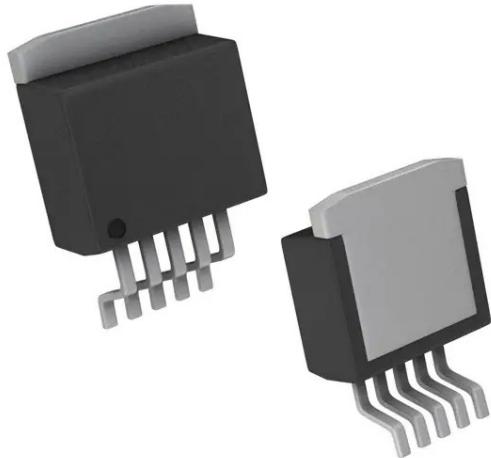


Figure 4.10: 3-D render²⁵ of the Texas Instruments LP3876 linear low drop out regulator in a 5-Pin DDPAK/TO-263 surface mount package (**LP3876**).

This regulator is capable of supplying up to 3 A. The device has a quick response to step changes and an adjustable output voltage thereby making it the ideal device to supply power. 10 μ C tantalum capacitors are used as decoupling capacitors to filter the power source as these capacitors have excellent robustness and transient responses especially at low temperatures. Some key characteristics for the device are given in Table 4.20.

Table 4.20: Key performance characteristics for the Texas Instruments LP3876 linear low dropout regulator (**LP3876**).

Performance characteristics	
Input voltage [V]	2.5 to 7.0
Voltage regulation over current	0.14%
Dropout voltage at 3 A [V]	0.8 to 1.2
Quiescent current at 3 A [mA]	14
Temperature Range [° C]	-40 to 125
Price ²⁶	R86 ²⁷

The LDO was placed on a customised PCB designed by Verrinder with the Texas Instruments INA219A current sensor as well as an indicator LED to show that the batteries have sufficient charge. The power board was supplied by 3.6 V C-cell LiSOCl₂ batteries. These batteries have ideal low temperature characteristics as well as a high capacity. Two cells were placed in series to create a 7.2 V power source which was placed in parallel with another 7.2 V array to increase the capacity. The batteries, battery holders and power board are connected to form a single subsystem which was placed in the bottom enclosure and connected to the microcontroller via a 7-pin cable.

4.3 Final assembly

The final electronics choice and configurations are shown in Figure 4.11.

²⁴Image source:<https://www.digikey.co.za/short/d95btr8d>

²⁵Image source:<https://www.digikey.co.za/short/d95btr8d>

²⁶Price as of February 2021

²⁷Source: <https://www.digikey.co.za/short/d95btr8d>

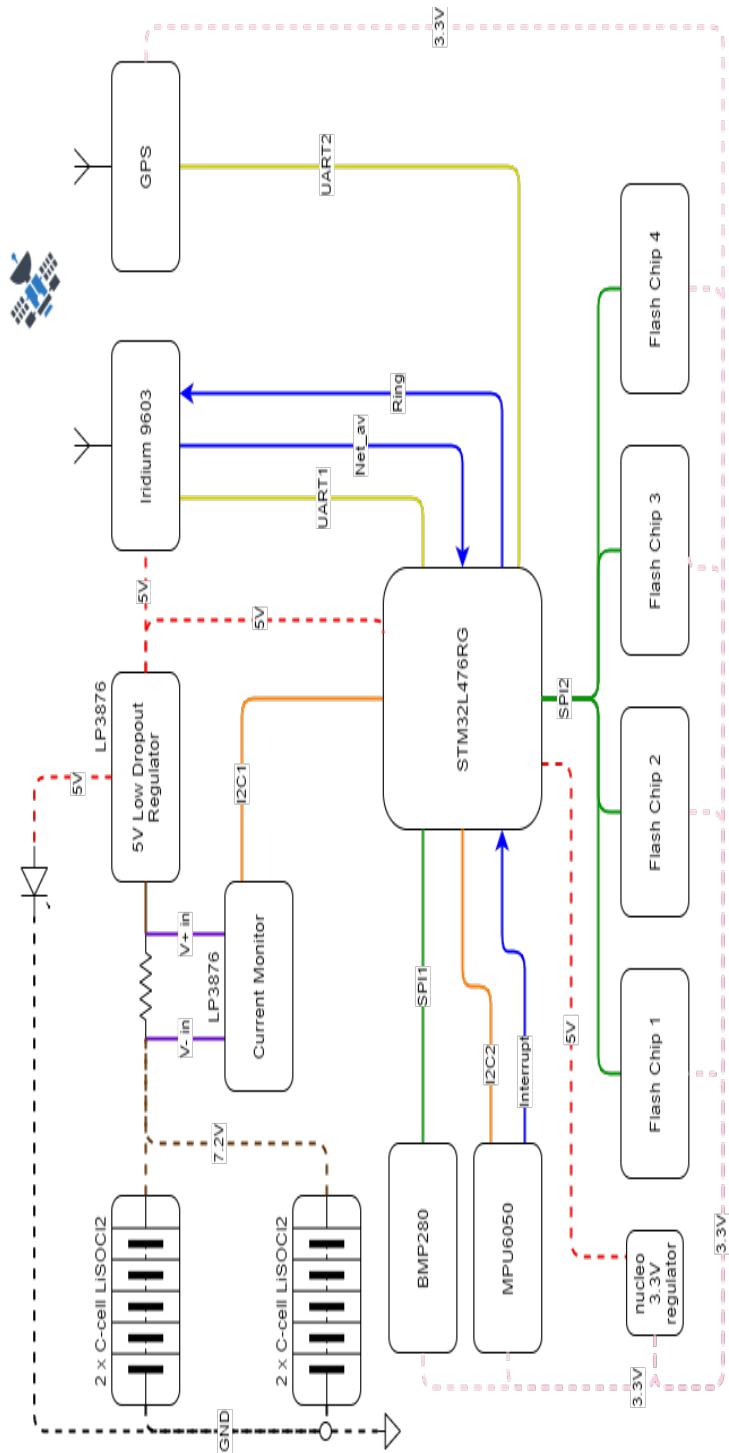


Figure 4.11: Simplified schematic of the final version of SHARC buoy showing (dash) power supply, (solid) communication and digital (arrows) connections. Also shown in the figure are lines indicating: (black) ground, (red) 5 V power, (pink) 3.3 V power, (Purple) analog inputs, (orange) I²C, (yellow) UART, (green) SPI and (blue) digital input/output.

A final costing of the system is provided in Table 4.21.

Table 4.21: Approximate procurement cost for a single SHARC buoy node.

Component name	Qty	Unit cost	Total
Buoy enclosure and stand	1	R1,206.84	R1,206.84
u-blox Neo-M9N	1	R1,195.45	R1,195.45
Rock7 Rockblock 9603	1	R3,278.144	R3,278.144
M1621HCT Helical Antenna	1	R1,411.15	R1,411.15
BMP280	1	R46.00	R46.00
INA219A	1	R17.77	R17.77
MPU6050	1	R40.00	R40.00
AT45DB641E	4	R65.307	R261.229
Nucleo-l476RG	1	R215.98	R215.98
Fanso C-cell 9000mAh Battery	4	R101.81	R407.24
BHC-2ND Battery Holder	4	R61.87	R247.48
LP3876 5V regulator	1	R95.19	R95.19
Wiring and Connectors	-	R136.46	R136.46
Grand total			R8,421.13

Customised PCBs were designed to connect the various subsystems together. The device was kept modular by separating PCBs and grouping devices by functionality. A circuit board was created for the LP3876 LDO and INA219 current sensor which was affixed to four C-cell battery holders. The battery holders have leads which were arranged in a 2-series, 2-parallel configuration.

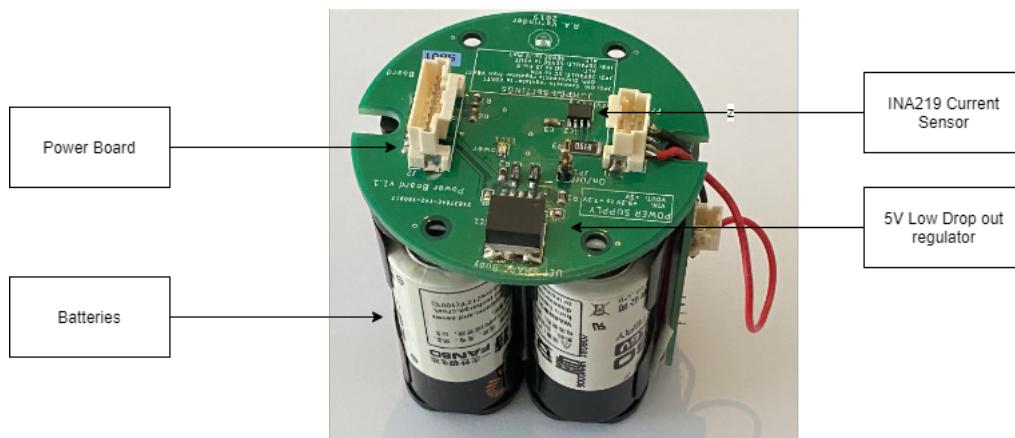


Figure 4.12: Power module for the SHARC BUOY. A custom PCB with a low dropout regulator and current sensor connected to a battery pack.

This module was placed in the bottom enclosure and fastened to the connector block

using a hex screw. A customised 7-pin Molex DuraClik cable connects the module to the modules in the top enclosure. A main connector board was developed with Molex DuraClik connections for each of the aforementioned devices. The board contains two 2x16 female header rows to fit the morpho connectors of the Nucleo-L476RG development board. Two more disc-shaped PCBs were developed with the first for a communication board containing a 4-pin female header to connect the u-blox NEO GPS module and two brackets to mount the RockBLOCK 9603 Iridium module vertically. A helical antenna connects to an SMA antenna on the Iridium module. The second board was developed as a sensor board for the IMU and environmental sensor. The boards were connected in a stack configuration and fastened to the connector block using M6 metal hex spaces with the communication board being placed at the top for direct line of sight with satellites. The environmental board was secured to the base of the connector block with the BMP280 placed face-down over a hole drilled through the connector block allowing it to interface with the environment. Figure 4.13 shows the configuration of the PCBs in the top enclosure of the buoy.

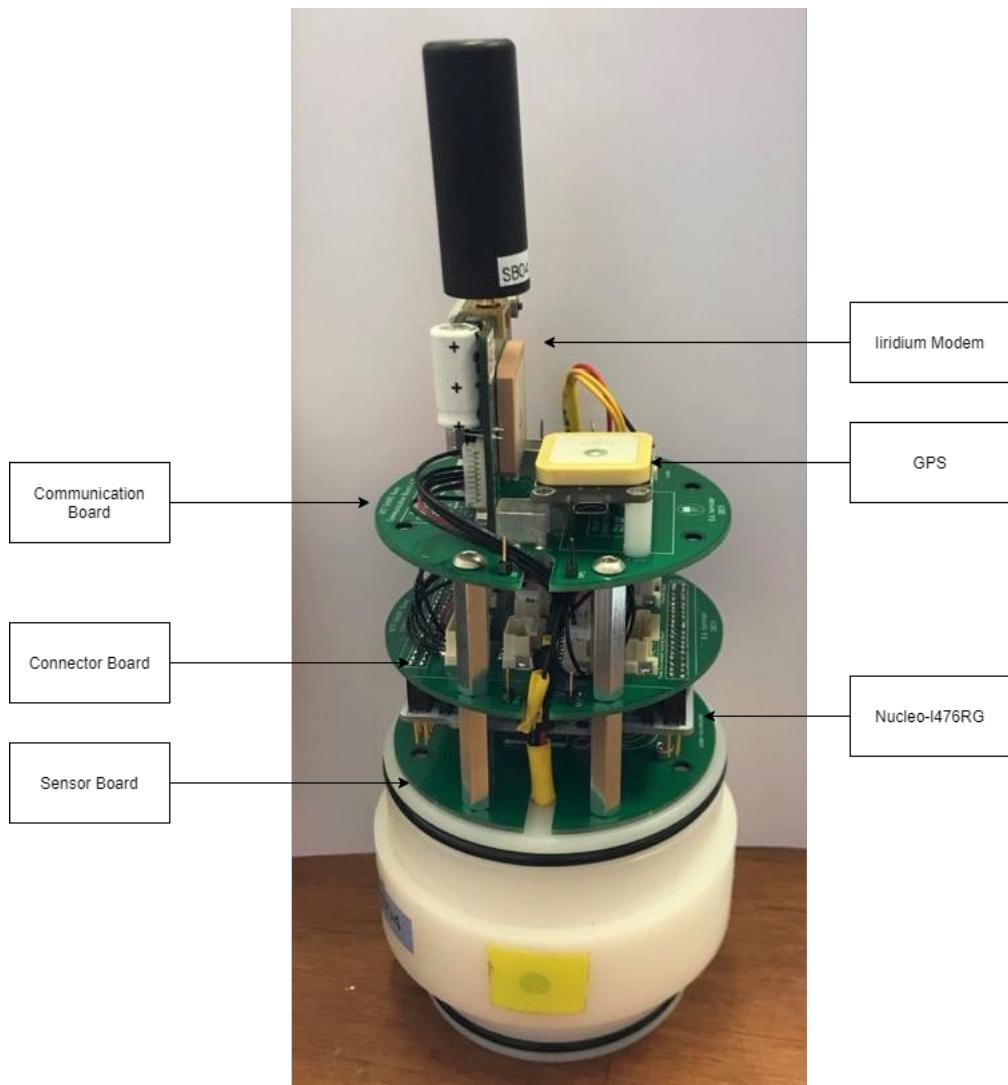


Figure 4.13: Electronic PCB stack for the top module consisting of connector board, microcontroller board and sensor board attached to the connector block.

This configuration greatly increases the robustness of the electronics and can overcome breaking caused by poor handling or improper deployment. The top enclosure is placed over the electronics and fastened to the connector block using hex screws. Finally, the system is placed in the stand housing and secured using another hex screw. Final assembly of the buoy can be seen in Figure 4.11.

4.4 Conclusion

In conclusion, this chapter outlined the construction of the SHARC buoy remote sensing device. Electronic components were selected and arranged into a system and connected together on customised PCB. This platform was then used to develop the firmware which will be discussed in the next chapter.

Chapter 5

Software Design

This chapter outlines the software design methodology for the SHARC buoy based on the hardware design and components selections in Chapter 4. This chapter describes the firmware written for SHARC buoy version 2. This version contains the STMicroelectronics STM32L476RG 32-bit microcontroller, (**stm32l4**) which contains the Arm Cortex-M4 processor (**ARMprocessor**); a low powered 32-bit microcontroller, based on the Armv7E-M central processing unit (CPU) architecture and Harvard bus matrix architecture. A high-level overview of this microprocessor is shown in Figure 5.1.

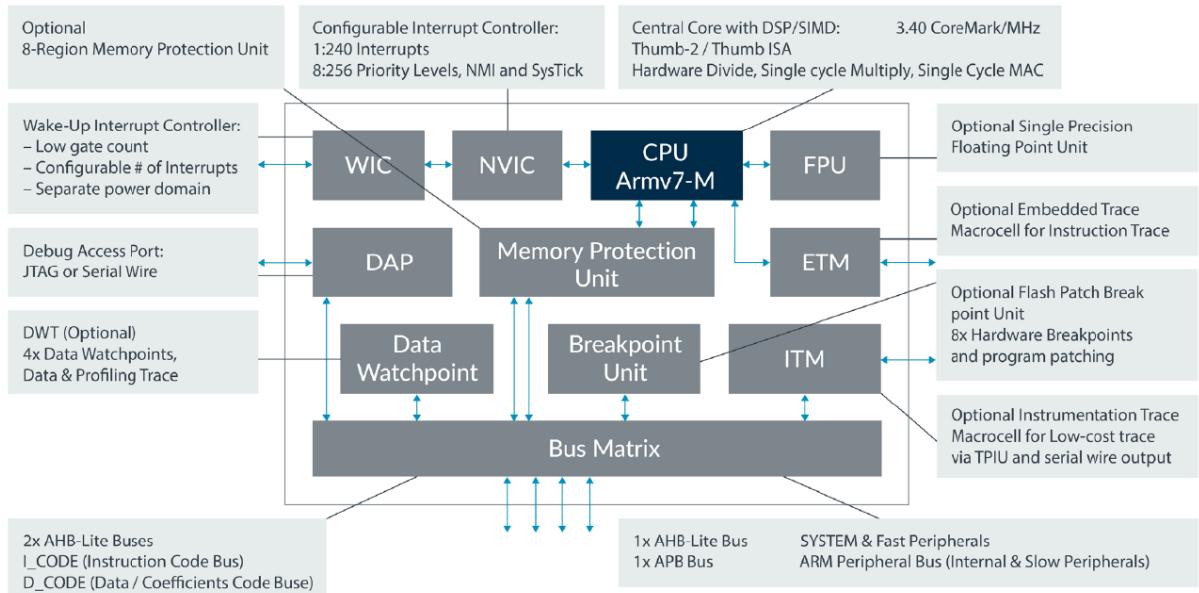


Figure 5.1: Block diagram of the Advanced RISC Machines (Arm) Cortex-M4 microprocessor. This device features a Harvard bus matrix and Armv7E-M central processing unit. The processor also contains a hardware floating point unit (FPU), nested vector interrupt controller(NVIC) for controlling interrupts and a wakeup interrupt controller (WIC) for low power operational control. This processor can be programmed through serial wire via STLink or via JTAG. Image source: (**ARMprocessor**).

The software was compartmentalised to improve the modularisation of the firmware.

This accommodated changes to the hardware subsystems during the development stages without writing entirely new software. Compartmentalisation was also necessary to accommodate microcontroller changes as the new microcontroller had a different pin map and processor architecture to the previous version of the buoy. The project was initially developed in C using the STMicroelectronics standard peripheral libraries (SPL). However, in 2019, these libraries had depreciated and were replaced with the STMicroelectronics Hardware Abstractionion Layer (HAL) libraries. This change also resulted in a change in firmware architecture due to a difference in the software project structure. Therefore, this section will focus solely on the firmware design for version 2 of the buoy.

This chapter begins with an overview of the development environment, tools, and libraries used to develop the firmware. An overview of the firmware architecture is presented, showing the layers and sections of the firmware. The timing requirements are presented, followed by the configuration parameters for the system clock real-time clock (RTC). These are followed by a discussion of the effects of these configurations on the overall system. Next, the software power optimisation techniques are presented, showing the selection of a suitable power mode and the impact during runtime. The main software is presented, showing careful consideration of timing requirements and the synchronous, and asynchronous behaviour during run time. The chapter closes with the systems data requirements followed by the techniques used to manage data between subsystems and data transfer from the device to a user.

5.1 Software architecture

The STMicroelectronic STM32 microcontrollers do not come loaded with any operating system. Therefore, firmware development had to occur at the base level. In addition, the firmware had to be tailored to the specific microcontroller architecture. A decision was made to write the firmware in C as it allowed for higher-level code to be implemented while still optimising for size and speed on the device. The Atollic Truestudio integrated development environment (IDE) allowed for development to take place in C/C++. This development platform includes an integrated Arm development tool chain and a GCC compiler allowing for easy code compilation and device programming via a USB cable. STMicroelectronics also provides a set of driver files and tools to assist with peripheral initialisation which reduced development time.

The firmware was designed using a top down approach. The overall system was decomposed into three distinct layers as shown in Figure 5.2.

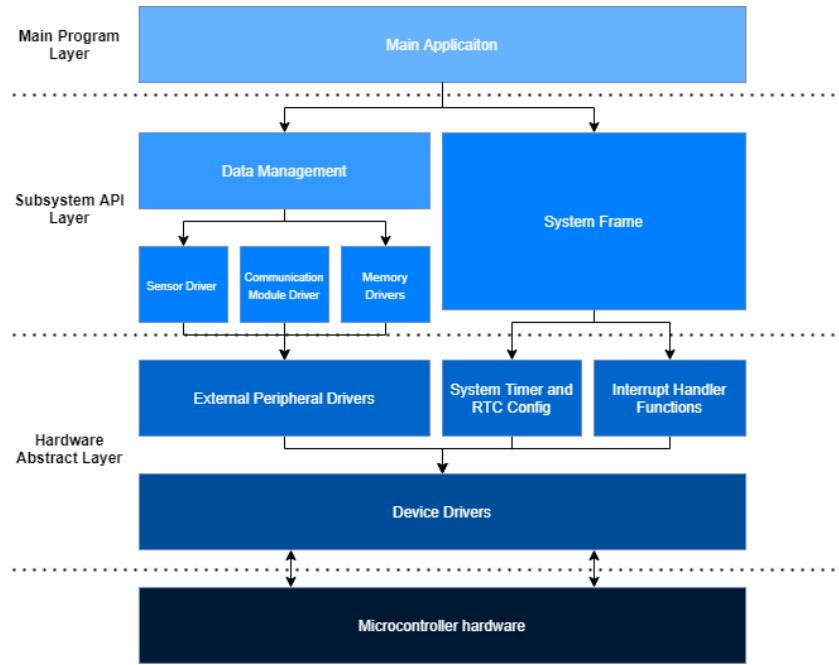


Figure 5.2: Diagram showing the decomposition of the overall firmware into distinct layers and the relationship between each part.

Figure 5.2 provides the basis for the firmware development. The main application is built on supporting layers with each layer providing an interface for a level closer to the hardware. These layers are discussed in Subsection 5.1.1 to 5.1.3.

5.1.1 Hardware abstraction layer

The hardware abstraction layer (HAL) consists of the driver files used to initialise and control the hardware of the microcontroller. This layer is processor specific and therefore needs to be tailored to the architecture of the selected microcontroller. This layer comes pre-written as a set of library files provided by STMicroelectronics which were included in the Atollic Truestudio IDE. The STMicroelectronics STM32L4 Standard Peripheral Library (SPL) was first used as the hardware abstraction layer in version one of the firmware. However, in 2019, the library had depreciated and was replaced with the STMicroelectronics Hardware Abstraction Layer (HAL) libraries. HAL libraries were used to form the foundation of the code as it allows for the code functionality to run independently from the hardware architecture. Should a new microcontroller be required, the HAL library simply needs to be replaced. Therefore allowing the firmware to maintain modularisation and increase portability. In addition, the HAL libraries offer robust error checking and flagging. If a peripheral fails at any point during run time, the libraries provide handlers and flag signaling to handle the error.

5.1.2 Subsystem application peripheral interface (API) layer

Next, hardware-specific driver files were written for each subsystem. These files formed the subsystem application peripheral interface (API) layer shown in Figure 5.2 which interfaces with the hardware peripherals through the HAL layer thereby reducing dependencies on the hardware. In addition, these driver files were created to abstract the initialisation and configuration process of each peripheral as well as the hardware routines that occur. Some external modules required the use of more than one peripheral such as timer channels for input capture or general purpose input/output (GPIO) pins for external interrupt detection. Finally, these drivers are critical for managing the flow of data too and from the module. Furthermore, these files contain functions that interpret incoming data bytes and convert them to the relevant data type.

5.1.3 Main program layer

The state machine layer is the highest level layer of the firmware architecture shown in Figure 5.2. In this layer the configuration files and other libraries are synthesised and sequenced into one main program. This program is responsible for controlling sensors and implementing data processing functions. This layer interfaces with sensor APIs through routines. Functions written in this layer are designed to control the behaviour of peripherals. For example, implementing a routine to initialise a sensor, sample the sensor and then turn it off. Further details of the main program will be discussed in Section 5.2.

5.2 Project structure

In this section, the tools used to develop the software are given. A description of the project files and project organisation for the main program, auxiliary files and libraries is provided.

5.2.1 Project tools and files

The project was set up using STMicroelectronics STM32CubeMX. This is a tool for creating an embedded firmware project that generates platform-specific initialisation code and creates a project folder with peripheral initialisation and handling functions. This tool was used to quickly set up a project as well as configure the clock settings and include the correct HAL libraries. The project code files are organised into the following folders:

1. Drivers
2. Source (src)
3. Start up

The drivers folder contains the HAL and Cortex microcontroller software interface standard (CMSIS) libraries for the device. These are a set of processor-specific libraries for the Arm cortex processor which provide a higher level interface between the microprocessor hardware and the firmware (see Figure 5.2). The source folder contains the main.c file which acts as the entry point for the program to run. The start up file contains assembly code that specifies the vector table, hard fault/reset handler entry Points as well as the entry point for the main code.

5.2.2 Program startup and entry

Program execution begins with the file *startup_stm32l476xx.s*. This is an assembly file that initialises the volatile (RAM) and non volatile (FLASH) memories and the main stack pointer (MSP) in the CPU. The file also contains:

1. Reset handler
2. Hard fault handler
3. Non maskable interrupt (NMI) handler
4. Interrupt vector table

Resets, hard faults, watchdog events and non maskable interrupts are critical interrupt events that cause the main program to halt execution. Hard faults and non maskable interrupts come from run time exceptions that result in critical failures while resets cause the microcontroller to restart code execution. The interrupt vector table contains references to interrupt handlers from interrupt requests generated by the NVIC. If the start up file executes successfully, the microcontroller will load the *main.c* and begin the main program. This program flow is shown in Figure 5.3.

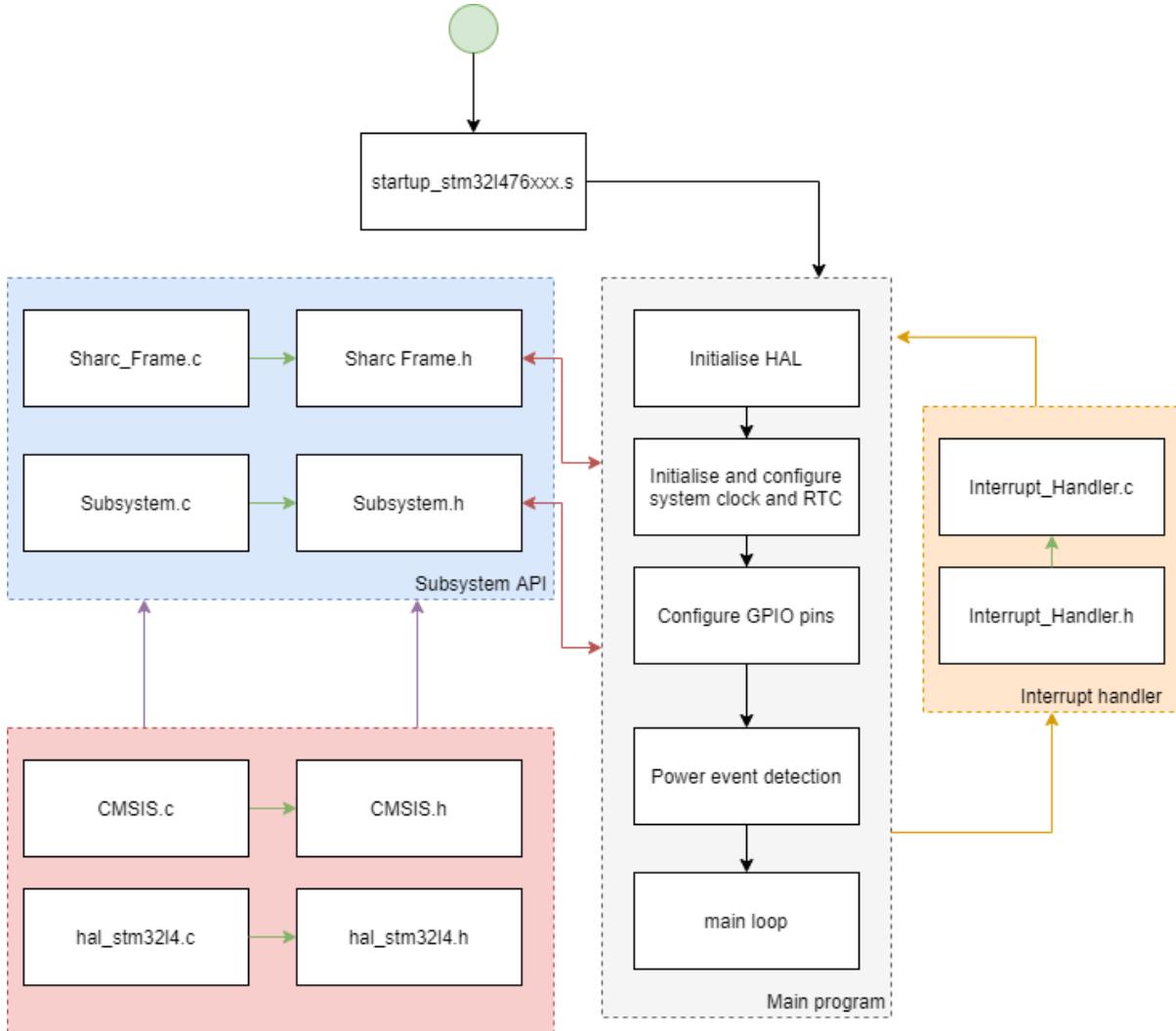


Figure 5.3: Diagram showing the firmware execution from (green) start. On start up, the `startup_stm32l476xxx.s` file executes and loads the main program. This program runs in a continuous loop until the power supply to the device is turned off. This diagram shows the interaction of the firmware layers (see Figure 5.2): (blue) subsystem API, (red) HAL and (grey) the main program. Also shown are arrows indicating (black) the main flow of the program, (green) references to functions, (red) calls to function prototypes and (orange) interrupts.

The program enters into the `main()` function which marks the start of the main application. Figure 5.3. The source folder contains the `Sharc_Frame.h/.c` files which are called by the `main.c` file. The `main()` function consists of a set up phase and a loop phase. During the set-up, the functions `HAL_Init()` and `systemClock_Config()` are used to reset the peripherals and the Systick timer, and set the system clock to the correct source and speed. For this project, the clock was configured to run from the 32.768 kHz low speed external (LSE) and mixed speed internal (MSI) oscillators in a phase-locked loop (Configuration). Additionally, the clock speed was set to 24 MHz. However, the reason for this discussed in the forthcoming sections. These two functions run in the set-up phase of the code and are called whenever the program re-enters the main function. To conserve power, the setup phase also included a function to configure the unused

GPIO pins to analog floating mode which greatly reduces the current consumption by the microcontroller (**stm32l4**).

After the setup, the power and reset state check is performed. If any power event occurs, a software reset is generated causing the program to restart from the *main()* function. When this happens, a flag is set in the *RCC_CSR*. This can occur in the form of a brown out, pin reset or low power event. This phase will check for the occurrence of any event and handle them before the program enters the main loop. Finally, if successful the program will enter the main loop and the firmware will begin.

5.2.3 Input clock selection

The biggest consideration with system operation is the system clock speed and source. The clock speed allows for faster execution of code as well as more operations during a cycle (**stm32l4**). However, this results in a larger current draw. Furthermore, the clock source is important as each input source has an associated drift and accuracy (**stm32l4ref**). The STM32L4 has five options: three internal oscillators (MSI, LSI, HSI) and two external crystal oscillators (HSE and LSE). Since the buoy will be inactive for long periods of time, an accurate 1 Hz reference signal is required to keep calendar date and time (**stm32l4ref**). In addition, The STM32L4 microcontroller features a variety of wake up options to transition from low power mode to run mode:

1. Internal configurable alarm
2. Periodic wake up alarm
3. External wake up pin

These options are made available through an internal RTC on the STM32L4 microcontroller. The peripheral can receive input from the LSE, LSI or HSI oscillators. This peripheral also allows for fast and simple data storage during extreme power down modes. When the device enters deep sleep (shutdown mode and standby mode), RAM is turned off, therefore all data are lost. The RTC has thirty two 32-bit back up registers capable of retaining 1 KB of data when the device is powered down. Therefore, The RTC back up registers were used to store system data and also act as a back up non-volatile storage location for sensor data.

As mentioned previously, the real time clock requires input from the LSE 32.768 kHz crystal provide an accurate calendar function (**stm32l4ref**). The external crystal oscillators provide high precision clock speed with extremely low drift however, the power consumption of these oscillators is much higher than the internal RC. The clock configuration of the STM32L4 allows for a combination of these oscillators in a phase-locked loop (PLL) which allows for a greater degree of accuracy at desired speeds. The final clock configuration parameters are shown in Table 5.1.

Table 5.1: Configuration parameters for the system clock and real time clock (RTC) including sources and frequencies.

	Run mode system clock	Low power mode system clock	Real-time clock (RTC)
Clock source	MSI and LSE in a PLL configuration	LSE	LSE
Clock frequency [MHz]	24	32.768 kHz	1 Hz

5.2.4 Power mode selection

As discussed in Chapter 3, a key design requirement of the firmware is to optimise the device for low power consumption. Furthermore, the criteria for sampling ice drift requires a minimum measurement frequency of at least once every 3 hours (**vichi2019effects**; **alberello2019drift**) while waves-in-ice measurements require sample rates of at least 2 Hz every four hours (**rabault2019open**). This shows that the software life cycle will be defined by short periods of sampling activity followed by long periods of inactivity. Therefore it is important to place the device in as low power mode as possible to minimise consumption.

The biggest consideration with system operation is clock speed and source. The external crystal oscillators provide high precision clock speed with extremely low drift. However, the power consumption of these oscillators is much higher than the internal oscillators. However, the clock configuration of the STM32L4 allows for a combination of these oscillators in a phase-locked loop (PLL) which allows for a greater degree of accuracy at 24 to 26 MHz with a reduced power consumption.

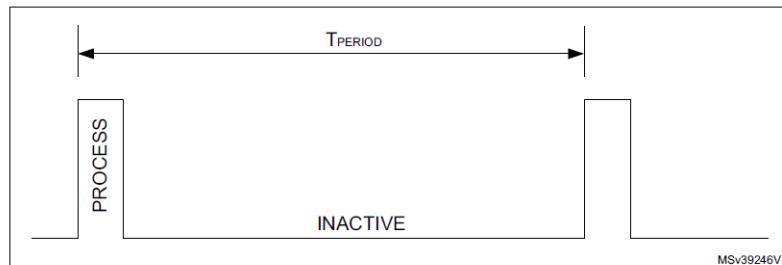


Figure 5.4: Diagram showing a general low power operation profile for the microcontroller with two distinct phases: process and Inactive occurring over a period T .

Figure 5.4 above details a typical low power operation which can be expected from the buoy. For a typical application, we consider two main phase:

1. Process phase: The system is in run mode with peripherals active at regular intervals.
2. Inactive phase: The system is asleep until triggered by a wake-up event (RTC) or interrupt (GPIO).

Once the buoy has finished an active routine, the system becomes inactive between samples. The ice drift sample period was selected to be once every half an hour to provide a higher temporal resolution to characterize the inter-sample behavior of ice floes described by **vichi2019effects** in Chapter 2. Therefore, once the routine is complete, the device will be inactive for tens of minutes until required. This is the inactive between sample mode and is our period of inactivity where we can place the device in the lowest possible state with very little concern for wake-up time or peripheral settings. Therefore, the following power modes were selected for each phase of the system's operation.

Table 5.2: Power mode selection for each phase of the buoy's operational cycle. Power modes are preset by STMicroelectronics and are based on the device's architecture. A full explanation of each power mode can be found in (**stm32l4ref**).

Phase	Power mode	Current draw
Process	Run mode	1.16 mA
Inactive	Standby mode	710 nA

Table 5.2 shows the selected power mode and the estimated current consumption of each mode (**stm32l4**). The run mode current was benchmarked using a Dhystone test with a system clock of 24 MHz and code loaded from Flash. The inactive current draw was estimated with a low speed external (LSE) oscillator supplying the real time clock.

5.3 Main program design

As outlined in the user requirements in Section 3.2, the SHARC buoy aims to increase remote sensing on both a temporal and spatial scale. Each sensor has its own sampling requirement which needs to be synchronised to the clock speed of the processor. Furthermore, the device needs to monitor and control the subsystems to ensure they are functional and operational. Finally, subsystem power consumption needs to be carefully controlled to ensure the device operates in a power efficient manner. The main program was designed to meet these requirements by providing a frame to sequence sensor activities. The firmware was also designed to provide time-critical execution of sensor routines to ensure that the temporal integrity of the data set was achieved. To achieve this, a state machine was selected and implemented on the STM32L4 microcontroller.

A state machine can be implemented to provide a high-level form of control over the system. This can be achieved by decomposing the overall function of the buoy into a series of finite states. These states are connected through a series of transitions which can be described using Boolean techniques. Through this, the buoy retains a modular structure both in firmware and in hardware which can allow for additional sensors and functions to be implemented seamlessly.

5.3.1 Execution

The goal of the buoy is to sample environmental, GPS and power data at fixed rates. T_{sample} will be used to describe the period between sampling the sensor devices. Each data point will be condensed into a byte packet and stored in the external Flash chips. A decision was made to synchronise the transmission period to four samples so as to reduce the time the Iridium modem is turned on. This device consumes the most power so having it used as infrequently as possible will allow the device to conserve more power. During the transmission phase, the device will read the data packets from memory into a buffer and transmit the data. When the device exits this state, it will reset the sample count and repeat until the buoy is turned off or loses power. The buoy can therefore be broken down into a set of finite states which are shown below:

1. **Initialisation state:** The device initialises the counter and verifies the sensors.
2. **Reset state:** During this state, the sample counter is reset and the Flash chips are formatted to prevent old data from corrupting the set.
3. **Sample state:** During this state, the device actively receives data from the sensors and stores them into a packet which is then saved to a memory location.
4. **Sleep state:** The device enters this state between samples and active states. Here, the device will remain in this state for a time T_{sample} . After which, the buoy will wake up.
5. **Transmit state:** The device will load the data from memory and transfer to the Iridium modem buffer. Upon successful transmission, it will enter the reset state.

Each state will control which routines are performed during the function and provide the system with information on the current status of the device. During run time, system information will be saved to the back up registers of the RTC. These registers will constantly be updated to show what state the buoy is in, what the last action was and what the memory status of the Flash chips is. Should the device encounter an unexpected hardware reset, the system can read the back up registers, load the last state and run as normal. A typical system run is shown in Figure 5.5:

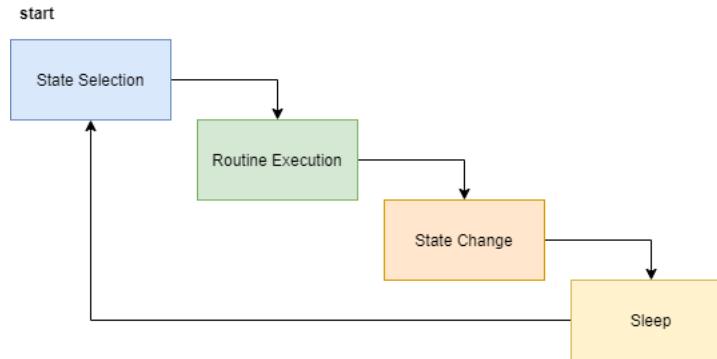


Figure 5.5: Diagram showing a typical run of the main program. The program is designed to run sequentially based on the previous completed state. The program begins from power on/wake up and reads the last known state from the RTC back up registers. Based on this state, the program will execute the corresponding routine. Finally, the program will determine the next state and write this to the back up register. If no activity is required, the device will enter low power mode.

The inputs to the state machine are:

1. **C**: a 2-bit integer signifying the number of samples performed ($0 \leq N < 4$)
2. **T**: Variable that matters when the system is asleep. Signifies whether the system has been in sleep mode for a time defined by the constant value T_{wake} .

The RTC contains a wake up unit which stores this value while a counter running at 1 Hz counts in a separate register. When the counter reaches T_{wake} , an interrupt is generated on an internal wake up line and the device exits low power mode.

The system has no explicit outputs. However, the state machine is used to control which routines will be executed during the execution phase of the program. Therefore, the outputs can be considered as routines **R_x** as shown below:

1. R_{sample} : Functions to initialise and sample a sensor. This routine is called for any sensor during its sample window.
2. R_{sleep} : Function to put device to sleep for a period T_{wake} . During this time, the device will not respond to any stimulus with, two exceptions:
 - (a) An interrupt generated on an internal wake up line.
 - (b) An external interrupt generated on one of five wake up pins.

After wake up, the device resumes functionality.

3. $R_{transmit}$: Routine to initialise the Iridium modem and prepare the data for transmission. After transmission, the modem is deinitialised and put into power saving mode.

Given the following information, the Algorithmic State Machine (ASM) chart is derived and shown in Figure 5.6.

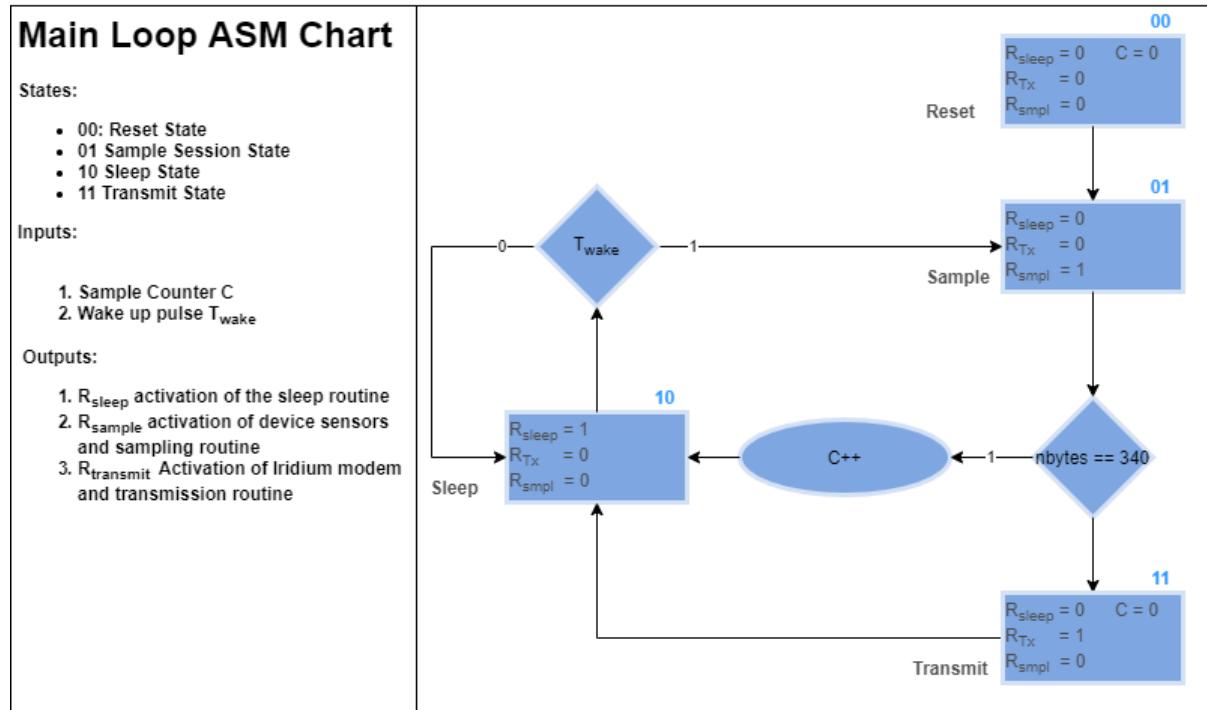


Figure 5.6: ASM chart for the proposed state machine to run on the processor showing entry/exit conditions and functions to be run during states.

Figure 5.6 shows an abstract representation of the logical flow of the program. A typical run from the system will have the buoy initialised and calibrated before entering the main loop where it will alternate between active sampling and inactive sleep mode until enough data have been collected to transmit. This allows the Iridium modem to only be turned on when needed thereby significantly conserving energy while allowing for the system to sample as much as possible. The variable T_{wake} is user defined and sets the sample rate of the system. As mentioned previously, the interval for measuring ice drift was set to 30 minutes with data transmitted on the fourth cycle.

The RTC periodic wake up unit is used as a counter in deep sleep mode. This is a 16-bit down-counting Auto Reload Register (*ARR*) that generates an interrupt on an internal wake up line when the system has slept for a length of time T as defined by the user. In addition, the sample counter gets reset after every transmission state and when the buoy enters a reset state. The number of samples before transmission was chosen to be four. Since the Iridium transmission buffer is 340 bytes long and the RockBLOCK data service charges £0.14¹ per 50 bytes, the goal is to transmit as much data that would fit into the buffer as possible. Too frequent transmissions incur a high data cost but result in data integrity. Too few transmissions can result in lost sample points if a transmission is not received. As a result, transmissions will occur every 2 hours with an expected total of 12 transmissions each day. The reason for this will be discussed in Section 5.4.

¹Source: <https://www.rock7.com/products/rockblock-9603-compact-plug-play-satellite-transmitter> as of March 2021

5.3.2 Asynchronous behaviour

Asynchronous behavior describes all functionality that occurs outside of the main loop. This can come from interrupts or external events which causes the system to exit the main loop regardless of state. For this version of the buoy, asynchronous functionality was considered for the Iridium modem to allow the device to receive messages. This would allow for users to interact with the device and receive data outside the scheduled transmission period. Further asynchronous functionality was added for the inertial measurement unit. This would allow for easier data flow from the sensor (see section 5.5) and allow for event-based sampling. The following are sources of asynchronous behavior:

1. Interrupts
 - (a) Iridium messages received (ring alert)
 - (b) IMU event detection (ice floe collision/free-fall detection)
2. Events
 - (a) Low power detection
 - (b) Brown out detection
 - (c) Software resets
 - (d) Watch dog resets

These events take precedence over the main loop function. A full description of events, interrupts, and the protocols for handling them are shown in Tables C.1 to C.5 in Appendix C.1. The figure below shows how the event handling procedure is sequenced in the main program:

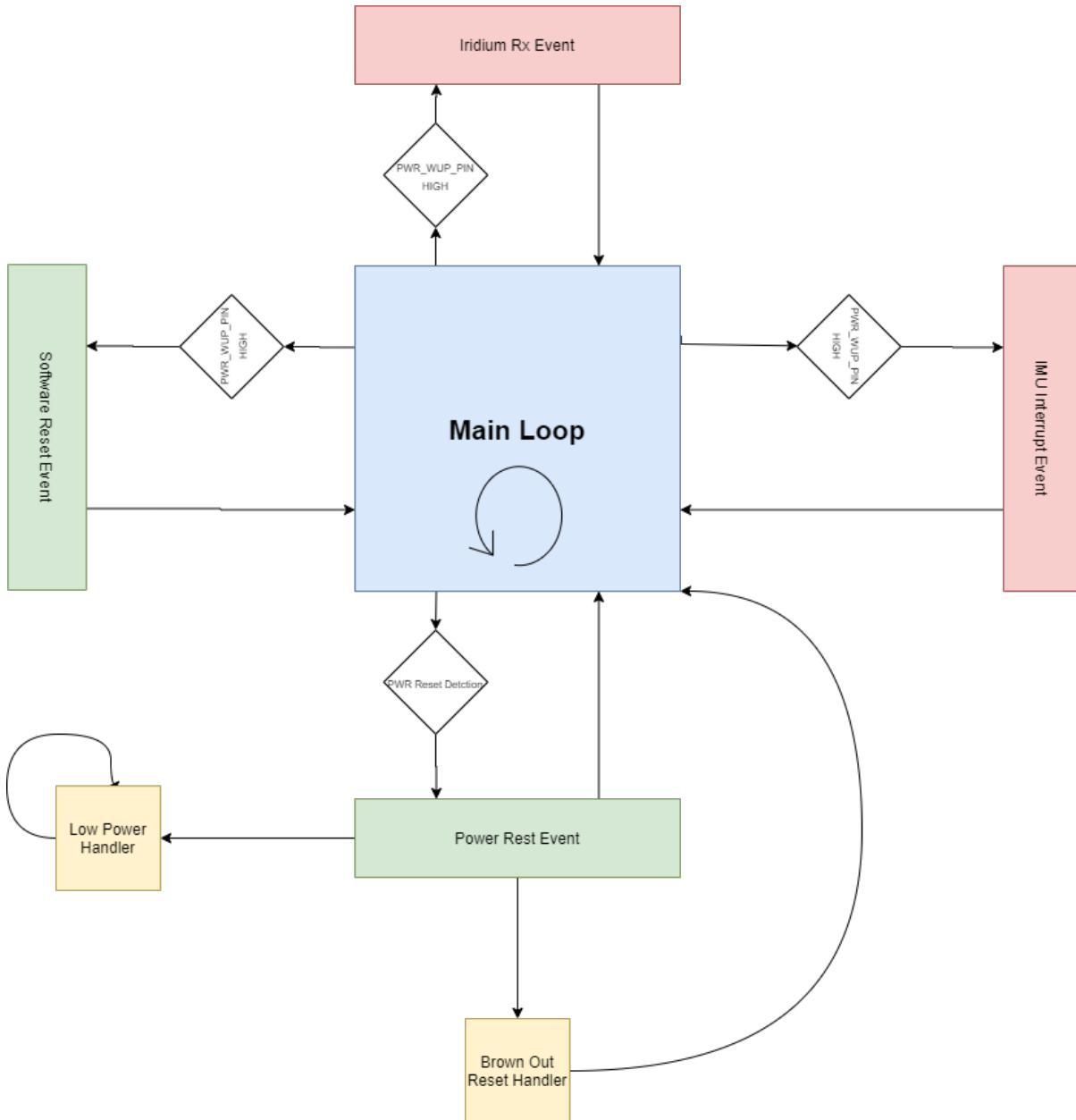


Figure 5.7: Application diagram with event and interrupt sequencing.

States are represented by unsigned 8-bit integers on the system and are represented in Table 5.3.

Table 5.3: Numerical and 8-bit unsigned integer representation of the synchronous states on the STM32L4 microcontroller.

State number	State name	8-bit representation
0	Reset	0b0000 0000
1	Sample	0b0000 0001
2	Sleep	0b0000 0010
3	Transmit	0b0000 0011

The backup are used to store state information while the buoy is in deep sleep. These registers are located at the memory addresses: 0x40002850 to 0x400028CC in the STM32L4 memory map ([stm32l4ref](#)). These registers keep data even when the device is in low power mode or a software reset has occurred therefore making them the perfect storage location. The state variable holds the value of the current state of the buoy. This variable is stored in two locations: When the system is in run mode, the value is stored in the global variable *Current_State*. When the device is in a deep sleep state, the variable is stored in the RTC back up registers at byte 0 of back up Register 0. Upon wake up, the value is loaded from the register and placed in the global variable.

The main loop follows a sequential state transition as described in Figure 5.7. To achieve this, at the start of each loop, the program reads the value stored in the state variable. This determines what the previous state was. Based on this value, the new state is determined and stored in the state variable. This process is shown in the figure below.

System State Check Algorithm

The following flow chart shows the algorithm for transitioning between states.

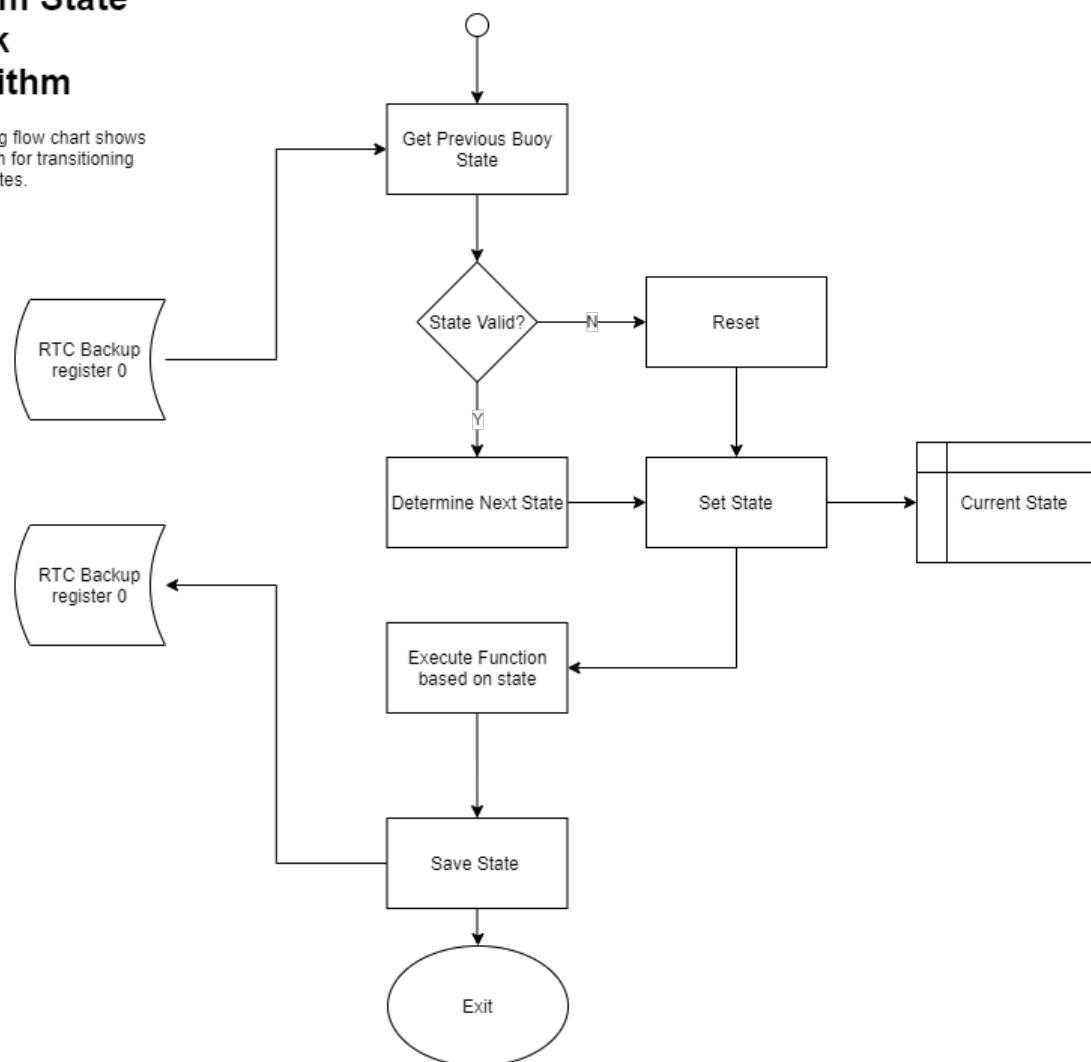


Figure 5.8: Flow chart for the state-check algorithm

Figure 5.8 above shows the algorithm for selecting and transitioning between states. This

algorithm allows for states to be linked in any order and, most importantly, separates the state selection from the state function. By separating these two concepts, a more modular framework is created. This allows for the addition of more states and transitions without modifying the routines that are currently in place. This allows for device functions to be turned on and off as desired without drastic changes to the firmware.

Finally, asynchronous states take a higher precedence over the main loop states and are checked before the state check shown above. The order of precedence is shown in the table below:

Table 5.4: Table showing the types of states that the system checks for ordered by priority with 1 being the highest priority and 3 being the lowest.

Name	Priority
Power event	1
Asynchronous interrupt	2
Sequential state	3

Power events generate a system reset and raise a flag in the *PWR* status register. When the flag is set, the program enters the handler and, if the event is non-fatal, returns to the main loop. The following flow chart shows an example of such a case for a brown out event.

Brown Out Reset Handler

A Brown-Out event occurs when the voltage to the micro controller drops below a set threshold.

Entry:

The Device is held under reset until the voltage rises above the threshold. This causes the BORST Bit to be set in the RCC_CSR. Software will detect the bit and enter the handler on this condition

Function:

Here, software will read the back up register to determine the state. If this value is corrupted, the device will reset the back up domain then perform a software reset. If the value is secure, the device will reload the state and start the main loop as normal

Exit:

Device exits the handler upon successful completion of the Handler Routine. Fatal brown out will cause the device to enter an infinite loop

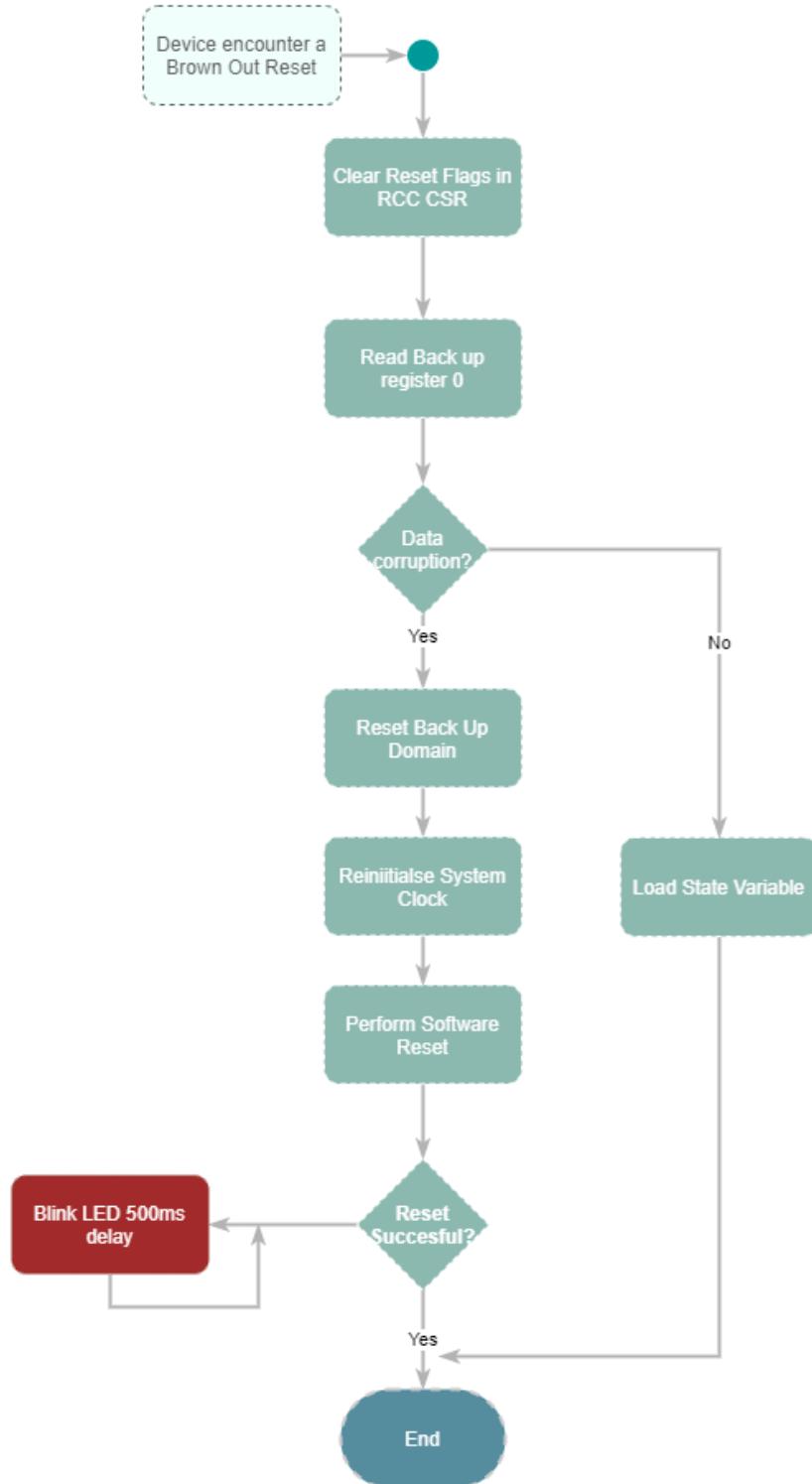


Figure 5.9: Diagram showing the algorithm for brown out event recovery and handling.

Some sensors have interrupt pins and can be configured to trigger upon detection of a specific event. When this happens, the sensor will send a digital high on the interrupt pin. On the processor side, a hardware interrupt is generated and the software handles the interrupt. An example of such a procedure is shown in Figure 5.10.

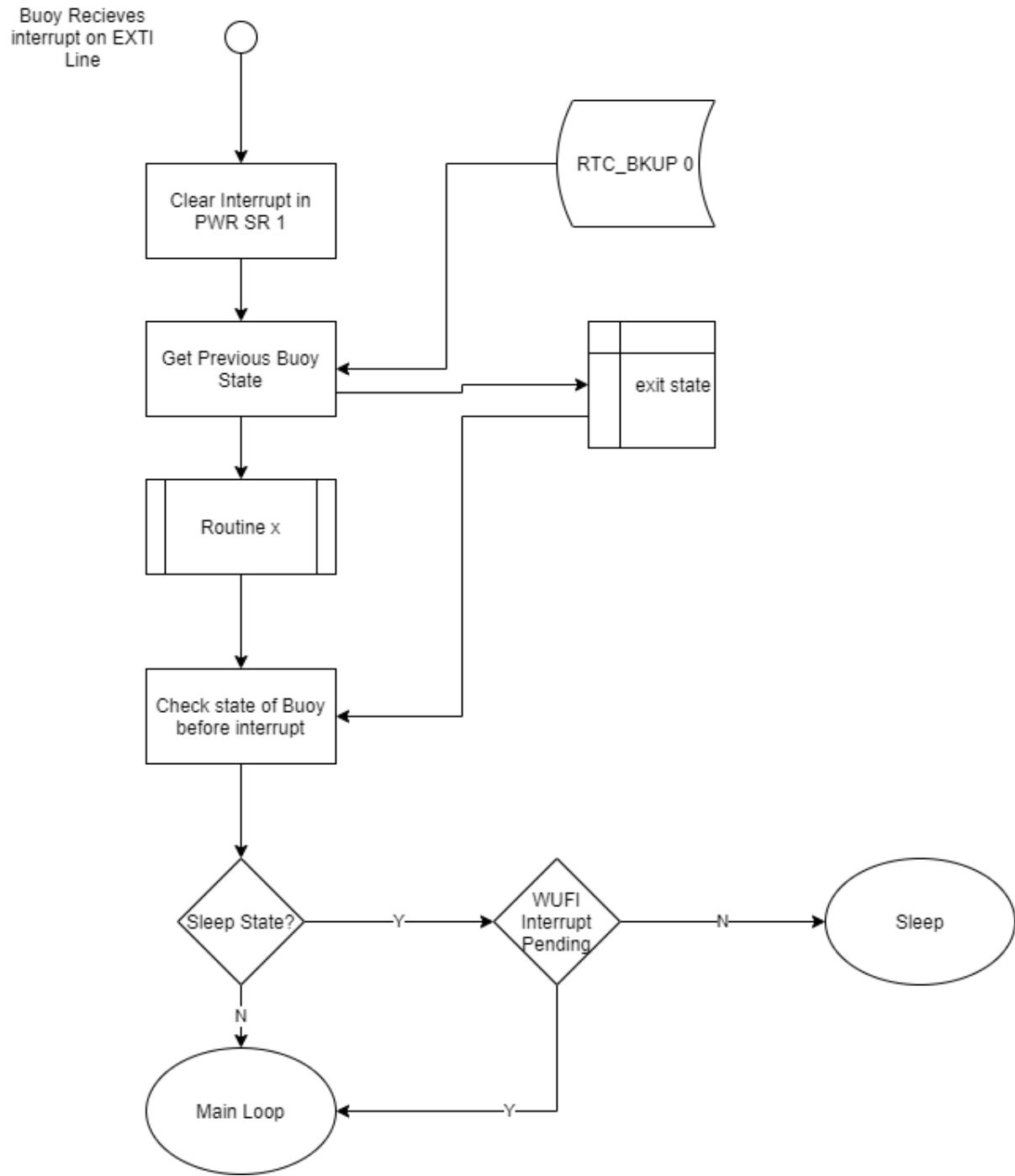


Figure 5.10: Diagram showing the algorithm for handling an external interrupt from a Wake Up pin connected to one of the modules

By connecting these pins to external wake up pins, the buoy is capable of event detection in deep sleep mode. If an event is detected while in deep sleep, the interrupt causes the buoy to wake up and resume from the beginning. No interrupt handler is entered at this point. A flag is set in the *PWR_SR* at the position of the wake-up pin it detected. The buoy will enter the asynchronous state depending on which flag is set and will execute the routine associated with it. When the buoy wakes up from an internal wake up timer, the pins are reconfigured into *GPIO EXTI* mode which allows the buoy to receive interrupts when active. It was found that by keeping these configured as wake up pins during run

time, the system reset when an interrupt was detected. Therefore, strict measures were taken to ensure that the pins were reconfigured at every point in the program that is entered when the device wakes up from deep sleep.

5.3.3 Subsystem execution

When a module is being used at any point in the program. The microcontroller will execute an initialise routine. This will enable any required peripherals for communication with the subsystem. This function will be called before every sample period in case the system encounters a power surge or an unexpected reset. Additionally, placing the microcontroller in deep sleep mode results in the registers being reset upon wake up. The initialisation routine is specific to each microcontroller and includes the following:

1. High level communication peripheral configuration.
2. Low level pin configuration
3. Sensor verification function
4. Sensor configuration function
5. Return status

The outcome of the initialisation routine is evaluated based on a return status from the function. This ensured that the status of the device was known at all times. If the device was offline, the program would either retry to connect to the module or continue execution without it. This ensured that the program was robust enough to deal with hardware failures as experienced by **doble2017robust** and **rabault2018investigation**. Items 1 to 3 are included to configure the subsystem to adhere to the functional requirements in Table 3.8. Sensor verification functions are included to satisfy acceptance tests AT001 and provide evaluations for acceptance tests AT002 and AT004. These functions also allow the program to continue execution if a sensor becomes corrupted or a communication channel becomes non-responsive. The initialisation function is designed handle fail modes by evaluating the system's failure return type and responding in accordance with the protocols outlines in acceptance test AT002. initialisation routines for each subsystem is provided in Appendix C.1 to C.6.

If the initialisation was successful, the program will continue using the module in the firmware. Should a failure occur, the system will attempt to reconnect with the device a predefined number of times. In case of a critical failure, the system will acknowledge that it can no longer use the device and will continue the main firmware without it. The resulting behaviour is shown in Table 5.5.

Table 5.5: Table Showing the device behavior in case of a critical failure in one or more of the subsystems. Critical failures are defined in AT006 (Table B.7) testing protocol.

Device failure case	Impact	Result
Iridium	Critical	No data will be transmitted from the buoy.
Flash chips	Critical	Data will be lost when power is reset.
GPS	High	Critical GPS data will not be captured.
MPU6050	High	Unable to measure critical accelerometer and gyroscope data.
Environment Sensor	Medium	Environmental data will not be captured.
Power monitor	Low	Current and voltage measurements will not be captured.

In conclusion, defensive functions were added to ensure the firmware interface with the hardware sensors is robust and recoverable in case of hardware failure.

5.4 Data management

A critical consideration for the system is the flow of data and memory management. The Flash chips provide a solution for permanent storage however, it is critical that data integrity be maintained. Some form of data organisation must be implemented for intelligent retrieval/storage of data in a meaningful way. In addition, the system requires some form of back up should the device be unable to connect to the Flash chips. This section outlines the data organisation and storage methods implemented in the firmware to ensure efficient retrieval and storage of data.

The Flash chips used are Adesto Technologies AT45DB641E SPI serial Flash chips. Each chip can hold up to 64 Mbit of data. Data can be read/written at speeds of up to 85 MHz or 15 MHz in low power mode. The device is low power with high data retention requiring a supply voltage of 1.7 V to 3.6 V and draws a maximum of 11 mA in active read mode thereby making it one of the lowest power consumption components in the system. In addition, the device comes with two 256 byte buffers that can store data while a read/write operation is taking place. Memory is organised into sectors (2 to 256 KBs long), blocks (2 KB long) and pages (256 bytes) with write, read and erase options at each level. In this section, the data requirements from each sensor are listed. The optimal storage strategy is to convert the measurements into binary data and store as an array of bytes at known locations in an array.

5.4.1 Debugging

As discussed in Chapter 3, a key user requirement for developing the SHARC buoy is ensuring the design is robust. This means that device must be capable of providing feedback of system and subsystem states either to the users or between the different firmware layers. Furthermore, from the stakeholders in Table B.1, users of the buoy may

not necessarily be involved in the design and development phases and therefore must rely on other methods to understand whether the device has deployed successfully or whether the system is online. To achieve this error checking methods were implemented in all algorithms to ensure that functions executed as predicted and current status of the buoy was known at all times. These subsections discusses the debugging methods both internal and external to the buoy.

Internal debugging

Internal debugging was achieved by ensuring that each layer of the system had defensive measures built into the algorithms. For example, on the device driver layer, the communication algorithms had built in timeouts to prevent the system from freezing while waiting for input. These functions returned error codes to the higher layers when called. On this level, different errors were accounted for by creating an enumeration of error types as shown in Figure 5.11.

```
typedef enum{
    GPS_Init_OK = 1,
    GPS_Init_Ack_Error = 2,
    GPS_Init_Baud_Config_Error = 3,
    GPS_Init_MSG_Config_Error = 4,
    GPS_Init_Ack_Tx_Error = 5,
    GPS_Init_Periph_Config_Error = 6,
    GPS_Init_Offline_Error = 7
}GPS_Init_msg_t;
```

Figure 5.11: Example of an enumeration of possible error sources for a peripheral library function. Each error code is a numeric value which is returned when a specific error condition is encountered thereby providing feedback to higher levels.

On the device peripheral level, if an error code was returned, the routine that called the functions would evaluate the error code and act accordingly. If the failure was critical, this feedback would be provided to the main program which would then break out of the routine and continue with the primary loop.

External debugging

Progress was monitored via visual indication. Having a visual indicator ensured that the buoy progress was constantly known without needing to interfere with the program execution or peak into the code. This allows the user to understand how the buoy operates without needing the firmware or references to the firmware. This was achieved by using a green LED onboard the STMicroelectronics Nucleo-L476RG development board. The green LED was programmed to execute Flashing sequences at various points in the code to signify information such as sensors were online or a routine was in progress. This

allowed for remote debugging of the platform. However, this method does not offer any detailed information about the status of the buoy.

To provide more information, a free UART port was programmed to output status information to a serial terminal. This allows a user to view more detail regarding the current state the buoy is in, the state of the sensors and the incoming data. However, this technique requires a constant connection and is not suitable for deployment.

5.4.2 Drift data acquisition

This section describes how data are acquired from the sensors to form an ice drift measurement packet. Readings are taken from the GPS and environmental sensor with the power monitor being sampled to provide an update on the buoys performance.

Overview of drift data

The GPS is sampled four times over a given interval at each sample point, the following data are recorded:

1. Time and date information
2. Geographical coordinates
3. Dilution of precision
4. Diagnostic information

By default, the u-blox NEO-7M and NEO-M9N receivers uses the National Marine Electronic Standards (NMEA)² to format data. These have the following format:

Table 5.6: Breakdown of a typical NMEA message string with fields indicating start/stop sequences and character information.

\$	Address	Data Field	checksum	End Sequence
	TT	SSS		

- \$ - Character denoting the start of the sequence.
- Address - This is a 5 byte character sequence that is used to provide information on the talker ID (TT) and the the type of information in the payload (SSS).
- Data field - Data in this field is formatted as a character sequence separated by commas. This field holds the payload specified by the payload information characters in the address field.

²Information about NMEA messaging on the u-blox Neo GPS is taken from (**UBLOX_M9N_INTERFACE**)

- Checksum - Sequence of characters denoted by a "*" and followed by two bytes in ASCII hexadecimal format. These values are calculated by performing an XOR operation on all the bytes between the "\$" and "*" characters.
- End Sequence - 0x0D, 0x0A denotes the end of the NMEA message.

NMEA messages are standardised in the format given in Table 5.6. However, each NMEA message holds different information. Therefore, the incoming message strings can have different lengths depending on the messages that have been enabled. To accommodate this in the firmware, a data streaming channel needs to be set up to allow for an unknown number of bytes to be transferred through serial communication in an efficient way. Furthermore, to ensure that the required GPS drift data were obtained, Tables 5.7 to 5.9 show the NMEA messages that were selected and the data as well as the format of each of the fields.

Table 5.7: Description of ZDA message string showing variables, description and how the example datum 5th September 2002 08:27:10 am is stored.

ZDA - time and date		
Description	Datum information in UTC representation	
Variable name	Format	Example
UTC time	hhmmss.ss	082710.00 - 08:27:10 am
UTC day	dd	05 - 5th
UTC month	mm	09 - September
UTC year	yyyy	2002
Time zone hours	hh	00 (+00)
Time zone minutes	mm	00 (+00)

Table 5.8: Description of GSA message string showing variables, description of parameters and how the variables are stored.

GSA - fix diagnostic		
Description:	DOP, number of satellites and fix type	
Variable name	Format	Example
Operation mode	A/M	A - Automatic
Navigation mode	Number (1-3)	1 - No fix
Satellite ID	Number	29 - Satellite number
Direction	C	E - East
PDOP	Float	1.91
HDOP	Float	1.18
VDOP	Float	1.14

Table 5.9: Description of GLL message string showing variables, description and how a set of coordinates e.g. (47°17.11364'N, 8° 33.91565') is stored.

GLL - Geographic coordinates and fix		
Description: latitude and longitude with positional fix information.		
Variable name	Format	Example
Latitude	ddmm.mmffff	4717.11364 - 47°17.11364'
Direction	C	N - North
Longitude	dddmm.mmffff	00833.91565 - 8° 33.91565'
Direction	C	E - East
Fix status	A	A - Valid

GPS sampling strategy

The u-blox NEO-7M/NEO-M9N module continuously outputs data at a fixed rate of 1 Hz (**UBLOX_M9N_INTERFACE**) through Universal Synchronous/Asynchronous Transmission. The device comes preset with not all NMEA messages active. The messages were enabled by writing to the *CFG-MSGOUT* register. This was achieved by transmitting a customised configuration string to the u-blox module. If successful, the receiver will return an acknowledgment message.

Then a UART data stream was created to read the incoming messages. To achieve this, one could poll the receiver register for the byte information. However this method requires additional processing power which increases the power consumption. Additionally, constantly polling the register can result in data being overwritten if it is not read fast enough. The STM32L4 microcontroller contains a direct memory access (DMA) controller (**stm32l4ref**) which allows for data to be streamed directly from a peripheral to a specified address in memory. This method was preferred as it allows for data to be read from the sensor without constraining processor resources.

Reading data through a DMA channel requires knowledge of the number of bytes being received. However, as mentioned previously, this value is not known. Therefore, a solution was to manually turn off the DMA stream when no data were received for a certain length of time (i.e. an idle line condition). This was achieved by setting up a timer channel in input capture mode on the receiver line in slave reset mode while another channel was configured in output compare mode. In this register, the expected idle period was stored. Incoming data would reset the counter in the input capture channel. If the counter reached the value in the output compare channel, an interrupt was generated which signaled the DMA to stop the transfer. The number of bytes transferred was stored, the DMA was reset and the data were stored.

As an additional precaution, data from the peripheral DMA stream was backed up to protect the data against accidental deletion or corruption. This was done by using another DMA stream set from the current data memory address to another data buffer via memory to memory DMA stream triggered on completion of the peripheral to memory DMA stream. This streaming technique formed a UART DMA circular buffer with a slave reset trigger.

Then, message parsers were written to extract the information from the message strings

and convert them into a binary representation. These message parsers contain a check for validity. This algorithm first checks that the data follows the correct NMEA formatting as shown in Table 5.6. Then it analyses the address to ensure that the talker ID and message-ID are valid. Finally, it calculates the two-byte checksum by performing an exclusive OR on all the bytes in the data field and compares them to the checksum bytes that were sent with the packets. Message parsers were written for GLL, GSA and ZDA messages and were called based on the return status of the validity check. Table 5.10 shows the memory allocation for each variable.

Table 5.10: Data collected from the GPS in a single sample session. Square brackets denote a numbered sequence of like variable types.

Variable name	Variable type	Size [bytes]
Epoch time	Unsigned 32-bit integer	4
Latitude	Signed 32-bit float	4
Longitude	Signed 32-bit float	4
HDOP	Unsigned 8-bit integer[2]	2
VDOP	Unsigned 8-bit integer[2]	2
PDOP	Unsigned 8-bit integer[2]	2
Diagnostic information	Unsigned 8-bit integer	1
Total:		19

Time and date information were combined and converted into Unix epoch time. This represents the number of seconds that have elapsed since a defined epoch (1 January 1970) which allows for a single, 4-byte variable to represent both time and date. Geographic coordinates have been converted into signed 32-bit floats with the sign representing the direction of the coordinate. The coordinates were then split into an array of 4 unsigned 8-bit integers and recombined using IEEE-754 as a standard. The dilution of precision represents a value between 0 and 99.99 therefore, the optimal storage solution is to allocate a byte for the digit and a byte for the precision. Finally, diagnostic information includes the fix type and the number of satellites. A maximum of 15 satellites can be used to determine a position. These data can be stored in the lower four bits of a single 8-bit integer. The fix type is a number from 1 to 3 therefore only taking up 2 bits of a byte.

5.4.3 Environmental data acquisition

The BMP280 contains two onboard Analog to Digital Converters (ADCs) which are used to convert the pressure and temperature measurements into unsigned 8-bit values. Each measurement is stored as three unsigned 8-bit integers in three registers and must be read sequentially to get the full measurement. Once retrieved, the data must be combined into a 24-bit word which results in the raw, uncompensated ADC value. The BMP280 also contains a configurable infinite impulse response (IIR) filter as well as configurable oversampling parameters for the pressure and temperature measurement. Data are read through an SPI communication interface into the microcontroller by performing a burst read of 6 bytes. To compensate for the mechanical effects of each sensing element, the device comes preloaded with a set of compensation parameters for the temperature and pressure reading ([BMP280 Datasheet](#)). The compensation algorithms are shown in

Appendix C.8 and C.7. The output of the compensation algorithm are shown in Table 5.11.

Table 5.11: Description of output values from BMP280 post processing.

Name	Type	Format	Example	Total [bytes]
Temperature	Signed 32-bit integer	CCcc° C	2508 to 25.08° C	4
Pressure	Signed 32-bit integer	PPPppp kPa	100653 - 100.653 kPa	4
Total				8

5.4.4 Power data acquisition

The INA219 samples current across a shunt resistor of a known value. In this application, the shunt resistor provided is $0.1\ \Omega$. The device also samples the voltage across the shunt resistor through a programmable gain amplifier (PGA) before being sampled by an ADC. The sensor stores data as 16-bit integers. Negative values are stored in two's complement format. Data are transferred via I²C to the microcontroller after the conversions have taken place. The sensor measures both shunt and bus voltage which, when combined, provide an estimate of the voltage across a load. The output resolution can be programmed as either 9-bit, 10-bit or 12-bit. When the device is initialised, it needs to be calibrated. A calibration routine was developed for the device based on the algorithm recommended by **INA219**. The bus range voltage was chosen as 16 V with a maximum expected current of 1.2 A (two times the expected peak current draw) to be safe. The output of the calibration procedure is a 16-bit word that is written to the calibration register. The algorithm used to calibrate the sensor for the SHARC Buoy application is outlined in Appendix C.9 with the following parameters:

Table 5.12: Description of parameters used to calibrate the INA219 current sensor.

Maximum bus voltage	16 V
Maximum expected current	1.2 A
Shunt resistor	$0.1\ \Omega$
Shunt voltage range	$\pm 160\text{ mV}$

The sensor calculates the power consumption as a signed 16-bit number by multiplying the bus voltage ADC reading with the current ADC reading and placing it in the power register. The microcontroller performs a burst read of the bus voltage, shunt voltage, current and power register and stores the values as signed 16-bit integers. The bus voltage register reserves the first three bits of the register for signal flags. Therefore, the bus voltage reading must be shifted by three bits to the right to remove them. Finally, the power reading is multiplied by the LSB step size calculated in the calibration routine which results in a signed 16-bit integer representation of the power in milliwatts (mW). The data requirements are shown in Table 5.13

Table 5.13: Description of output values from INA219 current sensor.

Name	Type	Format	Example	Total [bytes]
Shunt voltage	Signed 16-bit integer	vvmm	18049 to 180.49 mV	2
Bus voltage	Signed 16-bit integer	VVvv	08025 to 8.025 V	2
Current	Signed 16-bit integer	IIIii	51234 to 519.23 mA	2
Power	Signed 16-bit integer	PPPPpp	28130 to 2813.00 mW	2
Total				8

5.4.5 Total ice drift data

Table 5.14 below shows the total ice drift data collected from a single sample point.

Table 5.14: Total drift data collected during a single sample point.

Device name	Total data (bytes)
GPS	19
Environmental Sensor	8
Power Monitor	8
Total	35

Table 5.14 shows that 35 bytes of data are collected during a single sample session if all the sensors (excluding IMU) are sampled concurrently. Since the device has four Flash chips each with a capacity of 8 MB, the device contains 32 MB of permanent storage space. This is enough memory to store 914285 full packets of drift data. At 30 minute intervals, this is enough to store 152.19 years worth of drift data.

5.4.6 Wave measurement data

This section describes how data are acquired for waves in ice measurements. The inertial measurement unit (IMU) provides 3 axes of acceleration and 3 axes of angular velocity which are the components used to estimate the significant wave height, dominant wave frequency as well as the spectra and co-spectra over the sample period. IMU sampling occurs after the fourth drift measurement is taken during the buoy operational cycle. The sample frequency was chosen to be 5 Hz to satisfy the Nyquist sampling criteria.

IMU sampling strategy

The MPU6050 IMU is a micro electrical-mechanical system (MEM). The device measures the inertial axis reading which is then digitised using a 16-bit ADC for each axis of the accelerometer and gyroscope. Communication is performed using I²C where the pin AD0 is used to set the I²C address. In addition, the device is fully configurable allowing for programmable ADC full scale resolutions and sample rates. The device also contains an

on-board digital low pass filter, the bandwidth of which can be programmed through the *CONFIG* register.

The MPU6050 is an 8-bit device. Measurements from the ADC are split into 8-bit bytes and stored across 2 registers (one for the most significant byte, and one for the least significant byte). A burst read operation is performed to retrieve the data in the register. The two bytes are combined and stored as a signed 16-bit integer. This value is then multiplied by a sensitivity factor which results in a float representing either the acceleration or the angular velocity. The sensitivity factor is determined based on the selected full scale range of the accelerometer and gyroscope. Table 5.15 gives a breakdown of a single sample of IMU data.

Table 5.15: Description of output values from the MPU6050 IMU showing variable name, size and significance.

Name	Type	Total [bytes]
X-axis acceleration	Signed 16-bit integer	2
Y-axis acceleration	Signed 16-bit integer	2
Z-axis acceleration	Signed 16-bit integer	2
X-axis angular velocity	Signed 16-bit integer	2
Y-axis angular velocity	Signed 16-bit integer	2
Z-axis angular velocity	Signed 16-bit integer	2
Total		12

From our user requirements the sample period for collecting wave data are a minimum of 15 minutes. Average ocean wave sample periods are recorded at 20 minutes sometimes even as high as 30 mins for significant wave height. The dominant wave frequency occurs at about 1 Hz. Sampling at 2 Hz (**kohout2015device**) is a bare minimum. However, 5 Hz is recommended.

Table 5.16: Parameters of the IMU and their configured value for this application.

Sample period [minutes]	20
Sample frequency [Hz]	5
Accelerometer full scale range [g]	± 2
Gyroscope full scale range [$^{\circ}s^{-1}$]	± 500
Digital low pass filter bandwidth [Hz]	92

Finally, the total data accumulated over the required sample period is shown in Table 5.17.

Table 5.17: Breakdown of data accumulated from the IMU with the sample parameters mentioned in Table 5.16.

Sample frequency [Hz]	5
Sample period [s]	1200
Number of samples	6000
Bytes per sample	12
Total	72000 bytes

Therefore, a total of 72 KB is collected from each session. With the current memory configuration. A single sample can occupy 0.9% of a single Flash chip. With four 8 MB Flash chips, The buoy can record 444 raw IMU datasets. If recorded every 2 hours, this amounts to 37 days worth of data. However, due to the low bandwidth of the Iridium modem, IMU data would need to be split into packets of 340 bytes. This would require 212 transmissions to deliver a single set of data. This is not advisable due to the high current consumption of the modem as well as the long transmission times. To send all data, advanced compression techniques or a robust wave data processing algorithm needs to be implemented which falls outside the scope of this project. For testing purposes, as a proof of concept, the IMU sample period was reduced to 5.6 seconds which resulted in 28 samples or 336 bytes of data.

5.4.7 Timing requirements

Table 5.18 below shows the timing requirements for each sensor.

Table 5.18: Timing requirements for each sensor showing the interval between samples and the number of samples collected during each session.

Sensor	Sample interval	Number of samples	Sample period
u-blox NEO-7M/NEO-M9N	30 minutes	1	N/A
BMP280	30 minutes	1	N/A
INA219	30 minutes	1	N/A
MPU6050	200 ms	6000	20 minutes

As mentioned in the previous sections, ice drift measurements are sampled once every 30 minutes. **vichi2019effects; alberello2019drift** show that the conditions surround sea ice formation are important for understand sea ice dynamics. Therefore a decision was made to sample the environmental sensors directly after the GPS measurements. Furthermore, power monitoring was performed during each active sample cycle and transmitted with the environmental and drift data. This was done to simplify the timing requirements and reduce complexity in the firmware implementation. The Rock Seven RockBLOCK 9603 modem has a maximum transmission bandwidth of 340 bytes and the transmission periods were sequenced to be able to transmit one message containing as much drift data as possible. However, in future, the full IMU sample period will be implemented followed

by a data compression algorithm. Hence, Figure 5.12 shows how these activities are sequenced into a single buoy cycle.

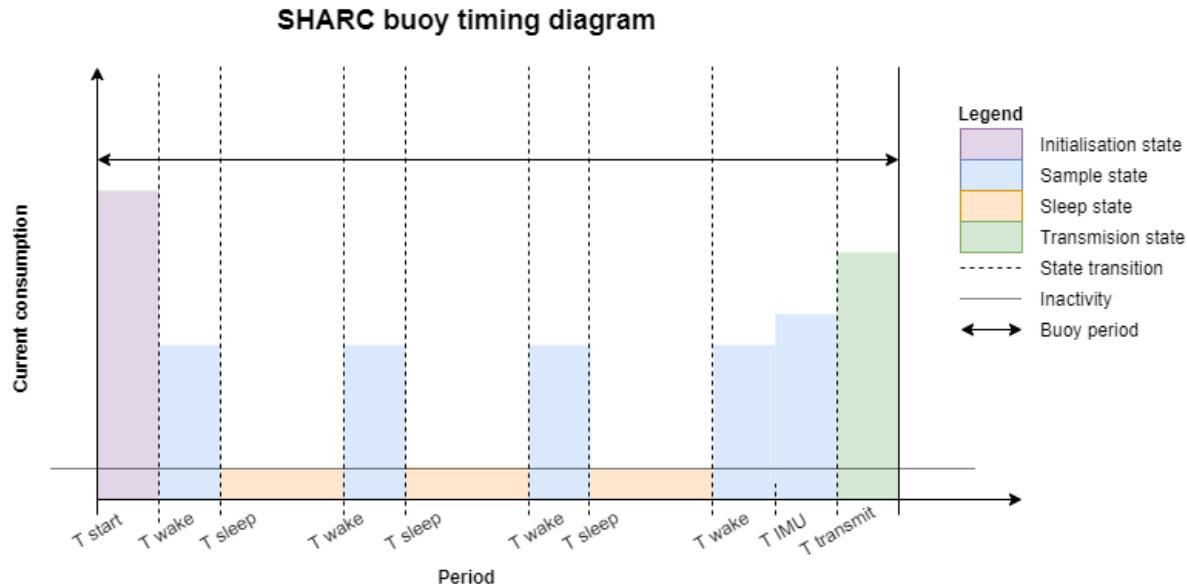


Figure 5.12: Timing diagram for the SHARC buoy showing the sequence and states as well as the expected current draw. The cycle begins with (purple) an initialisation phase where the sensors are verified and configured into the correct modes. Then (blue) the sample periods where the sensors are sampled followed by (orange) a period of inactivity. After four sample sessions, (green) the transmission state is entered where data are condensed into packets and transmitted. Diagram is a representation of the estimated current VS timing and is not to scale.

Figure 5.12 shows that the largest current consumption is expected to occur at start up. This is due to the large start up current of the Iridium modem (see Table ??). The device is also expected to consume more current during the transmission period. Once again, this is attributed to the power characteristics of the Iridium modem. Finally, the buoy is expected to be inactive for periods relatively longer than the periods of activity. This is dependent on the signal acquisition time of the GPS receiver (see Table 4.2) as well as the time taken to complete a transmission. The firmware is therefore developed to ensure careful tracking of state transitions and timings to ensure activities are completed in ordered sequence. Finally the fourth state will be the longest period of activity since this is when the IMU will be sampled.

5.5 Data flow

Based on the information above, drift data are collected every half an hour with a transmission occurring after four sampling intervals. The software interfaces with the GPS, environment sensor and power monitor sensor to create a single drift measurement. Since its important to understand the environmental conditions and give an update on the

power performance of the device where possible. Therefore, the order of sampling is laid out as follows

1. Activate sensor
2. Collect data
3. Order and compress data
4. Store in memory

This sequence is shown in Figure 5.13 below.

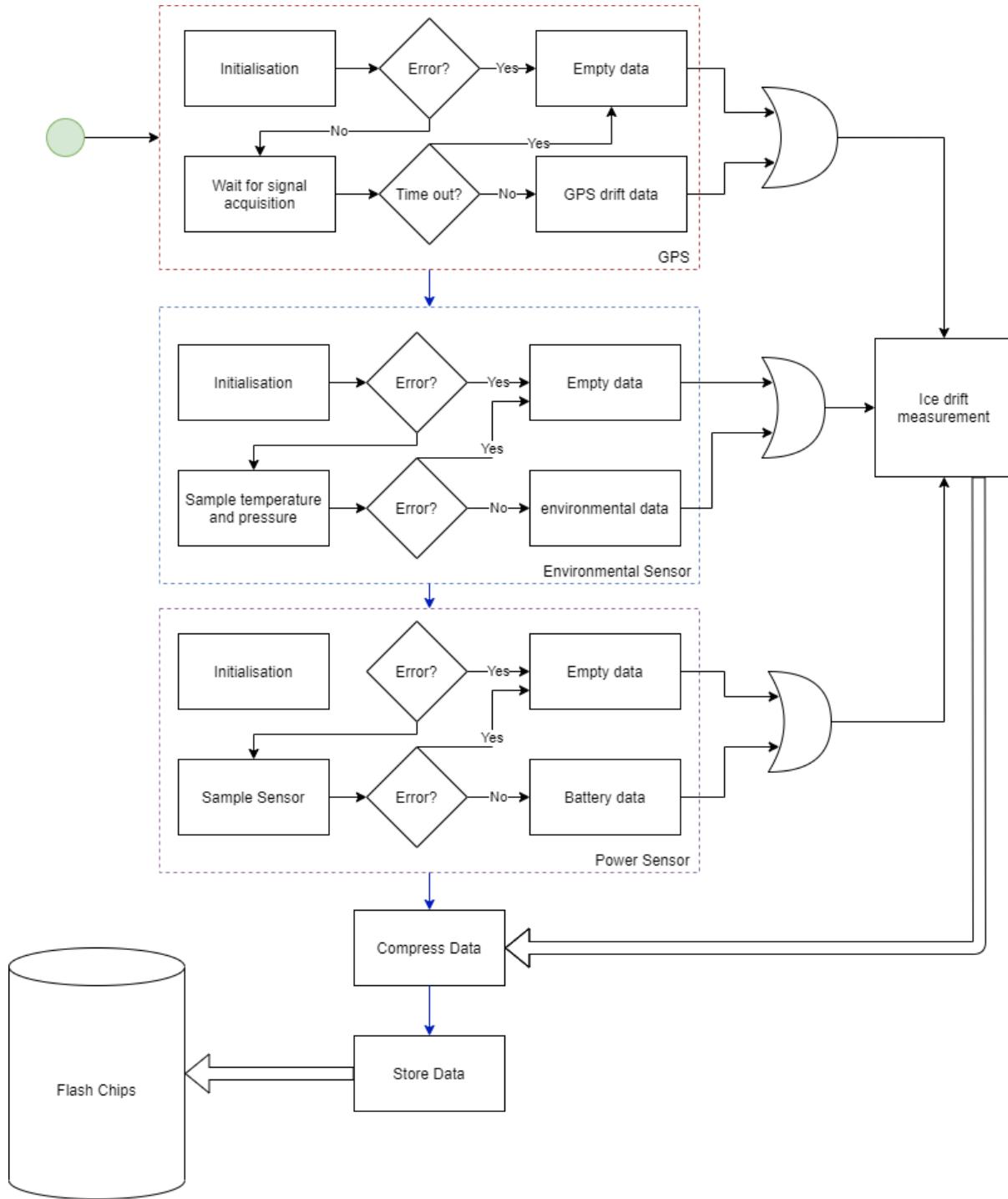


Figure 5.13: Flow chart showing the ice drift sampling routine called during the sample state. This diagram contains arrows showing: (blue) the main flow and sequence of steps in the routine, (black) the flow of steps for sampling data from sensors. (white) the flow of data from temporary storage (as an ice drift measurement) to permanent storage. Also shown are (red) GPS subsystem, (blue) environmental sensor and (purple) power sensor.

The total data collected during a two hour sample period is 140 bytes. The Iridium modem has a maximum transmission buffer of 340 bytes. Therefore, all data can be transmitted at once without any advanced transmission routines required. To accommo-

date centralised data storage in volatile memory, A custom struct was defined to hold all data in a central location.

```

/*
 * @brief: Structure to store data from GPS in an organised
 *          format. Note: custom data types from HAL_GPS.h
 */
typedef struct
{
    uint32_t Etime;      //UTC Epoch representation of time
    Coord_t coordinates; //GPS coordinates
    Diagnostic_t diag;   //Diagnostic information
    uint32_t env_Temp;   //Environmental temperature
    int32_t atm_Press;  //Atmospheric pressure
    int16_t shunt_v;    //Shunt voltage (mv)
    int16_t bus_v;      //Bus voltage (mV)
    int16_t current;    //Load current (mA)
    int16_t power;      //Power consumption (mW)
}GPS_Data_t;

```

Figure 5.14: Data struct for storing drift data collected from the sensors during a sample period where `Coord_t` and `Diagnostic_t` are shown in Appendix C.10 - C.11

The struct is populated with data as each sensor completes its sampling. If a sensor fails or is unable to return valid data, the field is left blank and the program continues to sample from other sensors. This ensures that the program is robust when handling sensor fail error to meet the criteria for acceptance test AT004 in Table B.5. Once the sensors have finished sampling, the data are condensed into a packet structure and stored in memory. The diagram below shows the structure of a single packet.

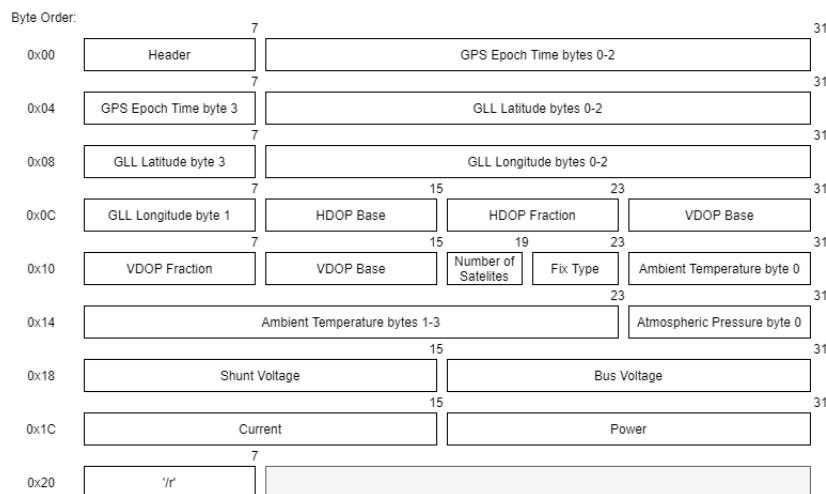


Figure 5.15: Diagram showing the structure of a drift data packet including byte position, size and data being collected

Each packet begins with a header. This is an 8-bit value to give information about the

data in the payload. This value consist of a 4-bit identifier (0x0 for drift data) as well as a 4-bit number indicating the sample number (1 to 4) before transmission. Data are stored sequentially in little endian format as shown in Figure 5.15 above. A '\r' character is used to indicate the end of the packet. This increases the total data requirement from 35 to 37 bytes per sample however, by adding the tail and the header, data integrity is maintained and allow for the standardisation of data transmission.

IMU data, however, is of uniform type therefore, no special structures needed to be created. Data from the IMU is stored in an 8-bit buffer array with the most significant byte of the measurement first. Much like the drift buffer, The data were combined into a packet with a header created at the beginning. The header was given the value 0x57 or "W" to identify the packet as an IMU data packet. Then the data occupies the remaining bytes with the final byte of the packet assigned to the '\r' character to indicate the end of the packet.

Data are stored in the Flash chips in packet structure form in the first page of the first available Flash chip. Packets are stored sequentially until the device enters transmit state. All data are downloaded from memory and uploaded to the Iridium transmission buffer. The device initiates two transmission sessions. First the drift data are uploaded and transmitted, Then the IMU data. Upon successful transmission, the data are sent via satellite network to the Rock Seven RockBLOCK server. The data are saved to a user's account and sent to their email address where the data can be downloaded as an attachment. The diagram below shows the flow of data from the sensors to the user.

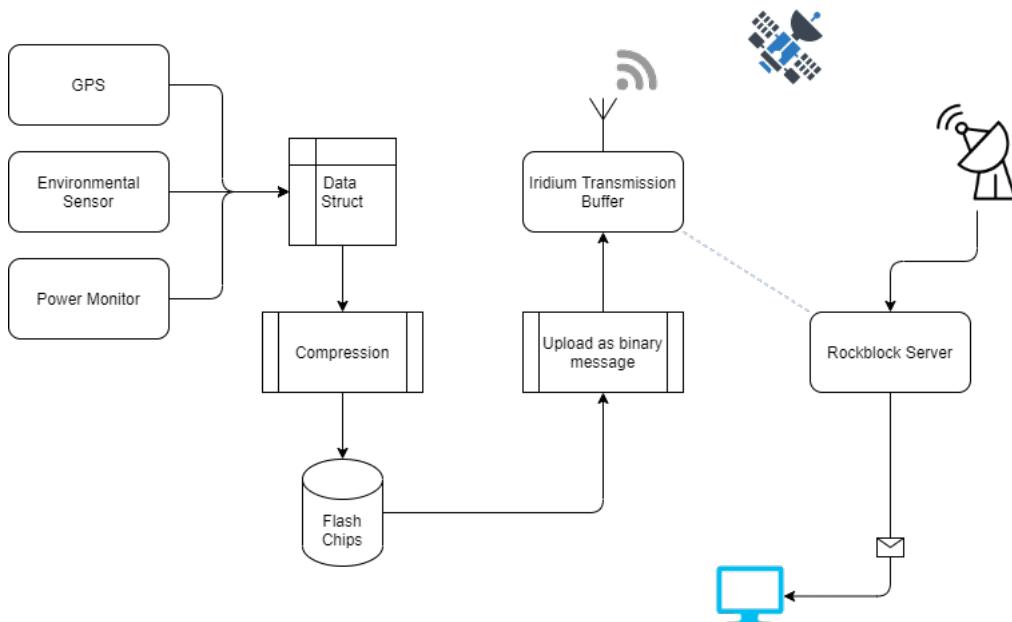


Figure 5.16: Diagram showing the flow of data during a cycle of the buoy. The data are sampled by the sensors and converted into packet form where it is stored until it is ready to be transmitted. The transmitted data arrives at a server and is sent via email to the user.

5.6 Conclusion

In conclusion, a robust set of firmware was designed to sequence and control the sensors in a time critical manner. Furthermore, the communication methods of each module were outlined with strategies shown to facilitate communication and manage the flow of data. Firmware validation and full system testing will be discussed in the next Chapter.

Chapter 6

Testing

From Chapter 5, a final set of firmware was developed for two versions of SHARC buoy. The firmware was written for the hardware platforms described in Chapter 4 to meet the user requirements (see Chapter 3). This chapter outlines the testing procedure undertaken to validate the device on the following levels.

1. low level
2. Subsystem level
3. Full system level
4. Remote test

The testing procedure initiated with low level testing. This was conducted on the firmware with unit tests at each layer (see Figure 5.2) to ensure that the functions written for each subsystem behaved as described in Chapter 4. Then, testing was conducted on a subsystem level to test the base functionality of the module against the system requirements. System level tests were conducted to test the buoy's response in a controlled environment followed by a remote test where the buoy was deployed in an unknown environment to validate the overall performance against the user requirements. Due to time constraints, rigorous data validation tests were not performed. In addition, IMU data and wave simulation testing falls outside the scope of the thesis. All subsystems are tested as a proof of concept using the unit tests outlined in the design methodology and the subsystem tests outlined in Appendix E.1 to E.8. Finally, due to the 2020 COVI-19D pandemic, final system evaluation of SHARC buoy version 2 could not be conducted during a final research expedition to the Southern Ocean Marginal Ice Zone.

6.1 Subsystem tests

In this section, the low level testing is described. As discussed in Chapter 5.4.7, each subsystem has a different data requirement and data rate. The firmware was designed

to cater to these requirements ensuring that the correct communication ports were initialised, the correct protocol was used and that an uninterrupted stream of data were created to prevent blocking and timing issues.

6.1.1 Unit tests

The testing procedure for each subsystem is outlined below. The base functionality of each device is given with the unit test used to verify the functionality. The unit tests were designed using IEEE1012¹ as a structural reference and were written to validate the low level functionality of the subsystems and ensure that each module conforms to the outlined specification. Included in each function is a numeric return status describing how the function exited. Table 6.1 gives an overview of the unit tests shown in Appendix E.1 to E.8.

Table 6.1: Objectives of the unit tests defined in Appendix E.1 showing how the test protocols help validate the firmware in the device peripheral layer.

Unit test	purpose
UT001	Test the initialisation and deinitialisation routine for each hardware sub module. The test ensures the firmware can run a successful initialisation routine, the function can recognise when errors occur and the function can detect when the device is offline.
UT002	Test the wireless modules line of sight by polling for signal strength or polling the device until it acquires data from a wireless source.
UT003	Tests the data flow from the microcontroller through to the submodule by polling to transmit or receive data. Test returns an error code based on how the test exited.
UT004	This test is used to verify functions used to safeguard against data corruption by checking that the function can recognise valid and invalid data based on the implementation of the function.
UT005	Tests the UART DMA circular buffer implementation to ensure the stream does not corrupt data and can handle errors.
UT006	Tests the functions that interface with the sensors to ensure that the data received is valid and no errors occur.
UT007	Tests a firmware function under fail conditions to ensure that the function responds by exiting with the appropriate status code without freezing or generating a hard fault.
UT008	Tests a firmware function designed to interface with a digital logic input/output pin.

6.1.2 GPS

Communication peripheral functionality

The GPS module communicates through a UART port with communication parameters shown in Table 4.2. As discussed in Section 5.4.2, a DMA circular buffer was implemented to provide a seamless stream of data directly to non-volatile memory. An input capture

¹IEEE_STDVV.

channel in slave reset mode was added to the receiver pin to turn the DMA stream off after the line was idle for a defined period. This allowed for a full message of unknown size to be transmitted without polling or generating an interrupt on a byte by byte basis. Table 6.2 shows the baseline functionality required to achieve this and the methods used to verify this functionality.

Table 6.2: Baseline functionality of the GPS UART communication module of the firmware and the unit test used to verify this functionality.

Baseline function	Validation
1 Initialise UART communication, TIM input capture slave reset and output compare, DMA peripheral to memory stream.	UT001
2 Transmit a message through UART DMA stream.	UT003 UT006
3 Receive a message through UART DMA circular buffer.	UT003 UT005 UT006
4 Handle interrupts generated from UART communication stream.	UT005
5 Handle errors from the data stream.	UT007

To ensure the base level functional requirements were met, the device was tested under the conditions outlined in acceptance test AT001 (Table B.2). Then, robustness testing was performed using the conditions outlined in acceptance test AT002 Table (B.3) for fail conditions and AT004 for robustness testing Table (B.5).

Subsystem functionality

On a subsystem level, the system needs to know the device is online and working. This can be achieved by requesting an acknowledgment. To do this, the processor transmits an acknowledgment string to the device. If successfully received, the device will either return an ACK-ACK (acknowledge) or ACK-NACK (not acknowledged). No response means the device is offline while a NACK means the device does not recognise the message sent. Then, the device messages and communication parameters need to be configured. The u-blox NEO-7M and NEO-M9N come preset with a UART baud rate of 9600 bits/s (**UBLOX_M7N_DATA**) and 38400 bit/s (**UBLOX_M9N_DATA**) respectively. The baud rate was increased to 115200 bits/s to allow for faster data reception resulting in a longer idle period between messages allowing for more efficient message detection. Then, the GSA, GLL and ZDA messages must be enabled. Finally, the incoming messages needed to be tested for validity before the data are extracted and placed in an ice drift packet. This results in the following subsystem functionality.

Table 6.3: Baseline functionality of the GPS UART communication module of the firmware and the test used to verify subsystem functionality.

Subsystem function	Validation
1 Request acknowledgement from the GPS.	UT006
2 Configure GPS Baudrate to 115200 bit/s	UT006
3 Configure GPS messages to output ZDA, GLL and GSA messages	UT006
4 Determine whether device has acquired satellite signal	UT002
5 Recieve NMEA ZDA, GLL and GSA message strings	UT005
6 Validate and classify message strings	UT004
7 Timeout if no signal acquired	UT007

Once complete, the full subsystem underwent robustness testing AT004 (Table B.5) and low temperature testing AT008 (Table B.9). Additionally, a power test AT007 (Table B.8) was performed to ensure the power characteristics matched to those given in the datasheet. Finally the positional data were verified using a reference GPS and accurate epoch time counter.

6.1.3 Iridium modem

Communication peripheral functionality

Much like the GPS, the Iridium modem transmits and receives data through a UART port. A circular buffer such as the one for the GPS was implemented for the Iridium modem to allow for efficient data transfer of messages of an unknown length. This would allow for variable-sized messages to be transmitted should any sensors need to be changed or data packets need to be resized. Furthermore, power control was required to keep the modem in a low power state. This was achieved through a digital microcontroller pin connected to the On/Off pin on the modem. The pin needed to stay active low in low power mode to ensure the device was switched off for the full cycle until necessary. This results in the following base functionality:

Table 6.4: Baseline functionality of the Iridium UART communication peripheral of the firmware and the test used to verify unit functionality.

Baseline function	Validation
1 Initialise/deinitialise UART communication on the serial port with interrupt generated when the line is idle.	UT001
2 Transmit a message through UART DMA stream.	UT003 UT006
3 Receive a message through UART DMA circular buffer.	UT003 UT005 UT006
4 Handle interrupts from UART communication stream	UT005
5 Handle errors from UART communication stream	UT006
6 Control the power mode with a digital output pin	UT008
7 Receive interrupts from ring alert pin	UT008

The base functionality was tested in ideal conditions described in AT001 (Table B.2), fault testing in conditions described in AT002 (Table B.3) and robustness testing in AT004 (Table B.5).

Subsystem functionality

On a subsystem level, the firmware controls the RockBLOCK 9603 modem using AT commands. These are command strings that begin with "AT" and finish with a "\r" character. The device comes preconfigured to communicate at 19200 bits/s with no flow control. To ensure the device is working, the microcontroller can request acknowledgement with the command "AT\r" which should return "OK" if successful. The subsystem should therefore be able to interpret the success of a command based on the return function. The modem accepts data in the form of binary or ASCII messages requiring routines to upload data in either ASCII or binary format. Finally, the microcontroller initiates a transmission by sending the command "AT+SBDIX\r". The transmission takes 10 seconds to complete before a return status is returned. Finally the device needs to be put into sleep mode to reduce current when inactive and turned on when required. This was done by interfacing with the modem through the digital on-off control pin. This results in the following subsystem functionality:

Table 6.5: Baseline functionality of the Iridium UART communication peripheral of the firmware and the test used to verify unit functionality.

Subsystem function	Validation
1 Request acknowledgement	UT006
2 Interpret return status*	UT004 UT005
3 Upload AT command	UT006
4 Upload ASCII message	UT006
5 Upload binary message	UT006
6 Acquire network signal	UT002
7 Initiate satellite transmission	UT006
8 Handle transmission errors	UT007

Once complete, the full subsystem underwent robustness testing AT004 (Table B.5) and low temperature testing AT008 (Table B.9), a transmission test and a power test AT007 (Table B.8) to confirm the power characteristics and to ensure that the device was turned off during all periods of inactivity.

6.1.4 Environmental sensor

Communication peripheral functionality

The microcontroller interfaces with the BMP280 sensor through SPI. As shown in the data requirements, the device will be sampling data from the sensor in short burst reads. A maximum of 24 bytes is transferred at a given time and this occurs when reading the compensation registers (**BMP280_Datasheet**). Therefore, the sensor would be operated in a simple polling mode with data read in bursts per the recommendations of **BMP280_Datasheet** to ensure that data do not arrive fragmented. This would also greatly simplify the communication peripheral functionality which is shown in Table 6.6. Finally, the software needs to be able to write to transmit data to specific registers on the sensors to control the configuration, output type and power mode.

Table 6.6: Baseline functionality of the BMP280 SPI communication peripheral of the firmware and the tests used to verify unit functionality.

Baseline function	Validation
1 Initialise/deinitialise SPI communication	UT001
2 Read from an 8-bit register	UT003
3 Write to an 8-bit register	UT003
4 Validate incoming data	UT004
5 Handle errors.	UT007

The functional requirements were further validated using test protocols AT001 (Table

B.2) for connectivity testing , AT002 (Table B.3) for fault testing and AT004 (Table B.5) for robustness testing. This ensured that the baseline functional requirements were met.

Subsystem functionality

On a subsystem level, interactions with the BMP280 occur by reading and writing to the onboard registers. The device can be tested for functionality by reading the value from the *whoami* register. This check ensures the device is online and functional. Furthermore, the BMP280 is fully programmable and can be configured by writing to the *ctrl_meas* and *config* registers at memory addresses 0xF4 and 0xF5 respectively. When a conversion takes place, the status register bits are set to 1 to signify that a conversion is taking place. Being able to read this register will allow for the microcontroller to synchronise with the BMP280 measurement cycle. Finally the factory calibration values need to be read and calculated to calculate the temperature and pressure values from their raw ADC values. This results in the following baseline functionality shown in Table 6.7.

Table 6.7: Baseline functionality of the BMP280 SPI communication peripheral of the firmware and the tests used to verify unit functionality.

Subsystem function	Validation
1 Read device ID.	UT004
2 Configure device.	UT006
3 Set power mode.	UT006
4 Trigger conversions.	UT006
5 Read measurement status.	UT006
6 Read raw ADC values.	UT006
7 Read factory calibration parameters.	UT006
8 Calculate pressure and temperature from ADC values.	UT004
9 Handle errors.	UT007

During testing, the sensor was configured in weather station mode which was recommended for environmental monitoring with sample rates greater than once per second (**BMP280 Datasheet**). These configuration parameters are shown in Table 6.8

Table 6.8: Table showing the configuration parameters for the BMP280 environmental sensor for the final version of the buoy firmware.

Temperature oversample	1
Pressure oversample	1
Infinite impulse response (IIR) coefficients	off
Measurement mode	Forced
Standby time [s]	off

Placing the device in forced conversion mode required the microcontroller to manually trigger a conversion. The standby time measures the time between each measurement completed by the sensor. Since the device is in forced mode, this parameter is not used. When the device was not performing a measurement, it was in sleep mode which conserved

power. The subsystem was fully tested using AT004 (Table B.5) and AT008 (Table B.9) to ensure the device performed under stress and in low temperature. Measurements were validated by placing the buoy in a controlled environment and comparing the measured temperature and pressure to an accurate reference.

6.1.5 Power monitor

Communication peripheral functionality

The INA219 communicates with the microcontroller over I²C. Unlike the BMP280 or the MPU6050, this device has a 16-bit architecture requiring a two-byte read of data. Data are sampled in 30 minute intervals with a single data point read from this device for shunt voltage, bus voltage, current and power. Therefore, in line with the approach taken with the BMP280, the microcontroller will read and write from the sensor in polling mode. The device begins communication upon reception of an address. Any errors that occur during communication result in error codes returned that give more information about the type of issue encountered. The baseline functionality for the INA219 power monitor is shown in Table 6.9.

Table 6.9: Baseline functionality of the INA219 I²C communication peripheral of the firmware and the tests used to verify unit functionality.

Baseline function	Validation
1 Initialise/deinitialise I ² C peripheral.	UT001
2 Transmit data to a 16-bit register on the sensor.	UT003
3 Read data from a 16-bit register on the sensor.	UT003
4 Validate incoming data	UT004
5 Handle errors	UT007

The functional requirements were further validated using test protocols AT001 (Table B.2) for connectivity testing , AT002 (Table B.3) for fault testing and AT004 (Table B.5) for robustness testing. This ensured that the baseline functional requirements were met.

Subsystem functionality

The microcontroller initiates communication with the INA219 by transmitting an address byte to the sensor. This address byte is set by the pins A1 and A0 (**INA219**) and can take on 16 values. For this version, both pins were set to ground generating an address byte of 0b10000000 (0x80). Transmitting this address byte allows the microcontroller to additionally check if the device is online. The device also comes preloaded with a configuration value in the register which resets during a power cycle (**INA219**). This value can also be used to check for previous configurations or corrupted chips. The INA219 is also the only sensor in the SHARC buoy system to require a manual calibration before use. The process for calibrating this sensor is given in **INA219** and was implemented

in the firmware to calibrate the sensor for 16 V bus range and a maximum 1.2 A. The function outputs an LSB value that scales the ADC readings from the sensor to match the actual power information being sampled. The device is fully programmable. Therefore, sensor configuration functions were written to easily configure the sensor through a parametric function. These functions were validated by reading the registers after a value was written to it to see if it had accepted the new configurations. Finally, the device was configured for a maximum bus range voltage of 16 V with an ADC resolution of 12-bits for both shunt and bus voltages. The device was also configured to operate in triggered mode where the microcontroller manually triggers a single measurement in the device. Once complete, the device is placed into sleep mode. This functionality is shown in Table 6.10.

Table 6.10: Subsystem functionality of the INA219 I²C communication peripheral of the firmware and the tests used to verify unit functionality.

Subsystem function	Validation
1 Request acknowledgment from the sensor.	UT004
2 Configure device.	UT006
3 Calibrate device.	UT006
4 Trigger conversion.	UT006
5 Read data and convert from ADC value.	UT006

The subsystem was fully tested using AT004 (Table B.5) and AT008 (Table B.9) to ensure the device performed under stress and in low temperature. Power, current and voltage values were verified by connecting the sensor across a load of known resistance and voltage and measuring the values.

6.1.6 Flash chips

Communication peripheral functionality

The library for the AT45DB641E Flash chips was written by Bowden and was designed to allow the microcontroller to interface with multiple Flash chips on the same SPI port. Each chip had a chip select pin connected to a unique GPIO pin on the microcontroller. This allowed for individual chips to be selected. This requires the microcontroller to keep track of which chips are connected to the lines to keep careful control over the flow of data between the chips. As a safeguard, the write protect pins of each chip were connected to a single control pin. This would prevent data from being corrupted or unexpectedly overwritten. The Flash chips communicate via SPI requiring the library to have the baseline functionality shown in Table 6.11.

Table 6.11: Baseline function of the AT45DB641E Flash chips SPI communication peripheral of the firmware and the tests used to verify unit functionality.

Baseline function	Validation
1 Initialise/deinitialise SPI peripheral.	UT001
2 Initialise chip select and write protect pins.	UT001
3 SPI write to a memory location.	UT003
4 SPI read from a memory location.	UT003
5 Enable/disable chips.	UT008
6 Enable/disable write protect.	UT008
7 Handle errors.	UT007

The functional requirements were further validated using test protocols AT001 (Table B.2) for connectivity testing , AT002 (Table B.3) for fault testing and AT004 (Table B.5) for robustness testing. This ensured that the baseline functional requirements were met.

Subsystem functionality

Each chip is monitored by the firmware to ensure that there is sufficient memory and the devices are online. The chips are given a number and a status written to the very first byte of memory on each chip. When one chip reaches full capacity, the status is set to full and the next chip is formatted and set to active. When the last chip is full, the first chip is selected again and the memory cycle restarts. If a memory chip suddenly stops working, the software will find the next working chip and set that to active. This method was tested by taking chips offline one at a time and writing to the last page of memory in the chip. This method forms a circular data buffer. The number of the active chip is stored in the first byte of the 32-bit back up registers in the RTC. The primary role of the Flash chips is to provide a permanent, ordered storage solution to large data sets. The firmware was designed to manage the data by constantly tracking the amount of data saved to memory and the latest occupied memory address. Memory addresses are three bytes long requiring 3 bytes of storage space in the RTC back up registers. Packets of data are written to Flash memory at the end of every sample routine in sequential order incrementing the memory address by the set number of bytes. The firmware reads the chip data by calculating the difference in the starting memory address and the last known memory address to determine the number of bytes to read and performing a burst read starting at the start address. This subsystem functionality is shown in Table 6.12

Table 6.12: Subsystem functionality of the AT45DB641E Flash chips SPI communication peripheral of the firmware and the tests used to verify unit functionality.

Subsystem function	Validation
1 Read chip status.	UT004
2 Set chip status.	UT006
3 Get memory address.	UT006
4 Set memory address	UT006
5 Read chip data.	UT006
6 Write chip data.	UT006
7 Delete chip data.	UT006
8 Get active chip.	UT005
9 Set active chip.	UT007
10 Handle errors.	UT007

The functional requirements were further validated using test protocols AT001 (Table B.2) for connectivity testing , AT002 (Table B.3) for fault testing and AT004 (Table B.5) for robustness testing.

6.1.7 IMU

Communication peripheral functionality

The IMU communicates with the microcontroller via I²C. The firmware transmitted data via polling methods. However, due to the long sample period required by the IMU (see Chapter 5), this method would not be viable as it would block the CPU responding to other critical functions (See Section 5.3.2). A decision was made to use an interrupt based sampling method by setting an external interrupt line on the microcontroller and connecting it to the interrupt pin on the IMU. Therefore the baseline functionality for this device can be shown in Table 6.13.

Table 6.13: Baseline functionality of the Iridium UART communication peripheral of the firmware and the test used to verify unit functionality.

Baseline function	Validation
1 Initialise/deinitialise I ² C.	UT001
2 Initialise/deinitialise GPIO pin for digital input and external interrupt line.	UT001
3 Write data to 8-bit register on the sensor.	UT003
4 Read data from an 8-bit register on the sensor.	UT003
5 Receive data in an interrupt-based sample stream.	UT005
6 Handle interrupts.	UT005
7 Handle errors.	UT007

Subsystem functionality

The IMU is fully programmable and requires functions to set the sample rate, power mode, digital filter and the full scale resolution. To verify that the device had been configured, the registers were read to ensure the data in the register matched the byte that had just been written. To verify that the sensor was online, a function was created to read the device ID and ensure it matched the value in **mpu6050**. To verify the sensor was functional, a self test was performed. This is a built in feature that can be activated by writing to the Self Test register. During this process, a value is produced which is measured relative to the factory trim values preloaded in the device. The test is passed if the difference falls within the parameters specified by **mpu6050**. In Section 5.4, the IMU was configured to output enough data to fit into a single transmission buffer (336 bytes). A single sample of the 6 axes results in a total of 12 bytes per sample. An interrupt would be generated on the interrupt line which triggered the processor to perform a burst read of the accelerometer and gyroscope data registers on the device. The data were saved to a buffer and the interrupt would be cleared. The device was configured for a 5 Hz sample rate and a full scale resolution of ± 2 g for the accelerometer and $\pm 500^\circ \text{ s}^{-1}$ for the gyroscope. The digital low pass filter was activated to remove high frequency noise and was set to a bandwidth of 94 Hz. Hence, the subsystem functionality is shown in Table 6.14.

Table 6.14: Subsystem functionality of the MPU6050 six DoF IMU and the tests used to verify subsystem functionality.

Subsystem function	Validation
1 Read sensor ID.	UT006
2 Perform self test.	UT009
3 Configure accelerometer.	UT003
4 Configure gyroscope.	UT003
5 Set sample rate.	UT003
6 Configure interrupt pin.	UT008
7 Configure low pass filter bandwidth.	UT003
8 Read raw gyroscope axis.	UT003
9 Read raw accelerometer axis.	UT003
10 Calculate acceleration.	UT004
11 Calculate angular velocity.	UT004
12 Handle errors.	UT007

The subsystem was fully tested using AT004 (Table B.5) and AT008 (Table B.9) to ensure the device performed under stress and in low temperature. However, IMU data handling falls outside the scope of this thesis and therefore was not tested further.

6.2 System tests

The system tests were conducted using version two of the SHARC buoy system. For these tests the following sensors were enabled for ice drift and diagnostic measurements.

1. u-blox NEO M9N
2. Bosch Sesnortech BMP280
3. Texas Instruments INA219

These sensors were sampled at an interval of 30 minutes with the device placed into standby mode in between samples. During the test, the LED was configured to Flash on when the device successfully activated a sensor. This LED was also configured to turn on when the buoy was in an active state. Finally debugging and diagnostic information was transferred to a computer through a USB cable. These data were timestamped and recorded to a log. Finally, the following sensors were enabled for waves-in-ice measurements.

1. TDK Invensense MPU6050

The MPU6050 was set to sample every 2 hours by synchronising the sample interval to four ice drift sample intervals. At the end of the 2 hour period, the transmission routine was completed. All sensors were configured as described in the subsystem tests and the following tests were performed.

6.2.1 SYS001 accelerated system test

Scope

The accelerated system test aimed to test the systems ability to complete a single cycle described in Chapter 5. This will verify the performance of the system against AT005 (Table B.6).

Protocol

The ice drift sample interval was set to 10 seconds with transmission occurring after four sample sessions. Batteries are inserted and the device is placed inside the enclosure. The buoy was placed in an area with an unobstructed view of the sky. The system was left to run for a single cycle from initialisation. Data were recorded from the buoy through a serial debug output to USB which connected to a serial monitor running on a computer to record and timestamp the output.

Configurations

GPS data were sampled at 1 Hz until a full packet of datum, diagnostic and positional information was achieved. The BMP280 and INA219 current sensors were sampled once per sample period and the MPU6050 was sampled at 5 Hz for 67.2 seconds.

Results

A sample of the Buoy output from SYS001 system test is shown in Table 6.15. Data are timestamped debug outputs from the buoy showing the behavior of the system as well as the subsystems. Environmental data are shown in 16-bit signed integer representation as discussed in Chapter 5.

Table 6.15: System output during an accelerated test as described in SYS001. Data output occurs over the period from initialiation to first sample and sleep with a failed GPS signal acquisition.

"11/21/2020 7:47:40 PM",Warning! Device encountered a Brown Out. Exiting Program...
"11/21/2020 7:47:40 PM",Software Reset Detected. Initialising main program...
"11/21/2020 7:47:40 PM",Setting Up Flash Chips...
"11/21/2020 7:47:40 PM",Allocating Chip Statuses...
"11/21/2020 7:47:40 PM",Chip 1 Status: 1
"11/21/2020 7:47:40 PM",Chip 2 Status: 2
"11/21/2020 7:47:40 PM",Chip 3 Status: 3
"11/21/2020 7:47:40 PM",Chip 4 Status: 3
"11/21/2020 7:48:02 PM",Error Connecting To Modem
"11/21/2020 7:48:02 PM",Environmental Sensor Online!
"11/21/2020 7:48:02 PM",Current Monitor Offline!
"11/21/2020 7:48:02 PM",IMU Online
"11/21/2020 7:48:02 PM",Device Online!
"11/21/2020 7:48:02 PM",Current State: RESET Next State: SAMPLE
"11/21/2020 7:48:02 PM",Current State: SAMPLE Next State: SLEEP
"11/21/2020 7:48:19 PM",GPS Online! Acquiring Signal...
"11/21/2020 7:48:49 PM",Failed to Acquire Signal
"11/21/2020 7:48:49 PM",Logging Data...
"11/21/2020 7:48:49 PM",local time: 1605988066, position: 0.000000 Lat, 0.000000 long
"11/21/2020 7:48:49 PM",HDOP = 99.99, PDOP = 99.99, VDOP = 99.160
"11/21/2020 7:48:49 PM",Number of Satellites 0, Fix Type = 1
"11/21/2020 7:48:49 PM",Environmental Sensor Online!
"11/21/2020 7:48:49 PM",Temp = 2346°C Pressure = 101285 Pa
"11/21/2020 7:48:49 PM",Successfully saved drift data to chip 1
"11/21/2020 7:48:49 PM",Current State: SLEEP Next State: SAMPLE
"11/21/2020 7:48:49 PM",Good Night!

Figure 6.15 shows an accelerated test from initialiation to the end of the first sample cycle and sleep mode. The system was unable to acquire a GPS signal and therefore timed out while resuming normal operations. A successful GPS signal acquisition is shown below during the phases: first wake up to the end of second sample.

Table 6.16: System output during an accelerated test as described in SYS001. The data output occurs over the period from first wake up to the end of second sample with successful GPS signal acquisition.

"11/21/2020 8:18:48 PM",Current State: SAMPLE Next State: SLEEP
"11/21/2020 8:18:50 PM",GPS Online! Acquiring Signal...
"11/21/2020 8:18:58 PM",Logging Data...
"11/21/2020 8:18:58 PM",local time: 1605989870, position: -3357.781250 Lat, 1822.458984 long
"11/21/2020 8:18:58 PM",HDOP = 7.50, PDOP = 1.86, VDOP = 7.160
"11/21/2020 8:18:58 PM",Number of Satellites 4, Fix Type = 3
"11/21/2020 8:18:58 PM",Environmental Sensor Online!
"11/21/2020 8:18:58 PM",Temp = 2391°C Pressure = 101215 Pa
"11/21/2020 8:18:58 PM",Successfully saved drift data to chip 1
"11/21/2020 8:18:58 PM",Current State: SLEEP Next State: SAMPLE
"11/21/2020 8:18:58 PM",Good Night!
"11/21/2020 8:48:57 PM",Current State: SAMPLE Next State: SLEEP

6.2.2 SYS002 power test

Scope

The power test aims to verify to test power consumption characteristics of the buoy to determine the survivability of the device in standard operating conditions to verify the software robustness against AT007 (Table B.8).

Protocol

The buoy was connected to a power source with all sensors enabled. The device sampling interval was set to 30 minutes between drift measurements as described above. The device was connect to an Agilent E3630A 0 to 6V, 2.5 A bench top power supply set to output a constant 7.2 V. A data logger was connected to the INA219 sensor and configured to sample current, supply voltage and power consumption. The test ran for a single cycle sampling data at a constant 1 Hz. The logger outputted data to a serial through USB to a computer which recorded the data. The test was completed after a single uninterrupted firmware cycle. This configuration is shown in Figure 6.1.



Figure 6.1: Diagram showing the configuration for the power test. All subsystems were connected and activated with version 2 sensors and firmware. Power was supplied by an Agilent xxxx bench top power supply set to a constant 7.2 V output. A data logger was connected to the INA219 sensor to sample the power information at a rate of 1 Hz

Configurations

The INA219 sensor sampled at 1 Hz while the drift measurements were sampled in intervals of 30 minutes, the IMU was set to a sample rate of 5 Hz for 67.2 seconds.

Results

Based on the test protocol outlined above, data were successfully collected over a complete buoy lifecycle. The current consumption characteristics for a constant load voltage are shown in Figure 6.2.

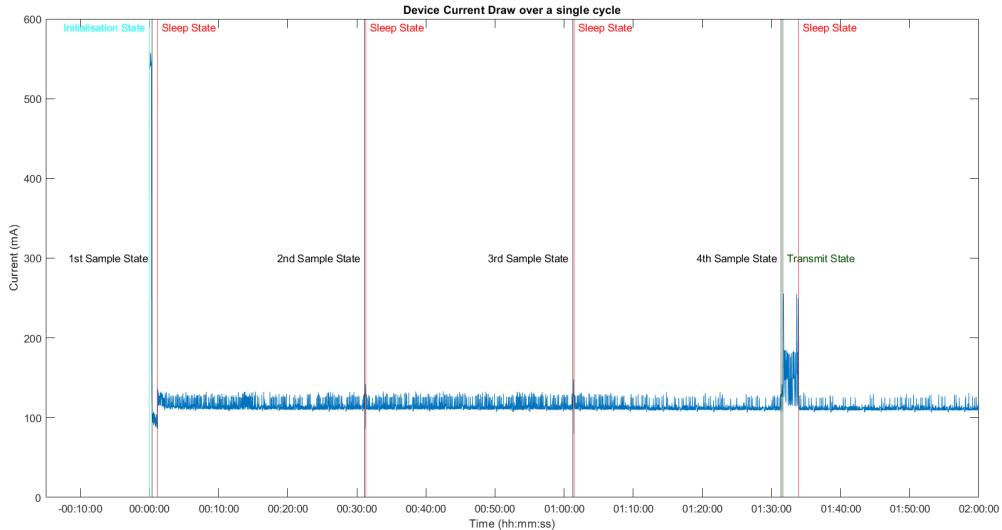


Figure 6.2: Graph showing a typical current cycle of the buoy during the various phases. (Cyan) initialisations phase, (Black) sample phase, (Red) sleep phase and (Green) transmit phase. Data were sampled at 1 Hz with all modules connected, sample intervals set to 30 minutes the INA219 sensor connected to an external data logger and the device placed in a partially obstructed environment.

State transitions were captured through a debug serial monitor and timestamped to synchronise the state changes to the current draw as shown in Figure 6.2. Then, the average current draw was calculated for each phase of the buoy with the period defined as the time between state transitions. The equation to calculate the average current is shown below.

$$I_{avg} = \frac{1}{T} \int_0^T i(t)dt = \frac{1}{T} \sum_{k=0}^N i(k)\Delta t \quad (6.1)$$

where the time step Δt is 1 Hz and T is the total time taken for the buoy to complete 1 sample cycle. Then, the average current consumption and cycle duration was calculated for each phase in the buoy cycle. The results are shown in Table 6.17.

Table 6.17: Average current draw (mA) and cycle duration over the full sample period.

Cycle phase	Phase duration [s]	Average current [mA]
Initialisation State	20	494.37
1st sample state	45	97.79
1st sleep state	1797	115.00
2nd sample state	8	127.96
2nd sleep state	1797	114.41
3rd sample state	7	128.17
3rd sleep state	1797	112.87
4th sample state (including IMU)	12	129.71
Transmit state	135	157.01
Full cycle	10033	114.09

This shows that the buoy completes a cycle in 10033 seconds (2 hours 47 minutes) which is 47 minutes longer than the predicted cycle. The average current consumption was 114.09 mA Finally, the average current consumption for each phase is displayed in Figure 6.3.

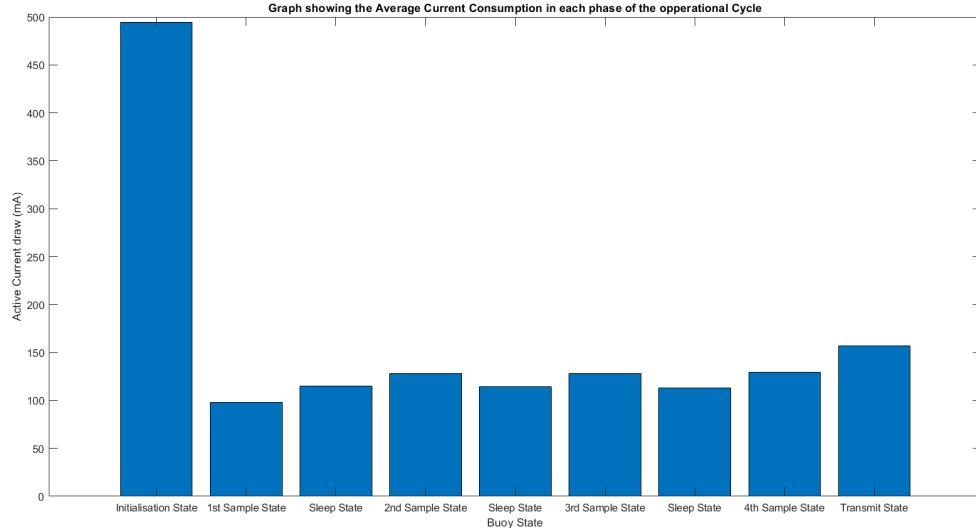


Figure 6.3: Average current consumption at each phase in the operational cycle of the buoy. Ordered chronologically.

From Figure 6.3, the longest current draw occurs at start up at 494.37 mA, the second largest current draw occurs during transmission (157.01 mA) while the lowest current consumption occurs during the first sample state (97.79 mA) with the sleep states drawing consistently the lowest current (112.87 mA to 115.00 mA).

6.2.3 SYS003 low temperature test

Scope

The low temperature test aimed to test the software performance in temperatures below 0° C and verify this performance against AT007 (Table B.8). Successful completion of this test validates a critical user requirement of being able to withstand the climate of the Southern Ocean.

Protocol

All sensors were connected to the system and the Iridium modem was turned off. Sensors were set to the sampling intervals described at the beginning of this section with the transmission routine removed from software. The device was powered with batteries and sealed with desiccant to prevent a build up of moisture. The device was placed in a commercial grade freezer to run for one hour saving data to the Flash chips during its

operation. The freezer was set to -20° C and the buoy status was visually monitored. After the test, the buoy was placed in a room-temperature environment where another accelerated test SYS001 was performed.

Configurations

GPS set to timeout after 30 seconds, sensors sampled in intervals of 30 minutes.

6.2.4 Results

During the test, the Flash chip routine failed to save data resulting in all test data being lost. The firmware contained functions to clear memory when the buoy boots up and enters initialisation phase. When the device was taken out the freezer, these functions may have deleted the data from the Flash chips. Another reason could be that moisture entered the device causing the chips to malfunction and not save any data as a result.

6.2.5 SYS004 field test

Scope

The goal of this test is to deploy the system in a controlled environment to test that the system can complete a full cycle from start up to transmission. This test verifies the length of time the buoy can survive for as well as the transmission capabilities and measured data against AT009 (Table B.10).

Protocol

All subsystems were connected with GPS data, environmental data power consumption data and IMU data enabled for transmission. Ice drift data were sampled in intervals of 30 minutes with IMU data sampled in intervals of two hours. Both IMU and ice drift data were transmitted after the fourth ice drift sample session in two separate consecutive transmissions. The sensors were configured as shown in subsystem with version 2 of the SHARC buoy firmware and hardware. The device was connected to an array of batteries and placed in an environment with good line of sight of the sky. The device was left to run until no more transmissions were received. The device was then removed and the data were analysed.

Configurations

GPS messages sampled every 1 Hz until a valid fix, timeout if no fix reached within 30 seconds. The power sensor and BMP280 sensors were sampled by triggering a measure-

ment once during the interval. Flash chips set to circular buffer mode with the first available chip used to store data.

Results

The data from the device was transmitted to the Rock Seven RockBLOCK data portal. Table 6.18 below shows the data logs downloaded from the data portal.

Table 6.18: Data transmitted from the SHARC buoy as compressed binary messages in ASCII hexadecimal format. This table contains ice drift data shown in packets beginning with 01 ("1") and IMU data in packets beginning with 57 or "W". Also shown are diagnostic data including the transmitting device, payload size and number of credits used to transmit the message.

Date (UTC)	Time	Device	Direction	Payload	Length (bytes)	Credits
12/31/2020 7:51		RockBLOCK 17285	MO	57022600e235f8d84ff84004602b0c008e3642d2aef70004c03c8162364cd21ff6b004d02ac01383690d2d3ff68004e01 92010c3628d2e3ff68004e01f600023672d2f4ff60004f02f001583608d2fbff63004c0396004c37a2d30bf5e004c03dcfc0 36acd313f5e004d039afea3838d31ff58004a02aaef038a0d324ff560048018sfdff393cd32ff5900430028ff6f39a4d3 33ff5e004000effd723a7cd3ccff610039028afe23ab4d341ff67003804eff883ba4d347ff69003402e4ff383b36d344ff6e0 02f0142fc43b2cd351ff78002c014fe723b3ad351ff7b002801c401e63c3ed357ff820024012402083bcd35cff82002101 e000863bdcd35ff86002001eefee43aeaf362ff9001b021001d235ed366ff87001e032c03de3ab8d367ff81001c0294043 a3b2ed36cff81001b02ea01e639d6436ff7b001c022c027a3a5ad37ff78001c0d	338	7
12/31/2020 7:51		RockBLOCK 17285	MO	01008789ed5f51b2a2c524d1cb3c014f01a000622bb90000000d9a0100007c46bc158703eb010d019b90ed5f12e7a2c5e b909b3c012b01a0004f2fc0c00000d5990100007c46b815ec0295010d02af97ed5fad1ca3c582a8db3e013a01a000532fd b00000ac990100007c46b8155303a3010d03c09ed5f2d52a3c5919b1931012301a0004c33e6000000a9990100007c46 b4156303a3012	152	4
12/31/2020 5:50		RockBLOCK 17285	MO	570526f5c39cd770ff4f9001203c6fd2437fed1b7ff3100130304fc2a3a561cdff3900190440fb6d384ed1e1ff40001b04b 6fb8393d21caff440024030ffce0396cd1ff8ff4a002704defdfbe393d0206ff52002c0404fc8839d0d213ff5a003203aaafb23 9fd6219ff620032043efc6439a0d223ff6900330fcfc6b31e2d22eff720036038ef4d839naed235ff7a0038037cfb1e3b4ed23 dff8300380232fa538x4d246ff8800340186f9a3a4ad247ff92003603ccfe0396d24ff9800380320ff23a3ad24ff9d00 370288f6e39d2d256ff6a003602f0fe0438ff2d25bffa9003403fafe943972d25eff200360382ff5239b8d261ff30035033a 010c38e0d265ff70036039800c23a32d26ff66003a026c01a838ead26ff4003d025602de39acd272ff6003e03a023c 3956d275fad0042038402123920d27bfae004200e6047e3992d27bffa6003a0430d	338	7
12/31/2020 5:49		RockBLOCK 17285	MO	0100e96ced5fb4a00c5022b073d015001a0005a2b7400000029a0100007c46c815f60293010d010174ed5fcfd3a0c53a afb13d013e01a0005927710000049a0100007c46e4150ff3c7010d02167bed5f2d07a1c5734bab0e011900a000512b8 20000001d9a0100007c46bc15f302d7010d032b82ed5f8a3ba1c5d13f41bd012c01a00057279a0000002c9a0100007c46 bc150703a301	152	4
12/31/2020 3:47		RockBLOCK 17285	MO	57019c01423c82d6ffff84001bf64ffdc3cd4d127ff6a000ffccfce3bd6d137ff6f0008ffac01a83b76d14cff6bffff0304045 83cbad15bf69000303d402083b24d169ff6e000000dfdee3a2ad179ff6dff0454023039fc1d7ff680006057803c43a04 d188ff6300905ba04303a16d194ff66001106f801303910d19aff650014045cffda37a2d1a1ff64001d0356ffd4373d1aaf 5fe0027056a00dc374ad1aff59002d05f00288382ad1b5ff7003904e028e373d1b9ff51003d023e02d43796d1c1ff4d0 03d026ff436a4d1c2ff4d003d02da02a3d61caff40044603ce01fc37141c9ff3d0043049e00043862d1cff340045023 6ffd036c2d1d4ff39004101a6fe3a375d21d6ff340042016aff743832d1dff36004001a2fe738sc0d1defff35003b01baff24 385dad1ff3d03700c6fe303910l1e0ff4603100c6ff23820d1e4ff52002a0d	338	7
12/31/2020 3:47		RockBLOCK 17285	MO	01006650ed5f4ae49fc5d2981641013301a0005d2768000000ff990100007c46d4150103af010d017d57ed5f950da0c57b 668140014401a001b276a00000e8990100007c46d15ec02b7010d02945eed5fd439a0c5be2363e020b01a001151f6 d0000001990100007c46d015f702db010d03a865ed5f446ca0c5f41abb3b014d01a00060276e000000fd990100007c46 d015f102af01	152	4

Table 6.18 shows a section of the data received during a deployment test. The data includes both IMU data (demarcated with a 0x57 or "W") and ice drift data (demarcated with a 0x01 or "1"). These payloads are compressed data packets as discussed in Chapter 5

6.3 Field testing

This section describes the field testing performed during the 2019 SCALE winter cruise in July. This test occurred early in the development cycle and used version 1 of the software and the hardware for the SHARC buoy system. The cruise took place onboard the SA Aghulas II and followed a trajectory along the Good Hope line towards the

Weddel sea shown in Figure 6.4. This system was deployed with sensors, processors and configurations shown in the table below.

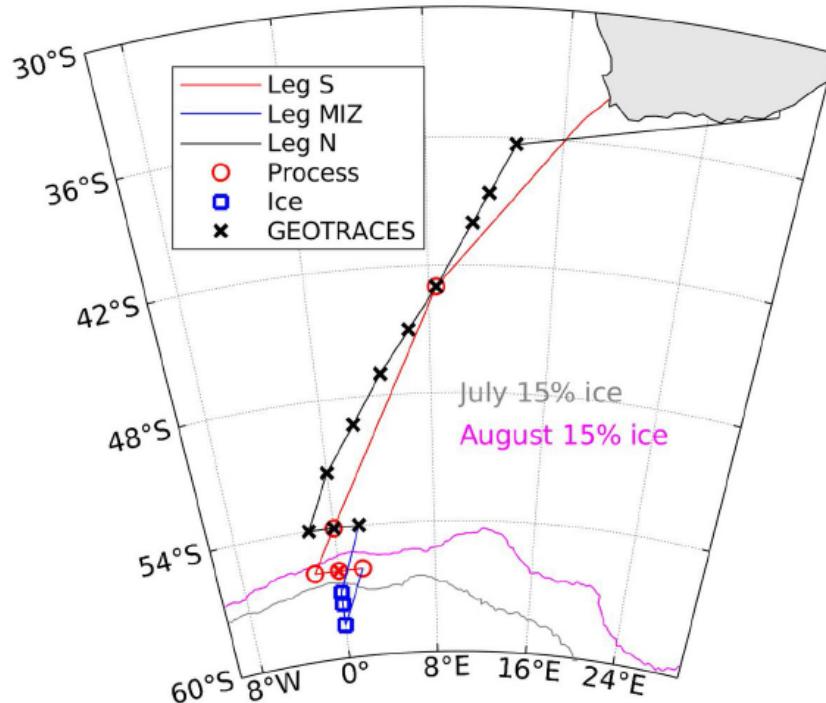


Figure 6.4: Diagram showing the trajectory of the SA Aghulas II for the 2019 SCALE winter cruise from Cape Town to the Southern Ocean Marginal Ice Zone along the Good Hope line. Diagram created by SCALE planning team.

Table 6.19: Hardware components used during field testing for version 1 of SHARC buoy deployed during the 2019 SCALE winter cruise to the Southern Ocean Marginal Ice Zone.

Subsystem	Component
GPS	u-blox NEO-7M
Temperature sensor	DS18B20
Microcontroller	STM32F407VG
Memory	Internal microcontroller flash memory
Remote communication	RockBLOCK 9603

These components were connected to a double sided PCB and mounted vertically to the base. The fully connected system can be seen in Figure 6.5.



Figure 6.5: SHARC buoy version 1 used for deployment testing during the 2019 SCALE winter cruise in July. The device contains an Iridium modem, GPS and temperature sensor for sea ice drift and ambient temperature measurements. Photo by R. Verrinder.

Six prototype systems were brought onboard and carried to the Southern Ocean with the objective of testing the suitability, basic sensing capabilities, remote communication capability and GPS signal acquisition capabilities. During the expedition, the initial power system began to experience instabilities resulting in system failures. Due to time and resource constraints, alternative power supplies were made for three of the systems. Two systems (2019-WC-SB01 and 2019-WC-SB02) were deployed in the first and second Marginal Ice Zone (MIZ1 and MIZ2) respectively. These locations can be seen in Figure 6.6. A final system was deployed on the helideck of the SA Aghulas II on the return journey to East London.

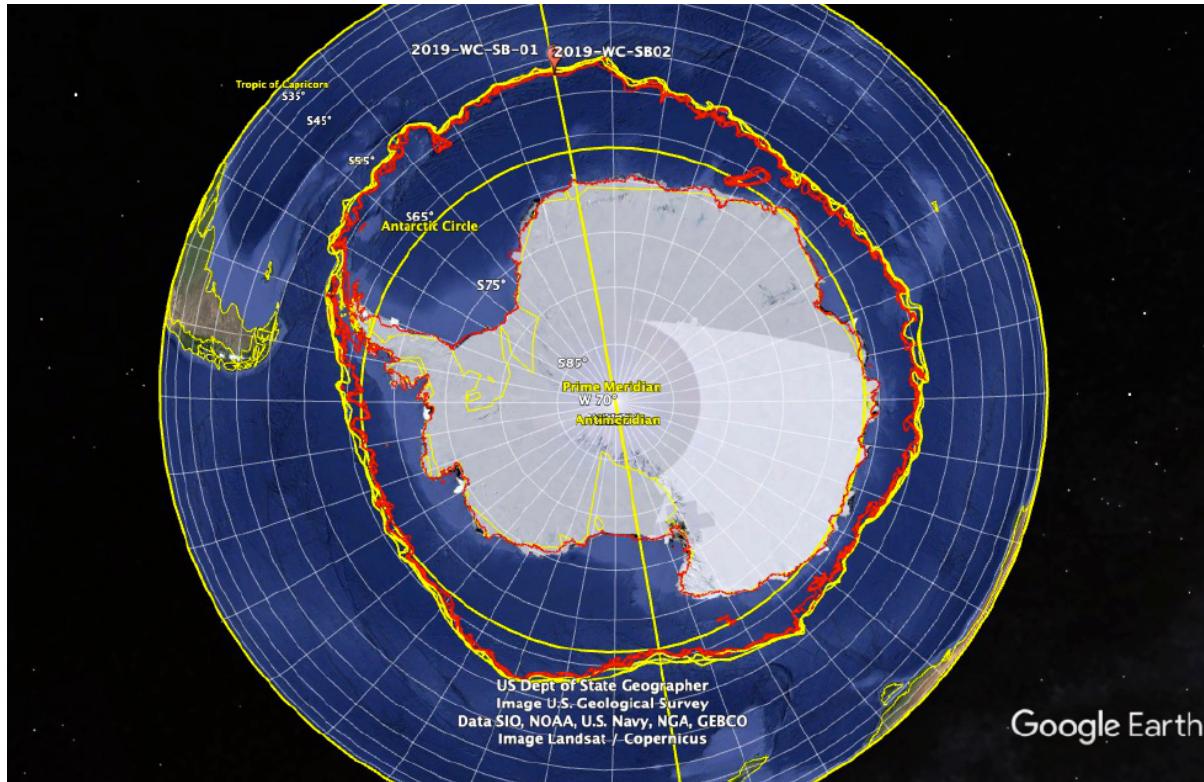


Figure 6.6: SB01 and SB02 buoy deployment locations. Ice edge on 26 to 28 July 2019. Ice edge data provided by E. Hepworth from https://www.natice.noaa.gov/products/kml_daily.html

6.3.1 Configuration

Before deployment, the system was testing using AT001 to AT004 for the subsystems and AT005 for the full system. Low temperature validation was performed on the deck of the ship while near the Southern Ocean. The device was configured to sample GPS and temperature data in intervals of 30 minutes with a single triggered temperature measurement and a GPS reading every 1 Hz until a signal was acquired. The configuration parameters are shown below.

Table 6.20: Table showing the parameters the GPS was configured with before deployment.

Model	Ublox Neo-7M
Baud rate [bits/s]	115200
Data bits	8
stop bits	1
parity	None
Active Networks	GPS, GLONASS
Satellites	3 to 6
NMEA Messages	GLL, GSA, ZDA

Deployment protocol

The buoy was switched on and sealed in the enclosure which was fastened to the buoy stand and placed on the deck of the ship. The buoy was placed in a personnel basket along with three crew members who were fastened to the basket with personnel harnesses. The basket was attached to a crane, hoisted over the side of the ship and lowered towards the surface of the ocean. The crew members then identified a suitable ice floe to place the buoy on. The floe had to have a diameter greater than 2 m and be visually capable of supporting the weight of the buoy (i.e. no visible rafting or flooding). Once an ice floe was selected, the personnel basket was maneuvered to hover 1 m above the desired location. Figure 6.7 shows the deployment of the buoys in the Marginal Ice Zone using this procedure.

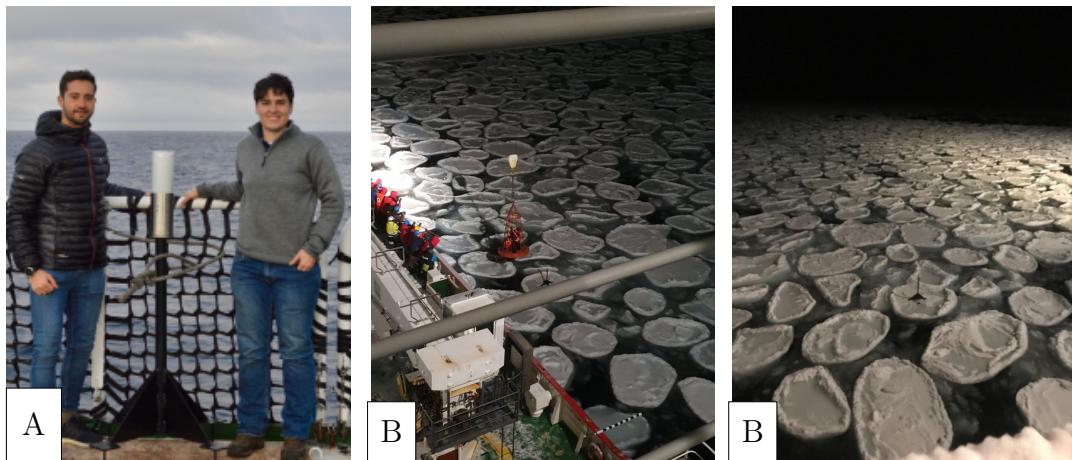


Figure 6.7: Photos taken during field testing of the SHARC buoy system showing (A) the device fully assembled and secured to the stand (photo: R. Verrinder), (B) the device deployed onto an ice floe via a basket and crane deployment procedure, (C) SHARC buoy on an ice floe following a successful deployment.

The buoy was then deployed from the basket with enough force for the spikes to penetrate into the sea ice thereby tethering the stand to the ice floe. Once complete, the buoy tracked GPS coordinates, signal diagnostics, datum information and ambient temperature. The conditions of the deployment are shown in the table below.

Table 6.21: Deployment conditions for buoy 1 (2019-WC-SB01) and buoy 2 (2019-WC-SB02) including deployment coordinates, time and environmental conditions

Buoy serial number	2019-WC-SB01	2019-WC-SB02
Latitude	56°59'59.70'' S	57°17'11.28'' S
Longitude	0°0'36.96'' E	0°1'18.30'' E
Date	26th July 2019	28th July 2019
Time [UTC]	22h15	03h15
Air temperature [° C]	-10.7	-17.5

Results

SB-WC-01 transmitted one message after deployment before losing contact possibly due to a power system failure. SB-WC-02 failed to transmit any messages which may have been due to mechanical failure as a result of deployment. SB-WC-3 survived on the helideck for one week until the batteries were changed. The buoy continued to transmit data continuously. GPS data collected from the transmission packets was compared to the GPS data recorded from the ship and the results are shown in Figure 6.8.

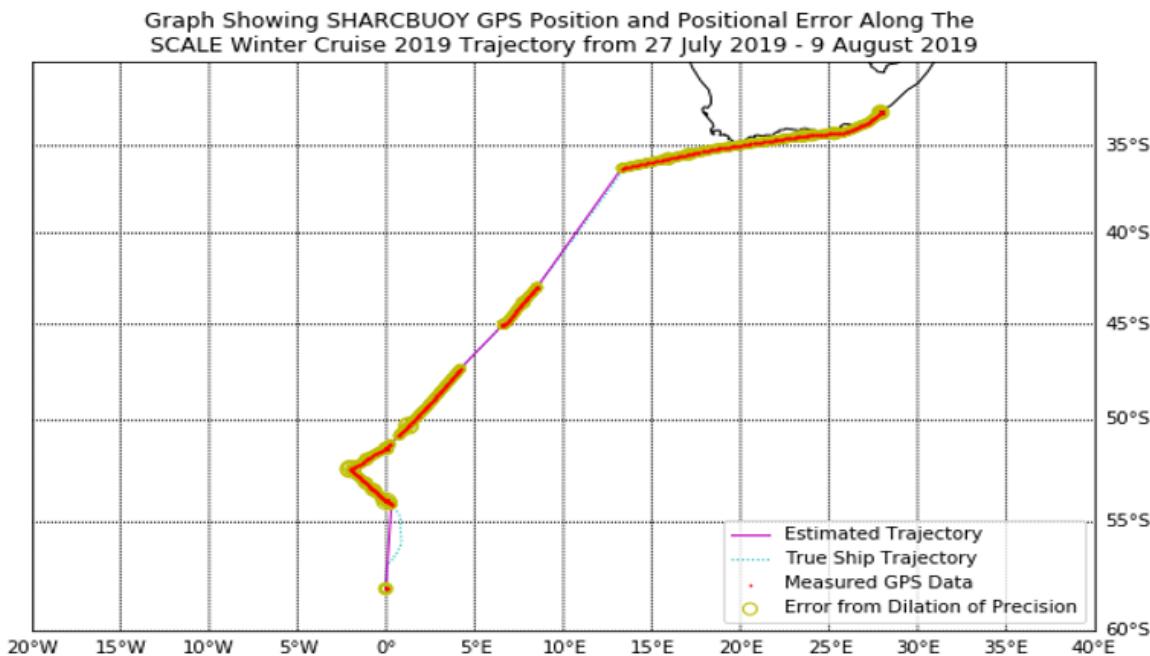


Figure 6.8: The GPS trajectory of the SA Aghulas II ship from the Marginal Ice Zone to East London. The plot shows the estimated position (magenta) taken from the buoy samples (red) compared to the actual trajectory (cyan). The positional error (PDOP) of each measurement is shown as an exaggerated area around the measured position.

Figure 6.8 shows the measured GPS trajectory closely matched that of the ship. However, there were significant gaps in the data set which was attributed to loss of GPS signal or failed Iridium transmissions. The dilution of precision appears the highest between 50 °S and 55 °S. As discussed in Section 2.3.3, this source of error may be attributed to strong ionospheric interference and poor satellite spread.

Finally, Figure 6.9 shows the ambient temperature sampled by the buoy during its journey from Antarctica to East London. The data collected was compared to the data from the ship's onboard weather station.

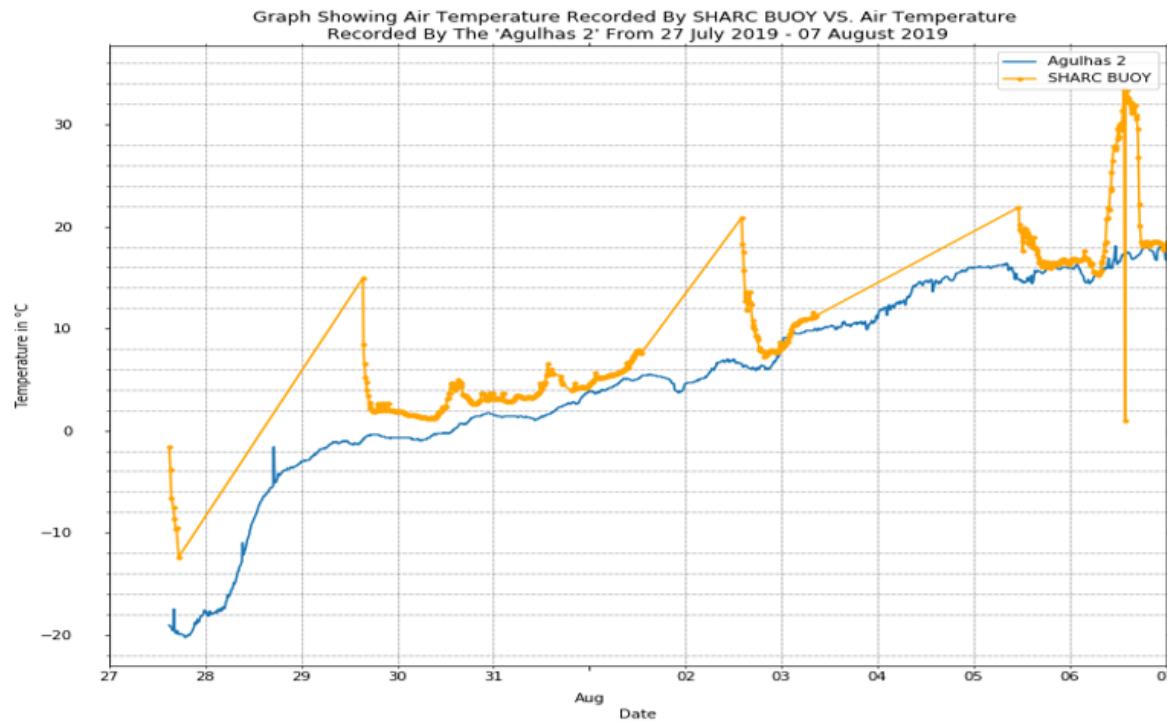


Figure 6.9: Air temperature recorded by the buoy (yellow) over 11 days compared to the air temperature recorded by the ship (blue).

Figure 6.9 shows that the temperature sensor readings show a significant offset from the temperature readings on the ship. Furthermore, the gaps in the data sets lead to "spikes" appearing. The DS18B20 was not calibrated before deployment which may have caused these readings. Furthermore, the DS18B20 may have been unresponsive when the GPS signal was active resulting in missing data points. Finally, while the data points are inaccurate at low temperatures, the data trends towards higher temperatures as the ship reached higher latitudes. This trend was confirmed in the temperature measurements of the SA Agulhas II showing that the device was functional but very inaccurate.

6.4 Final evaluation

In this section, the results of the platform system and subsystem tests are discussed. Due to timeline constraints, no calibration testing was carried out other than for the INA219 current sensor where this was necessary for the device to function. On a full system level, the results of the acceptance tests are shown in Table 6.22.

Table 6.22: Results of the full system acceptance tests indicated by a tick in the appropriate column.

Full system acceptance			
Test ID	Fully satisfied	Partially Satisfied	Not satisfied
AT006		✓	
AT007	✓		
AT008		✓	
AT009			✓

Full system calibration was partially satisfied. The power monitor circuit was calibrated successfully for the power supply and the IMU successfully passed the self-test. However, extensive IMU calibrations could not be performed. The extent of IMU functionality demonstrated by this platform is to prove functionality by initialising and sampling at a fixed known rate. Furthermore, long-term data logging and wave measurements fall outside the scope of this thesis. It is recommended for future research to create an acceptance test for extensive IMU calibrations. Finally pressure measurements are difficult to verify without a calibrated barometer. In future work, verification of the environmental sensor should be conducted at a location with a calibrated weather station.

The system completed low temperature tests in a -20° C freezer and could function normally afterwards. However, due to an issue with the Flash chips, all data from the test was lost. As discussed previously, this may have been due to a software malfunction that resulted in the Flash chips being wiped after the power was turned off. The only indication the buoy was operational was through a debugging LED. In the future, more extensive low temperature tests should be conducted. An additional improvement is to link the device to a data-logger and conducted low-temperature data validation tests to ensure proper operation of the buoy in low temperatures.

6.4.1 Functional requirement validation

Once the testing was completed, the system was evaluated against the system requirements. This section shows how the functional requirements were satisfied in the context of the overall thesis to increase remote sensing in the Southern Ocean. This ultimately proves if the steps undertaken in Chapter 3 had successfully fulfilled the requirements outlined by the stakeholders and evaluate the achievements of the device. These results are shown in Table 6.23.

Table 6.23: Results of the platform evaluation and how each functional requirement was addressed.

Functional requirement	Validation	Discussion
FR001	Fully met	<i>The system shall have a protective enclosure against precipitation and frost.</i>
FR002	Fully met	<i>Enclosure shall be from strong, corrosion resistant materials with strong thermal characteristics.</i> This requirement was validated from the deployment tests during the 2019 SCALE Winter cruise. The buoy continued to transmit during rain, ocean spray and low temperatures thereby providing sufficient protection to the electronics.
FR003	Partially met	<i>The device will protect electronics from internal humidity</i> - This requirement was partially met. When the device transitioned from sub zero temperatures to room temperature, condensation formed both inside and outside the device. While the electronics continued to work, this could result in unexpected failures and needs to be addressed in the next iteration.
FR004	Fully met.	<i>The electronics will be elevated above the ground by 1 m.</i> The buoy stand reaches 1.2 m to satisfy this requirement.
FR005	Fully met	<i>All subsystems shall be rated for extreme temperatures.</i> This was an inherent requirement for component selection which was validated with acceptance test AT005 (Table B.6). The devices would not have been included if this requirement was not met.
FR006	Fully met	<i>System will transmit data via Iridium satellite network.</i> Data were received through the Rock Seven RockBLOCK data portal thereby validating remote communication capabilities.
FR007	Fully met	<i>Device shall be battery powered.</i> The device successfully survived for days on a single battery charge in low temperatures.
FR008	Fully met	<i>Device shall measure ice drift using a global positioning system (GPS).</i> The GPS module successfully acquired satellite signal and produced time stamped coordinates. The average DOP was measured to be 5 below latitudes of 50° S which, according to Table 2.9 is a good measurement accuracy.
FR009	Partially met	<i>Device shall measure ambient temperature.</i> The BMP280 could not be verified in low temperatures. The DS18B20 produced wildly inaccurate values as it was not calibrated properly.
FR010	Partially met	<i>Device shall measure atmospheric pressure.</i> The BMP280 produced a pressure reading. However, the device could not be verified in low temperatures.
FR011	Partially met.	<i>Device shall contain an inertial measurement unit (IMU) to record acceleration (3-axes) and rotation (3-axes) of an ice floe.</i> A proof of concept was implemented with the IMU capable of sampling all 6 axes for a total of 336 bytes of data. This is an insufficient to calculate significant wave height.
FR012	Fully met	<i>Device to contain sufficient memory for data storage.</i> The four flash chips have a total storage space of 8 MB. As shown in Section 5.4, the chips have sufficient storage space to store 152.19 years worth of drift data which is enough to cover a seasonal cycle (barber2005microwave). The chips can store 37 days worth of raw IMU data sampled at 5 Hz for 20 minutes. Which allows for similar data collection periods experienced by kohout2015device who additionally implemented processing.
FR013	Fully met	<i>Device to contain a processing unit to control sensors and process data.</i>
FR014	Fully met	<i>Device to be optimised for low-power consumption and power event handling.</i> The buoy successfully entered low power mode during periods of inactivity which resulted in a lower current draw than the other phases.
FR015	Unsatisfied	<i>Device shall be factory calibrated prior to shipping and delivered in a state where it can be deployed at a moment's notice.</i> The device entered the main loop and started running as soon as the power was on. However, the sensors were insufficiently calibrated to fully meet this requirement.
FR016	Fully met	<i>Device will cost less than currently available systems.</i> - The overall cost for a single system is R8,421. From the review of remote sensing devices, a comparison of device cost is shown in Table 2.5. From this comparison, the SHARC Buoy is much cheaper than WIIB (rabault2019open) which costed R30,200 and had similar features. However, this device used more reliable components and implemented more complex data processing strategies resulting in better wave estimation calculations than the SHARC buoy.

6.5 Discussion

6.5.1 Power requirements

The Initialisation state was the most power intensive state drawing 494.37 mA. This was attributed to the Rock7 Rockblock 9603 modem which has a reported startup current of 450 mA to charge the onboard super capacitors. The effects of placing the modem to sleep can be seen throughout the data in Figure 6.2 and Table 6.17 where the average current barely increases above 130 mA. The effects of putting the buoy to sleep mode when the device is inactive results in a significant drop in current consumption as the average current consumption during sample mode was roughly 10 to 15 mA larger than the current consumption during sleep mode. However, this is not true for the first sample state which results in the lowest current consumption at any point in the operational cycle of the buoy. However, since this phase occurred so close to the power up of the buoy, it may have been a result of the GPS searching for satellite signal which consumes less current than when in acquisition state (**UBLOX_M7N_DATA**).

The fourth sample state had the largest average current draw and the longest phase duration of all the sample states. This was to be expected as the inclusion of IMU sampling results in a longer data acquisition time as well as a higher current consumption. Finally, The transmission phase was expected to have the second highest current consumption since the Iridium modem was turned back on. At this point, the current draw increased to 250 mA as shown in Figure 6.2 and occurred twice during the transmission phase. Despite this spike in consumption, the average current over the transmission phase was 157.01 mA despite multiple transmission attempts.

The duration of each state has a significant impact on the average current consumption of the buoy. While the initialisation state current and the transmission state had significantly higher current consumption, the phase duration of these states were significantly smaller than the sleep states. The long periods of inactivity dominated the power cycle resulting in an average current of 114.09 mA. The duration of the sample states were small as a result of fast data acquisition and sampling speeds. However, the first sample state had the longest duration. This was due to a failure to acquire a GPS signal within 30 seconds which resulted in a timeout. The fourth sample state also had a relatively long phase duration due to the inclusion of the IMU in the sample routine. Finally, the longest, active state was the Transmit State. During this state, multiple attempts were made to successfully transmit a packet of data and failed resulting in the relatively long phase duration. Overall, it took 10033 seconds or (2 hrs 47.217 min) whereas each sleep state was found to be extremely consistent. This shows that the sample and transmit states have a non-negligible duration which can affect the accuracy of the sampling resulting in time delays and desynchronisations. These can be attributed to sensor measurement time, GPS signal acquisition time, and the Iridium transmission time which runs for 10 minutes. Furthermore, the sleep periods were set using fixed, full time periods counting from when the sample phase is concluded. Therefore, to account for this, the system must measure the time taken between wake up and the end of the sample phase and remove this from the inactive period counter to synchronise the sample state to the required sampling interval.

6.5.2 System performance

To verify the sensor accuracy during the deployment test. Time, data, temperature and positional data sampled at 1 minute intervals was acquired from the SA Aghulas II. Figure 6.8 shows that data collected from the buoy's GPS correlated well with the data from the ship. However, large gaps appear in the buoy's data-set. This can be attributed to signal loss or failure to acquire GPS position. In addition, the positional error appears larger for coordinates greater than 50°S and smaller as the trajectory approaches East London. This could suggest that the GPS satellite signal is much weaker closer to the Antarctic continent and may be attributed to either the strength of the antenna or the spread of GNSS satellites in the region.

Figure 6.9 shows that the temperature measured by the sensor was wildly inaccurate. However, the sensor was not calibrated before deployment which resulted in the inaccurate measurements. Additionally, missing packets resulted in large "spikes" in the data. The data, however, does show a trend towards warmer temperatures which is also reflected by the ship data. Therefore, the sensor was able to characterise the change to warmer temperatures however, the data are too inaccurate to be valid. The data from version 1 of the buoy was captured with the DS18B20 thereby showing it was not practical for this application. The new sensor (BMP280) could not be verified by remote testing due to cancellations of the 2020 Antarctic expedition.

6.5.3 Mechanical features

The mechanical features of the system successfully met the functional requirements FR001, FR002, FR004. FR003 was partially met as preliminary freezer tests resulted in condensation both inside and outside the system. In spite of this, the electronics continued to work. However, a revision of the design should be made to reduce the internal humidity of the system when it transitions from a sub-zero environment to room temperature.

The mechanical features of the system, while robust, were quite bulky and heavy. The stacked PCB design allowed for robust, modular development of the system and resulted in increased mechanical strength. However, the result was increased physical size of the device and increased cost. For future iterations, a single PCB with all the components should be created. By reducing reliance on off-the-shelf development boards, the performance, size and power consumption of the system can be more carefully controlled. Moreover, a single PCB design requires less physical hardware to secure the system to the enclosure such as hex spacers, screws and washers. This can significantly reduce the price of fabrication. This design was also found to create points of failures within the device. By having separate PCBs, additional wires were required to connect the boards. If a wire loses contact or breaks, the device stops working. Finally, by reducing the electronics size, more batteries could be included which can provide more power for the system.

6.5.4 Power system

The power system was the largest constraint to the device. The physical size of the enclosure limited the number of batteries included in the system and therefore lifespan of the buoy. The initial decision was to use LiSoCl₂ D cell batteries. These were chosen for their high specific energy and low temperature resistance. Two 3.6 V batteries were connected 2 in series and 2 in parallel resulting in 7.2 V into the LDO. However, in low temperature environments, the internal resistance of the batteries dropped significantly resulting in system brownouts and unexpected resets. These were exchanged for AA LiFeS₂. These batteries had higher stability at lower temperatures at a cost of significantly reduced specific energy. In addition, four cells were required in parallel to produce the required voltage thereby increasing the battery requirement.

The result was a maximum survivable period of 8 days in low temperatures < 0° C and 10 days in standard temperature. The seasonal requirement for operation is at least a month. To meet this requirement, the power system requires significant revision. The average load current was estimated to be 114.09 mA over a two hour cycle. The total energy requirement $114.09 \times 30 \text{ days} \times 24 \text{ hours}$ or 82,114.8 mAh for a period of thirty days. Future improvements to meet this requirement would be to use batteries with higher specific energy, couple the power system with an energy harvester or use a rechargeable power source. Additionally, the load current can be reduced significantly by implementing more power saving features such as MOSFET switches to turn off unneeded sensors or configuring devices such as the GPS for power saving mode.

6.5.5 Future work on wave measurements

Section 1.1 shows that waves in ice strongly affect the formation and dynamics of sea ice in the Southern Ocean. **vichi2019effects**; **albarello2020drift**; **kohout2015device**; **rabault2019open** also show the importance of increasing remote sensing of waves in the Marginal Ice Zone. Therefore, for this thesis, IMU integration was a fundamental step. The IMU was successfully integrated into the thesis. However, the sampling requirements resulted in extremely large data sets requiring complex data management algorithms. Moreover, the constraints of the Iridium data buffer significantly impacted the type of data that could be transmitted. The recorded time series requires compression algorithms/software processing algorithms which fall outside the scope of this thesis. Therefore, in terms of the thesis goals, the IMU only partially satisfies the requirements and more firmware development is required for wave data measurements to become fully realisable.

The MPU6050 is a low cost, 6 axis inertial measurement. This more than satisfies the requirements for analysing waves in terms of spectra, co-spectra and significant wave height as shown by **kuik1988method** and **earle1996nondirectional**. The majority of devices in the field use high precision, expensive IMUs with low cost devices similar to the MPU6050 to verify the measurements. This shows that there is still room for investigation into the accuracy and performance of low cost IMUs for complex functions.

6.5.6 Short-burst data modems VS. Telephone modems

The current version of the SHARC buoy uses a short-burst data modem with a maximum transmission buffer size of 340 bytes. This resulted in extreme data constraints which reduced the functionality of the system. Despite this complexity, the device was well integrated into the system and was able to reliably transmit data even through the enclosure. Short Burst Data are a very data limiting protocol and is not a feasible solution for real-time, raw IMU data transfer. In future version, the Iridium 9522A would be a more feasible solution as the data buffer is much larger (1960 bytes). Alternatively, an Iridium device with a SIM card or continuous real time data transmission protocol.

The modem required the most design consideration. The device dominated the current sample and had the highest current consumption of all components. Hence, the majority of software optimisation was focused on optimising the power cycle of the device. Despite having an extremely large current cycle, the average current consumption over a two hour cycle was reduced significantly therefore successfully meeting the functional requirements.

6.5.7 Evaluation against the State of the Art

The final evaluation for the system was against other devices in the field. Most of these devices have been field tested to a larger extent than this device and have a higher technological readiness level. Significantly more testing is required to verify the field performance against the operation of the system.

However, SHARC buoy consumes significantly less power than the majority of devices in the field. The mode supply voltage is 12 V with some devices using a supply of up to 18 V compared to the buoy's 7.2 V operating voltage.

Finally, while the SHARC buoy has more primitive modules on board, the device can, more evenly, measure a wider range of variables. Most devices generate complex measurements from single modules such as a high-powered IMU or AHRS measurement system. Devices such as WIIOS and WII buoy only contain low powered modules to compliment the measurements of the higher-powered components. This provides a unique opportunity for SHARC buoy to provide a deeper insight into performance optimisation in this region. Overall, the system shows that it is unique and fits a niche as a low powered, modular sensing device however, more rigorous tests and calibrations are required to bring the device to an overall state of technological readiness.

6.6 Conclusion

In conclusion, the device was evaluated on a unit level, a subsystem level and a full system level to verify the functionality against the user requirements. Field testing was conducted during the 2019 SCALE winter cruise where the device performance was tested in unknown conditions to determine the viability of the device in the Southern Ocean climate.

Chapter 7

Conclusion

From Chapter 6, the capability of automated systems as a solution for long-term in situ remote sensing was shown through the testing of the first iteration of the Southern Hemisphere Antarctic Research Collaboration (SHARC) buoy. The extensive design methodology resulted in the procurement of a set of robust firmware which tested the first and second hardware generation of the device. The design process was heavily guided through active engagement with the key stake holders which lead to a set of user requirements to verify the performance of the system. A detailed set of specifications was derived allowing for component selection to take place. The buoy structure was designed to be modular allowing for fast, prototyping phases and long, testing phases in the lifecycle. A single, processor architecture was adopted. Hence, the firmware was designed to control the subsystems, sample and process sensor data as well as handle power events. A set of acceptance tests and unit tests were written to validate the firmware thereby ensuring robust performance.

7.1 Key contributions to firmware and validation

The software functional units were validated using unit tests while subsystem functionality was successfully validated against the acceptance tests. Furthermore, the integrity of data streams was maintained through techniques such as circular buffers, DMA streams and the implementation of an algorithmic state machine to sequence and control sensor functionality. Overall system performance was validated by a full system test suite followed by field tests and deployment tests. The firmware successfully handled non-critical errors and failures. Controlled exits and error code evaluations resulted in robust communication with the sensor successfully retrieving data under non-ideal circumstances. The device was optimised for power consumption by setting a relevant processor power mode for each distinct phase of the cycle. This resulted in a significant decrease in current consumption during inactive phases. Extensive calibration testing was not performed as part of the project scope. Therefore, this requirement was only partially satisfied. Device performance during freezer tests showed promise. However, more extensive testing is required to fully validate this performance.

7.2 User requirement verification

The average active current, and sleep current over a cycle was still extremely high failing to meet the current consumption outlined in specification SP012. The key contributors were the Iridium Modem's start-up current and the GPS operational mode. Additionally, the size of the mechanical enclosure physically constrained the size of the power source resulting in the requirements for survivability being left unsatisfied.

In spite of this, the full system software was successfully verified against the functional requirements of the project. The IMU was verified as a proof of concept. Additionally development is required to implement a wave measurement algorithm.

7.3 Full system testing

The project concluded with a long term testing phase at home. The project encountered heavy time constraints due to the timing of the expeditions. In addition, due to the COVID-19 pandemic, all Antarctica expeditions were canceled for 2020. Therefore, extensive field testing could not take place. Ultimately, the device passed the long term deployment test being able to execute code from start up and successful complete multiple sample cycles at the required sample frequency while accounting for signal acquisition and sensor integrity. The testing ended with the reception of data packets in the Rock7 RockBLOCK portal with the packet structure and integrity maintained. The full system is currently expected to be deployed in early 2021 through a German-led Southern Ocean expedition to the Marginal Ice Zone in the East Antarctic sea.

7.4 Verification's against the state of the art

In comparison to other devices, the system requires a higher level of technological readiness fully compare the performance. However, preliminary results show that the prototype has a lower procurement cost and lower operating voltage than devices with similar specifications. These devices have more complex structures than SHARC Buoy and include higher-powered sensors that contribute to the expense of those project.

7.5 Final Remarks

In conclusion, the work presented by this dissertation successfully lays the foundation for future work and expansion of the SHARC Buoy project to take place. Significant revision to the power system and sensor routines are required to bring the device closer to completion. Given more time and development, SHARC Buoy can create a strong presence in Antarctic as a multi-use system providing. Thereby contributing to improving remote sensing on a spatial and temporal scale of the life cycle of sea ice in the Southern Ocean Marginal Ice Zone.

Chapter 8

Recommendations

8.1 Improvements to the power system

As discussed in Section 6.5.1, the power system requires significant revision to improve the operational time of the buoy. Using batteries with a higher specific energy can be a viable solution. Additionally, the power system can be revised to include a boost converter thereby allowing more batteries to be placed in parallel increasing the capacity of the power supply. An investigation needs to be conducted into the use of energy harvesters or renewable sources to compliment the power the power source. This can significantly improve the buoy's life cycle.

8.2 Improvements to hardware

The current PCB stack configuration provides too many points of failures. It is recommended to design a single, horizontally mounted PCB with all the sensors and micro-controller. Additionally, low-powered LED arrays can be implemented to provide better visual feedback on the status of the buoy.

The device enclosure should be redesigned to allow for a larger power supply. In addition, the enclosure should include a mechanism for de-humidifying the internal electronics.

Finally, a dedicated power board and communication module board should be designed to replace the breakout boards that the GNSS and Iridium modem modules arrive on. This will greatly reduce the form factor and reduce the reliance on connectors that can act as points of failure.

8.3 Improvements to the communication modules

The gain of the GPS antenna should be increased to provide a higher positional accuracy and shorter acquisition time. Furthermore, the RockBLOCK 9603 module should be

replaced with either a SIM card based modem or an SBD modem with a larger data buffer.

8.4 Firmware improvements

The main control layer of the firmware was designed as a state machine. This technique is somewhat primitive since states are executed sequentially. This results in time delays as shown in Table 6.17. This can be revised by implementing a Real Time Operating System (RTOS) for critical time optimisation.

Additionally, the firmware power optimisation strategy needs to be expended to configure the GNSS for power saving modes. This will significantly reduce the current consumption of the overall system.

In this version of the firmware, a simple set of unit tests were implemented to verify the connectivity of the subsystems. It is recommended for future versions to include more extensive calibration tests for each subsystem built into the firmware or run through its own routine

Due to time constraints, a fully realisable wave data algorithm could not be implemented. This can be expended on in a future project conducting an investigation into the most suitable wave-measurement algorithm and can include full IMU calibration techniques, evaluation of the current IMU as well as open-ocean and open-ocean with rigid platform tests.

The latest hardware platform allows for critical components such as the IMU to connect an interrupt pin to an internal wake up line on the microcontroller. This feature can be expended in the future to allow for interrupts to be generated when a specific event is detected. These features can be expanded based on the following devices.

GPS The device can be put to sleep and woken up when a GPS signal is acquired. Thereby reducing the reliance on polling for signal acquisition.

IMU The interrupt pin can be configured to detect motion of a specified magnitude and frequency which can allow for more precise detection and measurement of significant wave height and dominant wave frequency. This feature can also be expanded to detect Ice Collisions.

Iridium The RockBLOCK 9603 has a ring indicator pin which produces a logic high when a message is incoming to the modem. This feature can allow for ad-hoc programming of the device as well as asynchronous data retrieval thereby allowing for more precise monitoring of the device.

The move to a fully interrupt-based system will significantly improve power performance as well as reduce the reliance on timed-sequences.

8.5 Expansion of nodes into a network

This project resulted in the design and procurement of a single sensing node. To increase the sensing capability, the devices can be expanded to form a network with an additional communication protocol (such as LORA) can provide inter-buoy communication. Future projects can include an investigation into optimal buoy topologies or designing firmware to facilitate inter-buoy communication.

8.6 Future deployments

Following the completion of the new design, arrangements are being made to test and deploy the device through other groups. Contacts have been made with Alfred-Wegener Institute to deploy the device from the Polarstern Research Vessel. An additional prototype has been taken onboard the SA Aghulas II transporting researchers and supplies to the SAAE IV base where the device will be tested on the continent before being deployed on sea ice.

Appendix A

Numerical modelling

A.1 Modelling of polar stochastic processes

In this section, an analysis of modelling techniques for the polar region is provided. Here, the focus is on developing models for Polar sea ice mechanics and dynamics. An overview of these models is given with a description of the variables and the scope of each model.

A.1.1 Numerical modelling of sea ice

The Hibler model is a numerical designed to investigate sea ice dynamics and thermodynamics in the Arctic region (**hibler1979dynamic**). This model attempts to couple the sea ice dynamics to Sea ice thickness and uses this relationship to investigate the relationship between the effects of sea ice and the climate. Work so far has largely studied these effects independently using factors that largely ignore the inherent mechanical properties of Sea Ice (**hibler1979dynamic**). Coupling these effects would allow for a more general descriptor of Sea Ice spread regions.

The model is based off **coon1974modeling** AIDJEX (**hibler1979dynamic**), who use plastic-elastic constitutive laws to describe large-scale sea Ice spreads. It is assumed that cracks, ridges, and leads are randomly distributed on large scales¹. While the Hibler model is not as complex, it is more robust as it allows for larger time-steps and simplifies system boundaries. Here, sea ice is modelled using similar viscous plastic laws (**hibler1979dynamic**) that allow for non-linear plastic flows to be modelled without severe limitations by large time-steps. The model uses the following components:

1. Momentum balance - air and water stress
2. Coriolis force
3. Inertial forces

¹100 km from **coon2007arctic**

4. Constitutive laws - ice stress, strain, strength
5. Ice thickness distribution - accounting for open water patches, changes in thickness and Concentration
6. Ice strength

$$\frac{mDu}{Dt} = -m fk \times u + \tau_a + \tau_w - mg \nabla H + F \quad (\text{A.1})$$

$\frac{D}{Dt}$ is the substantial time derivative, k is a unit vector, u is the sea ice velocity, m is the ice mass, and f is the Coriolis parameter. Forces in the equation τ_a , τ_w represent the respective stresses of the air and water. F is the force related to the internal ice stresses. H is the sea surface dynamic height, and g represents gravitational acceleration. Assuming constant turning angles, The air and water momentum equations are as follows

$$\tau_a = \rho_a C_a |U_g| (U_g \cos(\phi) + k \times U_g \sin(\phi)) \quad (\text{A.2})$$

$$\tau_w = \rho_w C_w |U_w - u| [(U_w - u) \cos(\phi) + k \times (U_w - u) \sin(\phi)] \quad (\text{A.3})$$

where ρ_a and ρ_w are the densities of air and water, C_a/C_w are the drag coefficients, U_g is the geostrophic wind, and U_w is the geostrophic ocean current

The Hibler model is the de facto numerical model for large scale ice process (**Rutger2019SmallScale**). The model is used to describe an area of 10 - 100km². However, small scale models are still in development (**Rutger2019SmallScale**).

Numerical modelling of ocean waves

Ocean wave data is described by using spectral components with different magnitudes and wave periods. These spectral components are important in understanding the wave attenuation model (**williams2013wave**) where, assuming the ice is modelled as a viscous fluid, wave energy is exponentially attenuated (**meylan2014situ**) (**williams2013wave**) with distance travelled into the ice due to partial reflections with the ice floes. The rate of attenuation is dependant on the wavelength. However, an exact mathematical relationship has not been found. The issue with verifying these models is the lack of robust data availability (**meylan2014situ**). Thereby reaffirming the need for increased in situ measurements.

williams2013wave describe three fundamental components of Waves in Ice Modeling. These are advection, attenuation, and ice breakage (**williams2013wave**). Advection and attenuation describe how energy transfer occurs between waves and ice and are dependant on the group velocity c_g , and the attenuation factor $\hat{\alpha}$. These values are dependant on the frequency of the wave (**williams2013wave**). Also, the properties of ice are significant. These include Young's modulus Y , Poisson Ratio ν , strain ϵ and viscous damping parameter Γ . Furthermore, the initial floe size distribution and sea ice

concentration in a region are considered. The assumption is that wave breakage feeds back into the model with a new floe size distribution (**williams2013wave**).

Wave advection is described by the following energy model:

$$\frac{1}{c_g}(\partial_t + c_g \partial_x)S(\omega; x, t) = R_{in} - R_{ice} - R_{other} - R_{nl} \quad (\text{A.4})$$

Where R_{in} is the wind input energy, $R_{ice}, R_{nl}, R_{other}$ represent the energy loss from ice, other sources as well as non linear energy exchanges. $S(\omega; x, t)$ represents the waves in terms of its energy spectral density (**williams2013wave**) For this model, the energy input is considered to come only from the rate of exchange between ocean and ice. Hence all other energy rates are considered 0 and R_{ice} is defined in terms of $\hat{\alpha}$ and S

$$\frac{1}{c_g}(\partial_t + c_g \partial_x)S(\omega; x, t) = -\hat{\alpha}(\omega, c, h, \langle D \rangle)S(\omega; x, t) \quad (\text{A.5})$$

$\hat{\alpha} = \frac{\alpha}{\langle D \rangle}$ describes the average attenuation per ice floe in terms of ice thickness and wave period (**williams2013wave**). By this definition, R_{ice} is quasi linear (**williams2013wave**) since a wave with a significantly large energy spectral density can break the floe decreasing the dimensions $\langle D \rangle$ and increase the dimensional attenuation factor $\hat{\alpha}$. The operator $(\partial_t + c_g \partial_x)$ serves as the lagrangian reference fram at a moving velocity c_g . Finally, by breaking the above model into:

$$\frac{dx}{dt} = c_g(\omega, t_*, x) \quad (\text{A.6a})$$

$$\frac{dS(\omega; x, t)}{dx} = -\hat{\alpha}(\omega, x, t_*, S_*)S(\omega; x, t) \quad (\text{A.6b})$$

we can describe the dynamics of the sea ice during a breaking event at a time t_* (**williams2013wave**).

The next step is determining the mathematical model for wave energy. A stochastic approach is taken to define key wave parameters (**williams2013wave**). The significant wave height is found using the formula

$$H_s = 4\sqrt{m_0[n]} \quad (\text{A.7})$$

$m_n[\eta]$ describes the mean square surface sea elevation of a particle and is derived from the Spectral Density S (**williams2013wave**).

$$m_n[\eta] = \int_0^\infty S(\omega)\omega^n d\omega \quad (\text{A.8})$$

The significant wave height can be considered four times the standard deviation of the surface elevation (**meylan2014situ**). Finally, by determining the significant wave height, the dominant wave period can be calculated as $\frac{1}{f_d}$ where f_d is the frequency at which the dominant wave period occurs (**meylan2014situ**).

A.2 Modeling of GPS dilution of precision

Given a user's position on the earth, the distance from the user to the satellite is characterised by the following equation.

$$r = s - u \quad (\text{A.9})$$

where r is the distance from the user to the satellite, s is the distance from the earth's centre to the satellite and u is the distance from the earth to the user. By measuring the propagation time from the user to the satellite, The absolute distance $\|r\|$ can be calculated and hence, the pseudo-range can be calculated as

$$\rho_i = \|s_i - u\| + ct_b + v_{\rho_i} \quad (\text{A.10})$$

where ρ_i is the pseudorange for satellite i , c is the speed of light, t_b is the clock offset and v_{ρ_i} is the noise of the pseudorange measurement and:

$$\|s_i - u\| = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} \text{ for } i \in 1, 2, 3 \dots N \quad (\text{A.11})$$

where N is the number of satellites and (x_i, y_i, z_i) is the 3-dimensional position of satellite i . This represents a non-linear relationship for the line of sight from a receiver to a satellite. **jwo2001efficient** explains that by creating a Taylor series centered on a nominal user position $(\hat{x}_n, \hat{y}_n, \hat{z}_n)$ and ignoring the higher terms (**jwo2001efficient**). It then follows that:

$$\Delta\rho_i = \rho_i - \hat{\rho}_i = e_{i1}\Delta x_u + e_{i2}\Delta y_u + e_{i3}\Delta z_u \quad (\text{A.12})$$

The terms e_{ij} represent the line of sight vector E_i whereas the term $\hat{\rho}_i$ represents the pseudo-range at the nominal user's position. It follows that the vector E_i can be calculated as follows (**jwo2001efficient**).

$$e_{i1} = \frac{\hat{x}_n - x_i}{\hat{r}_i} \quad (\text{A.13a})$$

$$e_{i2} = \frac{\hat{y}_n - y_i}{\hat{r}_i} \quad (\text{A.13b})$$

$$e_{i3} = \frac{\hat{z}_n - z_i}{\hat{r}_i} \quad (\text{A.13c})$$

$$\hat{r}_i = \sqrt{(\hat{x}_n - x_i)^2 + (\hat{y}_n - y_i)^2 + (\hat{z}_n - z_i)^2} \quad (\text{A.13d})$$

Given n satellites, the equation (A.11) can be written as a matrix with the following form:

$$\mathbf{z} = \mathbf{Hx} + \mathbf{v} \quad (\text{A.14})$$

$$\Delta\rho_i = [\Delta\rho_1 \ \Delta\rho_2 \ \Delta\rho_3 \ \dots \ \Delta\rho_n] \quad (\text{A.15})$$

where

$$\mathbf{H} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & 1 \\ e_{21} & e_{22} & e_{23} & 1 \\ e_{31} & e_{32} & e_{33} & 1 \\ \dots & \dots & \dots & 1 \\ e_{n1} & e_{n2} & e_{n3} & 1 \end{bmatrix} \quad (\text{A.16a})$$

$$\mathbf{x} = \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ c\Delta t_b \end{bmatrix} \quad (\text{A.16b})$$

$$\mathbf{v} = \begin{bmatrix} v_{\rho_1} \\ v_{\rho_2} \\ v_{\rho_3} \\ \dots \\ v_{\rho_n} \end{bmatrix} \quad (\text{A.16c})$$

The matrix \mathbf{H} is $n \times 4$ where $n \geq 4$ to calculate all the parameters for GDOP (**jwo2001efficient**). We can then solve for the vector \mathbf{x} by taking the psuedoinverse of \mathbf{H} i.e. $\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{v}$. Hence, given that the psuedorange is linearised, the quality of navigation is taken as the difference between the estimated position and the actual position (**jwo2001efficient**).

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{v} \quad (\text{A.17})$$

$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\}$ describes the covariance between the errors in the components of the estimated position (**jwo2001efficient**) and is calculated as follows.

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T E\{\mathbf{v}\mathbf{v}^T\} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H} \quad (\text{A.18})$$

Where $E\{\mathbf{v}\mathbf{v}^T\} = \sigma^2 I$. If all components of σ are uncorrelated then the covariance become

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \sigma^2 (\mathbf{H}^T \mathbf{H})^{-1} \quad (\text{A.19})$$

and thus the GDOP factor can be calculated from the RMS values of σ^2 i.e.

$$GDOP = \frac{\sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 + \sigma_{tt}^2}}{\sigma} \quad (\text{A.20})$$

where $\sigma_{xx}^2, \sigma_{yy}^2, \sigma_{zz}^2, \sigma_{tt}^2$ are the RMS values of the x,y and z time components respectively. The value GDOP can also be decomposed into the positional dilution of precision (PDOP), time dilution of precision (TDOP), horizontal dilution of precision HDOP, and vertical dilution of precision (VDOP) which characterise the effects satellite spread on the 3-dimensional position, time, horizontal position and altitude respectively.

A.3 Numerical techniques for modelling ocean waves

A.3.1 Kuik method

The Kuik method is a computational technique used for measuring and determining the directional characteristics of ocean waves from the pitch and roll of an ocean buoy

(**kuik1988method**). By using an accelerometer, gyroscope, or an inertial measurement system to measure the slope and heave of the three axes (**kuik1988method**), it is possible to reconstruct the sea state given a set of data of a specific length sampled above the Nyquist frequency of dominant ocean swells. A major advantage of the Kuik method is that the parameters are estimated directly from the Fourier transform of the measured signal (**kuik1988method**) without assumptions about the measurement platform model. Therefore this technique can be applied to wave in ice measurements because no information is required about the dynamics of the model of the ice floe. This greatly improves the accuracy of the wave estimations since ice floes can vary in width, distribution and area as well as change shape due to collisions, freezing and melting. Wind waves are described using a two-dimensional energy spectrum E with wave energy spread over a frequency f . The normalised distribution of energy over direction is defined according to **kuik1988method** as

$$D_f(\theta) = \frac{E(f, \theta)}{\int_0^{2\pi} E(f, \theta) d\theta} \quad (\text{A.21})$$

Finally, by computing the model per frequency, the distribution simplifies to $D(\theta)$ which can be approximated by a Fourier series with four terms (**kuik1988method**) derived from the pitch, roll and heave of a buoy. Finally, the model can fully characterise the wave spectrum by calculating the following parameters from the Fourier coefficients:

1. Mean wave direction θ_0
2. Directional width σ
3. Skewness γ
4. Kurtosis δ

The accuracy of the mean wave direction and width is affected by noise in the sampled data. small RMS values of noise can result in rapid increases of directional width by 1% to 5% (**kuik1988method**). Additionally, pitch and roll buoys are not free particles. They have an associated mass and, therefore, an associated inertia (**kuik1988method**). This results in a phase shift of the Fourier term by $\phi_i i \in x, y, z$ in the first harmonic (**kuik1988method**). This shift can result in an error of 0.5° for $\sigma > 25^\circ$ to 1° for $\sigma < 10^\circ$ (**kuik1988method**)

A.3.2 Welch-Earle method

The Welch-Earle Method is an algorithm for calculating either directional and non-directional wave data depending on the assumptions of the input data (**earle1996nondirectional**). Data is derived from the vertical acceleration, and the roll and pitch of a buoy oriented perpendicular to the surface on either a vertically stabilised platform or a hull-fixed platform (**earle1996nondirectional**). Directional wave data is determined from both the acceleration, roll, pitch and heave from the buoy while non-directional wave data is calculated from time-series acceleration only. In this method, a digital time series

representation of the vertical acceleration along with two orthogonal Gyroscope measurements and magnetometer readings relative to the earth's magnetic field is obtained. This method accounts for the response function of the buoy. Thereby providing corrections to phase differences caused by the inertia of the buoy (**earle1996nondirectional**). Directional and non-directional wave data is characterised by calculating the spectra and co-spectra of discrete-time series data. The first part of the method was developed by **welch1967use** and is used to calculate the power spectral density. Given a discrete time series data $X(j)$ with a power spectral density $P(f)$, $|f| < \frac{1}{2}$, this data segmented into a set of k -bins $X_k(j) | j \in 0, L - 1$ (**welch1967use**). Each bin is multiplied by a selected window function $W(k)$ of length L . Additionally, bins are taken with a 50% overlap to produce better statistical averages (**earle1996nondirectional**). The Fast Fourier Transform (FFT) of the result is taken to for a periodograms I_k (**earle1996nondirectional**). Finally, the new power spectral estimator $\hat{P}(f_n)$ is calculated by taking the average of the K periodograms as shown in **welch1967use**

$$\hat{P}(f_n) = \frac{1}{K} \sum_{k=1}^K I_k(f_n) \quad (\text{A.22})$$

where

$$f_n = \frac{n}{L} | n \in 0, 1, 2, \dots, \frac{L}{2} \quad (\text{A.23})$$

Detrending is used to account for the effects of buoy motion on the time series. The inertia of the platform results in non-zero mean trends often as a result of constant wind or currents acting on the hull of the buoy (**earle1996nondirectional**). These must be discarded before the spectrum and co spectra are calculated. The resolution of the accelerometer is important for accurately tracking acceleration (**kohout2015device**). If the resolution is too small, low accelerations will not be recorded resulting in incorrect vertical accelerations being calculated. **kohout2015device** found that a low-resolution IMU was unable to reliably flag that it had exceeded a boundary condition and hence it was discarded (**kohout2015device**).

The spectra and co-spectra of the directional and non-directional wave series can be calculated by computing the spectrum $S(x)$ as a function of frequency and direction (as is similar to the Kuik method). **earle1996nondirectional** show that characterisation of the co-spectra $C(x)$ and spectrum $S(x)$ can be achieved by calculating the first four Fourier coefficients. Finally, the sea state can be represented by calculating the following parameters

1. Longuet-Higgins directional parameters
 - (a) a_0
 - (b) a_1
 - (c) b_0
 - (d) b_1
2. Significant wave height H_0

3. Dominant wave period T_p
4. Total degrees of freedom TDF
5. Average zero-crossing period T_{av}
6. Zero-crossing period T_{zero}

This approach brings into account the possibility of spectral leakage however, this can be greatly minimised by sampling above the Nyquist frequency of the upper Wave frequency band (generally taken to be 0.5Hz) for a minimum of 1000 seconds (about 16 – 17 minutes). Additionally, spectral leakage can be reduced by selecting a window function with a gradual taper such as a half cosine or Hanning taper (**welch1967use**)

A.4 Temperature sensing measurement techniques

Thermistors

Modern thermistors have progressed significantly in the past decade. Until recently, they had been considered inaccurate with uncertainty ranges of up to 5% (**tong2001improving**). Thermistors are capable of providing accuracies of up to 0.01° C. They consist of a semiconductor that changes its resistance in response to temperature (**child2000review**). They have a faster response time than their resistive counterpart: the resistive temperature detector (RTD) which, works on the same principle for temperature measurement. However, where RTDs have a positive temperature coefficient, thermistors have a negative temperature coefficient (**tong2001improving**). These devices can operate over a substantial, albeit relatively limited, range of $-100^\circ - 300^\circ C$. The major trade-off with these devices is the lack of standards (**tong2001improving**). Operating the device involves a large degree of uncertainty. Also, these devices are not powerful enough to accurately reach the desired ranges alone and require additional circuits. Finally, the response curve is nonlinear. The relationship between resistance and temperature is (**child2000review**):

$$R_T = R_0 e^{1-B(\frac{1}{T} - \frac{1}{T_0})} \quad (\text{A.24})$$

where R_T is the temperature measured, R_0 is the resistance at a known temperature of T_0 , T is the temperature, and B is a coefficient based on the inherent properties of the thermistor. Finally, these devices are more prone to noise from excitation current.

Silicon temperature devices

Semiconductor temperature devices (STDs) are suited to applications where the temperature ranges from -55 to 150 ° C. These devices provide a stable output with a typical accuracy of 0.8° C. These devices typically consist of diodes and transistors with a bandgap voltage that changes with a change in temperature (**child2000review**). STDs are advantageous in electronic application due to their small form, high accuracy and stability. These devices are relatively simple and have a good sensitivity to changes

(**childs2000review**). Diodes are typically used in semiconductor devices. Here, the forward voltage drop across the p-n junction is linearly proportional to the Ambient temperature over a certain temperature range (typically 25K - 400K) (**childs2000review**). These devices consist of either silicon or Gallium-Arsenide. Silicon is preferred as it has better stability at low temperatures and is cheaper. However, this comes at the trade-off of a lower voltage output (**childs2000review**).

These types of devices are readily available in integrated circuit (IC) forms and are manufactured in a variety of packages, types and compositions for any application. Typical devices are DS18B20, LM355 or BMP2080. Recent innovations in Silicon sensing have seen the rise of CMOS devices and Micro Electrical-Mechanical Systems (MEMS) used more frequently (**mansoor2015silicon**). While these devices can suffer from deterioration due to self-heating, **mansoor2015silicon** discuss that the low-power operation of these devices can offset this issue which is advantageous for systems that constrained by power consumption. However, a major disadvantage with these devices is that STDs work ideally with a purely DC signal as an AC coupled signal can cause significant errors in the output (**childs2000review**) (**mansoor2015silicon**). These errors can be the result of improper shielding and poor grounding. Hence proper shielding and grounding are required to reduce these errors. Finally, these devices require careful calibration before use.

A.5 Pressure sensing techniques

Diaphragm based sensors

The current state of pressure sensing technology is driven towards the miniature MEMs version of large scale devices (**eaton1997micromachined**). Most large scale pressure sensors consist of a diaphragm mounted on a device in a known way. The diaphragm is coupled to a device that converts the pressure to a mechanical movement which is then measured using a gauge. These sensors often had a secondary sensor that converts the mechanical movement to an electrical signal which is measured (**eaton1997micromachined**). These sensors determine pressure by measuring the deflection of a miniature diaphragm. This deflection is converted to an electrical signal. Typically, a reference pressure is provided as a measurement of a sealed chamber or absolute pressure port. Assuming the simplest version of this sensor i.e. a plate of uniform thickness (**eaton1997micromachined**) The deflection w of the diaphragm is related to the pressure P by the following equation: (**eaton1997micromachined**)

$$w(r) = \frac{Pa^4}{64D} \left(1 - \left(\frac{r}{a}\right)^2\right)^2 \quad (\text{A.25})$$

where r is the deformed radius of the diaphragm, a is the original radius and D is the rigidity of the diaphragm governed by the equation:

$$D = \frac{Eh^3}{12(1 - v^2)} \quad (\text{A.26})$$

where E, h, v are Young's modulus, thickness and Poisson's ratio of the disc (**eaton1997micromachined**). This technique suffers from a multitude of problems. The diaphragm is susceptible to plastic deformation, while more robust diaphragms result in more complex relationships. The current relationship is nonlinear and can result in calculation errors. Eaten (1997) advocate for the use of MEMs based electronics on these principles.

Piezoresistive sensors

Piezoresistive sensors are electric devices constructed out of a semiconductor whose electrical properties change when stress is applied (**eaton1997micromachined**). These devices are mounted to a diaphragm and exhibit a linear change in resistance with a change in pressure. Currently, these sensors take the form of single-crystal diaphragms with piezoelectric resistors diffused through the materials. The advantage of these devices is robustness towards hysteresis and measurement drift. At low temperatures, silicon exhibits near-perfect elastic behaviour and is three times the tensile strength of strain gauges(**eaton1997micromachined**). However, these sensors are susceptible to thermal expansion resulting in temperature measurement drift (**samaun1971ic**). Additionally, these sensors require resistors with identical temperature resistance characteristics, or the pressure measurement will be inaccurate. Finally, compensation techniques are required to calibrate the accuracy of the sensor.

Capacitive sensors

These sensors consist of parallel conductive plates. Assuming a constant, known dielectric, an external pressure causes the plates to deform which changes the capacitance C according to the relationship (**eaton1997micromachined**)

$$C = \int \int \frac{\epsilon}{d - w(r)} dr d\theta \quad (\text{A.27})$$

Where $w(r)$ is the deformation of the plate, ϵ is the strain experienced by the plate, and d is the distance of separation. The pressure capacitance relationship can be approximated using a least-squares fit (**eaton1997micromachined**). However, this results in model errors of 1.5% and up to 11% at $w = \frac{1}{2}h$ for the plate height. These sensors are more advantageous over piezoresistive sensors as they have higher pressure sensitivity and reduced susceptibility to temperature drift. However, these sensors are significantly susceptible to parasitic capacitance, which can result in losses and errors. Additionally, these sensors are simple in design. However, they tend towards more complex circuit requirements.

Appendix B

Project design

B.1 Stakeholder analysis

For this project, a stakeholder is defined as any user or set of users who will impact the overall project or be impacted by the final design of the project (**varvasovszky2000stakeholder**). The stakeholders for this project can be considered as an individual directly involved in the designing/building of the project or a user: i.e. an individual responsible for using the system or any data it generates. These project stakeholders are shown in Table B.1.

Table B.1: Table showing the key stakeholders in the project, their level of involvement as well as their interests in the project.

Stakeholder	Function	Involvement
Lead scientist	Principal stakeholder: Initiates and funds the SHARC buoy project.	Sets the user requirements, provides feedback on progress, organises expeditions.
Project supervisor	Set the project timeline and provide feedback on progress.	Primary engagement with principle stakeholder.
Project engineer	Translate specifications to subsystems.	Selecting hardware, sourcing materials as well as designing firmware for the buoy.
Deployment team	Place the system in a selected location and ensure the device is communicating	Physical handling of the device, understanding how the system works.
User	Collect and archive data packets from the buoy.	Interact with the data portal and decompression software.
Collaborators	Work with users on interpreting data from the system.	Analysing generated data sets.

B.2 Acceptance test protocols

Table B.2: Acceptance test protocol for subsystem connectivity testing.

AT001	Connection test
Evaluation type	Software unit test
Target	Sensors
Test protocol	Microcontroller should be connected to the device on a specified communication port. The microcontroller then requests an acknowledgment from the device either by reading the ID register or by asking the device to return an acknowledgment string.
Pass condition	<ul style="list-style-type: none"> • Microcontroller receives acknowledgment • ID register read and valid
Fail condition	<ul style="list-style-type: none"> • Incorrect ID register value returned • NACK Returned • Invalid message string (timing error or framing error) • No data received (receiver timeout - malfunctioning device) • Failure to request read (transmission timeout - No device available)

Table B.3: Acceptance test protocol for fault testing.

AT002	Fault testing
Evaluation type	System recovery
Target	Hardware subsystems
Test protocol	<p>Connect subsystem to a microcontroller and run acceptance test AT001 under the following circumstances:</p> <ul style="list-style-type: none"> • Nack test: Change Acknowledgment string (USART peripheral) or device ID (SPI/I²C) to an incorrect value. • Corrupted message response test: Modify the baud rate to produce a corrupted message. • Disconnection test: Set the system to run a continuous cycle. Remove the device while the system is running. <p>Evaluate return status.</p>
Expected response	<p>Nack test: Controlled exit and return "NACK_ERROR". System clears message buffer and retries.</p> <p>Corrupted message: Controlled exit return "MESSAGE_ERROR". System reinitialises communication peripheral with different baud rate and retries.</p> <p>Disconnection test: Communication timeout, controlled exit and return "TX_ERROR". Critical failure: system recognises that device is missing and continues routine without it.</p>
Pass condition	<ul style="list-style-type: none"> • Device returns status and handles faults in a predicted manner • Critical failures handled without errors
Fail condition	<ul style="list-style-type: none"> • Device freezes during test • Device returns incorrect status • Segment Faults • Hard faults • Software Reset occurs during test

Table B.4: Acceptance test protocol for component selection.

AT003	Specification test
Evaluation type	Parameters
Target	Hardware components
Test protocol	Evaluate Specifications of the components from the data sheet to determine if the specifications meet the requirements for the system.
Pass condition	<ul style="list-style-type: none"> • Specifications meet the desired requirements
Fail condition	<ul style="list-style-type: none"> • Specifications do not meet the requirement

Table B.5: Acceptance test protocol for Subsystem Robustness Testing

AT004	Subsystem robustness test
Evaluation type	Software
Target	System and subsystem
Test protocol	<p>Connect subsystem to microcontroller and run a preset routine covering the following cycle:</p> <ul style="list-style-type: none"> • Initialisation • Processing function • Deinitialisation <p>Run this cycle 100 times consecutively.</p>
Pass condition	<ul style="list-style-type: none"> • Microcontroller successfully completes consecutive cycles
Fail condition	<ul style="list-style-type: none"> • Failure to complete more than 1 consecutive cycle • Failure to initialise (fails acceptance Test AT001) • Failure to correctly deinitialise after completing routine • Subsystem does not restart the cycle when reset

Table B.6: Acceptance test protocol for accelerated system testing.

AT005	Accelerated system Test
Evaluation type	Software
Target	System
Test protocol	System to run firmware with all sensors initialised. Routine loaded on system that cycles between all the sensors and communication modules turning them on and off then cycling through deep sleep mode. This occurs over a 1 hour period.
Pass condition	<ul style="list-style-type: none"> • System successfully cycles through sensors with no timing delays • System completes an hour of accelerated testing with no intervention • Power Reset do not cause the device to lock up or malfunction
Fail condition	<ul style="list-style-type: none"> • System freezes at any point during the test • System unable to turn on any sensor • system unable to enter sleep mode • system encounters unexpected reset • system unable to run for an hour

Table B.7: Acceptance test protocol for subsystem calibration testing.

AT006	Calibration test
Evaluation type	Statistical
Target	Sensor measurements
Test protocol	Connect device to a data logger and set the measurand to a static value record 100 sample points from the Device under test at a fixed frequency for a set amount of time. Measure against an accurate reference. Calculate mean and average value from data set and ensure it falls within the parameters given by the datasheet. Determine the disagreement between the average recorded value and average measured value and take the difference as the fixed offset bias. Repeat the test twice more first by adjusting the value halfway through recording then by varying the value at a fixed rate.
Pass condition	<ul style="list-style-type: none"> • Calibration produces a consistent output well within the accepted error range when measured against a reference • Step testing brings the measured value to the correct value • Device is capable of measuring over the specified range
Fail condition	<ul style="list-style-type: none"> • Test does not produce a predictable or consistent offset. • Calibration values produces an invalid data set. • Device under Test fails at any point. • Calibrated Data set unable to replicate the measurand.

Table B.8: Acceptance test protocol for power test.

AT007	Power system test
Evaluation type	Hardware
Target	Power System
Test protocol	Connect the power system to a load of a known resistance. Connect an ammeter and a voltmeter respectively in series and in parallel. Measure the current and supply voltage at a fixed rate for 1 hour. Record the battery voltage before the test and after the test. Then decrease the load to increase the current and run until the battery voltage drops below the threshold for the regulator. Measure the output current and supply voltage.
Pass condition	<ul style="list-style-type: none"> • Device cycles through routines for set period of time without failure • Device survives for specified period of time • Recorded values do not exceed the specs given from the data-sheet of the components
Fail condition	<ul style="list-style-type: none"> • Power System depleted before test has finished • Device fails to perform routine at any point during the test • Mechanical/Electrical failure occurs during test

Table B.9: Acceptance test protocol protocol for low temperature test.

AT008	Low temperature tests
Evaluation type	Hardware robustness
Target	Subsystem and full system
Test protocol	Connect subsystem to a datalogger and place in freezer. Set the freezer to -20°C and run the system through an accelerated subsystem test as per AT003. Then take the device out the freezer and wait for it to thaw. Then run another accelerated subsystem test. Finally connect all subsystems together and place in enclosure. Put device in the freezer and run an accelerated system Test as per AT004. Repeat in room temperature conditions.
Pass condition	<ul style="list-style-type: none"> • System completes routine cycles in both sub zero and room temperature environment • Subsystem passes AT003 in -20°C and room temperature environment
Fail condition	<ul style="list-style-type: none"> • Incorrect ID register value returned (SPI or I2C address incorrect) • Subsystem Fails AT003 in sub-zero and room temperature environment available • System fails AT004 in sub-zero and room temperature environment

Table B.10: Acceptance test protocol for final system deployment test.

AT009	Deployment test
Evaluation type	Performance
Target	Full system
Test protocol	This test is only to be performed once the requirements for all other tests have been met. Ensure device is calibrated before hand and in a deploy-able state. Ensure power is turned on and sensors have been initialised. Deploy the system in a desired location and monitor the transmitted packages.
Pass condition	<ul style="list-style-type: none"> • System acknowledges deployment and routinely transmits full packets of data. • System survives for 1 month or longer
Fail condition	<ul style="list-style-type: none"> • No acknowledgement received • No packets received • Empty data received <p>Failure within the first half an hour of deployment should result in immediate retrieval of the device.</p>

B.3 Schematics and renders

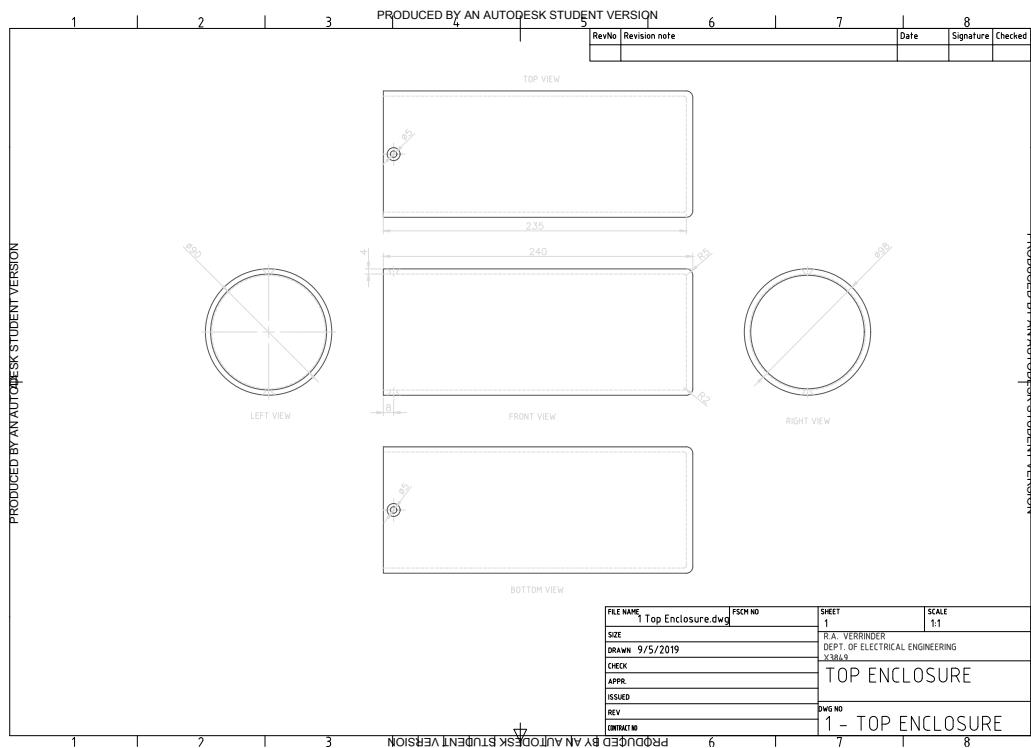


Figure B.1: Schematic of top enclosure which protects the electronics.

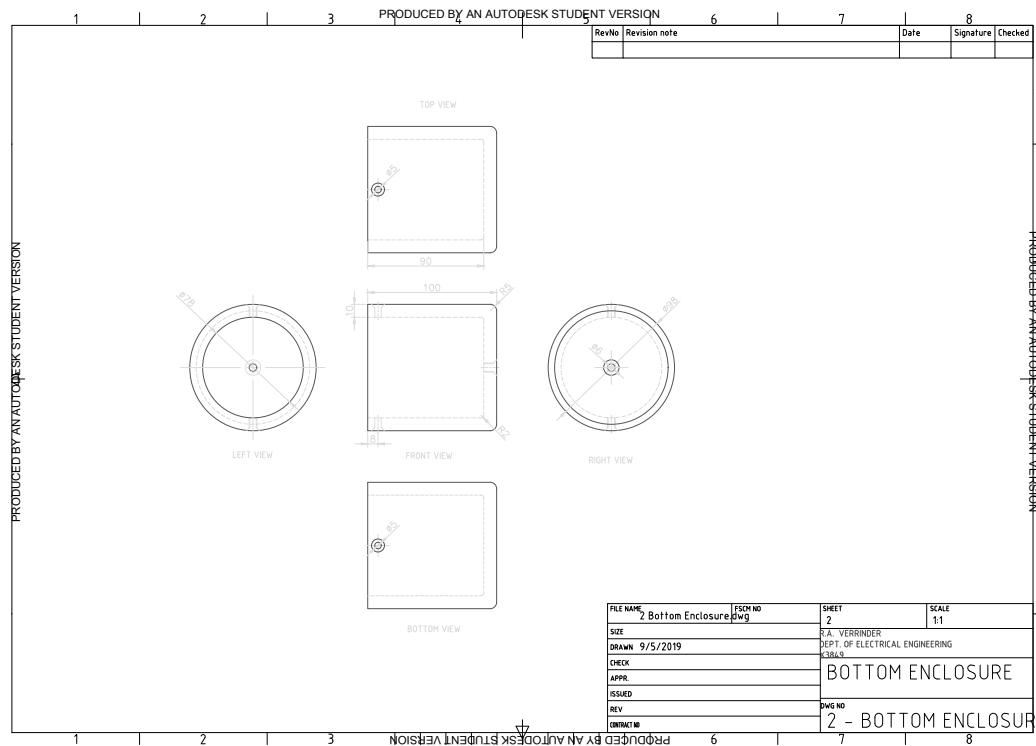


Figure B.2: Schematic of bottom enclosure for the batteries and power system.

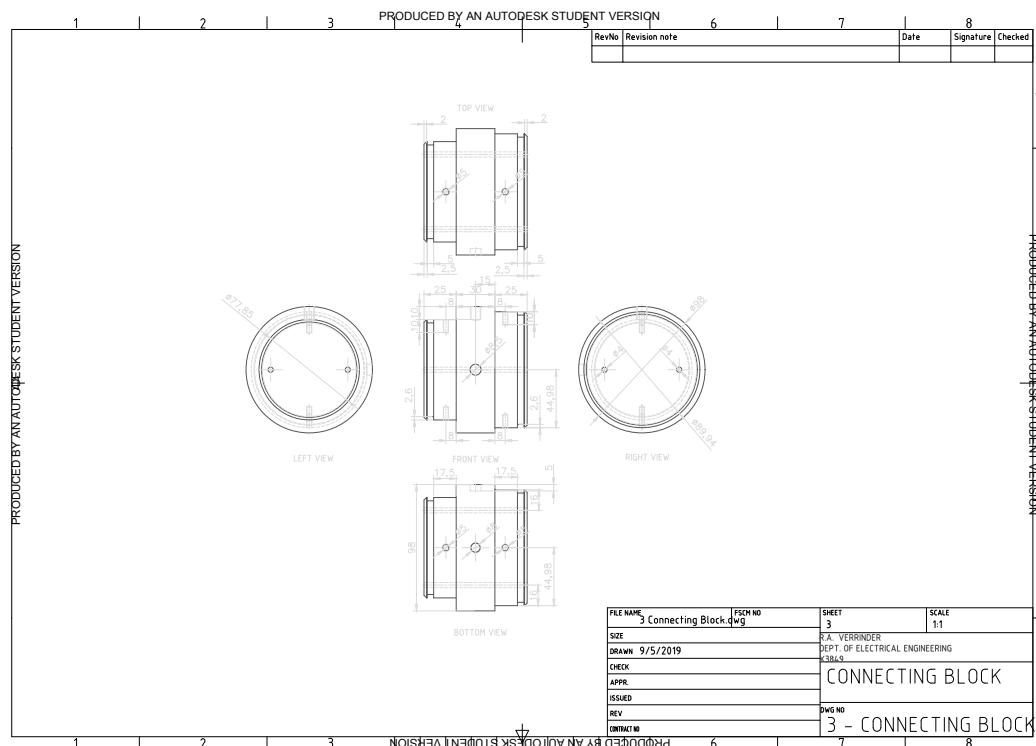
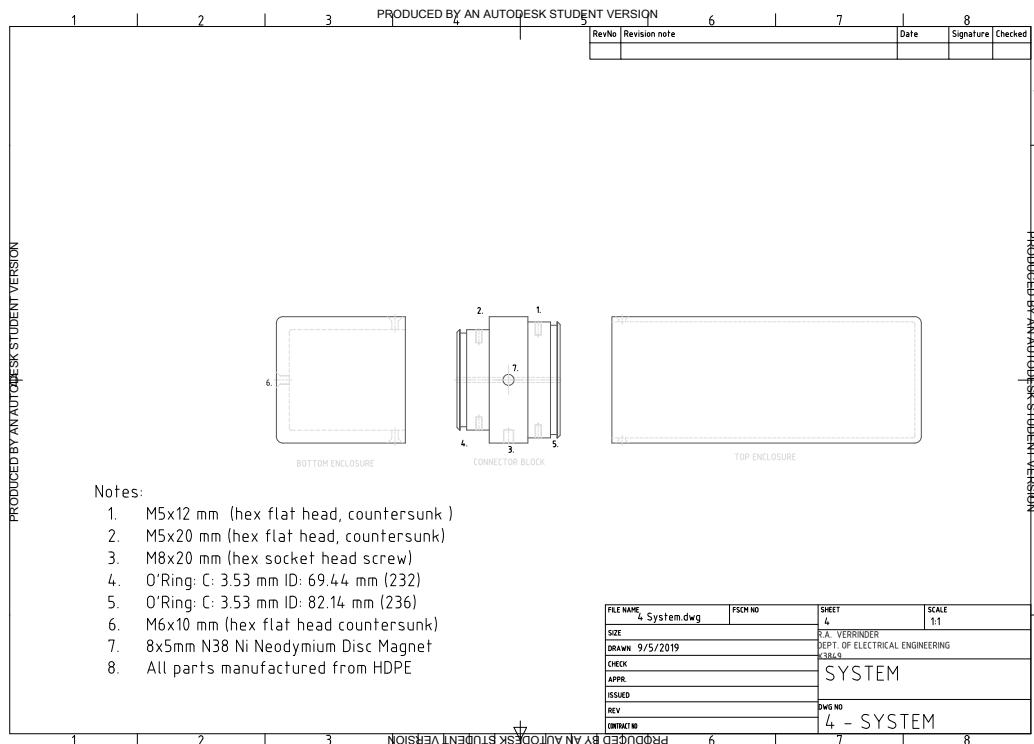


Figure B.3: Schematic of the connector block which provides a base for the electronics to be mounted on.

**Figure B.4:** Full enclosure schematic.

Appendix C

Software design

C.1 Events and interrupt handling protocols

Table C.1: Description of interrupt generated by the Iridium module on an external digital input line.

Ring Indicator	
Entry condition	Buoy in any state other than reset with GPIO mapped to EXTI, or with wake up from sleep mode on a wake up pin(WUP). The WUP pin receives a digital high from the ring indicator pin on Iridium
Function	The user has transmitted a packet to the buoy. Download the packet and execute/store the data based on the packet structure.
Entry condition	Device has downloaded user data which has been used to update the system and store data.
Return state	If entry source was a wake up, device will return to sleep, otherwise device will return to the main loop.

Table C.2: Description of routine for interrupts generated by the IMU on an external digital input line.

IMU motion event detection	
Entry condition	Buoy in any state other than reset with GPIO wake up pin mapped to an EXTI line, with wake up from sleep mode enabled. The WUP pin receives a digital high from an external interrupt pin.
Function	Device reads the interrupt source from the IMU, initializes I ² C peripheral and begins sampling IMU data. The interrupt source determines the sampling rate, period and mode.
Entry condition	Device will exit when the IMU has finished sampling and the data has been stored into memory.
Return state	If entry source was a wake up, device will return to sleep, otherwise device will return to the main loop.

Table C.3: Description of event handling routine for a brown out recovery event.

Brown out detection	
Entry condition	Buoy is in run mode or in standby mode with brown out detection voltage enabled and $V_{brownout}$ configured in option bytes. The event occurs when the voltage supplied to the microcontroller is less than $V_{brownout}$ causing the device to be held under reset. When the voltage rises above the threshold, the device will enter the handler.
Function	Device resets the relevant register flags and checks for data corruption. If no data is corrupted. Device will reload the last state and attempt to run it again. Otherwise the device performs a software reset
Entry condition	$V_{supply} > V_{brownout}$, device successfully executes code in handler.
Return state	Returns to main loop.

Table C.4: Description of routine for handling low power events.

Low power detection	
Entry condition	Device is in run or sleep, the power voltage thresholds are set in PWR and the low power interrupt is enabled. The event occurs when $V_{supply} < V_{power}$ generating an event interrupt.
Function	Device will read the current sensor and transmit a final packet to the user. All peripherals switched off, and the device is turned off.
Entry condition	No exit.
Return state	No return state.

Table C.5: Description of routine for handling a software reset event.

Software reset	
Entry condition	The Microcontroller reset line is pulled low for a few seconds. This is triggered in any state by triggering a software reset in the NVIC.
Function	Reset the buoy to an initial state, Clear any pending flags and reset system data in the RTC back up registers.
Entry condition	Successful reset of voltage domains.
Return state	Return to reset state and start of main loop.

C.2 Initialization Routines

Table C.6: Colour guide for the initialization routine flow diagrams.

Microcontroller (HAL) function
Sensor (API) function
Function return statement evaluation
Fail return status
Success return status

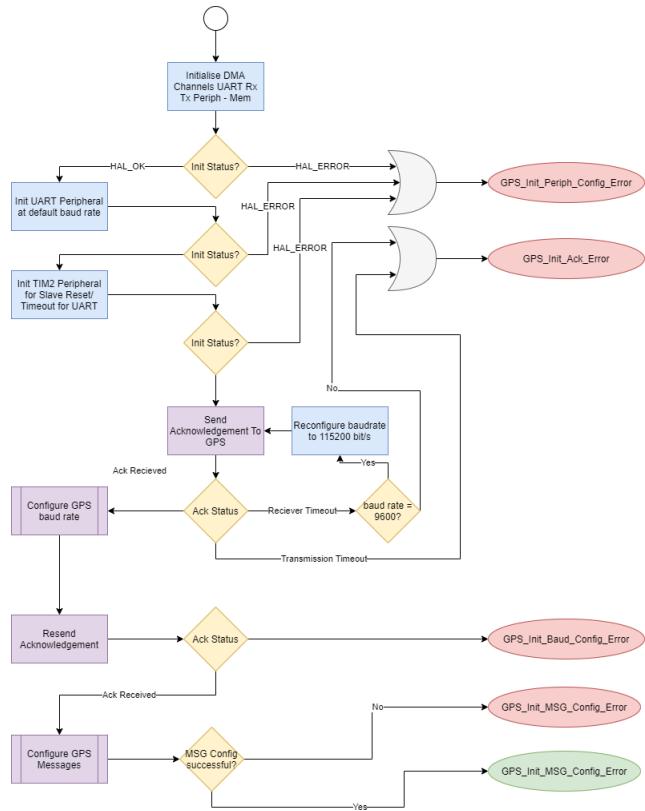


Figure C.1: u-blox NEO-7M and u-blox NEO-M9N initialisation routine.

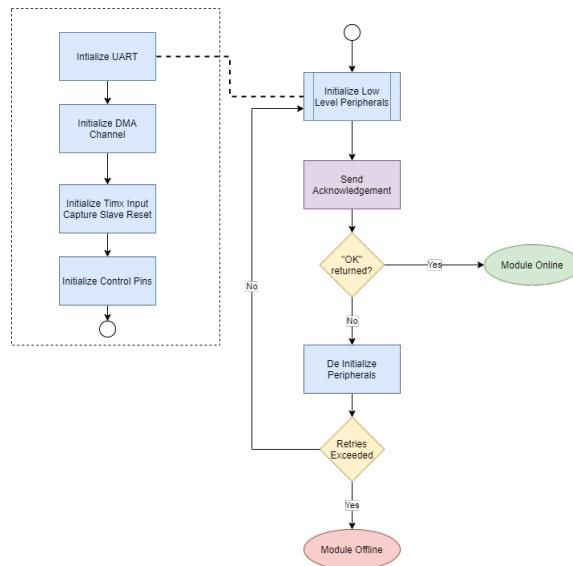


Figure C.2: Rock Seven RockBLOCK 9603 initialisation routine.

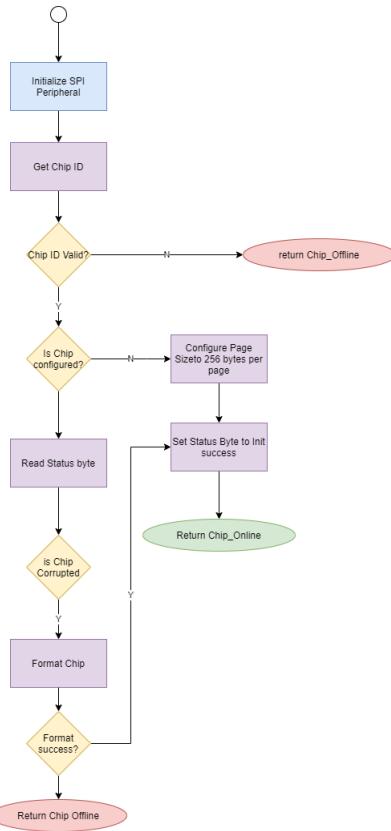


Figure C.3: Adesto Technologies AT45DB641E initialisation routine.

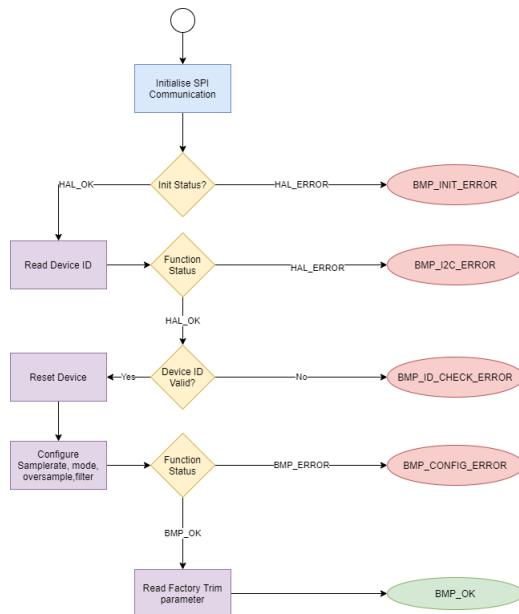
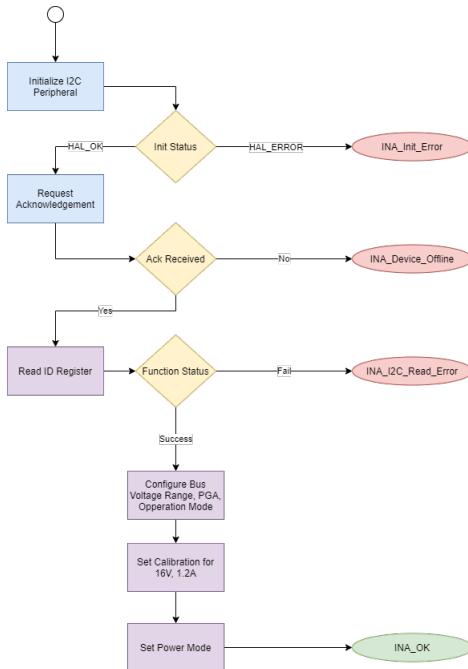
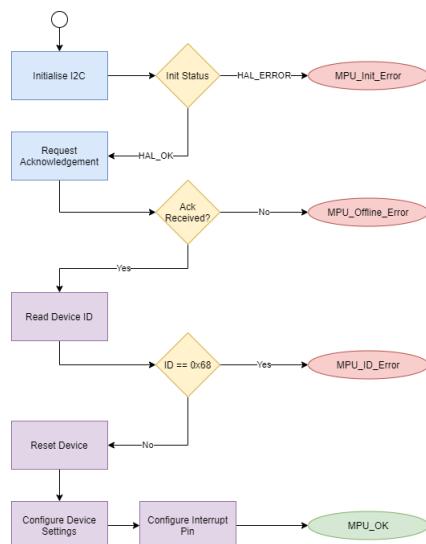


Figure C.4: Bosch Sensortech BMP280 initialisation routine.

**Figure C.5:** Texas Instruments INA219 initialisation routine.**Figure C.6:** TDK Invensense MPU6050 initialization routine.

C.3 Code

C.3.1 BMP280 temperature compensation formula

```

/*
 * @brief Temperature Compensation algorithm
 *
 * @param T_val
 * @param t_fine
 * @param bmp_trim
 *
 * @retval int32_t
 */
int32_t BMP280_Compensate_Temp(int32_t T_val, int32_t* t_fine,
    BMP280_trim_t bmp_trim)
{
    //compensate Temperature from datasheet
    int32_t var1 = (((T_val>>3)-
        ((int32_t)bmp_trim.dig_T1<<1)) * ((int32_t)bmp_trim.dig_T2))>>11;
    int32_t var2 = (((((T_val>>4) - ((int32_t)bmp_trim.dig_T1)) *
        ((T_val>>4) - ((int32_t)bmp_trim.dig_T1))) >>
        12) * ((int32_t)bmp_trim.dig_T3)) >> 14;
    int32_t temp = var1+var2; //for storage in global variable
    *t_fine = temp;
    return (temp*5 +128)/256;
}

```

Figure C.7: Function written to compensate a 32 bit Temperature reading for sensor irregularities using the 32 bit version of the recommended compensation formula from the datasheet (**BMP280_Datasheet**). The formula uses the compensation parameters stored on the sensor

```

/*
 * @brief Pressure compensation formula
 *
 * @param P_val
 * @param t_fine
 * @param bmp_trim
 *
 * @retval uint32_t
 */
uint32_t BMP280_Compensate_Pressure(uint32_t P_val, int32_t
    t_fine, BMP280_trim_t bmp_trim)
{
    //Compensation formula
    int32_t var1 = (int64_t)t_fine - 128000;
    int64_t var2 = var1*var1*((int64_t)(bmp_trim.dig_P6));
    var2 = var2 + (((int64_t)bmp_trim.dig_P4)<<35);
    var1 = ((var1 * var1 * (int64_t)bmp_trim.dig_P3)>>8) + ((var1
        * (int64_t)bmp_trim.dig_P2)<<12);
    var1 =
        (((((int64_t)1)<<47)+var1))*((int64_t)bmp_trim.dig_P1)>>33;
    //check for divide by 0 error
    if(var1 == 0) return 0;
    int64_t P = 1048576 - (int32_t)P_val;
    P = (((P<<31)-var2)*3125)/var1;
    var1 = (((int64_t)(bmp_trim.dig_P9)) * (P>>13) * (P>>13)) >>
        25;
    var2 = (((int64_t)(bmp_trim.dig_P8)) * P) >> 19;
    P = ((P + var1 + var2) >> 8) +
        (((int64_t)(bmp_trim.dig_P7))<<4);
    return (uint32_t)P;
}

```

Figure C.8: Function written to compensate a 32-bit pressure reading for sensor irregularities using the 32-bit version of the recommended compensation formula from (**BMP280_Datasheet**). The formula uses the factory trim compensation parameters stored on the sensor.

C.3.2 INA219 calibration algorithm

```

/*
 * Function Name INA_Status_t INA219_Calibrate_16V_1_2A(float
 *     *I_MBO, float *V_MBO, float *P_Max)
 * @brief: The following function writes a 16 bit value to the
 *         calibration register which
 *         is used to adjust the current, bias voltage and power.
 *         Here, A LSB value is
 *         calculated based on the user requirements and selected
 *         from a range. It would

```

```

*      be advisable to calculate the value manually and replace
*      it in the function below
*      please note: the following function has values calculated
*      manually. These can be
*      changed based on the configuration settings.
*      The values are calculated for 16V bus voltage range with a
*      2A expected current and
*      a 160mV shunt voltage range
*
*      Step 1: V_Bus_Max = 16V
*              V_Shunt_Max = 160mV
*              R_Shunt = 0.1 Ohm
*
*      Step 2: Max Possible I = 1.6A
*
*      Step 3: Let I Max Expected = 1.2A
*
*      Step 4: Min LSB = 36.6 uA/LSB
*              Max LSB = 292.97 uA
*
*              Choose LSB = 100 uA
*      Step 5: Set Calibration value = 4096
*/

```

```

INA_Status_t INA219_Calibrate_16V_1_2A(float *I_MBO, float
*V_MBO, float *P_Max)
{
    //set Current Step Size
    ina.INA219_I_LSB = 100.0/1000000.0;
    uint16_t I_cal_val =
        (uint16_t)(0.04096/(ina.INA219_I_LSB*INA219_R_SHUNT));
    ina.INA219_P_LSB = 20*ina.INA219_I_LSB;
    float I_max = ina.INA219_I_LSB*32767;
    if(I_max > 1.6) //max possible current
    {
        *I_MBO = 1.6;
    }else
    {
        *I_MBO = I_max;
    }
    float Vshunt_max = *I_MBO*INA219_R_SHUNT;
    if(Vshunt_max > 0.16)
    {
        *V_MBO = 0.16;
    }
    else
    {
        *V_MBO = Vshunt_max;
    }
    *P_Max = *I_MBO*16;
}

```

```

//write I_Cal_val to register
uint8_t temp[2] = {(I_cal_val&0xFF00)>>8, (I_cal_val&0x00FF)};
if(HAL_I2C_Mem_Write(&ina.ina_i2c, INA219_I2C_Address, CALIBRATION_REG, 1, temp,
    != HAL_OK)
{
    return INA_I2C_WRITE_ERROR;
}

return INA_OK;
}

```

Figure C.9: Calibration routine for INA219 current sensor for a maximum current of 1.2 A, maximum bus voltage of 16 V and maximum shunt voltage of 160 mV.

C.3.3 Data structs

```

/*
 * Coordinate Object
 *
 * Stores the Coordinates of GPS in the form DDMM.mmmm
 * where
 *     DD      - Degrees
 *     MM      - Minutes
 *     mmmm   - Fractional minutes
 * Variables:
 *     Name.....Type.....Description
 *
 *     lat.....float32_t.....GPS
 *     Latitude
 *
 *     longi.....float32_t.....GPS
 *     Longitude
 */
typedef struct
{
    float_t lat;
    float_t longi;
}Coord_t;

```

Figure C.10: Coord_t data structure to store incoming GPS coordinates as IEEE754 32-bit floats.

```
/*
 * Diagnostic Object
 *
 * Structure to Hold the GPS data signal diagnostics
 *
 * Variables:
 *   Name.....Type.....Description
 *
 *   PDOP.....DOP_t.....Positional
 *   dilution of Precision (3D)
 *
 *   HDOP.....DOP_t.....Horizontal
 *   dilution of Precision
 *
 *   VDOP.....DOP_t.....Vertical
 *   dilution of Precision
 *
 *   num_sats.....uint8_t.....Number
 *   of Satelites used to obtain positional Fix
 *
 *   fix_type.....uint8_t.....number
 *   between 1-3 describing the type of fix obtained
 *
 * Fix types
 * 1 - No Fix
 * 2 - 2D Fix (No altitude)
 * 3 - 3D Fix
 */

```

```
typedef struct
{
    DOP_t PDOP;
    DOP_t HDOP;
    DOP_t VDOP;
    uint8_t num_sats;
    uint8_t fix_type;
}Diagnostic_t;
```

Figure C.11: Data structure for storing GPS signal diagnostic information.

Appendix D

Supplementary tables

Table D.1: List of data services provided by Iridium for transmission of data over the satellite network including the bandwidth and purpose of the service taken from ([iridium_mobile](#)).

Service name	Purpose	Supporting modems	Bandwidth
Short Burst Data (SBD)	Sending short messages in bursts.	9603/9602	340 bytes upload & 270 bytes download
		9523	1960 bytes upload & 1890 bytes download
Iridium edge			
Router-based Unrestricted Digital Interworking Connectivity Solution (RUDICS)	Continuous transfer of large real-time data from a large array of devices to a host.	9523 9522B <i>(deprecated)</i>	6 to 10 KB/min
Circuit Switch Data (CSD)	Continuous transmission of large volumes of data over a dial-Up network using a SIM card.	9523 9522B <i>(deprecated)</i>	6 to 10 KB/min

Table D.2: Strategies used by the devices to transfer data from remote locations. Table includes transmission technologies and services used as well and transmission strategies and transmission intervals where given. Prices are converted to Rands (R) via (**usdcoversio**n).

Device name	Service	Modem	Bandwidth	Transmission strategy
WIIB	Iridium SBD	9602	340 bytes	Data condensed into one 340 byte packet (transmission intervals unavailable).
WIIOS	Iridium SBD	9602	340 bytes	Data condensed into one 340 byte packet transmitted every five hours.
NDWB	Iridium RUDICS	9522B	6 to 10 KB/min	Raw IMU data points transmitted every minute.
	Iridium SBD (long range)	9602	340 bytes	GPS data transmitted every 10 minutes.
SKIB	ZigBee (short range)	Xbee Pro	50 KB/second	Raw data transmitted when host is in range.
	Iridium	Geoforce SmartOne (tracking)	N/A	Not Reported/
SWIFT	Iridium SBD	Unspecified SBD Modem (telemetry)	1960 bytes	Data transmitted through SBD modem. variable packet sizes ranging from 4 to 1228 bytes in length.
	Ethernet	Digi Xpress ethernet bridge	935 KB/second	Not Reported.
SIMB	Iridium	9603	340 bytes	Single packet transmission of 275 bytes.
	ARGOS	Not Reported	Not Reported	Not Reported
Polar ISVP	Iridium SBD	9602	340 bytes	User configured packet sizes and transmission intervals.
Trident	Iridium SBD	9603	340 bytes	Single packet transmission of 16 bytes.

Appendix E

Test protocols

E.1 Unit tests

Table E.1: Protocol for firmware initialisation/deinitialisation functional test.

UT001	
Description	Initialisation/deinitialisation test.
Test protocol	The hardware module is connected to the microcontroller with the corresponding subsystem API libraries and driver files. The initialisation/deinitialisation function is run. After the function has completed, the return status from the function is evaluated. If the test was completed successfully, A "Module online!" is printed to the serial and the LED is switched on.

Table E.2: Protocol for the signal acquisition firmware test, This test is specific to receivers with wireless capabilities.

UT002	
Description	Signal acquisition test.
Test protocol	Test to be run after unit test UT001 in Table E.1. Module de initialisation function is to be run until completion. After the function has run, the configuration registers are checked for reset values. If the registers are reset, the function returns a success status and "Successfully deinitialised module!" is printed to the serial. If the LED was green, it is turned off. This test was also run before an initialisation function to ensure that the function does not result in hard faults.

Table E.3: protocol for testing the peripheral communication functional unit.

UT003

Description	Signal acquisition test.
Test protocol	Module to be connected to the microcontroller with firmware loaded with the required communication peripherals to be initialised. The microcontroller either requests to receive a byte of data or transmits a set of data in polling mode. If successful, the function will return a successful status. Otherwise an error code will be returned.

Table E.4: Protocol for testing a software function for validating sensor data.

UT004

Description	Transmission test
Test protocol	Test to be run as a unit test on the desired platform. An array of expected outputs from the sensor is written as a test case along with the expected output of the function. Then the input is passed through the function and stored after the function has been complete. The function output is compared to the expected output. The function passes if the output matches the expected output.

Table E.5: Protocol for testing data streams and direct memory access (DMA) functions.

UT005

Description	Data stream test
Test protocol	The hardware module is connected to the microcontroller with the DMA stream enabled. The microcontroller triggers the sensor to transmit a known amount of data through the DMA stream. The microcontroller waits for a signal to close the stream then stops the DMA transfer. The data from the stream is then compared to the data expected. The test fails immediately if errors or hard faults occur.

Table E.6: Protocol to test functions that interface with sensors by configuring settings in the register or reading data from a register.

UT006

Description	Sensor interface module functional test
Test protocol	Test requires the module library to be loaded onto the microcontroller with the hardware module connected. Function to be called after a successful initialisation. Load the function with a known parameter and wait for a return status. Evaluate the return status against a table to determine whether the function was successful.

Table E.7: Protocols for fault testing of software subsystem functions.

UT007

Description	Unit fault testing
Test protocol	Firmware to be loaded onto the microcontroller with the module placed in the configurations shown in AT002 (Table B.3). the test passes if each configuration results in the corresponding error code being returned. The test fails if the function freezes or does not return a predictable return code.

Table E.8: Protocols for testing digital input/output peripheral functional units.

UT007

Description	GPIO functional unit tests
Test protocol	Firmware to be loaded onto the microcontroller with an LED attached to the digital pin. Pin to be configured in digital output mode. A logic high must be written to the pin resulting in a pass if the LED turns on. Then a logic low is written resulting in a pass if the LED turns off. For input testing, a push button in normally open mode must be attached to the pin and configured in pull up mode. A logic high should be read while the button is open and a logic low should be read while the button is pressed.

E.2 System test results

Table E.9: Results of subsystem acceptance tests for each of the identified modules. Modules that were successfully validated were marked with a ✓, failed tests were marked by an X and tests that could not be applied to a subsystem were marked by an N/A.

	AT001	AT002	AT003	AT004	AT005
GPS	✓	✓	✓	✓	✓
Iridium Modem	✓	✓	✓	✓	✓
Flash Chips	✓	✓	✓	✓	✓
Power Module	N/A	N/A	✓	N/A	N/A
Env Sensor	✓	✓	✓	✓	✓
Power Monitor	✓	✓	✓	✓	✓
IMU	✓	✓	✓	✓	✓