UNIVERSITY OF SOUTHAMPTON

Standards-Based Internet of Things
Sub-GHz Environmental Sensor Networks

by

Graeme M^{c}Lachlan Bragg

Thesis for the degree of Doctor of Philosophy

in the

Faculty of Physical Sciences and Engineering

Electronics and Computer Science

November 2017

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCE AND ENGINEERING
ELECTRONICS AND COMPUTER SCIENCE

Thesis for the degree of Doctor of Philosophy

STANDARDS-BASED INTERNET OF THINGS SUB-GHZ ENVIRONMENTAL SENSOR NETWORKS

by Graeme McLachlan Bragg

In recent years there has been shift in the use of wireless sensor networks from standalone systems that use bespoke methods of communication and data transfer to systems that use Internet standards and can interact more directly with the Internet. This has allowed wireless sensor networks to become a key enabler of the Internet of Things; however, the same is not true for environmental sensor networks as the focus of most existing research into Internet of Things wireless sensor networks has been on 2.4 GHz designs for indoor, urban and agricultural applications. In these applications, power, Internet connectivity and physical access are less of a challenge when compared to a typical environmental sensor network. Environmental sensor networks are used for monitoring natural processes and are generally deployed in harsh, remote environments where these factors are more of a concern. Sub-GHz radios are commonly used for communication due to their increased range and desirable propagation characteristics. Unlike wireless sensor networks, environmental sensor networks have been slow to adopt Internet standards and have continued to rely on bespoke methods of communication and data transfer, keeping their usability low. This has impeded the adoption of environmental sensor networks for earth sciences research.

This thesis investigates whether the Internet standards that have helped to make wireless sensor network an important part of the Internet of Things can be applied to sub-GHz environmental sensor networks. It is demonstrated that 6LoWPAN can successfully be used with an 868 MHz network in a series of real-world deployments in the Highlands of Scotland that collected usable earth science data and facilitated research in other fields. Additionally, the suitability of these standards for real-world networks is assessed in terms of energy, throughput and latency performance and compared to a theoretical 2.4 GHz network. An publicly available open source Contiki radio driver for the CC1120 was developed as part of this work. Additionally, timing parameters for using ContikiMAC with 868 MHz radios were determines and shared with researchers at other institutions, facilitating further research into sub-GHz IoT ESNs by other researchers.

Contents

List of Figures

List of Tables

List of Acronyms

µC	Microcontroller. A small single-chip computer that contains a processing core, volatile and non-volatile memory. Microcontrollers also have programmable input and output peripherals for interacting with other hardware.
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks. 6LoWPAN is a method of efficiently providing global IPv6 addresses to resource-constrained devices. 6LoWPAN is defined by RFC 4944 and has been updated by RFC 6282 and RFC 6775.
ACK	Acknowledgement frame. Used to acknowledge successful reception by the receiver of an addressed packet.
API	Application Programming Interface.
CoAP	Constrained Application Protocol. A UDP-based web transfer protocol standardised by RFC 7252
COTS	Commercial off the Shelf. Usually used to refer to components that are already available.
dGPS	Differential Global Positioning System. An implementation of GPS that uses a fixed base-station to provide a greater degree of accuracy by being able to compensate for positioning errors.
DTP	Dynamic transmit power. A method of scaling the transmit power used to achieve an energy saving.
ESN	Environmental Sensor Network. A wireless sensor network used to monitor environmental processes and parameters.
EWoR	Enhanced Wake on Radio. A method of radio duty cycling incorporated into the CC1120 radio transceiver.
GPS	Global Positioning System.
I^2C	Inter-Integrated Circuit. A two-wire serial interface used to connect devices to a microcontroller or other processor.
IoT	Internet of Things.

IP	Internet Protocol.
IPv4	Internet Protocol version 4. The fourth version of the IP that is one of the core protocols underpinning the Internet. IPv4 provides 32-bit addressing and is described in RFC 791 (Postel 1981).
IPv6	Internet Protocol version 6. The current version of the IP. IPv6 provides 128-bit addressing and is described in RFC 8200 (Deering and Hinden 2017).
ISM	Industrial, Scientific and Medical. Used to refer to a set of radio frequency bands dedicated to ISM applications that are usually license-free for other uses.
ISR	Interrupt Service Routine. A function that is branched to when a specific hardware interrupt occurs.
LDO	Low-dropout regulator. A type linear voltage regulator.
LQI	Link Quality Indication. Indicates the quality of the communications link. This can be established with a combination of parameters including the RSSI and the number of errors.
MAC	Medium Access Control. The lowest sub-layer of Layer 2 (Data Link Later) in the OSI seven-layer model that handles physical addressing and channel access mechanisms.
MS1	Mountain Sensing 1. A sensor node designed for the Mountain Sensing project that comprises of a Zolertia Z1, a CC1120 and supporting hardware.
PCB	Printed Circuit Board.
PHY	Physical layer. The lowest layer in the OSI seven-layer networking model.
PHR	Physical Header.
PPP	Precise Point Positioning. A technique of post-processing data from a single GPS station to achieve centimetre level accuracy.
PSDU	PHY service data unit. The payload at the physical layer.
RDC	Radio Duty Cycling. A way to reduce the average energy usage of a radio by keeping the radio turned off for as long as possible.
RF	Radio Frequency
RFID	Radio Frequency Identification. A method of identifying objects through the use of a tag affixed to the object that can be interrogated by nearby readers.

RPL	IPv6 Routing Protocol for Low power and Lossy Networks. Defined by the IETF in RFC 6550.
RSSI	Received Signal Strength Indication. This is an indication if the strength of a received signal.
RTOS	Real-Time Operating System.
RX	Receive.
SD Card	Secure Digital Card. A type of removeable solid state storage.
SDN	Seismic Deep Node. A sub-glacial sensor node used in the Glacsweb project to monitor glacial movement using Geophones.
SFD	Start of Frame Delimeter. The sequency of bits at the end of a preamble that signify that the payload follows.
SHR	Synchronisation header. A fixed pattern of bytes transmitted before the PHY to synchronise the receiver to the transmitter.
SoC	System on Chip. An integrated circuit that includes a full system in a single package. SoCs can include a processor, memory, peripheral interfaces, radio and specialised co-processors.
SPI	Serial Peripheral Interface. A high-frequency four-wire serial interface used for connecting devices to a microcontroller or other processor.
SSDN	Slim Seismic Deep Node. A hardware re-design of the Glacsweb SDN.
SSN	Seismic Surface Node. A surface routing node used in the Glacsweb project to provide a wireless connectivity to an SDN.
TX	Transmit.
Wi-Fi	Wireless Fidelity. A method of wireless networking used in common desktop and laptop computers. Defined by the IEEE 802.11 series of standards.
WPAN	Wireless Personal Area Network.
LR-WPAN	Low-rate Wireless Personal Area Network.
WSN	Wireless Sensor Network. A system that consists of several nodes connected using a wireless technology for the purpose of sensing and actuating physical parameters. Typically, WSN nodes are small low-power microcontroller-based devices.

Declaration of Authorship

I, Graeme M^{c}Lachlan Bragg, declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

Standards-Based Internet of Things

Sub-GHz Environmental Sensor Networks

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:

- Bragg, G. M., Martinez, K., Basford, P. J., & Hart, J. K. (2016). *868MHz 6LoWPAN with ContikiMAC for an internet of things environmental sensor network*. Paper presented at the SAI Computing Conference 2016, London.
- Martinez, K., Hart, J. K., Basford, P. J., Bragg, G. M., Ward, T., & Young, D. S. (2017). *A geophone wireless sensor network for investigating glacier stick-slip motion*. Computers & Geosciences, 105, 103-112.
- Fabre, A., Martinez, K., Bragg, G., Basford, P., Hart, J., Bader, S., & Bragg, O. (2016). *Deploying a 6LoWPAN, CoAP, low power, wireless sensor network: Poster Abstract.* Paper presented at the 14th ACM Conference on Embedded Network Sensor Systems.

- Bragg, O. M., Basford, P. J., Black, A. R., Bragg, G. M., Hart, J. K., & Martinez, K. (2016). *Britain's Highest Bog: Can We Unlock Its Secrets?* Paper presented at the 15th International Peat Congress, Kuching, Malaysia.

- Martinez, K., Hart, J. K., Bragg, O., Black, A., Bader, S., Basford, P. J., Bragg, G. M., Fabre, A. (2016) *An Environment IoT Sensor Network for Monitoring the Environment.* AGU Fall Meeting Abstracts 2016.

- Martinez, K., Hart, J. K., Basford, P. J., Bragg, G. M., & Ward, T. (2013). *Using Internet of Things technologies for wireless sensor networks*. AGU Fall Meeting Abstracts 2013.

Signed: _____

Date: November 2017

Acknowledgements

This work would not have been possible without the input and collaboration of members of the recent Glacsweb and Mountain Sensing teams. The 2013 Glacsweb team consisted of Professor Kirk Martinez, Dr Phil Basford, and Tyler Ward. The Mountain Sensing team comprised of Professor Jane Hart, Professor Kirk Martinez, Dr Andrew Black, Dr Olivia Bragg, Dr Sebastian Bader, Dr Phil Basford, Tyler Ward, Arthur Fabre and Ed Crampin. Tyler Ward was the main contributor to the redesign of the SSN and Dr Basford was the main contributor in designing the Z1 interface board and the MS1. Arthur Fabre developed the CoAP application that was used for the later Mountain Sensing deployments. Thanks also to Thomas Macdonell and Wildland Ltd. for allowing access to the study site in the Cairngorm Mountains, access to their Internet connection and for providing access to vehicles for use on the site.

I would also like to thank my friends and family for their help and support throughout: Phil, for being there to bounce ideas off and to tell me when I was being silly; my mother, Olivia, for encouraging me to follow a path in research and for being on hand to carry out repairs and modifications to the test deployments in Scotland; my father, Kevin, for helping me get to grips with ArcGIS so that we could model the deployment before; my wife, Francine, for being my rock and understanding that late night finishes, weekends working and weeks away were a necessary part of this work even if she did not understand what I was trying to do; and finally my son, Logan, who has been a welcome distraction throughout this work and has brought many smiles and moments of relief.

Finally, I'd like to express my gratitude for my supervisor, Professor Kirk Martinez, for giving me the opportunity to carry out this research. I may not have always gone in the direction you encouraged but your support and advice has been continual encouragement.

Chapter 1

Introduction

Wireless Sensor Networks (WSNs) are a continually evolving area of research that has benefitted over the years from increased miniaturisation, decreasing cost and increasing availability of the electronics used to manufacture them (Tubaishat and Madria 2003, Healy, Newe et al. 2008). The basic, abstracted design of most WSNs consists of a single processor, usually realised as a single low-power microcontroller, a radio and a power supply plus attached sensors (Healy, Newe et al. 2008). The simple abstracted design and developments in technology have allowed WSNs to be used for many different applications including environmental monitoring, industrial monitoring & control, military purposes (Akyildiz, Su et al. 2002), agricultural management (Burrell, Brooke et al. 2004, Pierce and Elliott 2008) and as part of the "smart home" movement (Han and Lim 2010, Erol-Kantarci and Mouftah 2011).

WSNs are seen as a key part of the Internet of Things (IoT), as they bridge the gap between the physical and virtual worlds (Uckelmann et al., 2011), which is in turn perceived as being a key part of the Future Internet where everything from everyday objects, such as food packaging and furniture, to entire cities become "smart" objects that can autonomously interact with each other. Specific applications that are already benefiting from IoT developments include logistics management (Uckelmann, Harrison et al. 2011), supply chain management (Lou, Quan et al. 2011), facility management (Blackstock, Kaviani et al. 2010) and smart metering (Guinard, Trifa et al. 2011). More complex deployments have included a "smart community" (Xu, Rongxing et al. 2011), where pervasive health monitoring and neighbourhood security are carried out within a group of smart homes, and MagicBroker2 (Blackstock, Kaviani et al. 2010), which facilitates spontaneous interaction between people, smart displays and sensor/actuator networks. Traditionally, WSNs used bespoke communication protocols that have limited the ability of different node designs to operate in a single network however many contemporary designs now use standardised technologies that have become cheaper and more available in recent years. Technologies like 802.15.4, 6LoWPAN, standardised low-power routing and more capable and energy efficient microcontrollers have allowed IP to be a realisable concept on a low-power sensor node.

Environmental Sensor Networks (ESNs) (Martinez, Hart et al. 2004) are a sub-type of wireless sensor networks that are used to monitor the environment in harsh and remote environments. ESNs have been deployed for many different applications, ranging from habitat monitoring (Szewczyk, Osterweil et al. 2004) and forest fire detection (Lazarescu 2013) to glacier (Martinez, Basford et al. 2009) and volcano (Werner-Allen, Johnson et al. 2005) monitoring. The designs used for these deployments illustrate that the common requirements for ESNs are very low power efficient operation and resilience as the nodes need to operate for extended periods of time with limited, and usually finite, energy sources in environments where access for maintenance and replacement can be restricted. "Sleepy" networks using sub-GHz radios are a common theme.

1.1. Current Challenges

Broadly speaking, the limitations of current ESN designs can be split into two areas: processing efficiency and communications.

Limitations in processing are related to the capabilities of current microcontrollers with regard to the performance/power trade off. The basic WSN design, which uses a low-power microcontroller (Healy, Newe et al. 2008), works well for nodes that carry out non-resource-intensive tasks such as simple sensing and communications control; however, there can be compromises in processor selection where tasks of significantly different intensities need to be executed. Either an appropriate microcontroller for the most resource-intensive task is selected at the expense of energy efficiency and system lifetime, or a microcontroller that is appropriate for the lower intensity tasks and is very energy efficient is selected at the cost of being able to carry out the more resource-intensive tasks effectively or at all (McLntire, Ho et al. 2006, Martinez, Basford et al. 2009, Hong-Ling, Kun Mean et al. 2011). This is especially applicable as WSNs move to being part of the IoT where more processing power is required to handle the overheads involved (Zimmermann, Sa Silva et al. 2008). Most IP stacks require a scheduler to operate and run as software on top of a real-time operating system (RTOS) (NanoStack Project 2013b, RIOT 2013, TinyOS 2013) or other low-power operating system (Contiki 2015). Additionally, as wireless communications can account for up to 75% of the energy utilisation in some applications (Hong-Ling, Kun Mean et al. 2011), further processing capability may be required for signal processing, data compression (Sheltami, Musaddiq et al. 2016) or data aggregation (Fasolo, Rossi et al. 2007) methods to reduce the amount of energy expended on communication.

With regard to communications, ESNs have been slow to adopt IP-based standardised communication technologies that have been embraced by IoT WSNs. Hart and Martinez (2015) classified five

different families of sensor node as shown in Figure 1. In a true IoT system, the nodes will either be directly connected or indirectly connected Internet. Current 2.4 GHz-based IoT wireless sensor network designs fall into the indirectly connected classification; however, contemporary environmental sensor networks fall into the category of "not Internet connected" as non-standard methods of communication are used to communicate with a base station, which is then connected to the Internet. Standardised communications in the form of 6LoWPAN running on top of an 802.15.4 network is the method commonly used by contemporary 2.4 GHz WSN designs; however, 2.4 GHz radios are not appropriate for ESNs that require an increased range or specific propagation behaviour and ESNs have tended to rely on sub-GHz radios to meet these requirements (Werner-Allen, Johnson et al. 2005, Martinez, Basford et al. 2012, Lazarescu 2013). 802.15.4 (IEEE 2011), the standard required for the implementation of 6LoWPAN (Montenegro, Kushalnagar et al. 2007), does specify standardised sub-GHz communications but these have not been implemented in an actual deployment. Additionally, other standards required for a 6LoWPAN network, such as RPL (Winter, Thubert et al. 2012), have not been tested in a sub-GHz network.

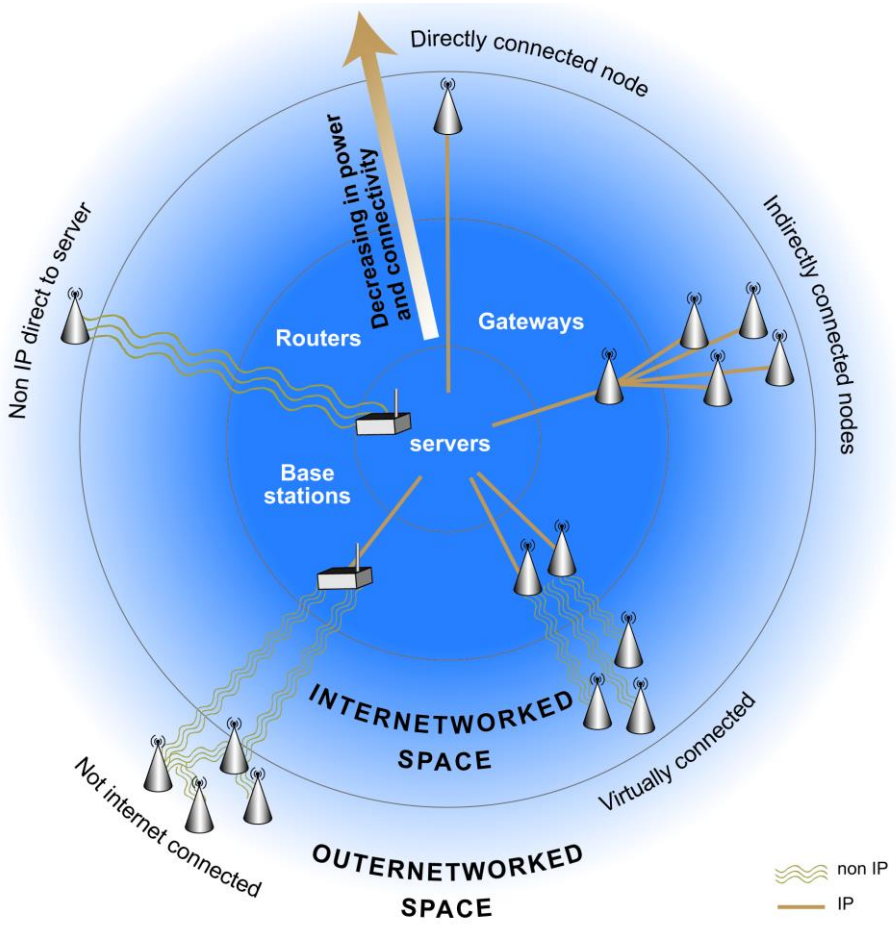


Figure 1: Schematic of the different types of sensor network with the availability of power and connectivity increasing towards the centre. Reproduced from Hart and Martinez (2015).

Additionally, the lack of standardised methods of communication in environmental sensor networks has hindered the adoption of standardised data transfer protocols, such as HTTP and CoAP (Shellby, Harke et al. 2014), and methods of presenting data, such as RDF, JSON and RSS (Guinard, Trifa et al. 2011). This makes it harder for users without specialist knowledge to interact with and manage the network.

The integration of ESNs into the IoT requires investigation of how existing Internet Standards at all layers of the networking stack can be applied to ESNs and whether acceptable performance, in terms of energy, resiliency and usability, can be achieved. Hardware designs also need to be reviewed to ensure that the additional overhead involved in standardisation can be implemented in a low-power and efficient manner.

This thesis investigates how 6LoWPAN and standardised data transfer protocols can be applied to sub-GHz environmental sensor networks and whether these standards have suitable energy, throughput and latency characteristics for real-world deployments. This work is validated through a series of real-world deployments with an 868 MHz environmental sensor network in the Highlands of Scotland. 868 MHz was chosen for these deployments as the frequency has appropriate propagation characteristics and has been used in several existing sub-GHz ESN deployments.

1.2. Research Questions

This thesis will address three research questions:

1. Can the standards used for 2.4 GHz IoT wireless sensor networks be applied to sub-GHz environmental sensor networks?
2. Can a sub-GHz standards-based 6LoWPAN network perform sufficiently, in terms of energy, for real-world IoT environmental sensor network deployments?
3. Can a sub-GHz standards-based 6LoWPAN network perform sufficiently, in terms of throughput, latency and reliability, for real-world IoT environmental sensor network deployments?

These questions have been identified through the review of the literature carried out in Chapter 2 and experiences with the Glacsweb project discussed in Chapter 3. The remainder of this section discusses these questions and the motivation for them.

Standards that allow sensor networks to communicate with the Internet have been successfully applied to 2.4 GHz wireless sensor networks and allowed them to become a key part of the Internet of Things. In contrast, environmental sensor networks have lagged behind in their adoption of these standards. This is partly due to a lack of implementations that support the sub-GHz bands that ESNs favour. While it is theoretically possible for 6LoWPAN to be applied to sub-GHz networks this has not been achieved on actual hardware as the focus of developers has been on 2.4 GHz based platforms that are more popular and available. Physical and addressing networking technologies are not the only area of standardisation that WSNs exploit. Standardised methods of data transfer, such as HTTP, have allowed existing utilities to be used to retrieve data from WSNs and interact with them. Determining whether these standards can be applied to sub-GHz environmental sensor networks is the first step in realising a real-world standardised IoT ESN.

While similar in concept, WSNs and ESNs have different requirements in terms of energy efficiency and performance. The techniques used to maximise energy efficiency for a 2.4 GHz IoT WSN may not apply to a sub-GHz environmental sensor network or if they do, they may not be as effective. The techniques of energy conservation used at 2.4 GHz, such as duty cycling protocols, need to be assessed in the context of a sub-GHz environmental sensor network.

Real-world environmental sensor networks rarely behave consistently with laboratory testing or theory. What may work well in a laboratory based testing may not work as well, or may even fail completely, when deployed in the real world. As such, real-world deployments are required to validate that a standards-based network using 6LoWPAN and other Internet standards can operate acceptably in the real world.

1.3. Research Contributions

Several contributions have been made with the work detailed in this thesis. The research questions posed in Section 1.2 are addressed through the course of this thesis. The first question is addressed in Chapter 4, where it is demonstrated that 6LoWPAN can successfully be used with a sub-GHz radio, and in Chapter 5, where acceptable operation of CoAP and protocol buffers is shown in a real-world deployment. The second and third research questions are addressed in Chapter 5 where the real-world performance of a sub-GHz environmental sensor network using the standards and duty cycling protocols commonly used for 2.4 GHz wireless sensor networks is assessed. It is shown that the latency and throughput are suitable for real-world deployments and the use of an asynchronous duty cycling protocol increases reliability when compared to a simple synchronous

duty cycling protocol. Energy performance was found to be acceptable where large batteries can be used however further work is needed to improve energy efficiency.

In addition to addressing the research questions, other contributions have been made. An open-source and publicly available Contiki radio driver was developed as part of the work detailed in Chapter 4. The timings for using ContikiMAC with 50 kbps 868 MHz radios have also been made available publicly and have been used by researchers for other similar platforms. The performance of HTTP is assessed and found to be inadequate when compared to CoAP in a real-world deployment. Finally, the work detailed in this thesis has helped to deploy the first real-world sub-GHz 6LoWPAN-based environmental sensor network and the data gathered has assisted researchers in other fields understand the natural processes occurring at the study site and facilitated their research. These contributions are discussed in more detail in Section 6.1.

1.4. Thesis Structure

This thesis is divided into six chapters that identify the current state of environmental sensor networks, identifies their short comings in the Internet of Things age and proposes ways to improve their design.

Chapter 2: Literature Review. The second chapter of this thesis reviews literature in the fields of the Internet of Things, wireless sensor networks and environmental sensor networks. The enabling technologies of the Internet of Things are identified and which of these technologies are required for an IoT environmental sensor network are discussed.

Chapter 3: Glacsweb. Glacsweb was a pre-existing research project and work was carried out to gain familiarisation with the challenges and requirements of a real-world environmental sensor network deployment. Work is carried out to prepare the existing network for the implementation of IoT technologies.

Chapter 4: Sub-GHz 6LoWPAN Communications. A true Internet of Things environmental sensor network requires low-power IP-based communications that have the propagation characteristics to allow for sparse node deployments over a large geographic area in harsh conditions. Existing 6LoWPAN implementations and research have focused on 2.4 GHz systems for short-range wireless sensor networks and not in the sub-GHz bands that are required for environmental sensor networks. This chapter details the implementation of 6LoWPAN at 868 MHz, determines the tim-

ing requirements for implementing a state-of-the-art asynchronous duty cycling protocol and assesses the performance of 6LoWPAN at 868 MHz compared to 2.4 GHz.

Chapter 5: Mountain Sensing. Networking is not the only requirement for an Internet of Things environmental sensor network. This chapter develops a platform for an Internet of Things environmental sensor network that uses the sub-GHz networking developed in Chapter 3 and standardised methods of data transfer. Theoretical modelling and laboratory based testing of technologies for environmental sensor networks rarely reflects what happens in reality so real-world deployments are carried out and refinements made to the platform to address the problems identified.

Chapter 6: Conclusions. The final chapter of this thesis summarises the contributions made by this work to environmental sensor networks and wireless sensor networks in general. The contributions that have been made are discussed and suggestions for areas of future work that would further enhance Internet of Things environmental sensor networks are made.

Chapter 2

Literature Review

Wireless Sensor Networks and the Internet of Things are two research areas where there is a significant amount of existing literature and on-going research. This chapter presents a review of the literature of the most relevant IoT concepts and WSN designs and discusses how environmental sensor networks differ from WSNs. It is split into five sections that cover general IoT concepts & history, a summary of technologies that help to enable IoT, WSNs, ESNs and a summary. The section reviewing WSNs includes a review of low-power networking and IoT operating systems.

2.1. The Internet of Things

Simply put, the Internet of Things (IoT) is about giving physical object a presence on the Internet that allows people and other things to find out information about the object and then interact with them. Put more complexly, IoT is about systems intelligently using data about the physical world that has been autonomously collected to influence the physical world (Farooq, Waseem et al. 2015). Beyond this basic definition, there are many different ideas on what the IoT is and what it encompasses (Ashton 2009) and the term has been miss-used extensively in modern vernacular to refer to everything that can interact with something else from non-Internet connected wearable technology (22 CityLink 2016) to home routers and IP connected webcams (Krebs 2016). For the avoidance of doubt, this Thesis interprets Internet of Things to refer to the principle of an object that interacts with the physical world that has some form of presence on or across the Internet, either directly or indirectly addresses, where its status is autonomously updated.

The term Internet of Things was first coined in 1999 in relation to autonomous supply chain management where objects were tagged with uniquely-identifiable RFID tags, which included details of the manufacturer, object type and a unique serial number, and read by a series of network connected RFID readers as they moved along the supply chain (Sarma, Brock et al. 2001, Ashton 2009, Lou, Quan et al. 2011). This concept is useful for supply chain management and stock control

where exact stock levels of individual components & products can be tracked and replacements automatically ordered from suppliers as needed but has a significant overhead in the amount of infrastructure required to read the tags affixed to items. The information that can be obtained using RFID technology is also limited to what is encoded on the tag – there is no facility for sensing other factors without additional infrastructure. Advancements in other fields of technology have resulted in the view on what IoT encompasses evolving since this initial vision and contemporary thinking includes "smart" objects that can autonomously report on their own status, sense the environment around them and directly interact with the environment (Kortuem, Kawsar et al. 2010). Early facetious examples of smart objects that pre-date the first use of the term include an internet connected drinks vending machine from 1982, which reported how often drinks were dispensed and their temperature, (CMUSCS 1998) and the IETF's 1998 April Fools RFC that details a protocol for interacting with Internet-connected coffee pots (Masinter 1998). Today, more serious consumer-focused home automation smart objects exist including Wi-Fi connected ambient notification electronic rabbits (Violet 2010), wirelessly controllable light bulbs (Phillips 2015) and smart thermostats (Hive 2015).

Guillemin, Friess et al. (2009) made a more encompassing prediction that IoT would become a key part of the "future Internet" where everyday objects become "smart" objects that more fully bridge the physical and virtual worlds. Aspects of this prediction are being realised as businesses look to use IoT technologies to enhance their current offerings (Gerpott and May 2016). As an example, Amazon have embraced IoT technologies to make it easier for them to sell products with the Amazon Dash Button (Crouch 2015), which allows consumers to reorder specific products from Amazon with a single press of a Wi-Fi connected smart button, the AWS IoT Button (Amazon 2016), a generic version of the Dash button that can be used to interact with other things through Amazon Web Services, and the Amazon Dash Replenishment Service that allows devices to reorder their own supplies as needed (Olivarez-Giles 2015). A common theme with consumer-focused IoT smart objects and IoT enhanced services is that many of them rely on a centralised cloud infrastructure that smart objects use to discover services and permit interactivity.

Outside of home automation and supply chain management, IoT concepts have been applied to other application areas including social, commercial, agricultural and environmental monitoring. Social applications range from smart communities (Xu, Rongxing et al. 2011) that ensure the health and wellbeing of their members through the use of pervasive healthcare and automated Neighbourhood Watch, to enabling spontaneous interactivity between devices and facilitating the creation of sharable user-created sensor mashups of data aggregated from multiple sources (Blackstock, Kaviani et al. 2010). IoT technologies have also been applied to dynamic and interactive art to give

novel and interesting art pieces. The eCloud installation (Goods, Hafermaas et al. 2010) uses real-time weather data from an Internet source to determine its visualisation state. Erica the Rhino (Basford, Bragg et al. 2016) interacts with people physically and online through social media with the online interaction having an effect on her physical interactions and physical state. Commercial applications cover a large number of areas and include the Smart Grid (Farhangi 2010), where data from many sources is used to more efficiently generate and deliver electricity; Smart Meters, which enable customers to easily monitor their energy usage, give suppliers an insight into their customers' usage patterns and allow for more automated and accurate billing; Smart Cities, where entire cities are instrumented and actuated (Zanella, Bui et al. 2014); food supply chain management; mining safety; healthcare service; logistics (Xu, He et al. 2014); and factory control (Zuehlke 2010). In agriculture, IoT concepts have been used for automated greenhouse management (Ji-chun, Jun-feng et al. 2010), irrigation control (Suping and Birui 2013) and animal management (Zhu, Dai et al. 2012, Ilapakurti and Vuppalapati 2015). Environmental monitoring using some IoT concepts has been carried out for glaciers (Martinez, Hart et al. 2004), forest fire detection (Lazarescu 2013), water monitoring (Chi, Yan et al. 2014) and climate change monitoring (Fang, Xu et al. 2014).

2.2. IoT Enabling Technologies

Attempts have been made by several different workers to categorise the enabling technologies of the Internet of Things. In Guillemin, Friess et al. (2009), the Cluster of European Research Projects on the Internet of Things identified thirteen different technology areas that support the Internet of Things vision. In contrast, Al-Fuqaha, Guizani et al. (2015) identified six key elements that are needed for the Internet of Things to be realised and Atzori, Iera et al. (2010) proposed eight and noted that different visions of the Internet of Things attributed importance to differing technologies. Despite the discontinuity in number and classification of enabling technologies, there are several common themes identified in the literature. Broadly speaking, these can be classified into four high-level areas of technology relevant to environmental sensing that: enable interaction with the physical world; enable things to be uniquely identified; enable communications with the Internet and other things; and enable people and other things to discover and interact with things. This section briefly reviews technologies in these four areas that have been used to enable the Internet of Things. Some of these technologies are pre-existing ones, which have been repurposed and combined with other existing technologies, and some have been specifically developed to address shortcomings with existing technologies. Certain technologies are only applicable to specific types of IoT application that are beyond the scope of this research, such as RFID and QR/Barcodes so not all technologies are reviewed here.

The first, and possibly most important, enabler of the Internet of Things is standardisation (Guillemin, Friess et al. 2009, Atzori, Iera et al. 2010, Al-Fuqaha, Guizani et al. 2015). Standardisation is of communications protocols, ways of translating between different communication methods and methods of transferring data are required to allow things to communicate with other things and the Internet.

Sensor and actuator networks enable the collection of information about and interaction with the physical world (Uckelmann, Harrison et al. 2011). Without these networks, the Internet of Things would not be a realisable concept as there would be no bridge between the physical and virtual worlds that allows for the autonomous collection of data and transmission of actuation. In turn, advances in microcontroller and low-power networking have made sensor networks more capable. Wireless Sensor Networks and the technologies that they rely on are discussed in more detail in Section 2.3.

Physical networking technologies, from Wi-Fi (Tozlu, Senel et al. 2012) and 3G (Zhiyong, Kui et al. 2010) to lower-power networking, such as low-power Wi-Fi, Bluetooth Low Energy (Kouche 2012) and 802.15.4, are required to enable connectivity between things and from things to the Internet to enable the transmission of sensor readings and actuation commands. Many of the consumer-focused IoT devices mentioned in 2.1 rely on the presence of an existing Wi-Fi infrastructure to operate and have been made possible by the prevalence of home Wi-Fi provision. To be able to communicate with the rest of the Internet these connections must be globally addressable, either directly or indirectly, with an IP address. Traditionally, this has been IPv4 however the near exhaustion of IPv4 addresses prior to the advent of the IoT and the prediction that there will be up to 50 billion Internet connected devices by 2020 has meant that a replacement is required to permit the continued expansion of the IoT and Internet in general (Evans 2011, Savolainen, Soininen et al. 2013, Levin and Schmidt 2014). IPv6 is the standard proposed to replace IPv4 that uses 128-bit addressing, allowing for a total of 3.4×10^{38} unique addresses, instead of the 32-bit addressing used for IPv4 (Postel 1981, Deering and Hinden 2017). Due to the longer addresses, IPv6 has a larger header of 40 bytes compared to the 20 byte minimum of the IPv4 header however it has a simplified structure as shown in Figure 2 to reduce the processing overhead for each packet and limit the bandwidth overhead of the increased addressing scheme. Despite being standardised in 1998, adoption of IPv6 by has been slow until recent years with some providers preferring to adopt stop-gap techniques, such as Carrier-grade NAT, to extend the life of IPv4 (Czyz, Allman et al. 2014). Despite the slow adoption, IPv6 is still seen as a key requirement for the IoT.

(a)

Octet	0								1								2								3							
Byte Offset \ Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Version				Header Length				DSCP						ECN		Total Length															
4	Identification																Flags			Fragment Offset											
8	Time to Live								Protocol								Header Checksum															
12	Source IP Address																															
16	Destination IP Address																															
20	*Options (Optional, if IHL > 5)*																															

(b)

Octet	0								1								2								3							
Byte Offset \ Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Version				Traffic Class								Flow Label																			
4	Payload Length																Next Header								Hop Limit							
8	Source IP Address																															
12																																
16																																
20																																
24	Destination IP Address																															
28																																
32																																
36																																

Figure 2: Comparison of the IPv4 (a) and IPv6 (b) header structure. Enhanced from ASCII depictions in (Postel 1981) and (Deering and Hinden 2017).

IPv6 has been brought to low-power Wireless Personal Area Networks with the 6LoWPAN standard (Montenegro, Kushalnagar et al. 2007). 6LoWPAN defines the frame format, header compression and address autoconfiguration methods to bring IPv6 services to low-power radio links using the 802.15.4 standard. The limited packet size of 802.15.4 radios and use of header compression requires an adaptation layer to bridge the low-power network to a standard IPv6-based network. 6LoWPAN has been used in many real-word deployments with several implementations available (Neves and Rodrigues 2010, Yibo, Hou et al. 2011). 6LoWPAN does not specify routing or neighbour discovery methods within the low-power network and even though standardised methods have been proposed by the IETF (Shelby, Chakrabarti et al. 2012, Winter, Thubert et al. 2012), many different routing protocols exist (Kumar and Tiwari 2012). 6LoWPAN and routing are discussed in more detail in Section 2.3.2.

Methods of automatic configuration and discovery are required for other parameters to enable the spontaneous machine-to-machine interaction of different IoT devices and platforms. Some visions of the Internet of Things see objects and services being able to discover each other and interact autonomously requesting data and services from each other without initial configuration (Garcia Macias, Alvarez-Lozano et al. 2011, Uckelmann, Harrison et al. 2011). Instead of relying on discovery of services at the device level, many of the applications mentioned in 2.1 rely on a central-

ised data base, cloud service or data warehouse to aggregate interactivity with multiple devices into one location. Al-Fuqaha, Guizani et al. (2015) imply that the use of centralised systems is a basic feature of the sensing element that they identified and focus on discovery of the overall service offered through the centralised system.

Additionally, efficient and adaptable methods of transmitting and presenting data are required to enable bandwidth and power-efficient access to data and actuators. Standardised examples of these methods include REST, RDF, JSON, RSS (Guinard, Trifa et al. 2011) and protocol buffers (Google 2015) along with CoAP (Shellby, Harke et al. 2014) and HTTP.

2.3. Wireless Sensor Networks

Wireless Sensor Networks are networks of nodes that combine sensing, data processing and communications technologies to autonomously sense something about the environment that they are placed in (Akyildiz, Su et al. 2002). The basic design of a WSN platform consists of a single processing element, a discrete radio and a power supply plus attached sensors as depicted in Figure 3 (Healy, Newe et al. 2008) and WSNs have been used in a range of applications to sense the physical world .

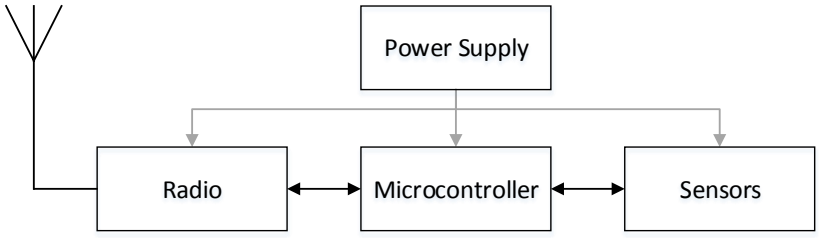


Figure 3: Block diagram of the basic elements of a basic WSN.

2.3.1. Hardware Architecture

Over the years there has been a progressive evolution of sensor node designs that has increased their capabilities. Healy, Newe et al. (2008) reviewed several WSN node designs used between 1998 and 2007 and found that the most common choice of processing element was an 8-bit microcontroller with 25 out of 43 reviewed designs using either a PIC or an AVR. Seven designs used a 16-bit MSP430 microcontroller while six designs used a 32-bit ARM processor. Several commercially available ‘motes’ were identified in their review including the Telos and Mica platforms. These platforms are pre-made using commercial off the shelf components (COTS) and include simple sensors, usually temperature and humidity, and/or some way of adding additional sensors.

They are relatively low-cost and are designed to enable experimentation in a wide variety of fields without the need to design and make a bespoke platform (Polastre, Szewcyzk et al. 2004).

Historically common mote designs have used either an 8-bit or a 16-bit microcontroller for the processing element as they are low-cost and low power. As shown in Table 1, 32-bit motes in the early 2000s tended to use more energy than the 8-bit and 16-bit designs of the time. This was reflected in the choice of microcontroller for popular commercial designs such as the Tmote Sky (Moteiv Corporation 2006), Zolertia Z1 (Zolertia 2010) and Telos Motes (Crossbow Technology Inc n.d.-b) all use a 16-bit MSP430 microcontrollers with a discrete radio.

Table 1: Power comparison of different WSN nodes including multi-core designs.

Platform	**Architecture**	**Power Usage**				**Data Source**
		Active (mW)	**Sleep (µW)**	**TX (mW)**	**RX (mW)**	
WINS	32-bit	727.6	6400	1080.5	751.6	Savvides and Srivastava (2002)
Telos mote	16-bit	3	2	45	-	Polastre, Szewcyzk et al. (2004)
MicaZ mote	8-bit	33	30	45	-	Polastre, Szewcyzk et al. (2004)
XYZ Node	32-bit	52.5 – 239.8	99	-	-	Lymberopoulos and Savvides (2005)
TelosB mote	16-bit	6.75	45	65.1	68.4	Prayati, Antonopoulos et al. (2010)
IoT Wildfire Detection	8-bit	2.34	16.92	108	-	Lazarescu (2013)
SENTIO-em	32-bit	24	5.4	150	49.5	Bader, Krämer et al. (2013)
Medusa MK-2	8/32-bit	92.22	20	107	104.7	Savvides and Srivastava (2002)
Gumsense	16/32-bit	900	7.2	-	-	Martinez, Basford et al. (2009)
E2MWSN	FPGA/8/32	62.7	3	65.4	-	Hong-Ling, Kun Mean et al. (2011)

The basic WSN design of a single low-power microcontroller works well for nodes that carry out non-resource-intensive tasks, such as simple sensing and communications control, however there can be compromises in processor selection where tasks of significantly different intensities need to be executed. Either an appropriate microcontroller for the most resource-intensive task is selected at the expense of energy efficiency and system lifetime, or a microcontroller that is appropriate for the lower intensity tasks and is very energy efficient is selected at the cost of being able to carry out

the more resource-intensive tasks effectively or at all (McLntire, Ho et al. 2006, Martinez, Basford et al. 2009, Hong-Ling, Kun Mean et al. 2011). The need for more capable microcontrollers in sensor network platforms is more pronounced in platforms that are used for IoT as there are additional processing, storage and memory overheads involved (Zimmermann, Sa Silva et al. 2008).

To address this limitation, some attempts have been made to develop multi-core nodes for sensor networks that combine processing elements of differing capabilities to improve overall system performance and efficiency. Three multi-core designs are included in Table 1 and this shows that multi-core designs have the potential for improving sensor node performance with comparable sleep energy; however, the existing implementations are more complex than a single-core design and are more costly to deploy as multiple processors are required. Existing multicore designs use a heterogeneous mix of different processor architectures that require different development environments, complicating software development and making code less portable across the platform. The increased complexity and cost has meant that use of multi-core designs for real-world sensor network deployments has been limited with most research and commercial platforms favouring single-core architecture.

Multi-core designs using multiple discrete processors have become less relevant with the advent of significantly cheaper and lower-power 32-bit microcontroller designs incorporating ARM Cortex processing cores. These microcontrollers have greater performance than 8-bit and 16-bit designs with comparable power consumption. One example is the SENTIO-em platform (Bader, Krämer et al. 2013) which is clocked at 32MHz, two to four times the clock rate of typical 8-bit or 16-bit based mote. As shown in Table 1, the SENTIO-em platform has power consumption comparable to commercial mote designs and significantly lower than earlier 32-bit designs such as the WINS node and the XYZ node. ARM cores have also been used for other sensor node designs with Cortex-M3-based microcontrollers being used for the subglacial probes in the Glacsweb project (Martinez, Basford et al. 2012) and the Sprouts WSN Platform (Kouche 2012). Even lower power designs should be possible with microcontrollers based on the Cortex-M0+ as this has a more energy efficient design than the Cortex-M3 (ARM Ltd. 2012) however no platforms have been found in the literature.

During the course of this PhD commercially available ARM-based motes have been designed that take advantage of other advances in microcontroller design. OpenMote (Vilajosana, Tuset et al. 2015) and the Zolertia RE-Mote (Zolertia 2016) use the CC2538 SoC (Texas Instruments 2015b) that includes the microcontroller and a 2.4 GHz radio in a single package, simplifying the overall design. While improvements in 32-bit microcontroller designs have generally meant that single-core platforms have remained the norm, more recent developments have seen some manufacturers

introducing heterogeneous multi-core SoCs targeted at applications that require low-power sensing. Examples of this include the CC1310 and CC2650 that include three different microcontrollers. A Cortex-M0 is used for the RF Core that managed the radio, a Cortex-M3 is used for the main CPU and a 16-bit microcontroller is used for a low-power sensor controller.

2.3.2. Networking

Networking technology is a fundamental requirement for sensor networks and encompasses methods of wireless communication, how energy is saved through duty cycling and how data is routed across the network. This sub-section looks at the methods key to delivering efficient wireless communications to WSNs and ESNs.

2.3.2.1. Methods of Communication

Many different methods of wireless communications have been used for inter-node communications in sensing applications with differing methods being more appropriate for different types of network. Acoustic methods have been used extensively for underwater sensor networks as the propagation characteristics of sound in water allow for a significant range however, they are throughput-limited due to interference and the attenuation of sound in air makes acoustic communications impractical for terrestrial applications (Zhang, Huang et al. 2005, Lanbo, Shengli et al. 2008). Optical communications are another method that can be used for both underwater and terrestrial applications (Mathews, Barnes et al. 2009). Optical methods can provide high data rates over a long range with low energy requirements but have to be very directional to achieve this so they require careful aiming and unobstructed line of sight between nodes (Kahn, Katz et al. 1999). This means that adverse weather that affects line of sight, such as fog, snow and heavy rain, can prevent successful communication. Shammaa and Verma (2011) proposed the use of optical communications for connecting two isolated WSN islands that were outside of radio communication range. Their implementation requires an additional high-power base station for each island increasing the deployment complexity and required energy. The optical link still suffers from the same general limitations of optical communications. Finally, the most common form of wireless communications used for terrestrial sensing applications and the focus of the remainder of this sub-section is radio.

Low-power radio in one form or another has been used in numerous different WSN designs (Akyildiz, Su et al. 2002). Of the potential frequency bands available, 2.4 GHz is the most commonly used for contemporary, commercially available WSN designs (MEMSIC Inc , Moteiv Corporation 2006, Zolertia 2010, Crossbow Technology Inc n.d.-b) and has been frequently used in

research deployments for urban and agricultural sensing (Stuntebeck, Pompili et al. 2006, Tongjuan, Jun et al. 2010, Tu, Liu et al. 2010, Horvat, Sostaric et al. 2011, Ling-ling, Shi-feng et al. 2011). 2.4 GHz designs are popular as radios are widely available, low-cost and well supported by existing software. 2.4 GHz is one of the bands supported by IEEE 802.15.4 and many of the radios used for these research deployments are compliant with 802.15.4, which is discussed in more detail in this Section 2.3.2.2.

Other bands that have been commonly used in the past for WSNs include 433 MHz and 868/915 MHz. 433 MHz was commonly used for WSNs prior to 2.4 GHz becoming common place and is still used in some contemporary designs that are not described as IoT and for applications that require increased range or specific propagation characteristics (Martinez, Ong et al. 2004, Werner-Allen, Johnson et al. 2005, Green, Nadimi et al. 2009, Lazarescu 2013). 433 MHz is not supported by the IEEE 802.15.4 standards (IEEE 2011, IEEE 2012b) meaning that there is no direct IP connectivity to nodes as is desirable in IoT deployments. IoT-like deployments are still possible using non-standardised implementations at the gateway to the network however these do not promote the same interoperability as a standards-compliant solution. The 868 MHz and 915MHz bands have been used in some research WSN designs (Savvides and Srivastava 2002) and commercial offerings (Libelium Comunicaciones Distribuidas S.L. 2012, Crossbow Technology Inc n.d.-a). The TinyNode (Dubois-Ferrière, Fabre et al. 2006) has been used in numerous Permasense deployments (Talzi, Hasler et al. 2007, Hasler, Talzi et al. 2008, Beutel, Gruber et al. 2009, Girard, Beutel et al. 2012). While they offer significantly increased range over 2.4 GHz designs these bands have not been as popular for IoT-like deployments because low-cost radios have not been as widely available or as well supported in software.

Other methods of communication that have been used for sensing applications include GPRS (Centenaro, Vangelista et al. 2016), Wi-Fi, Bluetooth (Healy, Newe et al. 2008) and LoRa (Centenaro, Vangelista et al. 2016). Wi-Fi and GPRS are high-power connections that are not appropriate for use on every sensor node. Bluetooth has been used for some sensing and IoT applications and a low-energy version exists (Ilapakurti and Vuppalapati 2015); however, while it can use IPv6 addressing (Nieminen, Savolainen et al. 2015) Bluetooth is restricted to short ranges and the 2.4 GHz band. LoRa is a relatively recent low-power radio technology that uses sub-GHz low power radio to form wide-area networks for low-power devices (LoRa Alliance n.d.). Despite being targeted at IoT applications, LoRa does not support Internet standard addressing.

2.3.2.2. 802.15.4

As discussed in Section 2.2, standardisation is a key requirement in realising the Internet of Things. IEEE 802.15.4 is a series of standards and amendments that specify the physical layer (PHY) and Media Access Control (MAC) layers of the OSI seven-layer model of computer networking for low-rate Wireless Personal Area Networks (LR-WPAN) and is a requirement for implementing 6LoWPAN, which brings IP-based communications to low-power radio links (Montenegro, Kushalnagar et al. 2007). 802.15.4 was first standardised in 2003 with the 802.15.4-2003 version of the standard (IEEE 2003). The standard was revised in 2006 (IEEE 2006) and again in 2011 (IEEE 2011) to enhance and correct the previous standards. The latest revision is 802.15.4-2015 (IEEE 2016) however the work in this thesis was carried out under 802.15.4-2011 so the remainder of this review is carried out with the 802.15.4-2011 standard and its amendments in mind. 802.15.4-2011 has been incrementally amended by numerous amendments that define new physical standards, add additional frequencies and modify the MAC sublayer. 802.15.4 specifies physical layers in a large number of frequency bands including 2.4 GHz, 868 MHz, 915 MHz, 433 MHz (IEEE 2012a), 169 MHz (IEEE 2012b) and numerous other regional bands.

Until the 802.15.4g amendment (IEEE 2012b), 868 MHz and 915MHz had different maximum bitrates specified in the 802.15.4 standards of 20 kbps and 40 kbps respectively. This difference meant that hardware, software and performance would be different depending on the country of operation. With the 802.15.4g amendment both bands gained unified support for 50 kbps and 200 kbps operation. These greater data rates mean that 868/915 MHz has the potential to be the most viable choice for future IoT ESN designs. The 802.15.4g amendment also allows for operation in the 169 MHz ISM band at data rates up to 9.6 kbps. This is close to the frequencies used by Martinez and Basford (2011) for sub-glacial probes meaning that there is the potential for 6LoWPAN-based IP communications to reach more extreme locations in the future.

In addition to specifying the physical layer, 802.15.4 specifies the hierarchy of headers and structure of frames as shown in Figure 4. The PSDU (PHY service data unit) has a consistent format across all PHYs with the Synchronisation and PHY headers are defined for each PHY. Despite being PHY specific, the PPDU follows a common structure for most PHYs and this is shown in Figure 5. 802.15.4 specifies four different frame formats with the general format shown in Figure 6. From these headers, the typical per-frame overhead for 802.15.4 can be established. This is 24 bytes for a header that uses 8-byte addressing with PAN ID compression, including the length byte. There will be additional overhead for the preamble and start of frame delimiter (SFD).

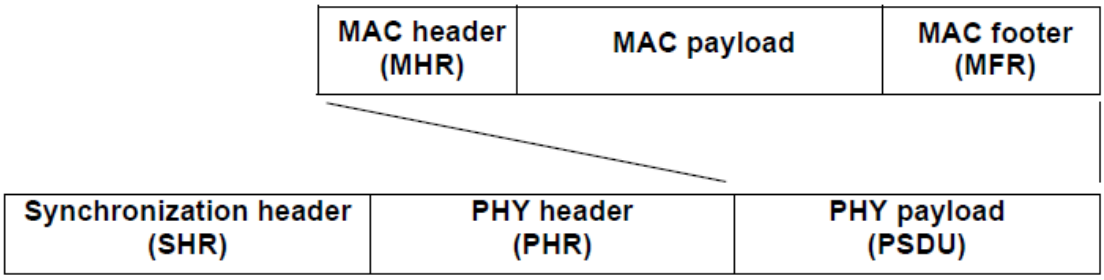


Figure 4: Schematic view of the 802.15.4 PPDU (PHY protocol data unit) Reproduction of Figure 6 from (IEEE 2011)

Octets		1	1	variable, max 127 bytes
Field	Preamble	SFD	Len	PSDU
	SHR		PHR	PHY Payload

Figure 5: Typical format of a PPDU header. The SHR includes the preamble and start of frame delimiter (SFD). The PHR includes the length byte. Modification of Figure 67 from (IEEE 2011)

Octets	2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variable	2
Field	Frame Control	Seq No.	Dest. PAN ID	Destination Address	Source PAN ID	Source Address	Auxilliary Security Header	Frame Payload	FCS
			Addressing Fields						
	MHR							MAC Payload	MFR

Figure 6: 802.15.4 general MAC frame format. Modification of Figure 36 from (IEEE 2011)

The standard also specifies the possible topology of the network. The two basic topologies are the start topology, where all of the devices communicate with a central co-ordinator, and the peer-to-peer topology, where any device can communicate with any other device that it is within range of. The peer-to-peer topology is the basis of a multi-hop mesh network, which is a requirement for larger deployments as it allows individual nodes to have a lower transmit power and can improve the connectivity and reliability of the network by providing redundant paths that data can take to reach the sink (Schwieger, Fettweis et al. 2004). 802.15.4 does not specify routing as this occurs at a higher layer in the OSI model. Routing is discussed in more detail in Section 2.3.2.4.

802.15.4 is used as the basis for several higher level protocols that have been used for sensing applications including ZigBee, WirelessHART (Lennvall, Svensson et al. 2008) and 6LoWPAN (Hui and Thubert 2011).

2.3.2.3. 6LoWPAN

6LoWPAN (Hui and Thubert 2011) is an IETF Internet standard that specifies how IPv6 packets can be transmitted over 802.15.4 networks. Compared to other high level protocols that utilise 802.15.4, 6LoWPAN is open and provides addressing that is easily translatable to IP, reducing the development overhead by allowing the use of existing network based tools for commission, configuration and management (Mulligan 2007).

The 6LoWPAN standard specifies the frame format, header compression techniques and stateless address autoconfiguration methods for 802.15.4 networks (Hui and Thubert 2011). Header compression techniques, where address information is inferred from the underlying 802.15.4 header, mean that 6LoWPAN can facilitate the transmission of IPv6 packets with an overhead as low as four bytes. The 6LoWPAN standard does not specify routing within the network so routing is discussed in more detail in Section 2.3.2.4.

6LoWPAN has been used as part of the THREAD networking protocol (Thread Group n.d.), which is used for commercial IoT applications, and many open source 6LoWPAN implementations exist (Yibo, Hou et al. 2011). Despite its standardisation, the literature does not describe and fully standardised sub-GHz deployments.

2.3.2.4. Routing

Multi-hop routing capability is a key requirement for larger deployments as it allows individual nodes to have a lower transmit power and can improve the connectivity and reliability of the network by providing redundant paths that data can take to reach the sink (Schwieger, Fettweis et al. 2004). These benefits enable deployments to cover a large or geographically complex area in a more energy efficient and reliable way than is possible with directly connected nodes that must be within range and line of sight of a base station. While multi-hop routing helps to increase range and reliability it also increases complexity as each node in the network needs to be able to act as a router in addition to its sensing duties. Overall energy consumption can also increase compared to a non-multi-hop deployment as nodes need to spend time in receive when they could otherwise be in a low-power sleep mode so that they can act as routing points for other nodes in the network.

Several different methods of routing over 6LoWPAN have been developed to provide energy and memory efficient routing for 6LoWPAN networks (Ee, Ng et al. 2010, Kumar and Tiwari 2012) and one method, RPL, has been standardised by the IETF (Winter, Thubert et al. 2012). When compared to another routing implementation that was considered for IETF standardisation,

LOADng (IETF n.d.), RPL was found to have better performance (Vučinić, Tourancheau et al. 2013) for home automation. As RPL is the Internet standard for routing on 6LoWPAN networks, it is the routing protocol of choice for a standards-based IoT network.

Multi-hop routing also acts to reduce the available bandwidth across the network with each additional hop. Gupta and Kumar (2000) demonstrated that the bandwidth attainable through a single-channel multi-hop network where nodes can cause interference with each other can be calculated with equation (1). The bandwidth available in a multi-hop 802.15.4 network is less than this due to packet copying and other overheads so equation (2) gives an appropriate approximation (Österlind and Dunkels 2008, Jain, Reena et al. 2013). The reduction in realisable bandwidth attributable to each additional hop poses potential issues for ESN deployments that rely on a number of "fill in" routing nodes to enable communication across large areas or in difficult terrain as the reduction in realisable bandwidth can be significant.

$$BW = \frac{1}{\sqrt{n \log n}} \quad (1)$$

$$BW = \frac{1}{n} \quad (2)$$

2.3.2.5. Duty Cycling Protocols

Keeping a radio permanently in receive mode so that a packet can be received at any time is not energy efficient as most radios use tens of mA in receive. The energy overhead of wireless communications can consume up to 75% of the available energy in some applications with the majority being expended on listening on the chance that a packet may be received (Shnayder, Hempstead et al. 2004, Hong-Ling, Kun Mean et al. 2011). One way to reduce the amount of energy is by using a Radio Duty Cycling (RDC) protocol to keep the radio turned off or in a low-power state when not required to reduce the average energy utilisation. RDC Protocols can range from having a short daily slot for communications, as used by Elsaify, Padhy et al. (2007) in GWMAC, to schemes where the radio is cycled on and off several times per second, as used by Dunkels (2011) for ContikiMAC. Different RDC protocols may not interoperate due to the different methodologies employed to manage the radio so all nodes in a network must run the same, or a compatible, RDC protocols to ensure reliable communications.

RDC Protocols can be broadly separated into two categories: synchronous and asynchronous (Kuorilehto, Kohvakka et al. 2008). In synchronous RDC protocols all of the nodes have a predetermined schedule where they turn on their receiver to receive data. Any node that wishes to transmit to another node needs to be aware of the receive schedule of the destination node so that it

only tries to transmit when the target node is awake and ready to communicate. The degree of complexity in synchronous RDCs can range from simple, manually pre-allocated schedules such as the single daily slot used in GWMAC (Elsaify, Padhy et al. 2007) to more complex self-allocating or fully self-synchronising implementations as used in Crankshaft (Halkes and Langendoen 2007) and S-MAC (Wei, Heidemann et al. 2002) respectively. Independently of the implementation, each node in the network needs to have a synchronised clock or other form of accurate timekeeping to ensure that each node maintains its schedule. Implementations that transmit time information to correct for clock drift and self-synchronising protocols have a greater network overhead than other synchronised protocols due to the additional data that needs to be sent over the network. In contrast, asynchronous protocols require no prior or on-going synchronisation between nodes, instead relying on superfluous transmissions and a suitable timing scheme for radio wakeup with each node maintaining its own schedule. Preambles longer than a node's sleep period have been used in B-MAC, some implementations of ALOHA and TSMP (El-Hoiydi 2002, Polastre, Hill et al. 2004, Pister and Doherty 2008) to ensure that other nodes are awake at some point during the preamble and can keep their radio on to receive the actual transmission. An alternative to using increased-length preambles is to either transmit a short preamble several times as used in X-MAC (Buettner, Yee et al. 2006) or to retransmit the entire packet until acknowledged as used in ContikiMAC (Dunkels 2011). Asynchronous RDCs can behave more like synchronous RDCs by learning the schedules of other nodes in the network and only transmitting when it is expected that the destination node is awake and listening. This approach can be used in WiseMAC (El-hoiydi, Decotignie et al. 2004) and ContikiMAC to improve energy efficiency by reducing the amount of excess data transmitted at the cost of increased memory usage. Nodes running asynchronous protocols can be of a simpler design with less memory than those used with synchronous protocols, as asynchronous nodes do not need to keep track of the schedules of their neighbours and do not need such accurate clocks. Memory usage in synchronous deployments can become problematic in larger networks where nodes need to communicate with each other rather than just with the base station, as in a multi-hop network.

The achievable duty cycle of a protocol depends not only on the pattern of traffic but also how mobile nodes are in the network. Synchronous protocols can achieve very low duty cycles significantly below 1% in static deployments (Pister and Doherty 2008) but this increases significantly in dynamic deployments where the schedule has to adapt to nodes joining and leaving the network randomly or where nodes can "move" by changing neighbours. Modern asynchronous protocols can achieve duty cycles approaching 1% or less depending on traffic types and network conditions (Dunkels 2011, Pinola 2012). Asynchronous protocols can be more conducive to spontaneous communications as the transmitting node does not need to wait until a set time to try to send data

and relaying nodes can immediately retransmit rather than storing for later transmission: however, the unscheduled nature of asynchronous protocols means that there is an increased chance of collisions. The flexibility of asynchronous duty cycling protocols makes them appealing for ESNs deployed in harsh conditions as static nodes may appear to be dynamic in nature due harsh and changing environmental and large bursts of traffic that won't fit into a synchronised slot may need to be sent if a node has been off the network for an extended period. Contiki includes drivers for several asynchronous RDCs (Contiki-OS 2014) that can be chosen at compile time. These are ContikiMAC, X-MAC (Buettner, Yee et al. 2006), CXMAC and LPP (Musaloiu-E., Liang et al. 2008). Contiki does not include any synchronous RDC protocols but it does include NullRDC which does not cycle the radio at all and is intended for testing purposes and non-energy-sensitive deployments. Of the available RDCs, ContikiMAC is the most energy efficient under most situations at 2.4 GHz with a radio that has similar transmit and receive energy profiles.

Another method that has been used in WSNs to reduce the energy expended on wireless communications is data aggregation (Wang, Heinzelman et al. 1999, Heinzelman, Chandrakasan et al. 2000). Data aggregation reduces the energy expended on wireless communications by the network as a whole by aggregating data from different nodes as it travels through the network, reducing the total number of transmissions. Early implementations aggregated data at the edge of the network however attempts have been made to carry out aggregation within the low-power network by sensor nodes as nodes have become more capable (Fasolo, Rossi et al. 2007). While data aggregation has been applied to urban and agricultural WSNs successful, its use for ESNs has not been established in the literature. In an ESN context, in-network data aggregation would appear to be less suitable as the transmission of data is essentially slowed down and left on a potentially vulnerable node for longer than necessary though further work is needed to establish the limitations and benefits of applying data aggregation to ESNs.

2.3.3. IoT Operating Systems

Certain enabling technologies, such as 6LoWPAN network stacks and routing implementations, benefit from being executed in under a scheduler rather than as bare-metal code. Several operating systems for low-power MCUs that are designed or adapted for IoT applications exist to fulfil this need with implementations ranging in nature from simple co-operative, non-pre-emptible non-real-time designs to more elaborate real-time operating systems (RTOS). Significant contemporary open-source operating systems that have been used for IoT applications include Contiki (Dunkels, Gronvall et al. 2004), Tiny-OS (Levis, Madden et al. 2005), FreeRTOS (Real Time Engineers Ltd. 2003), OpenWSN (Watteyne, Vilajosana et al. 2012) and RIOT (Baccelli, Hahm et al. 2012, Baccelli, Hahm et al. 2013). mbed OS (ARM Ltd. 2014) is an IoT-focussed OS that is currently

under development by ARM® but this will only target ARM® Cortex®-M-based MCUs. Unlike the other mentioned operating systems, FreeRTOS does not include a networking stack that supports IPv6 in its base distribution and instead relies on third-party, and usually proprietary, stacks to provide this functionality. One open-source 6LoWPAN stack available for FreeRTOS is Nanostack 1.1 however this is no longer developed (Nanostack Project 2013a). OpenWSN can run on its own scheduler, OpenOS, and support for using the FreeRTOS kernel as an alternative is currently under development (OpenWSN Project 2012).

A brief comparison of these operating systems is presented in Table 2. This shows that the Contiki and OpenWSN networking stacks are the only options that support TCP over 6LoWPAN and that Contiki and FreeRTOS support the largest range of processors. Tiny OS currently supports 8-bit and 16-bit MCUs and only have experimental support for TCP over 6LoWPAN. TCP is required for some higher level protocols, such as HTTP, so a lack of support can limit the utility of a node. RIOT is a fairly new OS and does not have the deployment history of the other options; however, it does have a significantly lower memory and flash footprint than the other operating systems.

Table 2: Comparison of IoT Operating Systems with 6LoWPAN stacks. * where OpenWSN is used with the OpenOS scheduler (Watteyne, Vilajosana et al. 2012). RAM and ROM information from (Yibo, Hou et al. 2011, Baccelli, Hahm et al. 2013)

OS	**Contiki**	**Tiny-OS**	**FreeRTOS (with NanoStack)**	**OpenWSN**	**RIOT**
Supported Processors	MSP430, AVR, Cortex®-M, ARM7, 8051, RL78, 6502, x86 (Contiki-OS 2012a)	MSP430, AVR (TinyOS 2011)	MSP430, AVR, Cortex®-M, Cortex®-A, Cyclone V SOC, ARM7, ARM9, Smart-Fusion, APS3, PIC32, V850, 78K0R, 8051, RM48, TMS570, PPC4xx, Zynq, NIOS II, Microblaze, x86 (Real Time Engineers Ltd. 2012)	MSP430, Cortex®-M *	MSP430, ARM7, Cortex®-M, x86 (RIOT 2013)
RAM Usage	10 kB	10 kB	4-8 kB		1.5
Flash Usage	30 kB	48 kB	32-64 kB		5
TCP Support	Y	Experimental	N	Y	Y
UDP Support	Y	Y	Y	Y	Y
RPL	Y	Y	N	Y	Y

Due to its deployment history, processor support and mature network stack Contiki was selected for further investigation. Had it been more mature in the early stages of this PhD, RIOT would have been investigated further.

Contiki is a multi-tasking operating system for networked applications that is designed to work on memory-constrained platforms such as low-power microcontrollers (Dunkels, Gronvall et al. 2004). Contiki has been ported to many different processing cores and platforms, ranging from 8-bit microcontrollers to running as an application on top of another operating system such as Linux. While Contiki includes some real-time capability through the use of "real-time timers", it is not an RTOS and instead provides co-operative multitasking by using Protothreads to implement processes (Dunkels, Schmidt et al. 2006). Protothreads are stackless threads that do not support pre-emption by other threads. Their stackless nature means that the use of local variables within threads can be problematic. To have the least impact on the currently running co-operative code, interrupts should be kept as short as possible and should generally poll a process or otherwise pass an event to Contiki's scheduler so that the event can be handled by code executing co-operatively. Contiki can support pre-emptive multithreading through the "mt" library (Contiki-OS 2012b) but this is not supported on all platforms and comes with a significant memory overhead.

Contiki uses several layers of abstraction. Specifically, microcontroller-specific code for core functions is kept isolated from the platform-specific code. This is in turn kept isolated from more generic code and interoperability is ensured through a well-documented API. The abstraction extends to the network stack, which has options for both IPv4 and IPv6, where each layer is kept separate from other layers with well-defined APIs existing between them. This allows network protocols, such as Radio Duty Cycling Protocols and network framers, to be changed with minimal work at compile time.

In summary, Contiki provides a memory efficient and consistent basis across numerous different devices with different processing capabilities and energy characteristics for IoT applications. The abstraction that is present within Contiki not only permits complete applications to be platform-agnostic but it also allows them to be protocol-agnostic. The wide processor support, open-source nature of the project, simulation options and significant research input have allowed Contiki to grow to be one of the most significant IoT operating systems. Given these features, Contiki was chosen as the operating system for the sensor nodes discussed in Chapter 4 and Chapter 5.

2.4. Environmental Sensor Networks

Environmental sensor networks can be seen as a subtype of WSNs that are deployed in harsher and more remote environments than WSNs to monitor natural processes (Martinez, Hart et al. 2004). ESNs have been used for a variety of environmental monitoring applications including habitat monitoring (Szewczyk, Osterweil et al. 2004), forest fire detection (Lazarescu 2013), glacier monitoring (Martinez, Basford et al. 2009, Smeets, Boot et al. 2012), volcano monitoring (Werner-Allen, Johnson et al. 2005, Song, Huang et al. 2009) and rock face monitoring (Talzi, Hasler et al. 2007, Hasler, Talzi et al. 2008, Beutel, Gruber et al. 2009, Girard, Beutel et al. 2012). These applications show that common requirements for ESNs are very low power & efficient operation and resilience as the nodes need to operate for extended periods of time with limited, and usually finite, energy sources in potentially harsh environments where access for maintenance and replacement can be limited. "Sleepy" networks using sub-GHz low-power radios are a common theme though 2.4 GHz radios, common to ordinary WSNs, Wi-Fi and GPRS have been used in some deployments.

In terms of standardisation, environmental sensor networks have been slow to adopt the Internet standards now commonly used by wireless sensor networks used for indoor and urban applications (Hart and Martinez 2015). In turn, the lack of standardisation has slowed the adoption of environmental sensor networks for Earth sciences research as deployments require specialised skills and knowledge to deploy, maintain and retrieve data from the deployment which are outside of the field of Earth sciences researchers.

While in theory standards like 802.15.4 and 6LoWPAN can be applied to the sub-GHz bands that have been used by some environmental sensor networks, there is a lack of literature on the subject. It has also been noted by other workers that theory and laboratory based testing do not reflect real-world environmental sensor networks (Langendoen, Baggio et al. 2006, Barrenetxea, Ingelrest et al. 2008, Burgess, Kranz et al. 2010). In the real world, conditions cannot be controlled and there will always be unexpected challenges that may mean that things that behaved perfectly in theory or in the laboratory do not perform as well in the real world or fail to function at all.

Additionally, standards and integration into the Internet of Things requires more processing power to handle the overheads involved (Zimmermann, Sa Silva et al. 2008). This is at odds with the general requirement for extremely low-power operation prevalent in environmental sensor networks.

2.5. Summary

Technological advances and standardisation in recent years have allowed wireless sensor networks to become cheaper, easier to deploy and easier to interact with by using standardised networking protocols that are translatable to standard IP networks such as the Internet. Wireless sensor networks are also seen as a key enabler of the Internet of Things as they bridge the gap between the physical and real worlds, allowing remote interaction with the environment and autonomous sensing and actuation. The Internet of Things is seen as part of the future Internet and IoT concepts have been applied to a wide range of applications that are intended to simplify and improve life. Because of their popularity, a large amount of research has been conducted on integrating wireless sensor networks into the Internet of Things.

This literature review has shown that environmental sensor networks use the same underlying technology as wireless sensor networks, but they have been slower in adopting the standardisation that has contributed to the success of WSNs. The underlying standards, specifically 802.15.4, do specify parameters for using sub-GHz radios as part of a standards-compliant networking stack; however, the use of 802.15.4 with sub-GHz radios in conjunction with other standards required for standardised networking, such as 6LoWPAN and HTTP, has not been demonstrated in the literature. This thesis details work that addresses this gap in the literature and has resulted in the first publications detailing real-world sub-GHz 6LoWPAN-based environmental sensor network deployments.

Standardised networking requires additional processing capability than the bespoke approaches used for existing ESNs and the literature has shown that complex approaches using multiple processing elements have been investigated to make sensor nodes more capable. Advances in microcontroller design mean that higher performance microcontrollers with similar, or better, energy performance characteristics are now available and show promise for increasing the performance capability of a sensor node without significantly affecting lifetime. The performance consideration drives the platform choice for the hardware used in Chapter 4 and the initial deployments in Chapter 5, where the performance limitation of a commonly used contemporary platform is demonstrated.

Additionally, contemporary sensor networks make extensive use of methods of duty cycling the radio in order to significantly decrease average energy and increase node lifetime. Synchronous duty cycling protocols can give the best performance in terms of energy as very low duty cycles can be achieved. Synchronous duty cycling protocols can fail, as occurred in Chapter 3, if synchronisation between the nodes is lost unless methods of re-synchronisation or recovery are implement-

ed at the cost of increased complexity. Contemporary IoT wireless sensor networks make use of asynchronous duty cycling protocols that ultimately give the appearance of having an always-on network, which is beneficial for IoT deployments. This makes a network easier to deploy as explicit synchronisation between nodes is not required to enable communications and changing network topologies can be accommodated more readily. Asynchronous duty cycling approaches can also improve reliability as communications can occur when possible, rather than being restricted to a set time slot that may be unsuitable due to environmental or other external factors. Given the characteristics of asynchronous duty cycling protocols they were used through the deployments detailed in Chapter 4 and Chapter 5.

In summary, the literature has shown that significant research has been undertaken in the fields of wireless sensor networks and environmental sensor networks. Wireless sensor networks have been able to become a key part of the IoT through the adoption of standardised methods of communication and other standardised protocols. In contrast, environmental sensor networks have been slow to adopt these same standards and there is a gap in the literature surrounding their use for sub-GHz ESNs deployed in harsh environments. Work is required to investigate whether standards can be applied to such a network successfully and whether they have suitable performance in terms of energy, throughput and latency for real-world deployments. This thesis goes some way to addressing the gap in the literature and the following chapters describe work that has been carried out to address these shortcomings.

Chapter 3

Glacsweb

This chapter details work carried out during 2013 as part of the Glacsweb project to improve the deployment network, update the deployed hardware and to carry out maintenance on the existing hardware. This work was used to gain a deeper understanding of real-world environmental sensor networks and the challenges surrounding their deployment and use.

Glacsweb was a research project that studied the Skalafellsjökull glacier in Iceland using an Internet-connected WSN (Martinez, Basford et al. 2012). The project used deployments of a heterogeneous mix of nodes to monitoring several different factors and an example of a typical ESN deployment. Sub-glacial probes monitor movement, pressure, strain and temperature in the glacier and in the till below the glacier. Seismic Deep Nodes (SDN) use geophones to monitor sudden seismic events within the glacier and Surface Sensor Nodes (SSN) sited above the SDNs relayed the data to a base station. Standalone non-connected Differential GPS (DGPS), with a reference station on the moraine near a camera node, were mapped with the SSNs to provide accurate monitoring of glacial movement.

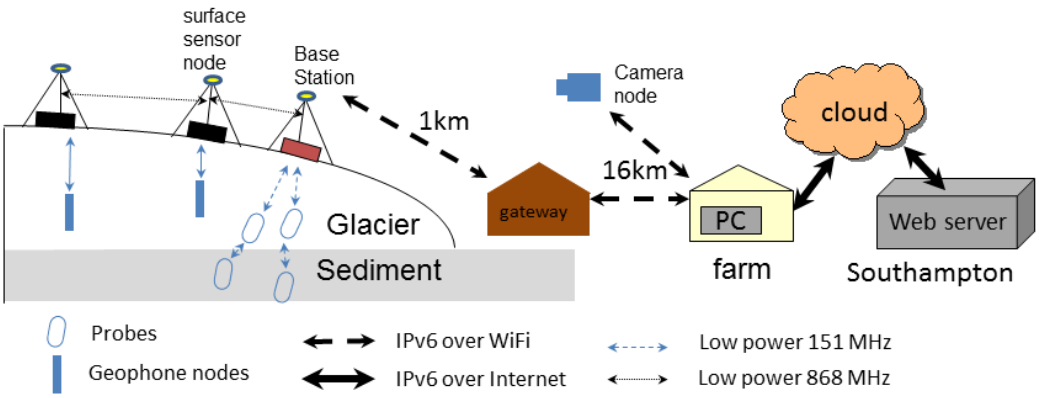


Figure 7: Sensor network deployed in Iceland with the types of network connection labelled. Reproduced from Martinez, Hart et al. (2017)

The deployment used a mix of different communications technologies to transmit sensor data from the glacier as shown in shown in Figure 7 (Martinez, Hart et al. 2017). A long-range Wi-Fi connection was used to connect the reference (referred to as gateway in Figure 7) and a camera node mounted at on the moraine beside the glacier to a local farm to provide Internet connectivity. A second long-range Wi-Fi link is used to connect the reference station to a battery powered base station located on the ice. 151 MHz low-power radio, realised with the RPM-1 radio module, is used to connect the sub-glacial probes to the base station and 868 MHz low-power radio, realised with a CC1120 (Texas Instruments 2015a), is used to connect the SSNs to the base station. The SDNs use a wired RS485 connection to connect to the SSNs though a fall back to 173 MHz with a CC1120 was included in the design. As the long-range Wi-Fi connections can be unreliable, the base station includes a GPRS modem for backup communications and was also designed to accommodate a satellite modem (Basford 2015). Data gathered by the deployment was sent to a cloud server and presented through a web server.

The remainder of this chapter is divided into four sections. Section 3.1 details changes that were made to the deployed network to evaluate new technologies and improve reliability. Section 3.2 details a re-design of sensor node hardware to extend the lifetime of the deployment. Section 3.3 details field work that was carried out to implement these improvements and describes the specific challenges that were encountered during deployment. Finally, Section 3.4 summarises the chapter and discusses the lessons learned during this work. Work carried out as part of this chapter has been published in Martinez, Hart et al. (2013) and Martinez, Hart et al. (2017).

3.1. Network

The Glacsweb project used a mixture of IP and non-IP based communications in the deployment. The 2.4 GHz Wi-Fi links used private IPv4 range via NAT to the Internet whereas the low-power radio links used a bespoke communications protocol with a very low duty cycle synchronous RDC. As identified in Chapter 2, IoT concepts have the potential to advance ESN deployments and IPv6 is one of the potential enablers in applying IoT concepts to ESNs. The use of NAT for IPv4 also meant that the IP-based nodes connected to the network could not be directly managed from outside of the network. This section details the changes that were made to the network to implement IPv6 and to prepare it for a future implementation of 6LoWPAN. An issue with the duty cycling protocol used for the sub-glacial nodes is also identified and briefly discussed.

3.1.1. Network Improvements

The networking equipment at the reference station was replaced to improve performance and enable the implementation of IPv6 discussed in section 3.1.2. The previous implementation used two access points at the reference station to connect to the Farm Wi-Fi network, using a high-power Wi-Fi amplifier (500 mW), and to provide Wi-Fi connectivity to the base station on the ice. These two access points were replaced with a single system containing two Wi-Fi cards. An Alix 3D1 x86 embedded computer (PC Engines 2008) was chosen due to its compact size, ability to have multiple Wi-Fi network cards and support for embedded Linux distributions. A high-power Atheros 9K-based Wi-Fi card was used for the 16 km connection to the farm and a standard Broadcom Wi-Fi card was used for the link to the base station. OpenWRT 12.09 "Attitude Adjustment" (OpenWrt Project 2004) was used as the operating system on the Alix board due to its support for the chosen hardware and its ability to be modified with the evolving needs of the project. The Alix board was installed in the reference station during a field trip in August 2013. Additionally, both Wi-Fi antennas were replaced due to damage over the winter months.

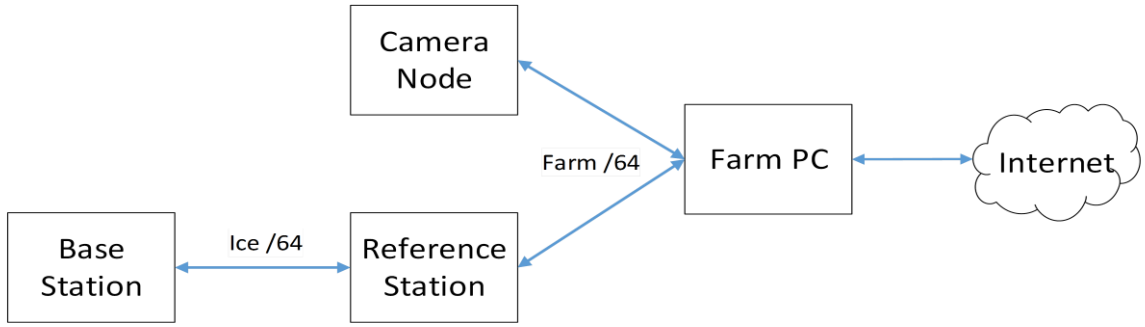
The change to a single device managing the two Wi-Fi networks reduced power consumption at the reference station. The two access points used approximately 4.5 W each, for a total of 9 W excluding the energy used by the high-power Wi-Fi amplifier. With a high-power Wi-Fi card the Alix 3D1 used 5.7 W, significantly less than the previous implementation.

3.1.2. IPv6 Implementation

IPv6 was added to the existing IPv4 networks in a dual-stack implementation. Global IPv6 addresses were allocated to the base station, reference station, camera node and PC at the farm. The introduction of IPv6 meant that the nodes that had previously been behind a NAT router could now be directly managed by authorised hosts without first having to log in through another system. Once implemented, all of the data collected by the deployment was transferred using IPv6 to the cloud server and then served on an IPv6-capable web server. The Internet connection available at the farm did not support native IPv6 so it was implemented on the network using tunnelled IPv6 through the SixXS tunnel broker (SixXS 2002). While not as performant as native IPv6, tunnel brokers provide a stop-gap solution to providing IPv6 until adoption increases.

Routing, rather than bridging, was implemented at the reference station to logically represent the network architecture and facilitate any future expansion to 6LoWPAN or other IPv6 capable nodes. An abstracted routing diagram is depicted in Figure 8 and shows that two /64 subnets are used. A stateful IPv6 firewall was implemented on the Farm PC to prevent unauthorised hosts from being

able to interact directly with the devices on the network. Implementation of a translation layer on the border router to bridge between the low-power networks used for the sub-glacial nodes and the SSNs to IPv6 was considered, however it was deemed to not be practical within the schedule. A translation layer would have to have been implemented on the base station, increasing the processing overhead and introducing additional complexity.



Base Station	
Route	**Via**
Default	Reference Station

Reference Station	
Route	**Via**
Default	Farm PC
Ice /64	Ice Interface

Camera Node	
Route	**Via**
Default	Farm PC

Farm PC	
Route	**Via**
Default	Internet
Farm /64	Farm Interface
Ice /64	Reference Station

Figure 8: Abstracted network diagram and routing tables of the IPv6 implementation on the Glacsweb network.

3.2. Sensor Node Redesign

In addition to upgrading the network, the SSN and SDN designs were modified to correct hardware issues and to accommodate the evolving needs of the project. This section details the work carried out in redesigning the SSN, to accommodate a new low-cost GPS module for evaluation and the SSDN to correct errors in the original design. The physical layout of the SSDN was also changed to permit deployment in a smaller bore hole that would not require bulky equipment to bore.

3.2.1. Surface Sensor Node

The surface sensor nodes use an 868 MHz low-power radio network to communicate with the base station and have the option to use either RS485 or 173 MHz low-power radio to communicate with the SDNs. The SSNs were responsible for relaying geophone data from the SDNs to the base station and were mapped with millimetre-accurate multi-band Leica System 1200 commercial differential GPS units that were used to precisely log the movement of the surface of the glacier for cor-

relation with geophone data from the SDNs. These GPS units were standalone and data had to be manually retrieved at least twice a year, meaning that comparisons with the SDN data could not be made as the data was gathered.

The need for manual intervention to retrieve data is undesirable in an ESN so a GPS that is connected to the network in some way is desirable. Recent advances in GPS receiver modules have reduced the cost of single-band receivers with raw data output. Raw data output is required for post processing GPS data against a reference station to improve accuracy. A low-cost single-band GPS module that supported raw data output over serial (Munaut 2012) was selected to trial the use of an integrated solution and to assess the feasibility of using single-band receivers for precise monitoring of glacial movement. Methods of post processing for single receiver scenarios also exist, one of which is Precise Point Positioning (PPP). PPP uses global reference data for satellite orbit and clock information to achieve centimetre accuracy (Bisnath and Gao 2009). Online resources exist for processing data so the use of PPP will also be explored.

The SSN was redesigned together with another member of the team to accommodate the low-cost GPS so that it could be assessed whether it is an appropriate alternative to a full differential GPS system. Figure 9 shows a photograph of an SSN and GPS board, which was deployed in 2013 and recovered in 2014, complete with the box that they were deployed in.

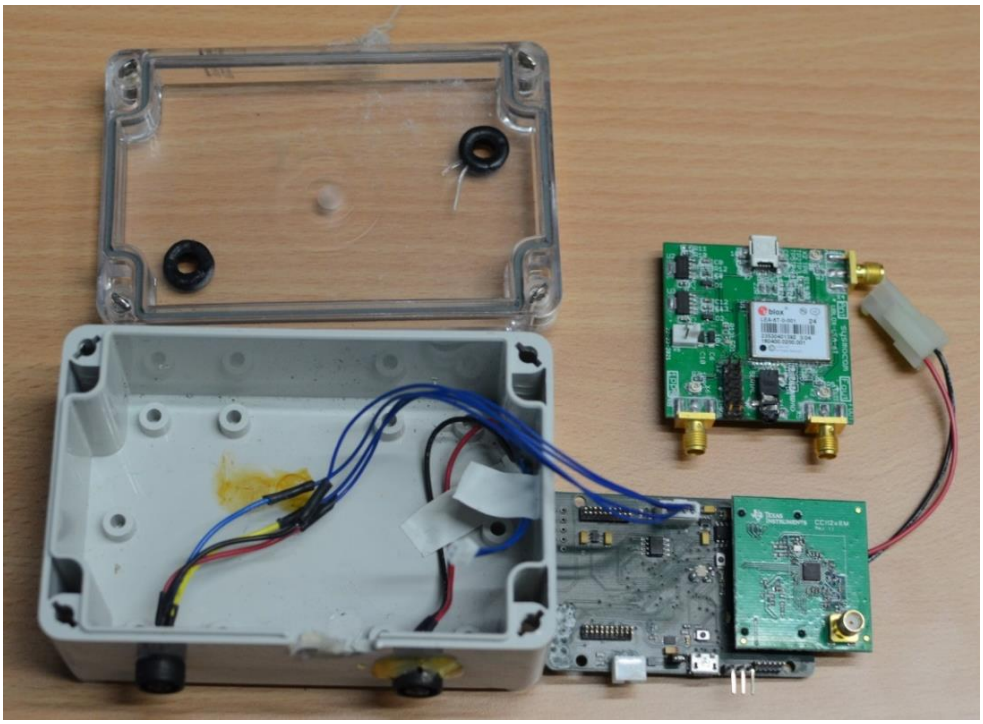


Figure 9: Photograph of an SSN and its Osmocom LEA-6T GPS board after it was recovered in 2014. Image reproduced from Ward (2017).

3.2.2. Seismic Deep Node

The SDNs that were deployed originally only communicated data over the network for a short time after they were deployed so it was decided that replacement nodes should be deployed to gather more data. The original nodes were deployed in bore holes cut with a hot water jet wash system however this would not be available for the deployment field work so the holes would have to be hand-bored, resulting in a smaller diameter borehole. The SDN was redesigned as the Slim SDN (SSDN) to correct design errors that existed in the original design and to enable the node to fit into a slimmer housing.

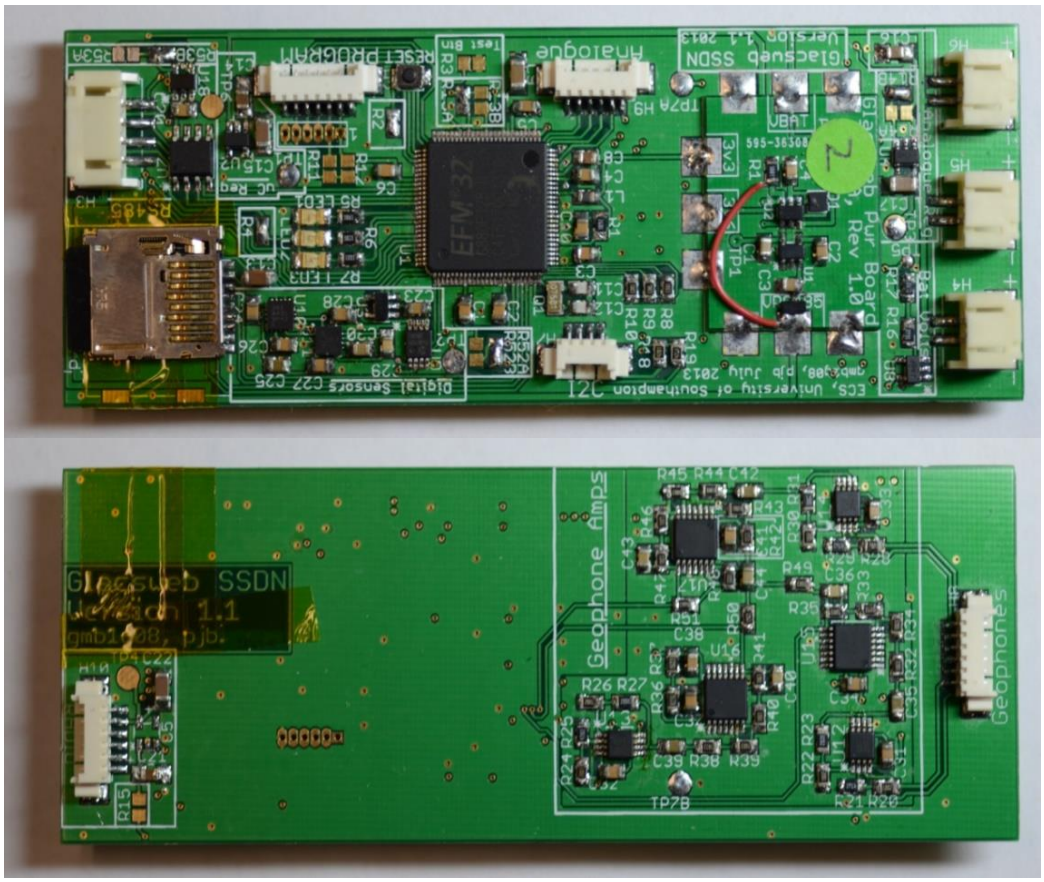


Figure 10: Photograph of top (upper PCB) and bottom (lower PCB) sides of the SSDN PCB. Two modifications were made during assembly: The SD card holder that the PCB was designed for was unavailable when the parts were purchased so an alternative was used and a modification was required to the power supply.

When deployed, the SDNs had two sources of power: a 5 V feed from the connected surface node and an internal 3.6 V lithium battery. The battery was intended to be used to keep the node sampling when power from the surface was lost, either due to the battery on the SSN running flat or the cable between the SSN and SDN breaking. The changeover was originally implemented as a sim-

ple combiner diode. During lab testing it was found that this design did not work as intended and the node failed to operate from the internal batteries due to the voltage drop across the diode reducing the input voltage below the dropout level of the on-board 3.3 V power supply. This issue was resolved by using a low-resistance single-pole double-throw TS5A3160 analogue switch (Texas Instruments, 2012) to switch between the 5 V supply from the surface, which was regulated to just under 4 V before being switched, and the internal battery. Other changes and improvements included replacing an analogue platinum temperature sensor with an I2C-connected digital thermometer and the removal of the gyroscope. External connectors were included for I2C and analogue inputs to make debugging easier and to enable future expansion. The PicoFlex connector for the RF connection was replaced with a PicoBlade to improve reliability and to save space. Finally, the PCB was changed from a 51mm round design to a 110mm by 40mm rectangular design to fit into the smaller diameter bore hole. A photograph of the assembled SSDN is shown in Figure 11. Schematics and PCB layouts are included in Appendix 2.

3.3. Field Work

Fieldwork was carried out in August 2013 to deploy the redesigned nodes and to carry out maintenance on the existing deployment. The network improvements detailed in 3.1.1 and the IPv6 implementation detailed in 3.1.2 were implemented as described. The aim of this work was to extend the working life of the deployment to allow additional environmental sensor data to be collected and to carry out initial in-field evaluation of IPv6 and the low-cost GPS receivers discussed in Section 3.2.1.

3.3.1. Geophone Nodes

The existing SDNs that were in situ were assessed to determine whether any data could be recovered. All of the existing SSNs had depleted batteries so an RS485 transceiver connected to a laptop was used to interrogate the SDNs. Some nodes were found to be operable and some data was recovered. The SDNs that were still operable were connected to the new SSNs and left in place to gather more data.

For a two-hour GPS recording, the low-cost GPS generated a 2.3 MB data file that was sent over the low-power 868 MHz network to the base station. PPP post-processing using the online facility provided by Natural Resources Canada (2016) gave an accuracy of 50 cm. The same data was processed with reference to the Leica reference station on the moraine by another member of the team and found to have an accuracy of 9 cm.

During final testing before deployment, it was found that the SSDNs suffered from intermittent reboots that also caused the internal clock to reset, offsetting the node's time and date. This issue was found to be due to a software issue in code that was developed by another researcher that manifested itself during SD card activity however there was insufficient time during the field trip to debug this issue so the SSDNs were not deployed. On further investigation, it was found that a watchdog reboot was triggered when an interrupt triggered task attempted to access the SD card while it was already in use. The original software used bare-metal code without a scheduler and no locking of resources was performed.

3.3.2. Sub-glacial Probe Radio Duty Cycling Failure

The wired probe used to communicate with the sub-glacial probes deployed in 2012 had failed preventing data being retrieved from any of the probes. A replacement wired probe was deployed in an attempt to contact the probes and retrieve any stored data as battery calculations indicated that the probes were reaching the end of their life but should still have adequate energy for communications. The nodes used a very simple synchronised duty cycling protocol with one communication period of one minute each day where the probes would turn on their radio and listen for solicitations from the base station before turning the radio off again. To account for potential clock drift due to the extended period of disconnection, the base station was set to transmit regular solicitations to all of the deployed sub-glacial nodes with the replacement wired probe however there was no response from any of the nodes.

The sub-glacial probes shared common code and microprocessor with the SDN so it is likely that the intermittent reboot issue identified in 3.3.1 also manifested in the sub-glacial probes meaning that any nodes that were still functional could have a more significant clock offset than originally anticipated. Because the synchronous RDC did not have a method to recover from a significant clock offset this would result in a complete failure of radio communications. To test this hypothesis a second base station was deployed and set to repeatedly send requests to all of the nodes every minute over a period of 24 hours in an effort to make contact. It was found that some of the probes were still active with varying degrees of clock drift.

The duty cycling used for the probes achieved a duty cycle of 0.069% with a very low processing overhead however the inability to correct for extreme clock drift resulted in a large amount of data not being retrievable. As the base station was also battery powered, the method of recovery used is not appropriate for use during the deployment on a regular basis.

3.3.3. Fieldwork Summary

Despite no new SDNs being deployed, the fieldwork described in this section was a partial success. Maintenance carried out extended the operational lifetime of the network, enabled additional sensor data to be collected and allowed new technologies to be assessed in an actual deployment. IPv6 was successfully implemented on the network and resulted in the realisation of IPv6 connectivity all of the way to the glacier. This served as an initial test of the IPv6 connectivity later used in the Mountain Sensing project as detailed in Chapter 5. The lower-cost GPS modules were also tested and found to have unacceptable accuracy for glacial movement monitoring. GPS data was successfully communicated over the deployment network.

The failure of the redesigned SDNs just prior to deployment demonstrated that the amount of pre-deployment testing carried out was not adequate. More thorough testing would have revealed the issue earlier and may have allowed enough time to adequately address the issues.

3.4. Summary

This chapter has described foundational work that was carried out to gain familiarity with a real-world Environmental Sensor Network deployment and to understand the challenges and issues involved. This section looks at the lessons learned from this work and discusses how this work drives the future work in this thesis. The experience with this deployment illustrates the truth in the sentiment of Langendoen, Baggio et al. (2006) and Barrenetxea, Ingelrest et al. (2008) that Murphy's law will make its way into a real-world deployment. Examples in the original deployment include the failure of the original SDNs after a short period and the failure of the sub-glacial probe RDC, which resulted in unexpected issues with the 2013 field work. These in-deployment failures can be mitigated in part by carrying out more extensive pre-deployment testing, including proof-of-concept deployments in a non-lab environment, and designing hardware to be more resilient.

Synchronous RDCs with a very low duty cycle can save energy and extend node lifetime, permitting more data to be collected; however, this is not useful if the protocol does not have a method of recovery for nodes that lose synchronisation if the nodes are physically unreachable as the data will be lost. As discussed in section 2.3.2.5, methods of self-correction and other methods of increasing the reliability of synchronised duty cycling protocols come at the cost of increased total energy consumption, either during normal operation or during a failure mode. An alternative is to increase the reliability of the sensor node hardware to make a loss of synchronisation less likely however this comes at the cost of increased node complexity, cost and power. For the SDN nodes, which

relied on a counter integrated in the microcontroller for time, the inclusion of a dedicated RTC with a battery backup would prevent the loss of synchronisation from reboots caused by the software issue or other random events. Another alternative would be the use of an asynchronous RDC so that no direct synchronisation between nodes is required. This would make multi-hop ESNs easier to realise as no complex synchronisation is required; however, an asynchronous RDC would mean an increase in overhead for each packet transmitted and an increased duty cycle. Low duty cycle RDCs reduce the amount of time that a node is on the network and this has the potential to interfere with the routing protocols required for a multi-hop 6LoWPAN network. These factors mean that there is a power-reliability-performance trade-off that requires further investigation into duty cycling protocols and routing for IoT inspired ESNs.

IPv6 adoption by ISPs is an obstacle to deploying full IoT-like ESNs that rely on existing infrastructure in remote locations. The existence of IPv6 tunnelbrokers, such as SixXS, goes some way to mitigating this obstacle as they make it relatively easy to implement IPv6 on networks that do not have native support. Tunnelled IPv6 introduces its own issues as it increases latency and increases the amount of data sent over the Internet connection. Tunnelled IPv6 also introduces potential security issues as it bypasses any existing network security and gives devices a globally routable IP address so care needs to be taken not to compromise the security of networks this solution is deployed on. The work to deploy IPv6 to the Glacsweb network has made it easier to manage the nodes that make up the deployment and has laid the foundations for a potential future implementation of 6LoWPAN for the low-power radio connections.

While the low-cost GPS did not perform adequately for the intended task, it did demonstrate that GPS can be integrated into an ESN and raw data transferred successfully. The data generated by GPS can have a significantly greater size than typical sensor data so higher bandwidth connections will be required in IoT ESNs to be able to cope with this data efficiently.

The experiences detailed in this chapter influenced the work detailed later in this thesis. An asynchronous RDC was investigated in Chapter 4 and assessed as part of a real-world deployment in Chapter 5. Hardware used in Chapter 5 followed an incremental design process with significantly more thorough testing being performed. Additionally, the same technique for implementing IPv6 on an IPv4-only network was used throughout the deployments in Chapter 5.

Chapter 4

Sub-GHz 6LoWPAN Communications

An IoT ESN requires appropriate communications to function properly; however, the 2.4 GHz radios that have been extensively used for IoT WSNs have a useful range of tens to hundreds of metres rather than the kilometres required for ESN deployments. Sub-GHz low-power radios can provide the required range at the cost of a lack of standardisation and reduced throughput. As discussed in Chapter 2, 6LoWPAN is one of the enablers of the Internet of Things and this relies on the 802.15.4 standard. 802.15.4 specifies several sub-GHz bands so it is theoretically possible to have a sub-GHz 6LoWPAN network however there is an absence of literature on the subject. This chapter partially addresses the first research question stated in Chapter 1: whether the standards used for 2.4 GHz IoT wireless sensor networks can be applied to sub-GHz environmental sensor networks.

The remainder of this chapter is divided into five sections: Section 4.1 describes the hardware and software used for the work detailed in this chapter; Section 4.2 details the development of an Contiki radio driver for the CC1120 low-power radio that functions with the ContikiMAC RDC, determines the timings required to use ContikiMAC at 868 MHz with a data rate of 50 kbps and makes initial comparisons to another asynchronous RDC; Section 4.3 assesses the theoretical performance of a sub-GHz 802.15.4 network and presents a method of accurately estimating the round trip latency; Section 4.4 experimentally validates the theoretical performance; Finally, Section 4.5 summarises the chapter.

The work detailed in this chapter has been published in Bragg, Martinez et al. (2016) and three main contributions are made:

- It is shown that 6LoWPAN can be successfully used with sub-GHz radios.
- An open source Contiki radio driver for the CC1120 low-power radio that functions with the ContikiMAC RDC has been developed and is publicly available (See Appendix 3).
- Timing parameters for using ContikiMAC with 50 kbps 868 MHz radios were determined.

4.1. Scheduler, RDC & Hardware Selection

The additional complexity of a 6LoWPAN network stack requires some form of scheduler to function correctly. Of the IoT operating systems reviewed in section 2.3.3 Contiki was selected due to its comprehensive support for different microcontrollers, active support community and integrated IPv6-capable network stack that supports TCP. Contiki has some support for sub-GHz radios, for example it has a driver for the CC1101 (Texas Instruments 2016a), however these radios do not comply with 802.15.4 so cannot be used for a standards-compliant implementation of 6LoWPAN.

An existing commercially available platform that is well supported by Contiki was selected to reduce the amount of time required to develop the driver. A Zolertia Z1 (Zolertia 2010) was chosen as the foundation of the test system as it is well supported and has been used for IoT WSNs.

The CC1120 (Texas Instruments 2015a) is a high-sensitivity, low-power sub-GHz single-chip transceiver that complies with 802.15.4g-2012 in the 169 MHz, 470 MHz, 868 MHz and 915 MHz bands with data rates up to 200 kbps. Like most 802.15.4-compatible radios, the CC1120 has separate 127-byte TX and RX buffers. 125 bytes are usable for a packet when the buffer is empty and the optional status bytes, which include the received signal strength indicator (RSSI) and the link quality indicator (LQI) of the received packet, are enabled. The radio can automatically handle CRC checks without waking up the microcontroller and silently drop packets that are incomplete or corrupt to help reduce the energy overhead. Unlike the common 2.4 GHz 802.15.4 radios that are used for IoT WSNs, the CC1120 is not able to handle automatic acknowledgement of 802.15.4 packets destined for the node so all ACK handling has to be carried out by the microcontroller during packet processing. Despite this limitation, the CC1120 is a suitable radio for sub-GHz IoT WSN deployments due to its frequency and throughput versatility.

As discussed in Chapter 3 synchronous RDCs can be complex if they include methods of recovery and asynchronous RDC have become more common. ContikiMAC is claimed to be the most energy efficient of the protocols included with Contiki however, this has only been demonstrated for radios whose transmit and receive energy are similar (Dunkels 2011). As such, ContikiMAC may not be the most energy efficient choice of RDC for radios where the transmit energy utilisation is significantly greater than the receive energy utilisation, such as the CC1120 sub-GHz radio (Texas Instruments 2015a), and a protocol that does not rely on retransmissions may be more appropriate. The CC1120 incorporates support for a duty cycling protocol called RX Sniff that operates with minimal input from the microcontroller (Texas Instruments 2013a), reducing the processing overhead of duty cycling. RX Sniff relies on only needing to receive a shorter preamble than what is

transmitted. This has the potential to reduce energy consumption beyond what is possible with ContikiMAC as the microcontroller can spend more time in a low-power state but still relies on extended length transmissions.

4.2. CC1120 Driver

At the time this work was carried out, Contiki did not include a driver for the CC1120. Thingsquare, a distribution of Contiki, includes a basic CC1120 driver based on the Contiki CC1101 driver that does not support ContikiMAC and has incomplete functionality. A driver for the CC1120 was developed for the Zolertia Z1 that follows the principle prevalent within Contiki of keeping as much code execution in the co-operative space as possible and limiting what occurs pre-emptively. A GPIO pin from the CC1120 set to the MCU Wakeup function provides a single interrupt source for the entire radio driver. Additionally, the driver has been written following the abstraction principles used throughout Contiki and in other radio drivers so that it can be easily ported to other platforms. Unlike the CC2420 driver, the CC1120 driver fully respects the relevant state diagram for the radio by verifying the current state of the radio before trying to transition to a different state. This comes at the cost of increased processing and communication overheads but makes it less likely that the CC1120 will be able to enter and/or remain in an unintended state. Rudimentary locking of the SPI bus is also used as it was found that the Coffee file system driver used on the test platform detailed in Section 4.4 could cause unintended state changes to the CC1120.

ACK handling was incorporated as part of the packet reception routine to provide consistent behaviour across multiple RDCs. In order to provide as fast a turnaround as possible and due to the variable length of packets, the first 13 bytes of each packet are read and an ACK sent, if required, before the rest of the packet is read. 802.15.4 requires that the transmission of ACKs begins within 12 symbol periods after the completion of packet reception. This translates to a period of 240µs when operating at 50ksps (for 50kbps at 868 MHz) and 120µs at 100ksps (for 100kbps and 200 kbps at 868 MHz). Compliance with this requirement is not achievable where the CC1120 wakes up the microcontroller after successful packet reception and the interrupt routine transfers immediately from pre-emptive to co-operative execution due to a delay in the transition of approximately 750 µs. This could be lessened significantly by handling ACK transmission within the pre-emptive ISR.

The CC1120 will be operated at 50 kbps rather than the maximum of 200 kbps to increase the attainable range and reliability of communications. A full code listing for the CC1120 driver and modified platform files are included in Appendix 3. Since this work was carried out a driver for the similar CC1200 has been released and included in the latest version of Contiki.

4.2.1. Range

Initial range tests were carried out for both the CC1120 and a common 2.4 GHz radio in a low-RF noise environment with the help of another member of the team. These tests consisted of establishing the maximum range at which reliable 6LoWPAN communications are possible using a combination of testing the latency and packet loss with Ping6 and loading node-hosted web pages from a node running the example websense application included with Contiki, modified to remove a dependence on Google's graph service. These initial tests found that the maximum range of the 2.4 GHz radio was 300 m at maximum output power with 16 dBi panel antennas. The CC1120 was found to have a range greater than 3.5 km with 3 dBi gain omnidirectional antennas.

4.2.2. ContikiMAC timings

ContikiMAC was originally developed for 2.4 GHz radios operating at 250kbps so the timings needed to be re-calculated for use with the CC1120. By default, not all settings that influence ContikiMAC's timings are accessible so some modifications to ContikiMAC were required. Table 3 lists the modified timing parameters while Table 4 details the internal parameters that were modified. As can be seen, the majority of timing parameters for the CC1120 are not simply five times the values used for the CC2420 as might be expected by operating at one fifth of the bitrate. Some key settings, such as t_r and T_{awake} are approximately five times the CC2420 value despite the parameters that they rely on not being so. The inter-packet-interval, t_i, is set to zero so that this timing can be handled within the radio driver as it was noted that the implementation of inter-packet timing within ContikiMAC could be inconsistent with regard to the handling of acknowledgements. The value of t_r observed during testing is greater than the value calculated in Table 3 due to timing inaccuracies in Contiki. These timings have been used by Dr George Oikonomou at the University of Bristol with the CC1310 SoC, which includes an 868 MHz radio, and found to give a stable operation (Oikonomou 2015).

4.2.3. EWoR/RX Sniff

As described in section 4.1, the CC1120 includes an on-board method of asynchronous duty cycling. This method puts the radio into sleep rather than idle and uses extended length preambles rather than retransmissions so it is possible that RX Sniff will be more energy efficient than ContikiMAC for radios with a large disparity in TX and RX energy consumption. Initial calculations using the Texas Instruments calculator (Texas Instruments 2013b) show that RX Sniff cannot be implemented at 802.15.4 data rates above 20 kbps with a standard length preamble as the time taken to send the preamble is less than the time taken for the radio to settle from its transition to RX

from sleep. Greater data rates can be achieved by increasing the length of the preamble, however this is at the cost of increased transmit energy utilisation and the loss of the ability to communicate with standard 802.15.4 networks.

Table 3: ContikiMAC timing constraints for the CC2420 at 250kbps operation and the CC1120 at 50kbps operation. CC2420 values reproduced from Dunkels (2011). t_r for the CC1120 has been calculated as per the CC1120 user guide (Texas Instruments 2013c).

Parameter	**Description**	**CC2420 Time** (ms)	**CC1120 Time** (ms)
t_a	ACK turnaround time.	0.192	0.7
t_d	ACK detection time.	0.16	0.8
t_i	Inter-packet interval.	0.4	4.5
t_c	Successive CCA interval.	0.5	5.1
t_r	Time for a stable RSSI.	0.192	0.462
t_s	Shortest packet TX time.	0.884	6.5
t_l	Longest packet TX time.	4.1	21

Table 4: Internal ContikiMAC settings that required modification for the CC1120

Internal Setting	**Related Constraint**	**Default value**	**CC1120 value**
SHORTEST_PACKET_SIZE	t_s	43 B	36 B
CCA_CHECK_TIME	t_r	0.12ms	0.63ms
CCA_SLEEP_TIME	t_c	0.5ms	4.8ms
INTER_PACKET_INTERVAL	t_i	0.4ms	0ms
LISTEN_TIME_AFTER_PACKET_DETECTED	T_{awake}	12.5ms	50ms

4.3. Theoretical Performance

Having methods of calculating the expected performance is important for evaluating the behaviour of a given implementation and for being able to compare a potential communication technology with one or more alternatives. This section details the application of existing methods for calculating the expected throughput and energy usage of the CC1120 operating at a bit rate of 50 kbps and presents an improved method of estimating the round-trip latency.

4.3.1. Throughput

Throughput is an important metric in establishing the performance of a network as it affects how quickly data can be transferred between nodes. A simple method of estimating the potential MAC-

layer throughput for an 868 MHz connection with a link-layer throughput of 50 kbps connection is to assume that the MAC-layer throughput is one fifth of what is achievable with a 2.4 GHz connection with a link-layer throughput of 250 kbps. The attainable throughput of a the CC2420, a common 2.4 GHz 250 kbps radio, has been reported as being 100.21 kbps (Jain, Reena et al. 2013) with 130 kbps being achievable with more optimal forwarding (Österlind and Dunkels 2008). This gives an estimate of 20.04 kbps to 26 kbps of attainable throughout for the CC1120 operating at 50 kbps.

A more accurate estimate of the throughput can be established with the method used by Jain, Reena et al. (2013) for establishing the throughput of non-fragmented packets for a 250 kbps 2.4 GHz 802.15.4 radio in a lightly loaded network. Using a $P_{inactive}$ of one for the CC1120 provides a means to establish the potential peak throughput, which is more useful for comparative purposes. Using the values from Table 5 results in a theoretical throughput of 24.45 kbps for a payload of 81 bytes, the maximum payload size achievable in Contiki before it fragments the packet. This value will be greater than what is actually attainable as the method used does not take into account the overhead involved in transferring data to and from the radio.

Table 5: Packet transmission time for the CC1120 operating at 50 kbps with $P_{inactive}$ = 1.

Function	Symbols	Time (s)
CSMA-CA	250	0.005
TX Packet	1064	0.02128
ACK Turnaround	35	0.0007
TX ACK	96	0.00192
LIFS	40	0.0008
SUM	1485	0.0297

The accuracy of this estimate can be refined by including hardware-specific overheads into the calculation. The hardware specific overhead can be calculated with Equation (3). This equation is a combination of the relevant parts of Equations (5) and (6) with all variables having the definitions stated in section 4.3.2. Of the 133 bytes that can be transmitted as a maximum size packet, only 125 bytes need to be written to and read from the radio as the CC1120 generates the preamble, sync and CRC bytes automatically. The platform specific overhead was calculated to be 1.087ms using the values from section 4.3.2. Taking this into account the revised theoretical throughput becomes 23.49 kbps. This is within the range established by assuming that the throughput will be one fifth of the achievable throughput for a 250 kbps connection.

$$t_{PlatOH} = \left(n_{bytes} \times (t_{SBTX} + t_{SBRX})\right) + t_{RLR} + t_{INT} + t_{SATX} + t_{SARX} + t_{TO} \quad (3)$$

4.3.2. Latency

Another important metric in analysing the behaviour of a network is the latency of the connection as it can influence the setting of timeouts and the performance of higher-level protocols. The latency in an 802.15.4 network is not only dependent on the data rate of the radio but also on how many packets the data is being fragmented into due to the number of bytes of per-packet overhead. Existing methods of establishing the performance of a low-power radio network theoretically have focused on throughput. This section proposes a method of estimating the latency when explicit knowledge of the platform is known. This approach ensures genericity across different platform and radio combinations as it takes into account platform-specific factors that influence the latency. Knowledge of these factors can be inferred from data sheets or measurement.

The round-trip single-hop latency for directed packets requiring an acknowledgement can be modelled using known data about the platform and radio using Equation (4). n_{BT} and n_{BR} are the number of bytes of payload to be transmitted and received, respectively, while n_{PT} and n_{PR} are the number of packets that the data will be split into to be transmitted and received, respectively.

$$t_{Latency} = \left((n_{BT} + n_{BR}) \times t_{BS}\right) + \left((n_{PT} + n_{PR}) \times t_{PO}\right) + (2 \times t_{CA}) + t_{RDCOH} \quad (4)$$

t_{BS} is the amount of time taken to load, transmit, and read each byte and is related to the radio bitrate and the speed of the interface from the radio to the microcontroller. t_{BS} can be calculated with Equation (5) where t_{SBTX} is the time taken to load each byte into the radio, t_{SBRX} is the time taken to read each byte from the radio and R_{BR} is the transmit bitrate of the radio.

$$t_{BS} = t_{SBTX} + t_{SBRX} + \frac{8}{R_{BR}} \quad (5)$$

$$t_{PO} = t_{SATX} + t_{SARX} + t_{TO} + (n_{HB} \times t_{BS}) + \left(n_{OB} \times \frac{8}{R_{Bitrate}}\right) + t_{ACK} + t_{RLR} + t_{INT} \quad (6)$$

t_{PO} is the constant overhead associated with the transmission of each packet. This can be calculated with Equation (6) where t_{SATX} and t_{SARX} are the time taken to access the SPI bus for burst write (TX) and read (RX) operations respectively. t_{TO} is the time taken to get the radio into TX from whatever state the radio uses as its "off" state. n_{HB} is the number of bytes of per-packet overhead that have to be loaded to and read from the radio. This includes header and length bytes and may include CRC bytes depending on the implementation. n_{OB} is the number of bytes of per-packet overhead that have to be transmitted and includes sync and preamble bytes. This may include CRC bytes but is dependent on the implementation. t_{ACK} is the time taken to process, load and transmit a MAC-layer ACK and t_{RLR} is the time taken to read the LQI and RSSI data from the radio. Finally, t_{INT} is the

time taken for the microcontroller to enter the receive interrupt vector and transition to co-operative space if applicable.

t_{CA} is the amount of time taken to access the channel. This is related to the channel access mechanism and is implementation dependent. For methods that have a variable channel access time, e.g. ETSI EN 300 220-1 (ETSI 2009) listen before talk (LBT), the minimum specified time can be used to establish the minimum expected latency.

t_{RDCOH} is the overhead associated with the RDC being used. For NullRDC, this is zero but for ContikiMAC it can be significant due to the reliance on retransmissions of the packet, or first packet in a fragmented transmission, to ensure reliable reception. t_{RDCOH} for ContikiMAC can be calculated with Equation (7) where n_{RTX} and n_{RRX} are the average numbers of retransmissions for transmitted and received packets respectively and t_i is the inter-packet interval with the same definition as in 4.2.2. n_{BT1} and n_{BR1} are the number of bytes that need to be retransmitted for transmission and reception respectively. For non-fragmented packets, where n_{PT} and n_{PR} are both equal to one, n_{BT1} and n_{BR1} will be the entire payload whereas for fragmented transmissions they are the number of bytes in the first packet. For direct communications between a border router and sensor node n_{RRX} may be zero as the border router may not be duty cycling the radio to improve performance. This is because the border router is generally less power constrained than a sensor node and this is the default behaviour for a Contiki border router.

$$t_{RDCOH} = (n_{RTX} + n_{RRX}) \times \left(t_{TO} + t_i + \frac{8(n_{OB} + n_{HB})}{R_{BR}} \right) + \frac{8\big((n_{BT1} \times n_{RTX}) + (n_{BR1} \times n_{RRX})\big)}{R_{BR}} \quad (7)$$

Multi-hop latency can be estimated for a network where each hop uses the same type of sensor node by multiplying the result of Equation (4) by the number of hops. For multi-hop calculations involving ContikiMAC, the latency of the first hop will need to be calculated separately from the latency of the subsequent hops as n_{RRX} is zero for the first hop only.

Table 6 lists the parameters and calculated latency for the CC1120 connected to a Zolertia Z1sensor node with the NullRDC and ContikiMAC duty cycling protocols. Additionally, an estimate is made for the two-hop average ContikiMAC latency. These values will be less than the real-world measurable latency due the calculations not taking into account the time that the far node will take to process the packet and prepare a reply. This processing time is not easily predictable or consistent as it depends on the method being used to measure the latency, the overhead of the ap-

plication processing the packet, the overhead of any operating system and on whether other processes are running on the microcontroller. The discrepancy between the calculated and measured values will increase with the number of hops as each additional hop adds processing time. Despite this limitation, calculating the theoretical minimum latency still has value because it allows direct comparisons to be drawn between different radios and duty cycling protocols as they will all be subjected to the same processing overheads in operation. Additionally, the latency of packets that traverse the Border Router will be greater than indicated by these equations due to the interface between the low-power network and the standard IP network. This work is validated experimentally in Section 4.4.2.

Table 6: Latency parameters for the CC1120 with NullRDC and ContikiMAC.

Parameter	**NullRDC**	**ContikiMAC** (Average Latency)	**ContikiMAC** (Minimum Latency)	**ContikiMAC** (two-hop)
$\mathbf{n_{BT}}$	94			
$\mathbf{n_{BR}}$	93			
$\mathbf{n_{PT}}$	2			
$\mathbf{n_{PR}}$	2			
$\mathbf{t_{BS}}$ (ms)	0.172			
t_{SBTX} (ms)	0.006			
t_{SBRX} (ms)	0.007			
R_{BR}	50000			
$\mathbf{t_{PO}}$ (ms)	7.765			
t_{SATX} (ms)	0.016			
t_{SARX} (ms)	0.022			
t_{TO} (ms)	0.012			
n_{HB}	22			
n_{OB}	8			
t_{ACK} (ms)	2.6			
t_{RLR} (ms)	0.044			
t_{INT} (ms)	0.001			
$\mathbf{t_{CA}}$ (ms)	0	34		
$\mathbf{t_{RDCOH}}$ (ms)	0	87.48	0	303.04
n_{RTX}	-	4.70	0	4.70
n_{RRX}	-	0	0	4.7
t_i (ms)	-	4.84		
n_{BT1}	-	56		
n_{RT1}	-	55		
$\mathbf{t_{Latency}}$ (ms)	**63.27**	**218.75**	**131.27**	**565.58**

4.3.3. Energy

Energy is another important factor in assessing performance. This section estimates the power that will be used by the CC1120 and compares it to the CC2420. This is not practically validated in this chapter.

The average receive energy utilisation can be estimated using the amount of time that the radio is expected to be in RX and the typical receive current from the relevant data sheet. For the default ContikiMAC settings of an 8 Hz channel check rate and two CCA checks per cycle, the radio will be turned on 16 times each second. This gives a total on-time of 10.08 ms in every second for the CC1120 giving a duty cycle of approximately 1%. The high-performance mode draws 22 mA in receive so with a 3.3 V supply the average RX energy utilisation will be 0.73 mW, or 43.8 mJ per hour. In the low-power receive mode, which draws 17 mA, the energy utilisation would be 0.56 mW, or 33.6 mJ per hour, at the cost of receive sensitivity. Comparatively, the CC2420 should have an average energy consumption of 0.12 mW. While these values are not necessarily fully representative of the general case where other specific radios are involved, the increase in energy consumption is broadly in line with expectation, with the CC1120 having an energy consumption of between 4.6x and 6.1x that of the CC2420.

The maximum retransmission energy overhead can be calculated in a similar manner by multiplying the amount of time it takes to transmit a full-sized packet by the TX power and the average number of times that the packet is transmitted beyond the first. With the CC1120, a full-sized packet will take 21.60 ms to transmit at a total energy cost of 3.21 mJ at 14dBM. This gives an average retransmission overhead of 11.87 mJ per packet. In comparison, the CC2420 takes 4.32 ms to transmit the same packet giving a transmit energy of 0.25 mJ per packet. As hypothesised, the transmission overhead is significantly greater for the CC1120 due to its increased transmit power and time spent transmitting.

The energy consumption when using RX Sniff is largely dependent on the length of preamble used and the data rate. Table 7 shows the relationship between preamble length and average energy consumption when operating at 50 kbps. The table also shows the average number of messages that a node would need to send per hour for RX Sniff to be more energy efficient than ContikiMAC. These figures take into account the increased energy required to send the extended preamble and the energy parameters where Smart Preamble is used with RX Sniff. Thus showing the shorter preambles require the transmission of a significant number of full-sized non-fragmented packets per hour. Actual deployment requirements will be greater as a number of packets sent by a node will be under 100 bytes in length. RX Sniff with Smart Preamble is more energy efficient with regard to

average receive energy consumption than ContikiMAC but the overhead incurred in transmitting a packet with such a long preamble means that RX Sniff becomes less energy efficient if more than one packet is transmitted per hour. It should be noted, however, that these calculations do not take into account the additional processing overhead associated with preparing and managing the overly long preamble used with Smart Preamble so actual savings are likely to be less.

Table 7: Modelled energy consumption when using different lengths of preamble with RX Sniff with the CC1120 at 50 kbps. The difference between the average energy consumption of ContikiMAC and the average number of full-sized packets that need to be transmitted per hour for RX Sniff to be more energy efficient is also presented. The final row models the energy of RX Sniff with Smart Preamble. Values calculated with (Texas Instruments 2013b)

Number of Preamble Bytes	**Average Power @ 3.3V** (mW)	**Average Consumption /hour** (mJ)	**Energy Δ from ContikiMAC /hour** (mJ)	**Preamble TX Overhead** (mJ)	**Energy Δ from ContikiMAC /packet** (mJ)	**Number of packets required per hour**
7	42.04	2522.4	2478.6	0.07	-11.80	211
8	36.51	2190.6	2146.8	0.09	-11.78	183
12	23.92	1435.2	1391.4	0.19	-11.68	120
24	11.56	693.6	649.8	0.48	-11.39	58
30	9.25	555	511.2	0.61	-11.26	46
1792	*0.16*	*9.6*	*-34.2*	*42.48*	*30.61*	*1*

4.4. Experimental Validation

The theoretical values obtained in section 4.3 needs to be validated practically to verify the accuracy of the methods when used for environmental sensor networks. The hardware used for these tests is shown in Figure 11 and consists of a Zolertia Z1, Texas Instruments CC1120 evaluation module and a bespoke interface board. The interface board was designed in conjunction with another member of the team and details are included in Appendix 4. A lab-based test network comprising of a border router and multiple nodes was created to carry out the tests detailed in the rest of this section to assess the actual throughput and latency achievable. The border router was connected to an Instant Contiki virtual machine using the USB Serial interface of the Z1. Global IPv6 addresses were not used with all tests being carried out from the virtual machine.

Figure 11: Test hardware consisting of a CC1120EM (left), Zolertia Z1 (right) and a bespoke interface board.

4.4.1. Throughput

The sustained single-hop MAC-layer mono-directional throughput between two nodes can be established by using modified UDP data transmission examples present in Contiki, two test nodes and a high-sample rate oscilloscope in a lab-based setting. The IPv6 UDP client and server examples transmit a string from the client to the server at a set rate with the server replying to every transmission received with an acknowledgement.

The client application was modified to remove debug prints to improve performance and to have a compile-time configurable send rate. A transmission string that resulted in a non-fragmented MAC payload of 81 bytes was chosen for the client as this is the largest payload that can be transmitted by a default Contiki build before fragmentation occurs. The server was modified so that it did not reply on packet reception and to alert on the console when missing packets were detected.

To determine the maximum attainable throughput, the send rate of the client was successively increased until the server indicated that it was not consistently receiving packets. Due to inaccuracies in timings under Contiki, the set send rate does not reflect the actual send rate. To obtain an accurate count of the number of packets transmitted per second, an oscilloscope was used to visually count the packets in a set period of time. NullRDC was used for these measurements as ContikiMAC has a significant impact on throughput and the aim of this experiment is to ascertain the throughput that is ascertainable at the link-layer. The result is not affected by the throughput or latency of the Internet connection as all traffic remains within the low-power network. As such, this experiment gives a good representation of the attainable throughput. The modified code for the UDP client and server examples is included in Appendix 5.

Figure 12: Example oscilloscope trace of the radio transmissions between the UDP client and server. 10 packets were transmitted in 270 ms. The yellow trace represents RF activity and the blue trace indicates when the server was transmitting and acknowledgement.

Figure 12 shows that 10 packets were transmitted in 270 ms including the tailing inter-packet interval. The Oscilloscope was set to trigger on a rising edge and the trace observed for stability before calculating the throughput and repeated tests gave the same result. This gives a sustained throughput of 23.44 kbps without a channel access mechanism. In the best-case, the ETSI EN 300 220-1 LBT mechanism would take an additional 5ms per packet resulting in a throughput of 19.78 kbps. This is 3.71 kbps below the theoretical value calculated in 4.3 but does fall close to the limits of the crude estimation. The inter-packet interval achieved during testing was 5.8 ms, 5 ms greater than the LIFS period of 0.8 ms. Taking this difference into account and re-running the calculation in 0 results in a predicted value of 19.88 kbps which is similar to the achieved throughput. The increased inter-packet interval is likely due to processing and other process overheads within Contiki.

For comparison, a similar test was carried out on the CC2420 present on the Zolertia Z1. Due to an inability to directly visualise the 2.4 GHz transmissions a GPIO pin on the radio, the SFD pin, that indicated when it was in the transmit state was monitored instead. Figure 13 shows that 10 packets were transmitted in a period of 72.5 ms giving a throughput of 87.28 kbps, between 12.93 kbps and 42.72 kbps below what is expected from the literature. As can be seen from the trace, there is a larger inter-packet gap of 6ms between every other packet that is transmitted. This reinforces the conclusion that process overheads and inefficiencies within Contiki are reducing the attainable throughput as shorter inter-packet intervals are shown to be possible.

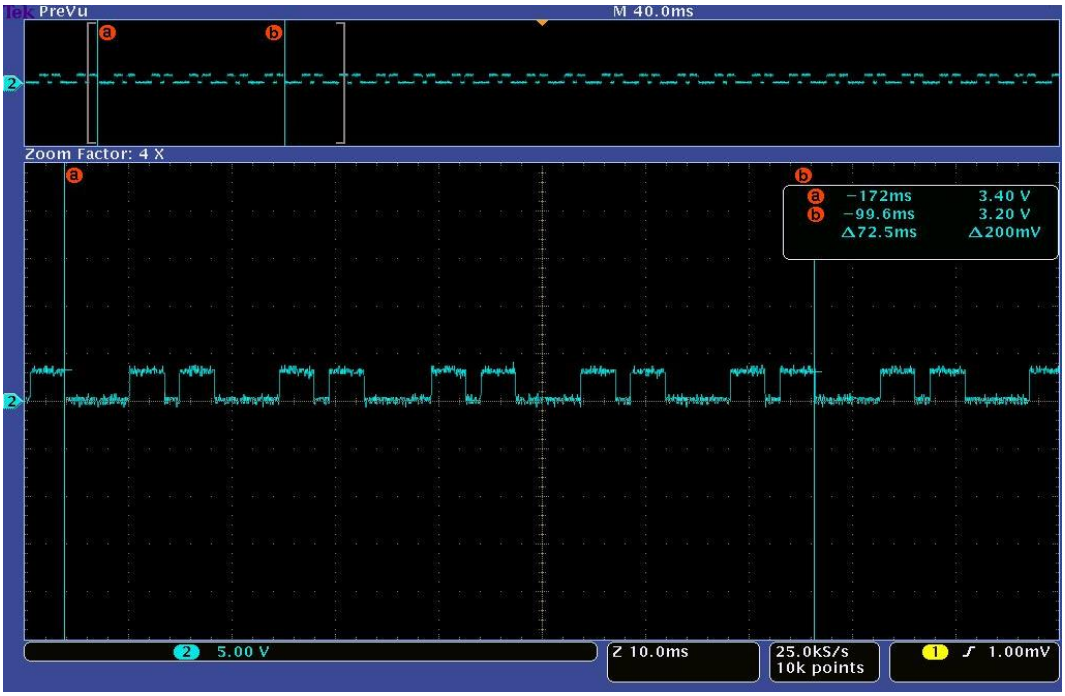


Figure 13: Example oscilloscope trace of CC2420 SFD pin during throughput testing at 2.4 GHz. 10 packets were transmitted in 72.5 ms giving a throughput of 87.28 kbps

4.4.2. Latency

The round-trip latency can be measured by carrying out a long-duration repeating Ping6 from the Instant Contiki VM. Any latency testing from outside the low-power network will include the latency of the interface between the low-power network and the standard network. For the test platform, this is a 6-Slip interface over a USB-to-serial connection. The latency of this interface was determined to be 21ms for the test setup by pinging the node acting as the border-router from the host system. This latency will be different for other test setups and interfaces.

Table 8 details the results from running a Ping6 every second for 43200 requests for a total test time of 12 hours. A 12 hour experimental period was chosen to collect enough data to account for variation in RF noise without being excessively long. A significant standard deviation is expected for tests involving ContikiMAC due to the variable number of retransmissions required. The results demonstrate that the measured latency in all cases is greater than the expected latency by a comparable amount. As the discrepancy is systemic and not proportional to the value of the latency it is likely that it is due to processing overheads in handling within the microcontroller. Comparative measurements for the CC2420 are included and demonstrate that a significant proportion of the latency exhibited on 6LoWPAN networks is not related to the throughput of the radio as the CC1120 results are less than five times that of the CC2420, even when the 6-Slip latency has been subtracted.

Table 8: Ping6 latency results from 43200 echo requests. The minimum and average latencies shown include the latency of the 6-Slip connection.

Test	Minimum Latency (ms)	Average Latency (ms)	Standard Deviation (ms)		Δ from Calculated Values (excluding 6-Slip latency)	
					Min (ms)	Avg (ms)
868 MHz NullRDC	102.60	105.02	15.51		18.33	-
868 MHz ContikiMAC	169.50	258.33	83.42		17.23	18.58
2.4 GHz NullRDC	41.86	43.00	1.73		-	-
2.4 GHz ContikiMAC	51.04	209.77	198.35		-	-
868 MHz CM 2-Hop	295.98	740.13	324.23		-	-
2.4 GHz CM 2-Hop	57.55	431.67	508.90		-	-

The average two-hop latency was also measured for ContikiMAC at 868 MHz and 2.4 GHz. For 868 MHz, this was found to be 153.55 ms greater than expected and requires further work to determine why.

4.5. Summary

This chapter has shown that 6LoWPAN can be successfully applied to 868 MHz sub-GHz networks in a lab-based setting using a common IoT operating system. A contemporary, commonly used asynchronous duty cycling protocol that was originally designed for 2.4 GHz networks has also successfully been applied to an 868 MHz lab-based network and achieved a comparable duty cycle of about 1%. Compared to its use at 2.4 GHz, when used at 868 MHz ContikiMAC has a comparable impact on the average latency and a more significant impact on the minimum observed latency.

Attainable MAC-layer throughput was in line with expectations but was less than the peak throughput that should be achievable due to limitations within the operating system. These overheads also reduce the attainable throughput on 2.4 GHz networks so are not specific to a sub-GHz network and show that the existing methods of throughput estimation are limited and require improvement.

A method of estimating the latency has also been presented. This method produces results that have a consistent variation from the values obtained experimentally so further work is needed to validate the accuracy of the method and its applicability to other frequency bands.

Chapter 5

Mountain Sensing

Mountain Sensing was a NERC funded proof of concept research project (NERC 2014) that investigated the feasibility of applying standards-based IoT technologies to ESNs monitoring periglacial and peatland processes. This chapter details work that was carried out as part of this project to investigate the second and third research questions posed in Chapter 1, which are restated here for convenience, through a series of real-world deployments:

2. Can a sub-GHz standards-based 6LoWPAN network perform sufficiently, in terms of energy, for real-world IoT environmental sensor network deployments?
3. Can a sub-GHz standards-based 6LoWPAN network perform sufficiently, in terms of throughput, latency and reliability, for real-world IoT environmental sensor network deployments?

The Mountain Sensing project moves away from the high-energy long-range Wi-Fi links and high-energy GSM connections used in other ESN deployments, such as the Glacsweb deployment discussed in Chapter 3 and deployments discussed in Chapter 2, and attempts to address the challenges of using a multi-hop long-range 6LoWPAN low-power radio network for deployments in harsh conditions with a sparse node density. Multiple deployments of sensor nodes were carried out at the Glen Feshie estate in the Highlands of Scotland during the course of this project.

The study site comprises of four areas of interest spread across an area of approximately four kilometres by one kilometre in mountainous terrain as shown in Figure 14. Table 9 details the type of terrain of each study area and the specific geographical features of interest in each area. The site has intermittent weak GPRS coverage in areas one and two but there is no coverage at areas three and four. Satellite-based Internet connectivity and permanent power are available at an estate office three kilometres north-west of area one as shown in Figure 14. Only areas one and two have partial line of sight to the estate office and there is no line of sight between areas two and three. The scale of the study site, lack of GPRS coverage, restricted line of sight and distance to permanent power and reliable internet connectivity make a multi-hop low-power radio network a viable option for providing long-term connectivity to sensor nodes in the intended study areas.

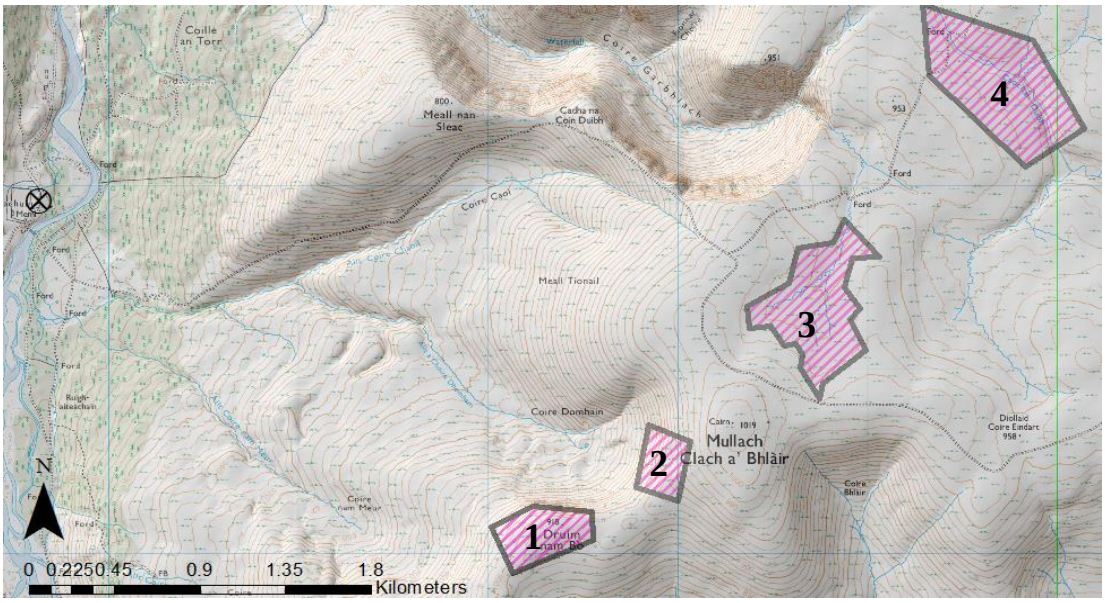


Figure 14: Cairngorm site areas of interest are shown by the outlined hatched boxes. Satellite-based Internet connectivity is available at an estate office marked by the ⊗ symbol. DTM data from (Intermap Technologies 2007) and Crown Copyright Ordnance Survey.

Table 9: Terrain classification of areas of interest.

Area	Terrain Type	Ground Cover	Features of Interest
1	Ridge	Upland heath and bare rock	Periglacial lobes
2	Steep hillside with periglacial features and pools	Upland heath/grassland and bare rock	Periglacial lobes, lochan
3	Headwater basin	Eroded peatland	Peat, streams
4	Stream valley	Upland heathland and eroded peatland	Peat, streams

A mix of different sensors are required to properly instrument these different features: Water level sensors were used for measuring the depth of bodies of water, with rain gauges to measure rainfall; chains of movement and temperature sensors were used to detect movement and events of interest in periglacial lobes; and temperature spiders and soil moisture sensors measure temperature profiles and water table depth of peatland. In-depth discussion on their design is beyond the scope of this thesis as the specific sensors were selected and implemented by another member of the team.

This chapter is divided into six subsections: Section 5.1 details pre-deployment RF propagation modelling that was carried out to determine suitable node placement prior to deployment to make deployment easier and to increase the chance of success; Section 5.2 discusses the initial platform that was designed with another member of the project team and discusses an initial failed deployment; Section 5.3 details software and protocol changes that were made to improve reliability and a discusses the second deployment; Refinements to the hardware design and a third deployment are discussed in section 5.4; Section 5.5 evaluates and compares the relative performance of the system, including a comparison of two different methods of data transfer, HTTP and CoAP; and Sec-

tion 5.6 discusses how this work has addressed the research questions posed in Chapter 1 and 5.7 summarises the chapter. The remainder of this section introduces the Mountain Sensing project. Work detailed in this chapter has been published in Bragg, Martinez et al. (2016), Bragg, Basford et al. (2016), Fabre, Martinez et al. (2016) and (Martinez, Hart et al. 2016).

5.1. Deployment Modelling

The topology of the site and disparate nature of the areas of interest means that there is no direct line of sight across the entire site. An initial site visit showed that dedicated routing nodes, with larger batteries and higher gain antennas, would be required to provide reliable communications. Coverage modelling gives an indication of the number and location of routing nodes required to provide coverage of the study areas before field work is undertaken, making deployment quicker and increasing the chances of a successful deployment. Basic line-of-sight RF modelling was carried out in ArcMap using viewsheds prior to the deployment detailed in Section 5.2.3. Line-of-sight modelling is less accurate than full propagation modelling as it does not take into account Fresnel diffraction caused by intervening terrain, fading or multipath interference caused by surrounding terrain. Despite these limitations, line-of-sight modelling is appropriate for initial deployment planning as it is easier to perform and requires less input data than full propagation modelling. The accuracy of the modelling will also be affected by the vertical accuracy of the elevation data: in this case the accuracy is one metre, which may have a significant impact for nodes close to the ground.

The parameters used to constrain the model to achieve a representative coverage are shown in Table 10. Each type of node will have different antennas mounted at different heights. It was assumed that: the border router would use a Yagi antenna mounted at a height of three metres; the routing nodes would use high-gain omnidirectional “can” antennas mounted on small tripods at 1.5 metres; and that the sensor nodes would use small “stick” antennas at approximately half a metre above the ground. The ranges determined in Section 4.2.1 were used to constrain the model; however, these ranges were determined with the same type of antenna at the stationary node and the model does not take into account a difference in gains at each end so it was assumed that a high-gain antenna could communicate with a lower-gain antenna beyond the maximum range of the lower-gain antenna. Figure 15 shows the modelled line of sight from a border router at the estate office and the proposed location of the first routing node. This location was chosen as it is between two areas of interest and has good coverage to the ridge between area two and area three as shown by the coverage plot. The second routing node was placed on this ridge however the gentle gradient of the ridge means that it does not provide any coverage to the third are of interest as shown in Figure 16. A third routing node was placed on a local summit to the north-east of the second routing node and

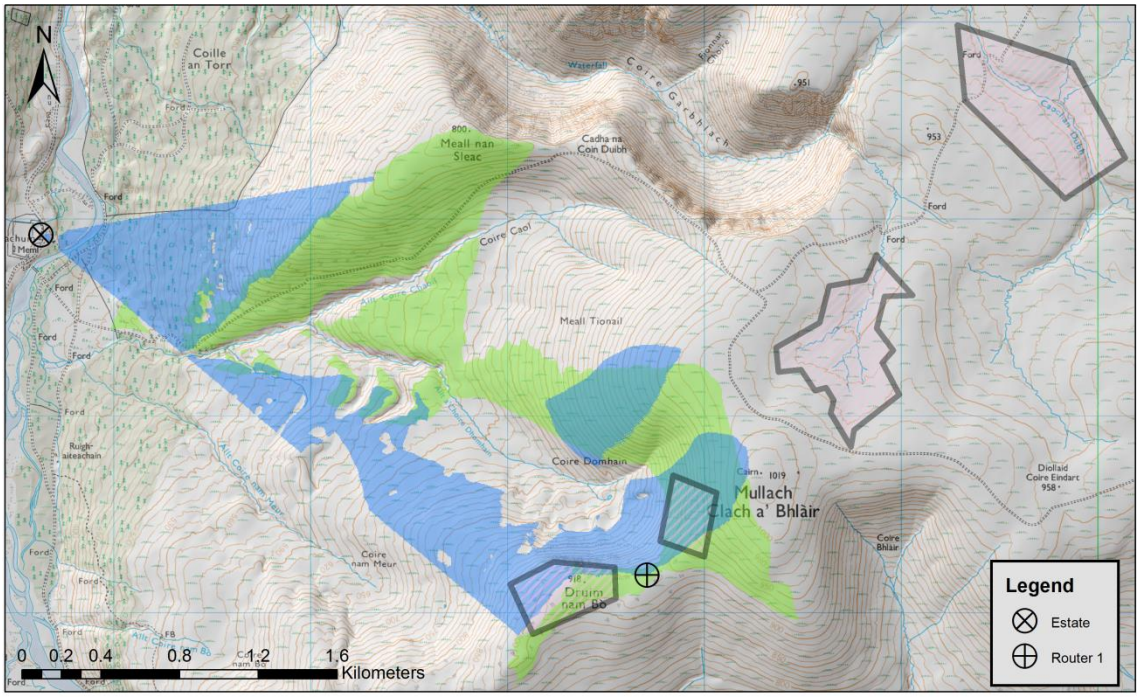


Figure 15: Predicted line of sight from the border router to nodes with a H_{Ant} of 1.5 m, the proposed location and coverage of the first routing node. The border router's coverage is shown in blue and the first node's coverage is shown in green. Areas of interest are shown by the hatched boxes. DTM data from (Intermap Technologies 2007) and Crown Copyright Ordnance Survey.

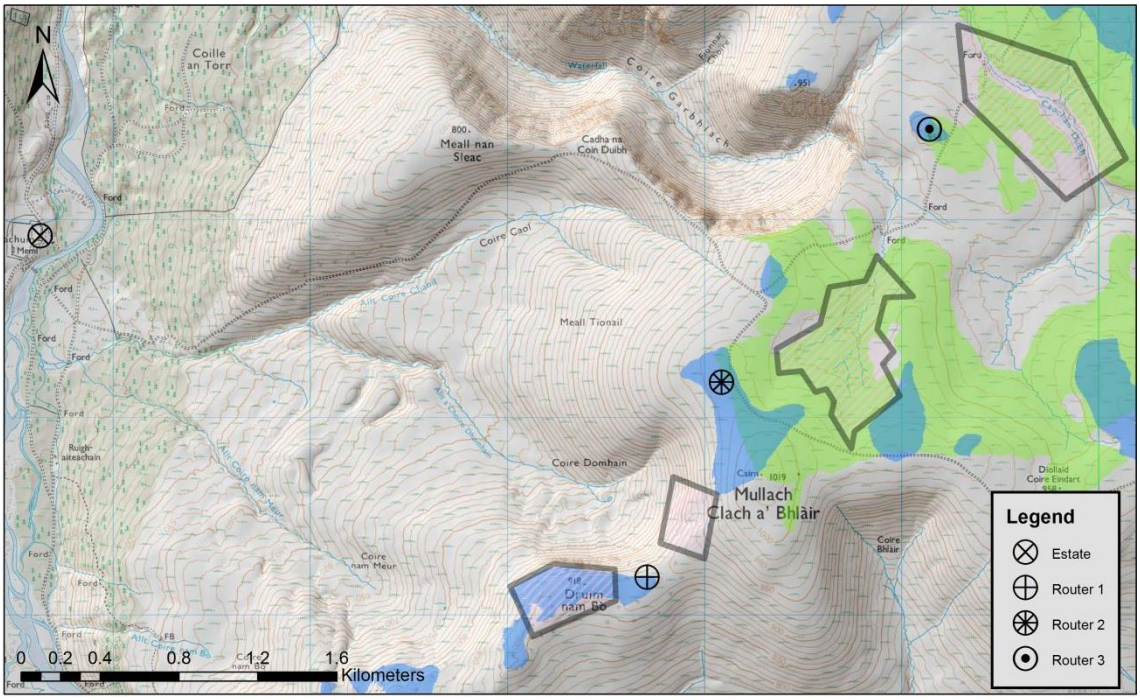


Figure 16: Proposed location of the second and third routing nodes with predicted coverage to nodes with a H_{Ant} of 1.5 m. Router 2's coverage is shown in blue while Router 3's is shown in green. Areas of interest are shown by the outlined hatched boxes. DTM data from (Intermap Technologies 2007) and Crown Copyright Ordnance Survey.

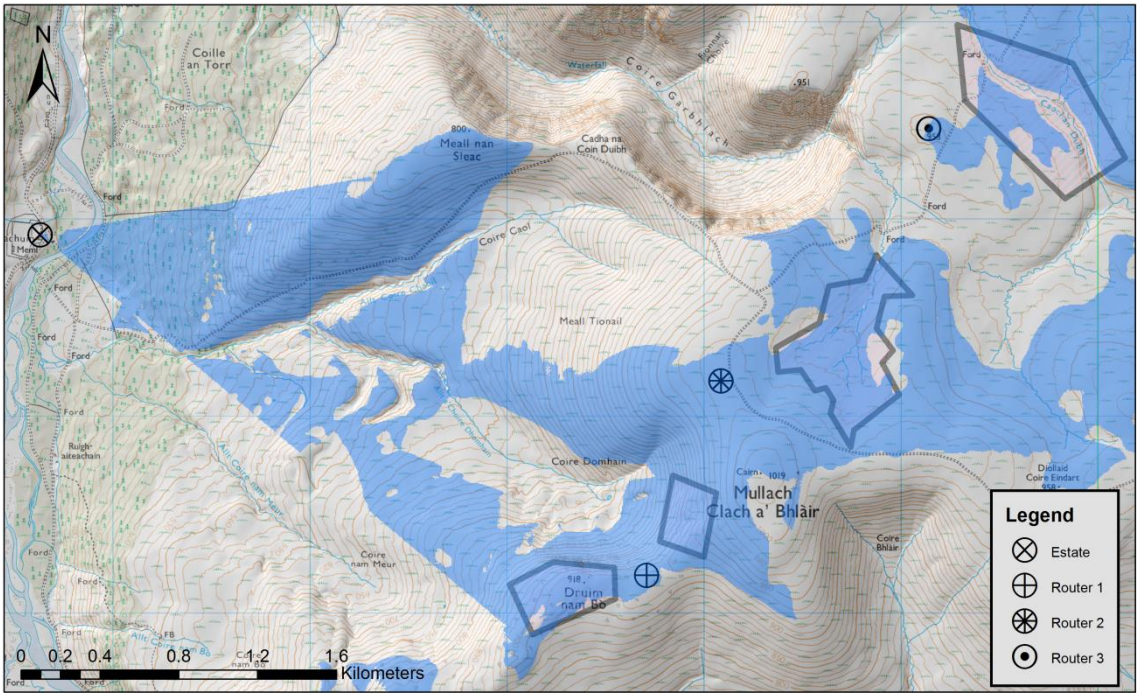


Figure 17: Final proposed deployment of three routing nodes. Coverage is plotted for sensor nodes that have a H_{Ant} of 0.5 m. Near complete coverage of the four areas of interest is provided. Areas of interest are shown by the outlined hatched boxes. DTM data from (Intermap Technologies 2007) and Crown Copyright Ordnance Survey.

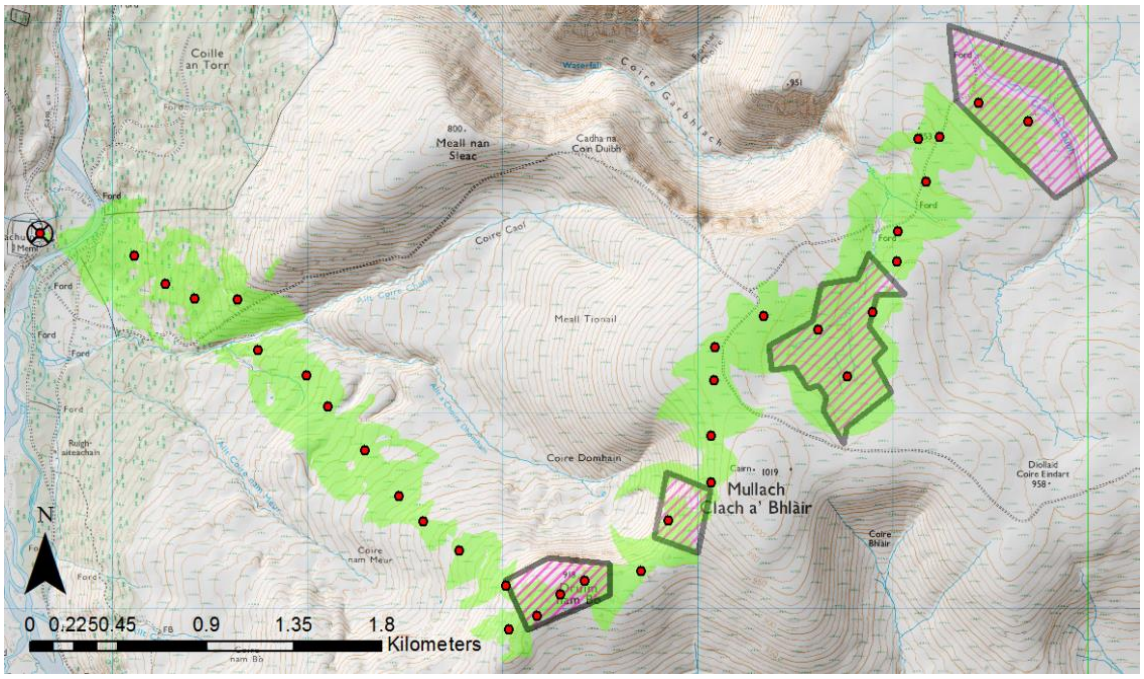


Figure 18. Modelled coverage of a 2.4 GHz-based network. The outlined areas indicate the study areas. Node locations are indicated by solid red dots. Estimated radio coverage is shown by the shaded (green) areas. DTM data from (Intermap Technologies 2007) and Crown Copyright Ordnance Survey.

this provides coverage for the second and third areas of interest. The final proposed deployment of routing nodes and predicted coverage for sensor nodes is shown in Figure 17. Router 1 is 3.5 km from the border router, Router 2 is 1 km from Router 1 and Router 3 is 1.75 km from Router 2. This deployment gives near-complete coverage of the intended study areas with a minimum number of dedicated routing nodes.

Table 10: Parameters used to constrain the line of sight modelling. The maximum ranges are as determined in section 4.2.1

Node Type	**Antenna Height (H_{Ant})** (m)	**Antenna Gain** (dBi)	**Horizontal Field of View** (°)	**Maximum Range** (km)
Border Router	3	9	50	4
Routing Node	1.5	4	360	3
Sensor Node	0.5	3	360	1
2.4 GHz Node	0.5	16	360	0.3

RF modelling can also be used to indicate the number of nodes required to obtain equivalent coverage with a 2.4 GHz network. Such a network was modelled in addition to the 868 MHz network and the predicted coverage is shown in Figure 18. This demonstrates that at least 33 routing nodes would be required to provide less coverage of the study areas with a total of 14 hops to study area one and 32 hops to study area four. Table 11 details the expected throughput and latency performance of both models using throughout and latency data determined in Chapter 4 and demonstrates that an 868 MHz deployment should have at least 2.4 times the throughput at the farthest site and less than one fifth of the latency. The 868 MHz deployment should also have a lower overall energy budget and cost as there are significantly fewer nodes involved than in a 2.4 GHz deployment.

Table 11: Expected MAC-layer throughput and post RDC latency to the furthest research site for both 868 MHz and 2.4 GHz based deployments with the ContikiMAC RDC.

Frequency Band	**Number of hops to Area One**	**Throughput to Area One (kbps)**	**Average Latency to Area One (ms)**	**Number of Hops to Area Four**	**Throughput to Area Four (kbps)**	**Average Latency to Area Four (ms)**
2.4 GHz	14	6.23	3094.47	32	2.73	7088.67
868 MHz	1	19.78	258.33	3	6.59	1221.93

5.2. Initial Platform: Zolertia Z1 + MS1

An initial deployment was carried out at the site in the Cairngorm Mountains to investigate the feasibility of using standardisation in a real-world ESN. This deployment uses node-initiated HTTP for data transfer of data from the sensor nodes to the edge of the network. This section details the design of hardware and software used for the proof of concept deployment and is divided into three sub-sections that describe the implementation used for the border router, the design of the sensor nodes and the deployment that was carried out.

5.2.1. Border Router

The border router consists of a node using the same MSP430-based hardware that was used for network performance evaluation in Chapter 4 connected by USB to a mini ITX computer running Ubuntu 14.04. A modified version of the Contiki RPL-Border-Router example runs on the gateway node with unmodified tunslip6[1] running on the ITX. An IP connected camera is also installed below the antenna to capture photographs of the study site from the estate. The ITX acts as the router and firewall for the deployment network and runs a webserver that accepts and stages data from the sensor nodes before retransmitting it to a server away from the deployment, effectively operating as a proxy server. The software was developed in Python by another member of the team.

The Internet connection available at the site is an intermittent satellite connection that only has support for IPv4. As in the Glacsweb deployment detailed in Chapter 3, the SixXS tunnelbroker was used to provide tunnelled IPv6 to the deployment. Two /64 subnets are used for the deployment: one for the other devices connected by Ethernet to the gateway; and one for the low-power sensor network. The Internet connection is high latency and is intermittent during adverse weather so all sensor readings are cached at the gateway until it is possible to transfer them to the next server. This means that nodes can offload their data whenever the low-power network is available even if the Internet connection is unavailable so that data is not stored on vulnerable storage for longer than necessary. All of the sensor nodes are globally addressable and authorised hosts can interact with them directly.

5.2.2. Sensor Nodes

Sensor nodes consisting of an off-the-shelf sensor mote running Contiki 2.7, a bespoke carrier/interface board and external RS485-connected sensors were designed for the deployment with

[1] tunslip6 is a standard tool in Contiki for creating a network connection over a serial interface

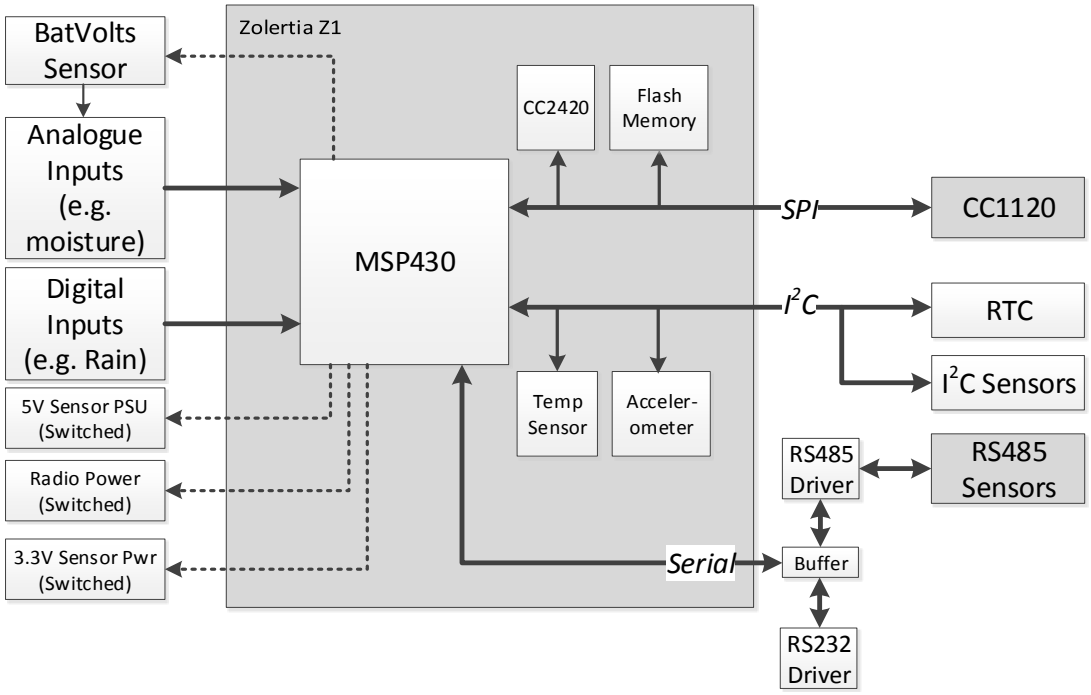


Figure 19: Block diagram of the sensor node. Mono-directional inputs and outputs are shown on the left while bi-directional busses are shown on the right. Dotted lines indicate control lines from the microcontroller. Shaded boxes indicate elements that are external to the carrier board.

another member of the team. A Zolertia Z1 was used as the sensor mote as it is well supported by Contiki and is the platform used in Chapter 4 for the sub-GHz radio driver development. The same radio as used in Chapter 4 was retained and a deployment specific carrier board was created. Deployment specific software to handle configuration, sensing and data transfer was also developed. A block diagram of the functional elements of the sensor nodes is shown in Figure 19 and a photograph of one of the nodes is shown in Figure 20. The remainder of this section details the hardware design of the carrier board and the software design of the node.

5.2.2.1. Hardware

The carrier/interface board, named MS1, includes headers for the Zolertia Z1, headers for the CC1120 EM radio module, a battery-backed-up real-time clock (RTC), power supplies and serial level translators. Due to the variety of factors being measured, the sensor nodes do not include any environmental sensors beyond what is built into the Z1 to allow for one base design to be used for all of the nodes. It was decided that a modular approach with a selection of different options for connecting sensors would be most appropriate. These include analogue and digital single-pin inputs, access to the I^2C bus and a dedicated RS485 bus. The digital and analogue inputs are for simple sensors like tipping bucket rain gauges and conductivity probes for measuring soil moisture.

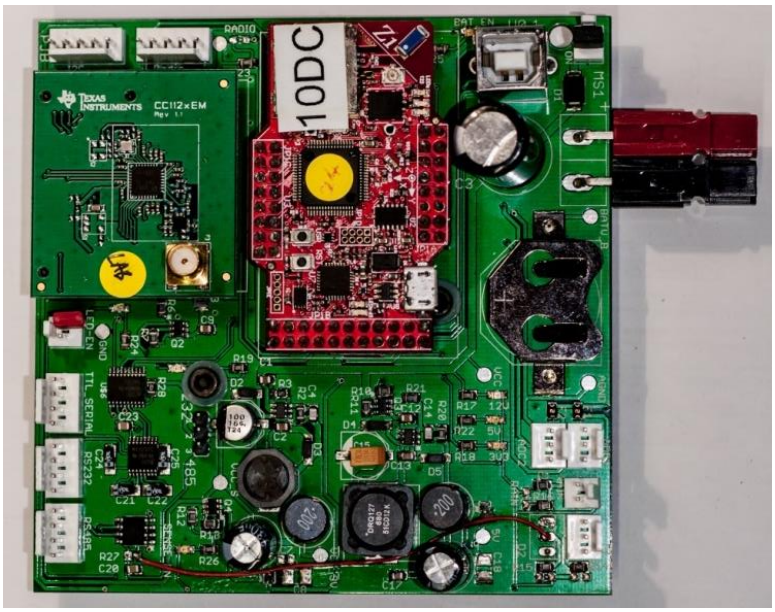


Figure 20: Photograph of the MS1 sensor node design with attached Zolertia Z1 and CC1120 radio board.

The RS485 interface is for more complex sensors, such as pressure sensors for water depth, temperature chains for soil temperature profiles and accelerometer chains for monitoring ground movement. RS485 was chosen as it connects to a standard UART, is resilient to noise due to its balanced transmission line and the interface hardware is readily available. The RS485-connected sensors were developed by another member of the team so their design is beyond the scope of this thesis.

Switch mode power supplies were used as they are more efficient than a low-dropout regulator for stepping down from 12 V to 5 V and 3.3 V. The board was divided into several different power domains with FET switches controlling power to the sensors and the radio as shown in Figure 21 to allow different parts of the board to be completely powered down when not required. In practice, the radio was never powered down and it was instead left in a low power sleep state to allow ContikiMAC to operate correctly. A separate, off-board FET switch was used to switch a 12 V feed for pressure sensors and was triggered from the 5 V sense line.

An independent temperature compensated RTC with battery backup, a DS3231 (Maxim Integrated 2015), was included in the design to allow the node to keep accurate time through varying environmental conditions and unexpected reboots, as occurred in Chapter 3, or if the node lost power.

Circuit diagrams and the PCB layout for MS1 are included in Appendix 6.

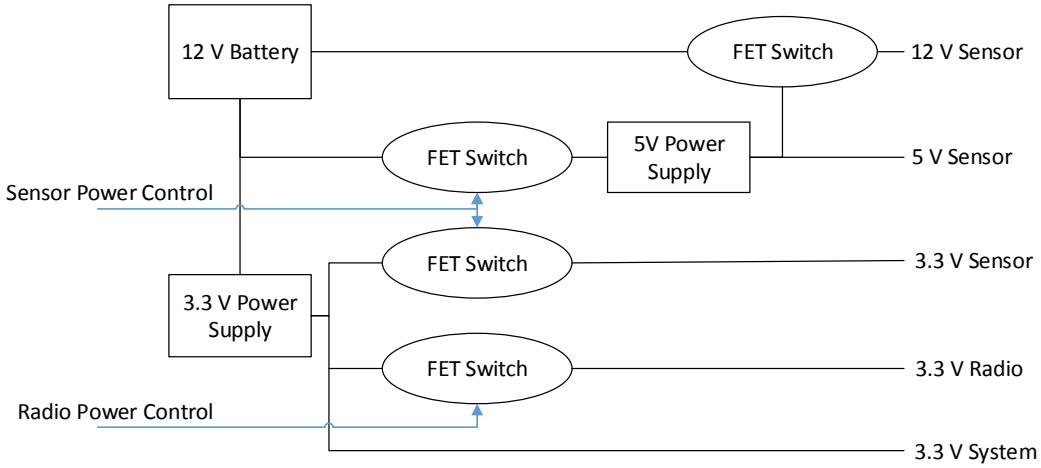


Figure 21: Power domains of the MS1. Blue lines indicate control lines. All three sensor supplies are controlled from a single GPIO from the Zolertia Z1.

5.2.2.2. Software

The nodes run a custom application on the Contiki 2.7 operating system that handles sensing communications and storage management. It is expected that the low-power network will be intermittent at times due to weather conditions and that nodes will run out of power or suffer damage that renders them inoperable. These expectations make relying on a sample and send paradigm or storing sensor readings in volatile memory inappropriate due to the high likelihood of losing data. As such, all readings are stored in local flash memory and are only removed when reception by the destination has been confirmed. Sensor readings are captured on a remotely configurable schedule and packed into protocol buffers (Google 2015) to allow the structured data of a sensing event to be stored and transmitted efficiently. Protocol buffers are an ideal solution for storing numerical sensor data in ESNs and other memory-constrained applications as they have a small size overhead, are fast are simple to implement and are cross-platform. Additionally, protocol buffers are flexible and extensible. Each field can be defined as optional, meaning that nodes do not have to send data, or null data, for factors that they are not measuring. Additional fields can be added in future with existing nodes not needing to be updated. The readings from the RS485-connected "smart" sensors are transferred to the Zolertia Z1 as pre-packed protocol buffers and incorporated directly into the main protocol buffer.

HTTP was used as the application layer protocol for transferring data from the sensor node as it is widely used with good software support and it uses TCP, which is generally more reliable on lossy connections than UDP. Initial testing with a lab-based 2.4 GHz network demonstrated acceptable performance. Communications are a node-initiated HTTP POST that occurs on a different schedule to sensing events to allow multiple sets of readings to be communicated at the same time, to allow

nodes that have been disconnected for a period of time to transfer data as soon as they have a connection and to allow new nodes to be added to the network without having to change the configuration of the gateway. Both the destination address and communications schedule are reconfigurable at runtime to allow for changes to be made while the sensor network is deployed. This is achieved by running a very lightweight webserver on the node that serves configuration webpages as show in Figure 22, which were designed by another member of the team. These page can be accessed from any IPv6-capable device with a web browser that has access to the network allowing the nodes to be reconfigured if the gateway changes or if a different sample interval is required.

Details of the application are included in Appendix 7.

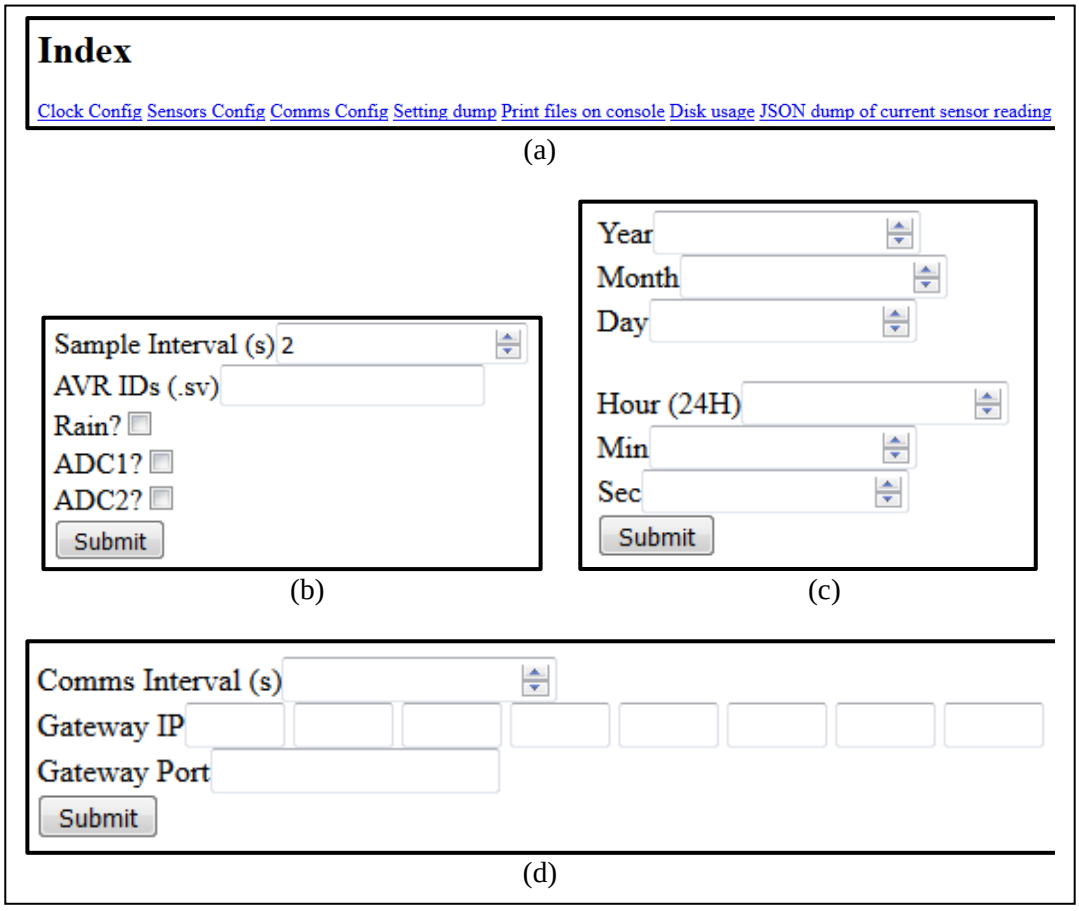


Figure 22: Configuration webpages served by a sensor node. (a) is the Index page that contains links to show the other configuration forms, show a dump of the current settings, trigger the node to print files over serial for debugging, show a print of the flash space used and to dump all of the data stored on the node in JSON format. (b) is the sample configuration page that is used to configure the sample interval and connected sensors. (c) is the clock configuration page that is used to set the current date. (d) is the communications configuration form that is used to set the communications schedule and configure the address that the node sends data to.

5.2.3. Deployment

A small proof-of-concept deployment consisting of five dedicated sensing nodes, two dedicated routing nodes, a combined routing/sensing node and a border router was carried out at the site during the autumn of 2014. Two of each of the different types of sensor node were deployed to increase the chance of retrieving data for other project partners. The name, identifier and connected sensors of each of the deployed nodes are shown in Table 12.

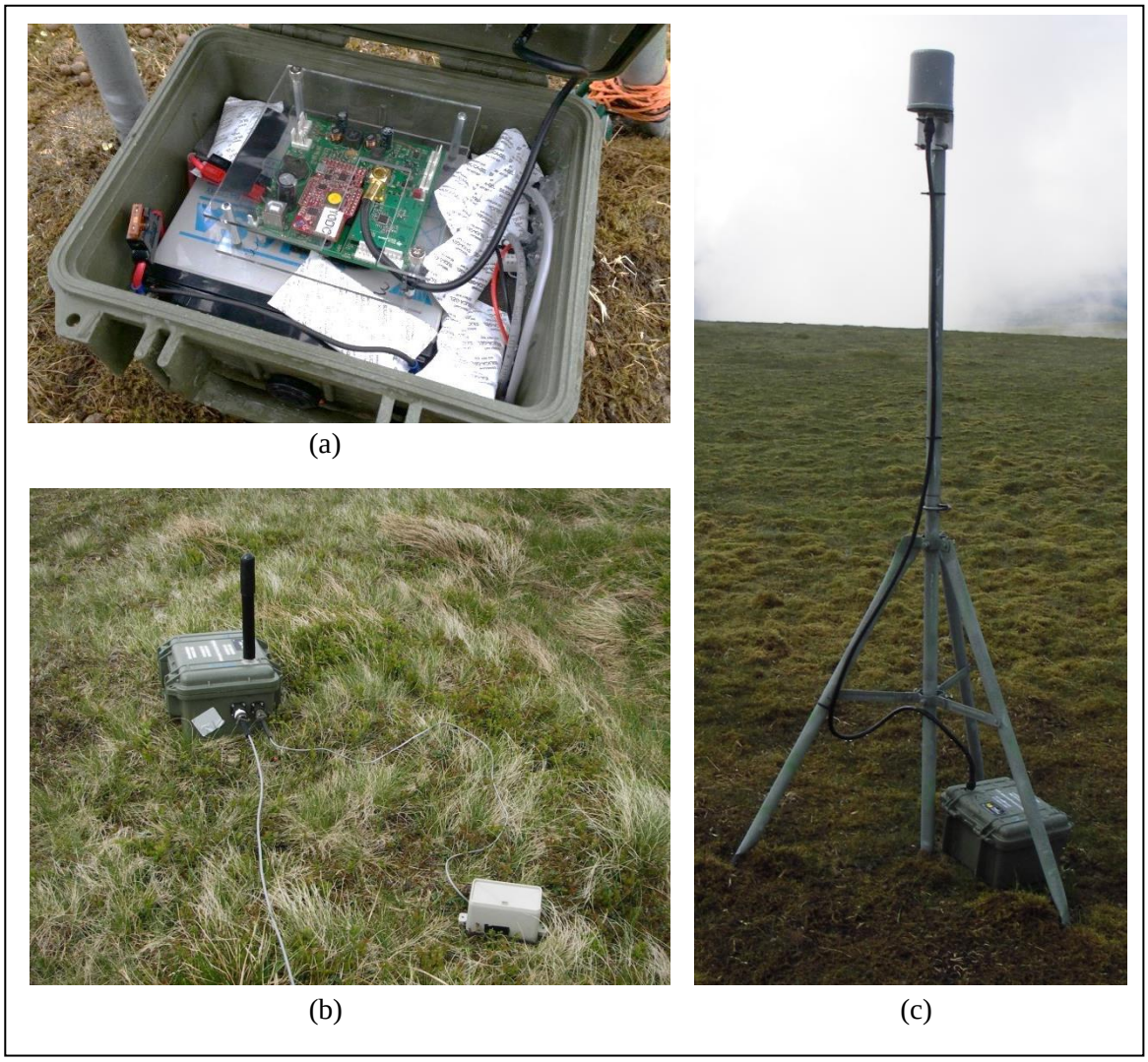


Figure 23: Photographs of the deployed nodes. (a) shows the box that the Z1 + MS1 were deployed in. (b) shows a typical sensor node and (c) shows a typical routing node with its tripod.

All of the nodes were placed in water resistant boxes with lead acid batteries for the deployment as shown in Figure 23(a). Waterproof MIL spec connectors were used for connecting external sensors. The sensor nodes were deployed with the boxes sitting on the ground using a short monopole antenna as shown in Figure 23(b). Routing nodes, including Router 3, were deployed on tripods with

4 dBi gain omnidirectional can antennas as show in Figure 23(c). The routing nodes used higher capacity lead acid batteries to account for the expected increased energy overhead of routing traffic for the network but were otherwise identical in hardware and software to the sensing nodes.

Table 12: List of deployed nodes with their name, function and details of connected sensors. Table modified from Bragg, Martinez et al. (2016)

Node	Function	Identifier	Sensors
Estate	Border Router	0FA1	-
Router 1	Routing Node	107C	-
Router 2	Routing Node	105C	-
Router 3	Routing/Sensing	106A	Temperature spider & soil moisture
Turf	Sensing	1084	Temperature & strain gauge chain
Lochan	Sensing	1068	Water level & rain gauge
Hummock	Sensing	1099	Temperature & strain gauge chain
Peat	Sensing	106B	Temperature spider & soil moisture
Stream	Sensing	1058	Water level & rain gauge

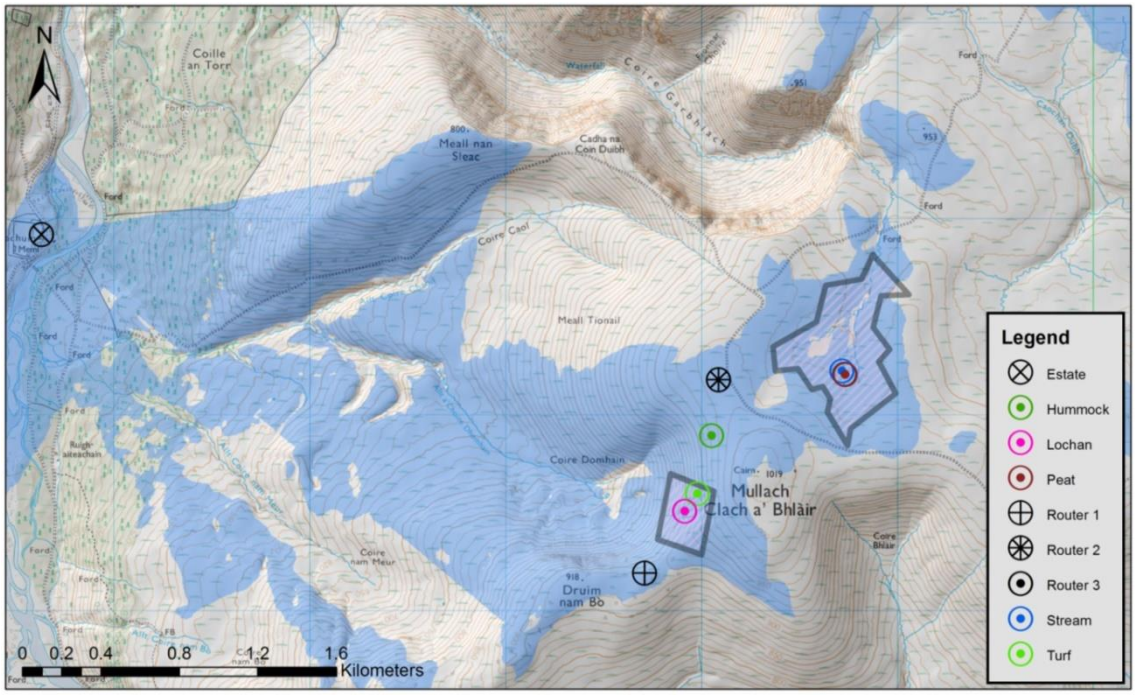


Figure 24: Modelled coverage for the initial deployment in Scotland. Estimated radio coverage is shown by the blue shaded areas. At this scale, Router 3 is co-sited with the Peat sensor node. Original areas of interest are shown by the hatched boxes. DTM data from (Intermap Technologies 2007) and Crown Copyright Ordnance Survey. Figure reproduced from Bragg, Martinez et al. (2016).

Routing node placement was guided by the deployment modelling carried out in Section 5.1; however, due to the limited number of nodes, the deployment was restricted to using two of the four areas of interest. Target features were identified in areas two and three, with an additional feature of interest that was identified between the two. Router 1 and Router 2 were deployed in the positions modelled in section 5.1. During the deployment, it was found that the routing nodes had a greater range than expected due to the low-noise environment. Direct communication between Router 2 and parts of area three were possible so the third routing node was moved into area three and used to provide additional sensor coverage. No nodes were deployed in Area one or Area four. Figure 24 shows the actual positions of the deployed nodes and shows the modelled coverage for this deployment, taking into account the increased achievable range. This additional modelling was carried out to verify that the coverage was appropriate. Table 13 shows the performance for the implemented deployment compared to a 2.4 GHz deployment and shows that the 868 MHz deployment has greater performance despite the reduced area.

Table 13: MAC-layer throughput and post duty cycling latency to the sensor nodes closest to and farthest from the border router for. Data is shown for both the CC1120–based deployment and the CC2420-based deployment modelled in section 4.3. The throughput and latency are calculated using values from sections 4.4.1 and 4.4.2. Table reproduced from Bragg, Martinez et al. (2016).

Radio	**Number of hops to Nearest Node**	**Throughput to Closest Node** (kbps)	**Average Latency to Closest Node** (ms)	**Number of hops to Farthest Node**	**Throughput to Farthest node** (kbps)	**Average Latency to Farthest node** (ms)
2.4 GHz	18	5.13	4203.97	25	3.49	5757.27
868 MHz	1	19.78	258.33	3	6.59	1221.93

Once deployed, the networking proved to be less reliable than expected from the lab testing. Data transfers were not reliable and those that were successful took much longer than desired. Nodes would regularly drop off the network and Router 1 had permanently disconnected the network by winter. It is assumed that Router 1 ran out of energy due to an excessive number of retransmissions and that the rest of the nodes ran out of energy shortly thereafter due to constant failed attempts to transfer data. No useful environmental data was gathered from the initial deployment.

5.3. Second Deployment

The method of data transfer was found to be the cause of the failure of the first deployment due to significant overheads that resulted in decreased node lifetime, increased numbers of retransmissions and reduced reliability. HTTP uses TCP for transport layer encapsulation and while TCP im-

proves reliability in typical network scenarios that have a high bandwidth, it can reduce the chance of communicating and significantly reduce battery life on low-bandwidth, high-latency lossy links such as those used for the deployment in 5.2 and other ESNs. This is due to the number of packets that need to be transmitted to establish the connection and acknowledge communications. The sequence diagram for an example HTTP POST from a directly connected sensor node to the receiver on the border router is shown in Figure 25. In addition to any transport layer retransmissions that may be occurring, retransmissions will be occurring at the MAC layer for each packet due to the use of the ContikiMAC RDC.

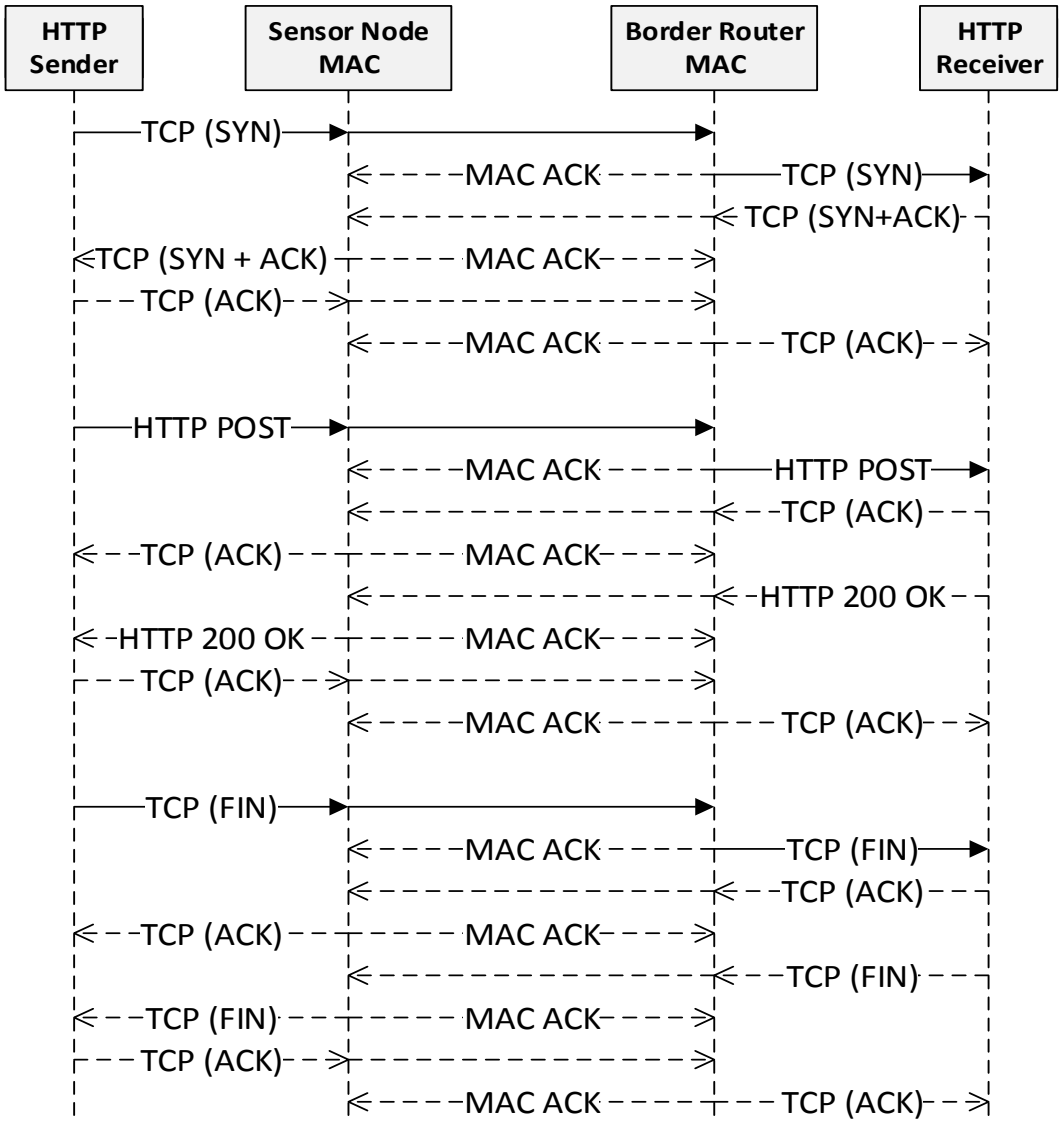


Figure 25: Sequence diagram for a HTTP POST from a sensor node to the border router over a single-hop network. The central column shows communications at the Network Interface layer. The left and right columns show the Transport layer communications. Solid lines indicate an initiated communication while dashed lines indicate a communication triggered in response to an initiated communication. A total of 22 transmissions are required assuming that there are no transport layer retransmissions or fragmented packets.

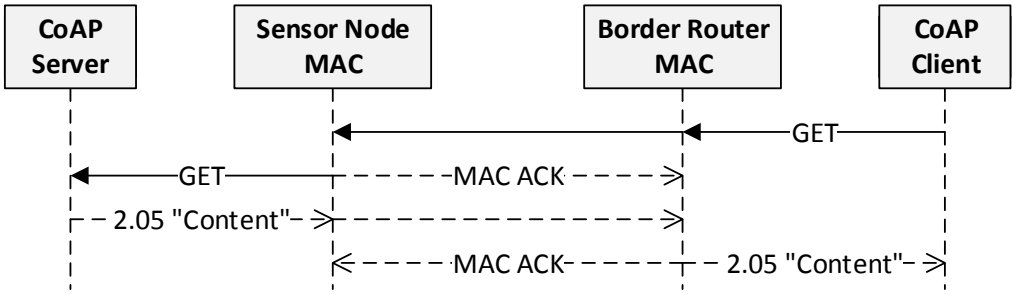


Figure 26: Sequence diagram for a CoAP GET from the border router to a sensor node over a single-hop network. The central column shows communications at the Network Interface layer. The left and right columns show the Transport layer communications. Solid lines indicate an initiated communication while dashed lines indicate a communication triggered in response to an initiated communication. A total of four transmissions are required assuming that there are no transport layer retransmissions or fragmented packets.

In the actual deployment, most nodes were indirectly connected by at least one intermediate routing hop, meaning that everything between the Sensor Node Mac and Border Router MAC in Figure 25 would be transmitted again for each hop increasing the latency of each packet. It is believed that this caused timeouts in the network stack that resulted in TCP retransmissions and ultimately communications failures. If the communications failed at any point for any of the packets the entire communication would have to be initiated again resulting in significant wasted energy. The HTTP-based data transfer appeared to operate acceptably when tested in the lab prior to the deployment and when tested on a lab-based 2.4 GHz deployment due to the reduced ranges involved and the increased per-link bandwidth of a 2.4 GHz network.

CoAP (Shellby, Harke et al. 2014) was chosen to replace HTTP for the second platform as it has a significantly reduced overhead compared to HTTP. CoAP is specifically designed for constrained applications and uses connectionless UDP for transport layer encapsulation. CoAP is HTTP-like and is designed to be translated easily to HTTP though this feature was not used. The implementation of CoAP in Contiki at the time this work was carried out did not fully support node-initiated transfers so gateway-initiated transfers were used instead. A sequence diagram for a CoAP GET request between a directly connected sensor node and the border router is shown in Figure 26. This demonstrates that CoAP requires 18 fewer transmissions on non-duty cycled networks and even on duty cycled networks, there will be significantly fewer transmissions. When multiple hops are involved, the reduction in number of transmissions is more pronounced.

The interaction options offered by the HTTP settings interface were retained as CoAP resources and an additional resource was added for requesting sample data. The resources available on each

node are shown in Table 14. While not as widespread as HTTP, CoAP is well-standardised and there are several implementations available, including plugins for desktop browsers such as the Copper (Cu) plugin for Firefox (Kovatsch n.d.), allowing for simple interaction with the nodes. The change to gateway-initiated transfers means that the gateway needs to be reconfigured any time a new sensor node is added to the network or when an existing node is replaced. In contrast, the nodes no longer require reconfiguration if the destination for sensor data changes. Details of the application are included in Appendix 7.

The limited amount of RAM and flash available on the Z1 proved to be a limiting factor during development of the CoAP based application. Care had to be taken when adding resources and debugging prints had to be turned off to ensure that the firmware fitted in the available flash as the application was larger than the previous deployment. The RAM usage was also higher and the nodes suffered from stack and heap collisions after being on for an extended period of time or when a lot of network activity was occurring. This was mitigated on the routing nodes by removing all sensing code from them and just running the networking stack. The reboots did not cause significant issues on the sensing nodes as all data was stored in flash and the external RTC meant that they retained their schedule.

Table 14: The CoAP resources available on each sensor node. The URI of each resource is shown along with a description of the functionality of a GET request and other available methods.

URI	**GET Functionality**	**Functionality of Other Method(s)**
/date	Get the current date and time as set on the node's RTC in seconds since the Unix Epoch.	POST: Set the date and time of the node to the Epoch specified in the request.
/sample	Get the sample specified as an additional parameter (e.g. GET /sample/23 would get sample 23). If no sample is specified, the latest sample is returned.	DELETE: deletes the sample specified as an additional parameter (e.g. DELETE /sample/23 would delete sample 23. An error is returned if no sample is specified
/config	Gets the current configuration of the sensor node	POST: Sets the node configuration to the configurations specified in the request.
/routes	Get a list of the current neighbours a sensor node can communicate with and a list of the routes that the node has	-
/uptime	Get the current uptime in seconds	-
/reboot	Get the current reboot count	POST: Reboot the sensor node

A second deployment was carried out in 2015 with the CoAP based application developed with other members of the project team running on the nodes. The same hardware as in Section 5.2 was used and all of the node IDs from Table 12 were retained. Figure 27 shows the routing diagram of the deployed network and shows that it is a cluster-tree topology. Router 1, Router 2 and Router 3 are single points of failure beyond the border router that can take down significant sections of the network if they fail, which occurred over the winter months during the deployment when Router 2 failed and disconnected the farthest nodes from the network. The nodes continued to sense and save readings to flash until their batteries ran out in early 2016. These readings were retrieved when Router 2 was repaired and their batteries replaced during a subsequent site visit. It was also found that the connection between Router 2 and Router 3 was only reliable during periods of good weather when there was no snow. The link between Router 1 and Router 2 became less reliable during periods of fog or low cloud.

The nodes were configured to sample every 15 minutes and data retrieval was attempted by the border router every hour. Despite the nodes operating for less than a year and intermittent networking, some useful environmental data was gathered and an initial analysis of segments of the peatland temperature profiles is presented in Bragg, Basford et al. (2016). Analogue data was affected by noise from the switch mode power supplies.

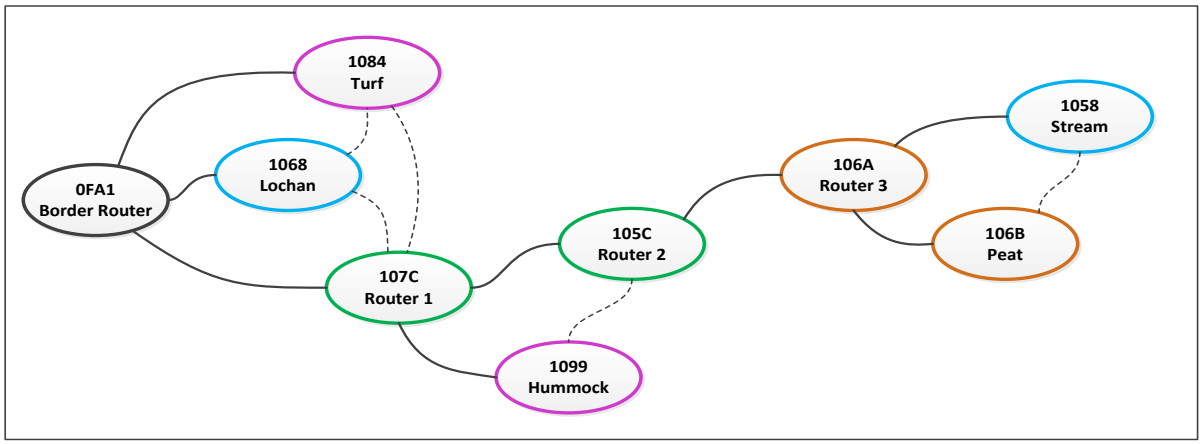


Figure 27: Routing diagram of the second deployment network. Solid lines indicate the routes being used at the time of deployment while dashed lines indicate potential routes between neighbours. The coloured outline indicates the node type: Black is the border router, green are routing nodes, blue are fluvial nodes, pink are periglacial nodes and orange are peat nodes.

5.4. Second Platform: Muntjac + MS2

During the second deployment it was found that performance and reliability were less than desired due to reboots caused by the stack and heap collisions. The memory and flash constraints of the Z1

were even more pronounced with Contiki 3.0 due to the larger memory footprint and changes made in the way that neighbours are stored in memory. Continued use of the Z1 for sensing nodes would have resulted in the number of neighbours and routes having to be excessively reduced, limiting the functionality of the network and preventing any future expansion. A second platform was developed to address these limitations and to address other hardware issues with the original platform.

5.4.1. Hardware

The RAM and flash resources of the MSP430 were a significant limiting factor of the original platform. The MSP430 was also relatively performance limited compared to more modern microcontrollers so such as an ARM Cortex. A Cortex M0+ microcontroller was chosen as the basis of a new platform as it is the most energy efficient ARM processor available (ARM Ltd. 2012); however, Contiki does not included support for any processors based on the M0+ core. A port to a Freescale 48MHz M0+ core with 128 kB of flash, 16 kB of RAM and a sub 3 µA sleep current (Freescale Semiconductor Inc. 2014), was attempted. This was not completed in adequate time before the second deployment so was abandoned in a partially completed state. Currently the port will execute Contiki with networking however the networking is not reliable due to issues with the internal timers and the OS crashes periodically. Details of the source code for the partial port are included in Appendix 7.

As an alternative, the CC2538 was chosen for the basis of a new platform. The CC2538 has a clock speed of 32 MHz, has up to 512 kB of flash with 32 kB of RAM and includes a 2.4 GHz radio that will not be used for the deployment (Texas Instruments 2015b). It is well supported in Contiki and is used in multiple existing platforms such as the Openmote, CC2538-DK and RE-Mote. The Zolertia RE-Mote (Zolertia 2016) includes a CC1200 sub-GHz radio module but at the time this work was carried out the version of the RE-Mote that was available had hardware issues that resulted in reduced reliability and there was incomplete support for the CC1200. Other options that include a sub-GHz radio exist, such as the CC1310 (Texas Instruments 2016b), but at the time this work was carried out they were only just coming to market and were not well supported by Contiki. As the CC2538 incorporates a Cortex M3 and a radio, its sleep current is significantly above that of the MSP430 and M0+ at up to 600 µA depending on power mode.

A new sensor mote, called the Muntjac, based on the CC2538 was designed to replace the Zolertia Z1. The Muntjac was designed to be as close to a pin-compatible drop-in replacement for the Zolertia Z1 as possible but pin number limitations mean that there are fewer analogue inputs available. The Muntjac includes the same SPI-connected flash as the Z1 for storage and has the option of using a removable SD card that can be fully powered off. A CC1120 is included on the mote

with an SMA antenna connection in addition to a U.FL connector for the inbuilt 2.4 GHz radio. The CC1120 was included on-board to allow future designs to be more compact and to allow the Muntjac to be used in a sub-GHz network on its own without a carrier board. The CC1120 is pin compatible and almost software compatible with the CC1200, allowing the Muntjac to be populated with a CC1200 if required. An antenna connector for the 2.4 GHz radio was included to allow the node to be used on either network or to be able to act as a cross-band router.

A new carrier/interface board, the MS2, was also designed to address issues with the MS1 and add additional functionality. The power supplies were redesigned to address noise issues by implementing a two stage design. Switch mode power supplies were used to step the 12 V source down to just above the drop out level of an LDO, which is used to provide a smoother output. The 12 V sensor power switch was moved onto the MS2 and switched using the sensor enable GPIO rather than the 5 V sensor power line. The RS232 driver was removed and two additional, but optional, RS485 drivers were included to allow more smart sensors to be connected in future if required. The headers for the CC1120 were left on-board but rotated 180° to allow Z1s or Muntjacs without radios to be used if required. Another project partner was carrying out research into peak point tracking for solar energy harvesting so an alternative power connector that bypassed the on-board power supplies was included to facilitate this research.

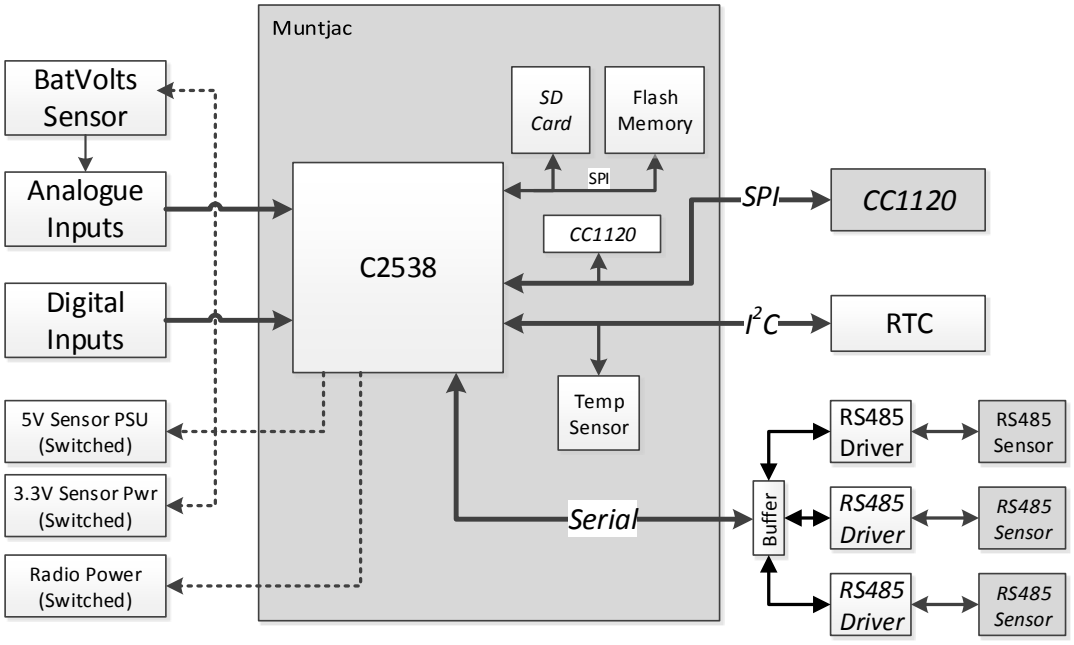


Figure 28: Block diagram of the Muntjac and MS2. Optional components are labelled with italics.

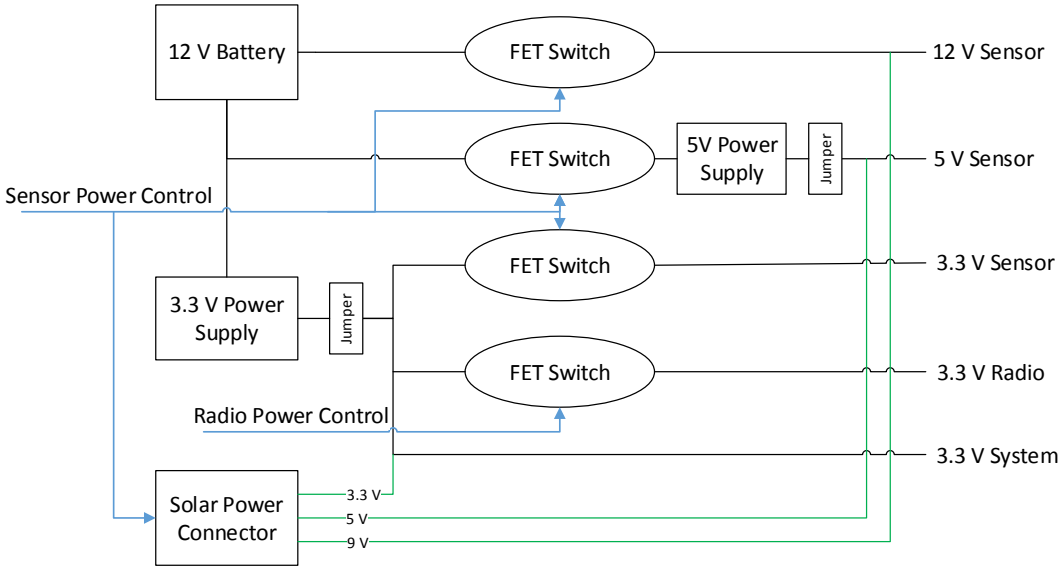


Figure 29: Power domains of the MS2 and Muntjac. Either a 12 V battery or the Solar Power Connector is used to power the system, with the two jumpers removed when powered from the latter. Blue lines indicate control lines and green lines indicate the power input from the solar power connector. The 5 V and 9 V power outputs of the solar power unit are switched by the sensor power control GPIO.

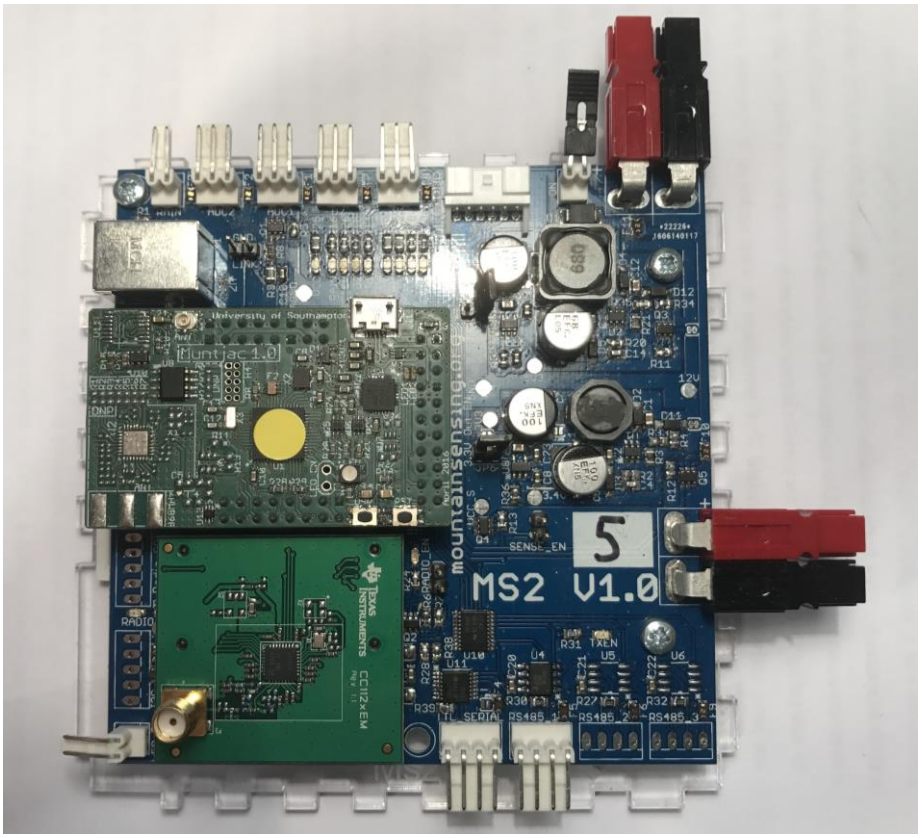


Figure 30: Photograph of the Muntjac sensor mote mounted on an MS2, as deployed. The Muntjac in this photo does not have its on-board CC1120 populated and is using a CC1120 EM module instead.

Figure 28 shows a block diagram of the Muntjac with MS2. The power domains are shown in Figure 29, which are effectively the same as for the MS1. A photograph of a Muntjac without the on-board radio populated mounted on an MS2 is shown in Figure 30. A mix of Muntjacs with and without the on-board radio populated were assembled due to time constraints before the deployment detailed in Section 5.4.3.

Circuit diagrams for the Muntjac and MS2 are included in Appendix 8.

5.4.2. Software

The second generation of nodes were upgraded to Contiki 3.0 and still use protocol buffers for data serialisation. Data is stored in flash as the SD card would use significantly more energy to store and retrieve readings. The same CoAP resources detailed in Section 5.3 are available on the new nodes and the border router software is unchanged.

The single-hop latency for the Muntjac was determined in the lab in the same manner as in 4.4.2 and is shown in Table 15. The Muntjac has a 23.5% lower average latency with Null RDC than the Z1 and a 13.6% lower average latency with ContikiMAC than the Z1. An improvement in performance is to be expected due to the improved processing capabilities of the CC2538.

Table 15: Muntjac latency for NullRDC and ContikiMAC duty cycling protocols.

Duty Cycling Protocol	**Minimum Latency** (ms)	**Average Latency** (ms)	**Standard Deviation** (ms)		**Δ from Z1 values**	
					Min (ms)	Min (ms)
NullRDC	75.85	85.04	5.01		26.75	19.98
ContikiMAC	146.12	227.32	294.53		23.38	31.01

5.4.3. Deployment

A third deployment was carried out with all of the sensor nodes being replaced with Muntjac + MS2s deployed in the same location as the previous nodes. The same hardware used for previous border routers was retained.

Changes were made to the network to address the intermittent communications between the routing nodes. The Hummock sensor node was mounted on a tripod to have better line of sight between Router 1 and Router 2 so that it could act as an intermediate relay node during foggy conditions.

An additional routing node, Router 4, was placed with line of sight of both Router 2 and the nodes in area 3. The location of the nodes in the latest deployment is shown in Figure 31 along with predicted coverage. Figure 32 shows the routing diagram for the network on the day of deployment. Router 1 and Router 2 are still single points of failure in the network but Router 4 adds redundancy for Router 3 and its unreliable link to Router 2. The only way to create redundancy in the deployment network for Router 1 and Router 2 would be to add at least two additional nodes.

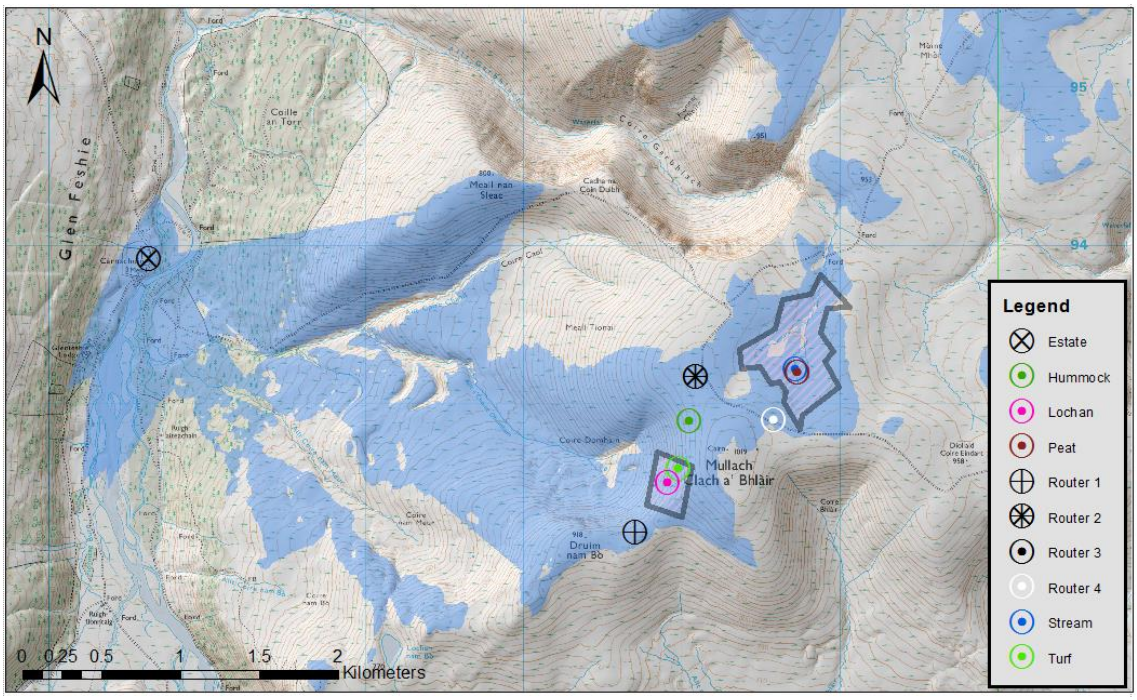


Figure 31: Modelled coverage for the third 868 MHz ESN deployment. Estimated radio coverage is shown by the shaded (blue) areas. At this scale, Router 3 is co-sited with the Peat sensor node. Original areas of interest are shown by the outlined hatched boxes. DTM data from (Intermap Technologies 2007) and Crown Copyright Ordnance Survey.

During the deployment the nodes equipped with energy harvesting failed due to a bug in the energy harvesting implementation that resulted in the sensors being powered permanently, depleting the batteries. The energy harvesting modules were replaced by lead acid batteries by another member of the project team on a subsequent site visit and all of these nodes re-joined the network and continued to operate.

5.5. Platform & Deployment Performance

This section presents the results of energy and latency performance testing that was carried out on the two generations of deployed hardware and the third deployment network.

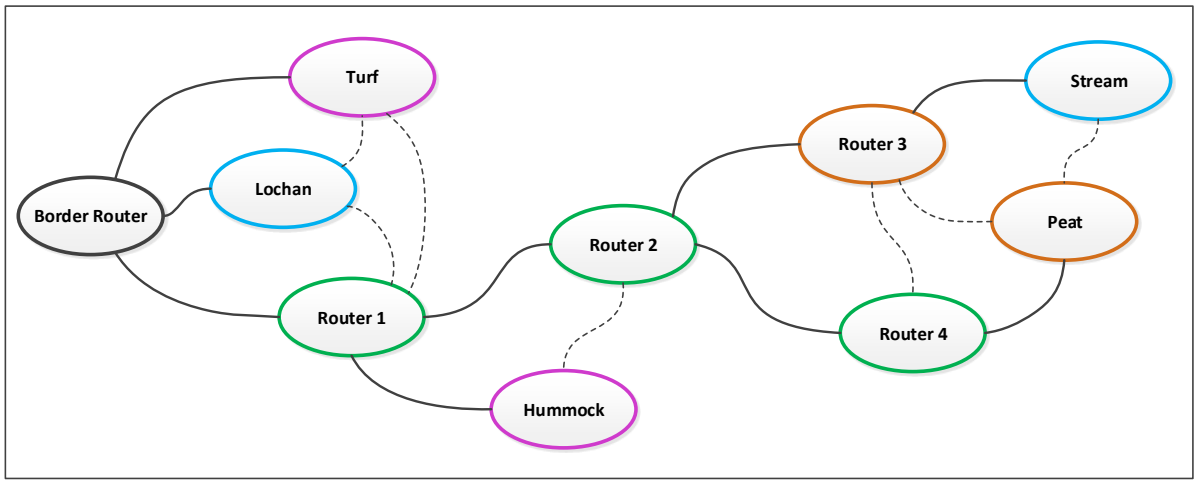


Figure 32: Routing diagram of the third deployment network. Solid lines indicate the routes being used at the time of deployment while dashed lines indicate potential routes between neighbours. The coloured outline indicates the node type: Black is the border router, green are routing nodes, blue are fluvial nodes, pink are periglacial nodes and orange are peat nodes.

The energy performance of the hardware that was deployed was measured in the laboratory to establish the average energy consumption. Table 16 details the average current measured for both of the hardware platforms used for the deployments detailed in this chapter with the NullRDC and ContikiMAC duty cycling protocols. These readings reflect the idle current draw as they were taken with the nodes running an application that does nothing on top of the base OS operation with an operational network stack that responds to ping. The standard deviation of the MS1-based platform is lower than the MS2-based platform due to the large smoothing capacitor on the 12 V input of the MS1. As expected, the Muntjac + MS2 draws more current than the Z1 + MS1 and ContikiMAC draws significantly less average power than NullRDC. ContikiMAC draws about 5% of the current as NullRDC for both platforms and a greater saving could have been realised with phase synchronisation. Phase synchronisation was not implemented in either deployment as it was found to require modification to operate with the adjusted ContikiMAC timings.

Values for the Muntjac + MS2 in two no radio scenarios are also included in Table 16 to give a baseline for the platform to allow the viability of alternative duty cycling methods to be assessed. The first, referred to as NullRadio in the table, uses a dummy radio driver and the CC1120 in the sleep state to approximate a situation where the CC1120 is left in a sleep state for a prolonged period. The second, referred to as no Radio, uses the NullRadio driver and turns off the CC1120 with the hardware power control to approximate a situation where the CC1120 is completely powered off for a prolonged period.

Table 16: Total node average current consumption with a 12 V input voltage for both node types with NullRDC and ContikiMAC. The current draws of the Muntjac + MS2 with the NullRadio radio driver selected and with the CC1120 fully turned off are also included. 1,000,000 readings were gathered at a sample rate of 10,000 per second with an Agilent 34411A Digital Multimeter and then averaged.

Hardware	**Average Current** (mA)	**Standard Deviation** (mA)	**Average Power** (mW)
Z1 + MS1 NullRDC	8.207	0.847	98.484
Z1 + MS1 ContikiMAC	0.425	0.266	5.100
Muntjac + MS2 NullRDC	10.817	1.021	129.804
Muntjac + MS2 ContikiMAC	0.581	1.886	6.972
Muntjac + MS2 NullRadio	0.484	1.337	5.808
Muntjac + MS2 no Radio	0.394	1.329	4.728

From these values, it is possible to calculate the expected node lifetime on the batteries that were used for the deployments when energy harvesting was not in use. Sensor nodes used 12 Ah lead acid batteries and routing nodes used 17 Ah lead acid batteries (Yuasa 2005). Ignoring any self-discharge, a sensor node using a Muntjac and MS2 with a 12 Ah battery should last up to 860 days; however, network activity and self-discharge, which will vary with temperature across the year, will decrease this. Rough calculations can also be performed to assess the impact that other forms of duty cycling can have on network lifetime. Table 17 presents a brief comparison of different duty cycling concepts and the impact that they have on node lifetime. The table shows that the duty cycling algorithm used for the sub-glacial probes in the Glacsweb project would increase node life-

Table 17: Predicted life time with a 12 Ah battery for different duty cycling protocols.

Duty Cycling Protocol	Duty Cycle (%)	Average Current (mA)	Node Lifetime (days)
NullRDC (Z1 + MS1)	100	8.207	60.9
ContikiMAC (Z1 + MS1)	1	0.425	1176.5
NullRDC (Muntjac + MS2)	100	10.817	46.2
ContikiMAC (Muntjac + MS2)	1	0.581	860.6
Sub-glacial node Synchronous protocol from Chapter 3 (Muntjac + MS2)	0.069	0.401	1246.1
Two hour-long periods with ContikiMAC per day (Muntjac + MS2)	0.083	0.410	1220.7
One hour-long period with ContikiMAC per day (Muntjac + MS2)	0.042	0.402	1244.4

time by just over a year. Combining synchronous and asynchronous RDC concepts in a hybrid protocol could bring the benefits of both with a reasonable duty cycle and longer communication windows. Two possible hybrid schemes are included in the table and the values for these show that at extremely low duty cycles, the amount of energy expended on communications becomes less significant and the amount of energy expended carrying out processing becomes more significant.

Figure 33 presents a plot of a 200 ms segment of the current data for the Muntjac + MS2 with ContikiMAC that shows two ContikiMAC channel check events and a large number of system tick wakeup events. The plot shows that the node spends most of its time in a low-power state but is waking up very regularly.

The current draw after the 3.3 V power supply was also measured to establish the efficiency of the redesigned power supply. The average current draw was measured to be 1.423 mA at 3.27 V, an efficiency of 68%. Figure 34 demonstrates that the redesigned power supply is less noisy and more stable than the original design.

The single-hop and two-hop round-trip latency of networks deployed in sections 5.3 and 5.4 was measured to determine the latency performance in real-world deployments. Table 18 presents the results and shows that there is an increase in the latency when compared to the laboratory based testing. A very small amount of data was gathered to limit the impact to overall node lifetime as this was carried out on the deployed networks. The latency during deployment is higher and less consistent than achieved in the laboratory environment due to the longer ranges involved and variable weather conditions that affect the quality of the radio link. A ping test during a period of low cloud where all of the sensor nodes were above the base of the cloud level gave an average single-hop RTT of over 600 ms and an average 2-hop RTT of over 1400 ms with standard deviations in the hundreds of milliseconds and over 10% packet loss.

Table 18: Latency of the two deployed hardware platforms running the CoAP application. Data was gathered for a single-packet ping during good weather conditions with no other network activity. Data was gathered with “ping6 –c 10 –i 5” to limit the amount of energy expended.

Hardware	**Single-hop RTT** (ms)	**Δ from Lab Values** (ms)	**2-hop RTT** (ms)	**Δ from Lab Values** (ms)
Z1 + MS1	303	45	820	80
Muntjac + MS2	274	47	664	-

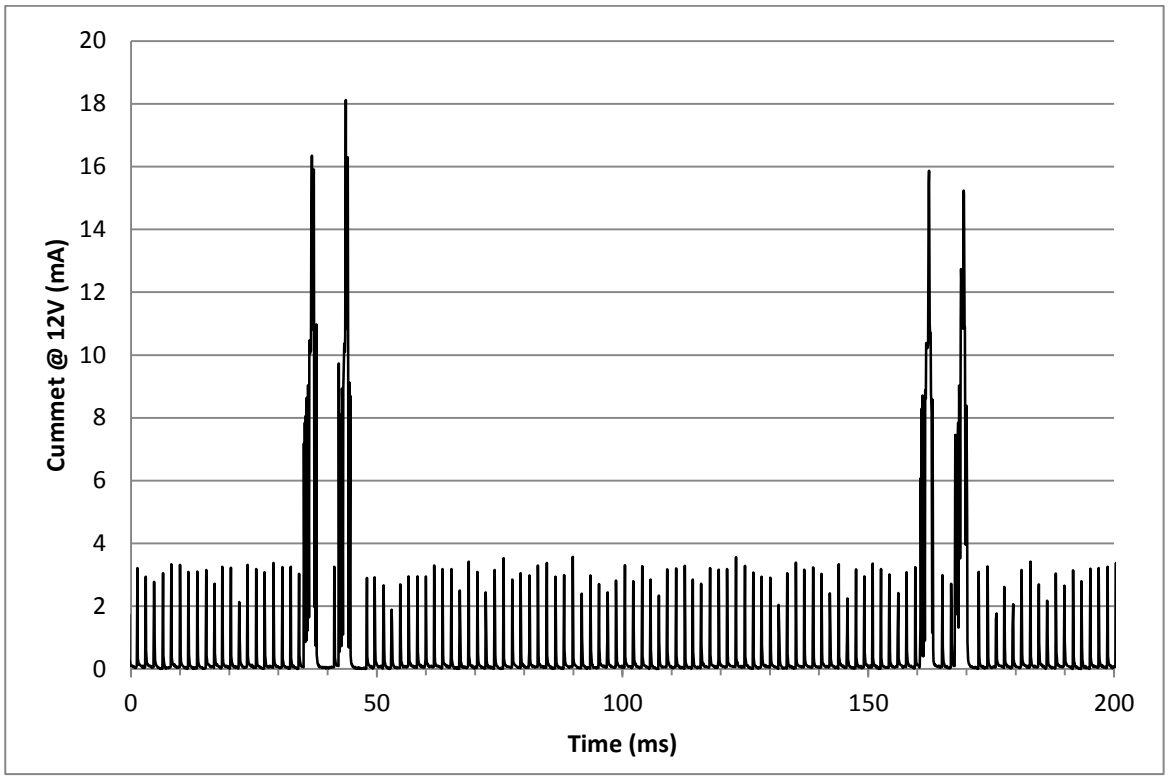


Figure 33: Plot of a 200 ms segment of the Muntjac + MS2 current data. The four large peaks are two ContikiMAC channel checks. The shorter peaks are the Contiki systick wakeup events.

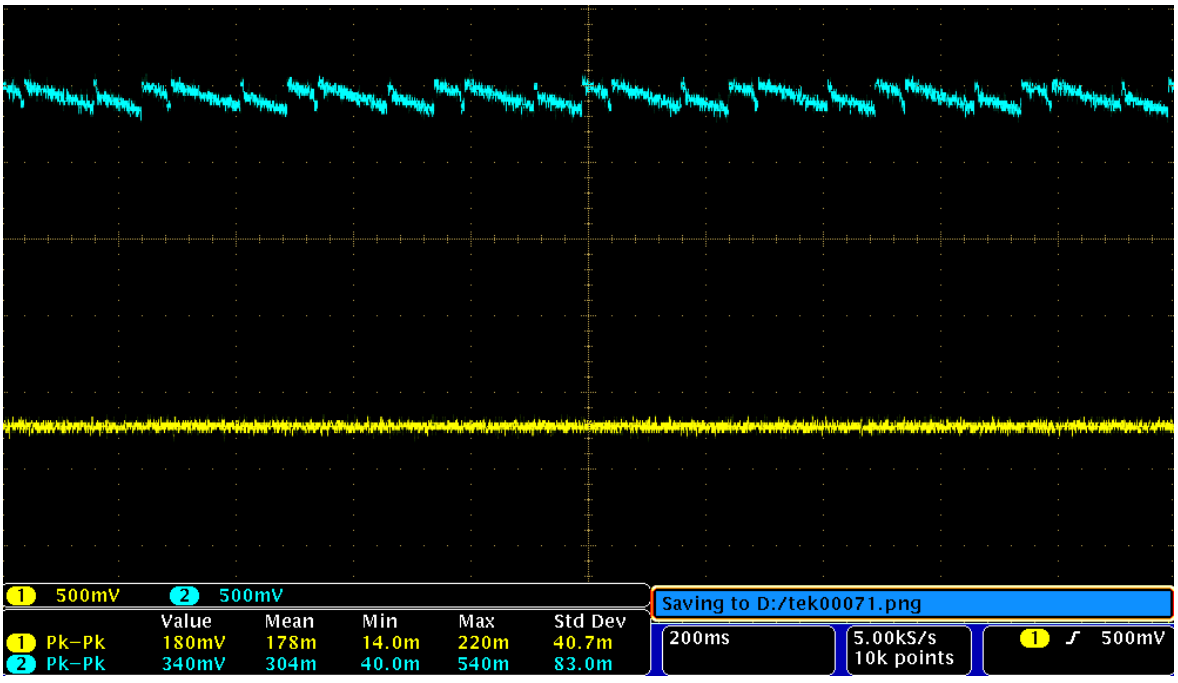


Figure 34: Power supply output noise comparison with an idle node. The top (blue) trace represents the output from the MS1's power supply. The bottom (yellow) trace represents the output of the MS2's power supply.

The Muntjac had a latency improvement of approximately 10.6% for single-hop and 23% for two-hop communications compared to the Z1 however this increase in performance comes at the cost of increased power. The Muntjac + MS2 draws 32% more power than the Z1 + MS1 with NullRDC and 37% more with ContikiMAC. This is a significant increase but is less than might have been expected as the CC2538 has a sleep current of 1.3 µA to 600 µA (Texas Instruments 2015b), depending on power mode, compared to the 0.5 µA of the MSP430 used in the Z1 (Texas Instruments 2012).

5.6. Results Discussion

This chapter has shown that Internet standards can be applied to real-world sub-GHz environmental sensor networks successfully with care and some modification. Heavyweight protocols such as HTTP are not appropriate as they were not designed with lossy networks in mind and in the case of TCP-based protocols, the overheads involved in establishing and maintaining reliable communications make the network as a whole less reliable and less energy efficient. Lighter weight protocols have been developed and standardised in recent years with WSNs in mind. It has been shown that one of these lightweight protocols, CoAP, can be applied to ESNs despite their different performance characteristics and deployment challenges. When used with a sub-GHz 802.15.4 radio with the ContikiMAC RDC and a 6LoWPAN multi-hope network, CoAP and protocol buffers proved to be reliable and appropriate for this ESN deployment.

In addition to partially addressing the first research question posed in Chapter 1, this chapter also addresses the second and third research questions. The next two sub-sections discuss how these questions have been addressed and the metrics that they have been assessed against.

5.6.1. Second Research Question

The second research question asked whether a sub-GHz standards-based 6LoWPAN network can perform sufficiently in terms of energy for real-world IoT environmental sensor network deployments. The lifetime calculations in Table 17 demonstrated that the Muntjac + MS2 has a predicted lifetime in excess of two years of receive-only operation with the ContikiMAC duty cycling protocol and a 12 Ah battery. While this is an acceptable lifetime, the 12 Ah lead acid battery is physically large compared to the node side and has undesirable self-discharge characteristics that reduce the lifetime below the calculated value. These factors make a large lead acid battery unsuitable for nodes that need to be physically small and/or are physically inaccessible once deployed, such as the

sub-glacial nodes discussed in Chapter 3. Common 2.4 GHz WSN deployments generally use smaller batteries that will not give an acceptable lifetime for the Muntjac + MS2.

Compared to non-IP ESNs, the IP-based deployments detailed in this chapter use significantly more energy. Recent advances in networking for WSNs have focused on providing an always-on network experience where data can be transmitted in near real time. Compared to the use of no RDC, ContikiMAC did save a significant amount of energy but it is not the most optimal method of duty cycling for ESNs where spontaneous transmission of data is not required and the data is not time sensitive so further work is required to investigate possible optimisations in this area. Additionally, the scheduler used in Contiki wakes up the microcontroller frequently to check for any pending tasks, increasing the energy over a bare-metal implantation. Frequent wakeups are not the most efficient paradigm for deployments where sensing occurs in the order of minutes and communications in the order of hours.

5.6.2. Third Research Question

The third research question asked whether a sub-GHz standards-based 6LoWPAN network had suitable throughput, latency and reliability for real-world environmental sensor network deployments. The latency measured during the deployment was greater than what was measured in the laboratory. Despite the increased latency, the latency and throughput performance was acceptable as it was possible to extract data from the nodes in a reasonable time when CoAP was used as the application layer protocol for data transfer. Compared to a theoretical 2.4 GHz deployment, the sub-GHz deployment discussed in Section 5.2 had significantly lower latency and increased throughput. This demonstrates that a sub-GHz 6LoWPAN network is more appropriate than a 2.4 GHz-based design for environmental monitoring in isolated areas with large distances between nodes.

In terms of reliability, all three of the deployments experienced periods where individual nodes were unavailable or the network as a whole was not functional due to environmental conditions, battery depletion or other node failure. Despite these issues, the use of an asynchronous RDC allowed the network to recover and rebuild itself without intervention when the underlying issue had resolved and the use of sample-and-store, rather than sample-and-send, allowed nodes that were isolated to continue to gather data that could later be retrieved. Several single points of failure existed in all of the deployments and these negatively impacted on the reliability as the failure of any one of these single points disconnected multiple nodes from the network, as occurred with the failure of Router 2 discussed in Section 5.3; however, this reduction in reliability was due to decisions during the specific deployments rather than an issue inherent to standards-based designs.

5.7. Summary

In summary, this chapter has demonstrated that a standards-based sub-GHz 6LoWPAN-based environmental sensor network can perform sufficiently in terms of energy, throughput, latency and reliability for certain deployment scenarios through series of real-world deployments in the Highlands of Scotland. Further work is required to increase energy efficiency to improve node lifetime and enable operation on physically smaller batteries. Section 5.5 presented initial values for a hybrid duty cycling protocol based on using ContikiMAC with defined communications windows and shows that this is comparable in terms of node lifetime to the basic synchronous RDC used in Chapter 3.

Even when using a network intended to give always on connectivity, nodes need to be able to cope with long periods of isolation and be able to re-establish connectivity when environmental conditions allow. Because of their inherently vulnerable nature, ESNs should avoid any single points of failure in their routing design with redundant communications paths where possible.

In addition to addressing the research questions posed in Chapter 1, the work detailed in this chapter has resulted in the first published real-world deployment of a sub-GHz 6LoWPAN environmental sensor network. Additionally, this sensor network gathered real-world environmental data that demonstrated how the temperature profile of peat varied with weather events.

Chapter 6

Conclusions

The Internet of Things is a key part of the of the future Internet and wireless sensor networks are seen as a key enabler of this vision as they facilitate interaction with the physical world. Advances in WSN design and the adoption of standards such as 6LoWPAN and RPL have allowed WSNs to fully participate in the Internet. The ability to interact with the Internet comes at the expense of increased processing overheads and the low-power microcontrollers that have been used in common commercial WSN platforms are becoming too limited in terms of RAM and flash for fully standards-based IoT applications. Additionally, the processing performance of these microcontrollers limits the latency and throughput that can be achieved. 32-bit ARM Cortex-based microcontrollers are becoming common in WSN designs as they are more capable than the 8-bit and 16-bit microcontrollers used by older designs but the increase in performance come at increased energy.

Environmental sensor networks are a key tool in helping to facilitate the work of researchers in the field of Earth sciences. ESNs allow researchers to have more continuous monitoring of natural processes that are not well understood. The adoption of ESNs has been hindered by a lack of standardisation, with contemporary designs relying on bespoke communications and interface implementations, and a need for expertise with the technology for deployment, maintenance, configuration and data retrieval. Integration with the Internet of Things, or at least the adoption of standardised methods of communicating with the Internet, is seen as the next step in improving ESN design. Significant research has been carried out in by other workers to improve the state of the art of WSNs for applications where 2.4 GHz radios suffice for communications, where power is more readily available and where Internet connectivity is commonplace; however, limited progress has been made in the applications of standards to ESNs.

Common themes in contemporary environmental sensor networks are nodes that spend most of their time in a low-power state with a sleepy network using lossy sub-GHz radio connections. In contrast, contemporary IoT WSNs use networks that appear to always be connected and standardisation has been developed with this assumption.

The remainder of this chapter is divided into three sections. Section 6.1 discusses the contributions made in this thesis and how the research questions posed in Chapter 1 were addresses. Section 6.2 explores areas of future work and Section 6.3 presents a closing summary.

6.1. Thesis Contributions

The worked detailed in this thesis can be assessed against the research questions detailed in Section 1.2, which are restated here for clarity:

1. Can the standards used for 2.4 GHz IoT wireless sensor networks be applied to sub-GHz environmental sensor networks?
2. Can a sub-GHz standards-based 6LoWPAN network perform sufficiently, in terms of energy, for real-world IoT environmental sensor network deployments?
3. Can a sub-GHz standards-based 6LoWPAN network perform sufficiently, in terms of throughput, latency and reliability, for real-world IoT environmental sensor network deployments?

The first question has been answered positively in Chapter 4 and Chapter 5. In Chapter 4, the theoretical performance of 6LoWPAN at 868 MHz was established and verified experimentally. This was achieved by developing a driver for an 802.15.4g-compatible sub-GHz radio for an MSP430-based platform running the Contiki operating system. The throughput and latency of a 50 kbps 868 MHz 6LoWPAN network are less than an equivalent 250 kbps 2.4 GHz network as expected; however, the results showed that the bitrate is not the only significant factor that affects the attainable throughput and latency. Chapter 5 described several real-world deployments that were carried out using a standards-compliant 6LoWPAN network with RPL routing using CoAP and protocol buffers.

Radio communications account for up to three quarters of the energy utilisation of a typical sensor network. Of the possible methods of reducing communications activity, duty cycling has the most significant effect on the energy utilisation as the radio is turned off or kept in a low power state as much as possible. Chapter 4 investigated the theoretical performance of two asynchronous different duty cycling protocols: ContikiMAC, which claims to be very efficient when used with 2.4 GHz WSNs; and RX Sniff, which is incorporated into the CC1120 sub-GHz radio. Of these two, RX Sniff would not operate at 50 kbps and ContikiMAC achieved a duty cycle of approximately 1%, comparable with operation at 2.4 GHz.

ContikiMAC was used for the deployments in Chapter 5 where the second research question was addressed. Predicted lifetimes for the two sensor nodes were calculated and showed that both nodes should operate for in excess of two years. This lifetime would appear to be sufficient performance however it is achieved using a physically large battery and the nodes used more energy than non-IP environmental sensor network designs. Initial calculations for alternative duty cycling approaches were carried out to determine the potential for further energy efficiency gains. Additionally, the use of solar energy harvesting is being investigated to allow physically smaller batteries to be used.

The second research question can thus be answered positively with some qualification: a sub-GHz standards-based 6LoWPAN network can perform sufficiently in terms of energy for real-world IoT environmental sensor network deployments where high-capacity batteries can be used; however, further work is required to improve energy efficiency as current performance is below non-IP environmental sensor network designs and 2.4 GHz wireless sensor network designs.

The third research question was also answered positively in Chapter 5 where it was shown that a sub-GHz standards-based 6LoWPAN network using CoAP for data transfer and protocol buffers for data serialisation can be successfully deployed as part of a real-world ESN deployment in an environment with varying harsh conditions and sparse node deployments. The throughput and latency achieved were adequate for the deployment scenario discussed. Additionally, the use of a 6LoWPAN network stack and asynchronous duty cycling protocol that gave the experience of an always-on network proved to be reliable as nodes that left the network due to environmental or power issues were able to re-join the network when conditions allowed.

Aside from addressing the research questions posed earlier, other contributions have been made. The work detailed in this thesis has helped to deploy the world's first real-world trial of an 868 MHz 6LoWPAN sensor network for environmental sensing in the Highlands of Scotland and the data gathered with this network is helping researchers in other fields understand the processes involved and facilitating their research in those fields.

The timings determined for ContikiMAC in Chapter 4 have already been used by researchers at other institutions for their own investigations and with other platforms that have been released during the course of this work.

The driver for the CC1120 developed in section 4.2 is open-source and publicly available for other workers to use as they see fit. The author is aware that it has already been used as part of an undergraduate group design project, in collaboration with an industrial partner, that investigated decen-

tralised mesh networks for disaster response applications where data on any node in the network can be accessed from any other node with an arbitrary node acting as the sink at any period of time.

6.2. Future Work

As is common with most doctoral theses, the research carried out as part of this thesis has highlighted areas where further work can be carried out. These areas are summarised in this section. Compared to using no duty cycling protocol or the method integrated into the CC1120, ContikiMAC used less energy and achieved a duty cycle of about 1%. This is greater than the duty cycle achieved for the Glacsweb sub-glacial probes with their simple synchronous implementation but the asynchronous RDC proved more reliable for the Mountain Sensing deployment due to the frequently changing environmental conditions. Nodes that had lost power or connectivity were able to re-join the network when they were able and recover if they had lost their timing information. A hybrid RDC, which uses a combination of specified periods of communication and an asynchronous duty cycling within those periods, would reduce the duty cycle of a purely asynchronous approach while retaining the flexibility of an asynchronous approach. A hybrid duty cycling protocol with daily windows for communication would remove the appearance of an always-on network that can be achieved with ContikiMAC. The always-on experience makes managing the network more convenient for end users as they are not confined to a set schedule. Some of this convenience could be retained by implementing caching of commands at the gateway, delay tolerant networking or other similar techniques.

While outside the scope of the work detailed in this thesis, additional energy efficiency could be realised by implementing on-node processing techniques, such as in-network data compression (Sheltami, Musaddiq et al. 2016) and data aggregation (Fasolo, Rossi et al. 2007), to reduce the amount of data that needs to be transmitted. The increased processing requirement of a full network stack aligns with the increased processing requirements of more intelligence at the node as the increased intelligence will reduce the amount of processing required for networking. This requires additional work to identify the specific challenges and constraints associated with combining these technologies with standardised IP-based communications in an environmental sensor network context.

Operation of 6LoWPAN at lower frequencies, such as 169 MHz, was not assessed and investigation would be required to assess if 6LoWPAN is feasible for these bands. If the use of ContikiMAC is desired, then the timings for each band will have to be established. The performance of RPL will also need to be assessed and it is likely that the parameters will need to be optimised for operation

at the significantly lower data rate available. If 6LoWPAN at 169 MHz is a realisable concept then a fully IP-capable glacial monitoring system could be developed.

All of the nodes using the same radio operating on the same frequency with the same transmit power is not optimal in terms of energy consumption. Nodes with lower range requirements, such as sensor nodes close to routing nodes, could operate with a lower transmit power for a shorter effective range. Dynamic transmit power (DTP) has been used for other types of sensor network including deployments that use 802.15.4 and 6LoWPAN (Lin, Zhang et al. 2006, Xiao, Sivaraman et al. 2008, Rukpakavong, Phillips et al. 2011, Vallejo, Recas et al. 2015) so this could be implemented for a sub-GHz 6LoWPAN ESN however this does not address all of the issues of single-frequency operation.

With the use of an asynchronous RDC, a node will receive all of the packets transmitted that it is in range to receive whether they are for that node or not. This means that in a cluster tree network, all of the nodes in a cluster will receive all of the transmissions from the cluster root even if the transmissions only relate to traffic transiting through the cluster, resulting in the leaf nodes expending energy on receiving unnecessary traffic. In the specific case of the Mountain Sensing deployment, this means that any of the sensor nodes connected to Router 1 will receive any traffic that is transmitted from Router 1. This would still be the case with DTP as the routing nodes would be using a transmit power appropriate for long-range communications. A synchronous RDC with different transmission slots for routing nodes and sensing nodes could be developed to prevent sensing nodes receiving traffic between two routing nodes but this would increase software complexity and would need to be dynamic to cope with nodes that can be a routing node depending on weather conditions, such as Router 3 and Hummock in the third deployment.

Another possible solution to this would be to use different frequency bands for different tasks: for example, the long-range links could be implemented using 868 MHz while the local connections from a routing node could be implemented with 2.4 GHz. This would improve the energy profile of sensor nodes as common 2.4 GHz radios have a lower receive energy than current 868 MHz radios but would increase the complexity of routing nodes and constrain routing to nodes equipped with the correct hardware. Research into extending RPL to cope with multiple interfaces in a graceful manner without relying on flooding incoming transmissions to all interfaces would be required to achieve this as existing RPL research and development has focused on networks that operate in a single band. The ability to extend an 802.15.4 network across different frequency bands is a requirement for a heterogeneous ESN like Glacsweb that uses different frequency bands for different purposes.

The method of latency estimation proposed in Chapter 4 showed some promise as the results were consistent with what was achieved experimentally, with a systematic error. Further work is required to validate the accuracy of the method and to establish its applicability more generally to other frequency bands and other IoT operating systems.

6.3. Closing Summary

Standardisation of networking technologies in recent years has allowed WSNs to interact with the Internet using variations of Internet Standards. The ability to interact with the Internet has in turn made it easier to deploy WSNs for a variety of applications in urban and agricultural environments. This thesis has shown that the same standardised technologies can be applied to ESNs at the cost of increased energy and processing capabilities. These technologies have been successfully deployed as part of the Mountain Sensing project where environmental data has been sensed and communicated using a sub-GHz 6LoWPAN sensor network that used CoAP for data transfer. The deployments carried out as part of the Mountain Sensing project demonstrated that a standards-based sub-GHz network can perform acceptably for a real-world ESN deployment however further work is required to increase network lifetime and improve communications efficiency.

Bibliography

22 CityLink. (2016). "The Internet of Things Is Here: Fitness Wearables, Smartwatches & Smartglasses [Wearables Part 1/2] " Retrieved May 2017, from https://22citylink.com/future-internet-of-things-fitness-wearables-smartwatches-smartglasses/.

Akyildiz, I. F., W. Su, Y. Sankarasubramaniam and E. Cayirci (2002). "Wireless sensor networks: a survey." Computer Networks **38**(4): 393-422.

Al-Fuqaha, A., M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash (2015). "Internet of things: A survey on enabling technologies, protocols, and applications." IEEE Communications Surveys & Tutorials **17**(4): 2347-2376.

Amazon. (2016). "AWS IoT Button." Retrieved April 2017, from https://aws.amazon.com/iotbutton/.

ARM Ltd. (2012). "Cortex-M0+ Processor." from https://www.arm.com/products/processors/cortex-m/cortex-m0plus.php.

ARM Ltd. (2014). "Technology / mbed OS." from http://mbed.org/technology/os/.

Ashton, K. (2009). "That 'Internet of Things' thing." Retrieved October 2013, from http://www.rfidjournal.com/article/view/4986.

Atzori, L., A. Iera and G. Morabito (2010). "The Internet of Things: A survey." Computer Networks **54**(15): 2787-2805.

Baccelli, E., O. Hahm, M. Gunes, M. Wahlisch and T. Schmidt (2013). RIOT OS: Towards an OS for the Internet of Things. INFOCOM. Turin, Italy, IEEE.

Baccelli, E., O. Hahm, M. Wahlisch, M. Gunes and T. Schmidt (2012). RIOT: One OS to Rule Them All in the IoT. , Centre pour la Communication Scientifique Directe.

Bader, S., M. Krämer and B. Oelmann (2013). A Domain-Specific Platform for Research in Environmental Wireless Sensor Networks. SENSORCOMM 2013, The Seventh International Conference on Sensor Technologies and Applications. Barcelona, Spain**:** 200-207.

Barrenetxea, G., F. Ingelrest, G. Schaefer and M. Vetterli (2008). The hitchhiker's guide to successful wireless sensor network deployments. Proceedings of the 6th ACM conference on Embedded network sensor systems, ACM.

Basford, P. J. (2015). Using an energy aware adaptive scheduling algorithm to increase sensor network lifetime Doctoral, University of Southampton.

Basford, P. J., G. M. Bragg, J. S. Hare, M. O. Jewell, K. Martinez, D. R. Newman, R. Pau, A. Smith and T. Ward (2016). "Erica the Rhino: A Case Study in Using Raspberry Pi Single Board Computers for Interactive Art." Electronics **5**(3): 35.

Beutel, J., S. Gruber, A. Hasler, R. Lim, A. Meier, C. Plessl, I. Talzi, L. Thiele, C. Tschudin, M. Woehrle and M. Yuecel (2009). PermaDAQ: A scientific instrument for precision sensing and data

recovery in environmental extremes. Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on.

Bisnath, S. and Y. Gao (2009). "Precise point positioning." GPS world **20**(4350): 4.

Blackstock, M., N. Kaviani, R. Lea and A. Friday (2010). MAGIC Broker 2: An open and extensible platform for the Internet of Things. Internet of Things (IOT), 2010.

Bragg, G. M., K. Martinez, P. J. Basford and J. K. Hart (2016). 868MHz 6LoWPAN with ContikiMAC for an Internet of Things environmental sensor network. 2016 SAI Computing Conference (SAI). London**:** 1273-1277.

Bragg, O. M., P. J. Basford, A. R. Black, G. M. Bragg, J. K. Hart and K. Martinez (2016). BRITAIN'S HIGHEST BOG: CAN WE UNLOCK ITS SECRETS? 15th International Peat Congress, Kuching, Malaysia, International Peat Society.

Buettner, M., G. V. Yee, E. Anderson and R. Han (2006). X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. Proceedings of the 4th international conference on Embedded networked sensor systems. Boulder, Colorado, USA, ACM**:** 307-320.

Burgess, S. S., M. L. Kranz, N. E. Turner, R. Cardell-Oliver and T. E. Dawson (2010). "Harnessing wireless sensor technologies to advance forest ecology and agricultural research." Agricultural and Forest Meteorology **150**(1): 30-37.

Burrell, J., T. Brooke and R. Beckwith (2004). "Vineyard computing: sensor networks in agricultural production." Pervasive Computing, IEEE **3**(1): 38-45.

Centenaro, M., L. Vangelista, A. Zanella and M. Zorzi (2016). "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios." IEEE Wireless Communications **23**(5): 60-67.

Chi, Q., H. Yan, C. Zhang, Z. Pang and L. D. Xu (2014). "A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment." IEEE Transactions on Industrial Informatics **10**(2): 1417-1425.

CMUSCS. (1998, February 2005). "CMU SCS Coke Machine." Retrieved December 2015, from https://www.cs.cmu.edu/~coke/.

Contiki-OS. (2012a). "Contiki Hardware." Retrieved Novemeber 2014, from http://www.contiki-os.org/hardware.html.

Contiki-OS. (2012b). "Multithreading, Contiki-OS Wiki." Retrieved September 2016, from https://github.com/contiki-os/contiki/wiki/Multithreading.

Contiki-OS. (2014). "Radio duty cycling, Contiki-OS Wiki." Retrieved September 2016, from https://github.com/contiki-os/contiki/wiki/Radio-duty-cycling.

Contiki. (2015, 26/08/2015). "Contiki: The Open Source OS for the Internet of Things." Retrieved January 2016, from http://www.contiki-os.org/.

Crossbow Technology Inc (n.d.-a). MICA2 Wireless Measurement System.

Crossbow Technology Inc (n.d.-b). TelosB Mote Platform.

Crouch, I. (2015). "The Horror of Amazon's New Dash Button." Retrieved January 2017, from http://www.newyorker.com/culture/culture-desk/the-horror-of-amazons-new-dash-button.

Czyz, J., M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil and M. Bailey (2014). Measuring IPv6 adoption. Proceedings of the 2014 ACM conference on SIGCOMM. Chicago, Illinois, USA, ACM**:** 87-98.

Deering, S. and R. Hinden (2017). RFC 8200: Internet protocol, version 6 (IPv6) specification, IETF.

Dubois-Ferrière, H., L. Fabre, R. Meier and P. Metrailler (2006). TinyNode: a comprehensive platform for wireless sensor network applications. Proceedings of the 5th international conference on Information processing in sensor networks, ACM.

Dunkels, A. (2011). The ContikiMAC Radio Duty Cycling Protocol. Sweden, Swedish Institute of Computer Science.

Dunkels, A., B. Gronvall and T. Voigt (2004). Contiki - a lightweight and flexible operating system for tiny networked sensors. Local Computer Networks, 2004. 29th Annual IEEE International Conference on.

Dunkels, A., O. Schmidt, T. Voigt and M. Ali (2006). Protothreads: simplifying event-driven programming of memory-constrained embedded systems. Proceedings of the 4th international conference on Embedded networked sensor systems. Boulder, Colorado, USA, ACM**:** 29-42.

Ee, G. K., C. K. Ng, N. K. Noordin and B. M. Ali (2010). "A review of 6LoWPAN routing protocols." Proc. Asia-Pac. Adv. Netw **30**: 7181.

El-Hoiydi, A. (2002). Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. Communications, 2002. ICC 2002. IEEE International Conference on.

El-hoiydi, A., J.-D. Decotignie and J. Hernandez (2004). Low Power MAC Protocols for Infrastructure Wireless Sensor Networks. In Proceedings of the Fifth European Wireless Conference, Citeseer.

Elsaify, A., P. Padhy, K. Martinez and G. Zou (2007). GWMAC- A TDMA Based MAC Protocol for a Glacial Sensor Network. 4th ACM PE-WASUN 2007, Sheridan Printing Company, Inc.

Erol-Kantarci, M. and H. T. Mouftah (2011). "Wireless sensor networks for cost-efficient residential energy management in the smart grid." IEEE Transactions on Smart Grid **2**(2): 314-325.

ETSI (2009). Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW.

Evans, D. (2011). "The internet of things: How the next evolution of the internet is changing everything." CISCO white paper **1**(2011): 1-11.

Fabre, A., K. Martinez, G. Bragg, P. Basford, J. Hart, S. Bader and O. Bragg (2016). Deploying a 6LoWPAN, CoAP, low power, wireless sensor network: Poster Abstract. 14th ACM Conference on Embedded Network Sensor Systems.

Fang, S., L. D. Xu, Y. Zhu, J. Ahati, H. Pei, J. Yan and Z. Liu (2014). "An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things." IEEE Transactions on Industrial Informatics **10**(2): 1596-1605.

Farhangi, H. (2010). "The path of the smart grid." IEEE Power and Energy Magazine **8**(1): 18-28.

Farooq, M., M. Waseem, S. Mazhar, A. Khairi and T. Kamal (2015). "A Review on Internet of Things (IoT)." International Journal of Computer Applications **113**(1).

Fasolo, E., M. Rossi, J. Widmer and M. Zorzi (2007). "In-network aggregation techniques for wireless sensor networks: a survey." IEEE Wireless Communications **14**(2): 70-87.

Freescale Semiconductor Inc. (2014). Kinetis KL25 Sub-Family, Data Sheet: Technical Data**:** 59.

Garcia Macias, J. A., J. Alvarez-Lozano, P. Estrada-Martinez and E. Aviles-Lopez (2011). "Browsing the Internet of Things with Sentient Visors." Computer **44**(5): 46-52.

Gerpott, T. J. and S. May (2016). "Integration of Internet of Things components into a firm's offering portfolio – a business development framework." info **18**(2): 53-63.

Girard, L., J. Beutel, S. Gruber, J. Hunziker, R. Lim and S. Weber (2012). "A custom acoustic emission monitoring system for harsh environments: application to freezing-induced damage in alpine rock walls." Geosci. Instrum. Method. Data Syst. **1**(2): 155-167.

Goods, D., N. Hafermaas and A. Koblin. (2010). "eCloud." Retrieved August 2016, from http://www.ecloudproject.com/index.html.

Google. (2015). "Protocol Buffers." Retrieved December 2015, from https://developers.google.com/protocol-buffers/docs/overview.

Green, O., E. S. Nadimi, V. Blanes-Vidal, R. N. Jorgensen, I. M. L. D. Storm and C. G. Sorensen (2009). "Monitoring and modeling temperature variations inside silage stacks using novel wireless sensor networks." Computers and Electronics in Agriculture **69**(2): 149-157.

Guillemin, P., P. Friess, O. Vermesan, M. Harrison, H. Vogt, K. Kalaboukas, M. Tomasella, K. Wouters, S. Gusmeroli and S. Haller (2009). Internet of Things Strategic Research Roadmap, Cluster of European Research Projects on the Internet of Things (CERP-IoT).

Guinard, D., V. Trifa, F. Mattern and E. Wilde (2011). From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices. Architecting the Internet of Things. D. Uckelmann, M. Harrison and F. Michahelles, Springer Berlin Heidelberg**:** 97-129.

Gupta, P. and P. R. Kumar (2000). "The capacity of wireless networks." Information Theory, IEEE Transactions on **46**(2): 388-404.

Halkes, G. P. and K. Langendoen (2007). Crankshaft: An energy-efficient MAC-protocol for dense wireless sensor networks. Wireless Sensor Networks, Springer**:** 228-244.

Han, D. M. and J. H. Lim (2010). "Design and Implementation of Smart Home Energy Management Systems based on ZigBee." Ieee Transactions on Consumer Electronics **56**(3): 1417-1425.

Hart, J. K. and K. Martinez (2015). "Toward an environmental Internet of Things." Earth and Space Science **2**(5): 194-200.

Hasler, A., I. Talzi, J. Beutel, C. Tschudin and S. Gruber (2008). Wireless sensor networks in permafrost research-concept, requirements, implementation and challenges. 9th Int'l Conf. on Permafrost (NICOP).

Healy, M., T. Newe and E. Lewis (2008). Wireless Sensor Node hardware: A review. Sensors, 2008 IEEE.

Heinzelman, W. R., A. Chandrakasan and H. Balakrishnan (2000). Energy-efficient communication protocol for wireless microsensor networks. System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on, IEEE.

Hive. (2015). "Hive Active Heating." December 2015, from https://www.hivehome.com/.

Hong-Ling, S., H. Kun Mean, Z. Hai-ying and L. Xing (2011). Energy Efficient and Fault Tolerant Multicore Wireless Sensor Network: E2MWSN. Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on.

Horvat, G., D. Sostaric and D. Zagar (2011). User authorization system using ZigBee WSN and AVR architecture. Telecommunications Forum (TELFOR), 2011 19th.

Hui, J. and P. Thubert (2011). RFC 6282: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, IETF.

IEEE (2003). IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Std 802.15.4-2003**:** 1-670.

IEEE (2006). IEEE Standard for Information technology - Local and metropolitan area networks - Specific requirements - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs). IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)**:** 1-320.

IEEE (2011). IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)**:** 1-314.

IEEE (2012a). IEEE Standard for Local and metropolitan area networks-- Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 2: Active Radio Frequency Identification (RFID) System Physical Layer (PHY). IEEE Std 802.15.4f-2012 (Amendment to IEEE Std 802.15.4-2011)**:** 1-72.

IEEE (2012b). IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks. IEEE Std 802.15.4g-2012 (Amendment to IEEE Std 802.15.4-2011)**:** 1-252.

IEEE (2016). IEEE Standard for Low-Rate Wireless Networks. IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)**:** 1-709.

IETF. (n.d.). "The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)." from https://datatracker.ietf.org/doc/draft-clausen-lln-loadng/.

Ilapakurti, A. and C. Vuppalapati (2015). Building an IoT Framework for Connected Dairy. Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on.

Intermap Technologies (2007). NEXTMap British Digital Terrain Model Dataset Produced by Intermap. NERC Earth Observation Data Centre.

Jain, D. P. C., Reena and D. Gautam (2013). "MAXIMUM THROUGHPUT OF 10 NODES FOR IEEE 802.15.4 WPAN (WIRELESS PERSONAL AREA NETWORKS)." International Journal of Engineering Science and Technology.

Ji-chun, Z., Z. Jun-feng, F. Yu and G. Jian-xin (2010). The study and application of the IOT technology in agriculture. Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on.

Kahn, J. M., R. H. Katz and K. S. Pister (1999). Next century challenges: mobile networking for "Smart Dust". Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, ACM.

Kortuem, G., F. Kawsar, V. Sundramoorthy and D. Fitton (2010). "Smart objects as building blocks for the Internet of things." IEEE Internet Computing **14**(1): 44-51.

Kouche, A. E. (2012). Towards a wireless sensor network platform for the Internet of Things: Sprouts WSN platform. Communications (ICC), 2012 IEEE International Conference on.

Kovatsch, M. (n.d.). "Copper (Cu)." from https://addons.mozilla.org/en-GB/firefox/addon/copper-270430/.

Krebs, B. (2016). "Did the Mirai Botnet Really Take Liberia Offline?" Retrieved November 2016, from https://krebsonsecurity.com/tag/mirai-botnet/.

Kumar, V. and S. Tiwari (2012). "Routing in IPv6 over low-power wireless personal area networks (6LoWPAN): a survey." Journal of Computer Networks and Communications **2012**.

Kuorilehto, M., M. Kohvakka, J. Suhonen, P. Hämäläinen, M. Hännikäinen and T. D. Hamalainen (2008). Ultra-low energy wireless sensor networks in practice: Theory, realization and deployment, John Wiley & Sons.

Lanbo, L., Z. Shengli and C. Jun-Hong (2008). "Prospects and problems of wireless communication for underwater sensor networks." Wireless Communications and Mobile Computing **8**(8): 977-994.

Langendoen, K., A. Baggio and O. Visser (2006). Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture. Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, IEEE.

Lazarescu, M. T. (2013). "Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications." Emerging and Selected Topics in Circuits and Systems, IEEE Journal on **3**(1): 45-54.

Lennvall, T., S. Svensson and F. Hekland (2008). A comparison of WirelessHART and ZigBee for industrial applications. 2008 IEEE International Workshop on Factory Communication Systems.

Levin, S. L. and S. Schmidt (2014). "IPv4 to IPv6: Challenges, solutions, and lessons." Telecommunications Policy **38**(11): 1059-1068.

Levis, P., S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer and D. Culler (2005). TinyOS: An Operating System for Sensor Networks. Ambient Intelligence. W. Weber, J. Rabaey and E. Aarts, Springer Berlin Heidelberg**:** 115-148.

Libelium Comunicaciones Distribuidas S.L. (2012). Waspmote Datasheet.

Lin, S., J. Zhang, G. Zhou, L. Gu, J. A. Stankovic and T. He (2006). ATPC: adaptive transmission power control for wireless sensor networks. Proceedings of the 4th international conference on Embedded networked sensor systems, ACM.

Ling-ling, L., Y. Shi-feng, W. Li-yan and G. Xiang-ming (2011). The greenhouse environment monitoring system based on wireless sensor network technology. Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2011 IEEE International Conference on.

LoRa Alliance. (n.d.). "LoRa Alliance™ Technology." from https://www.lora-alliance.org/What-Is-LoRa/Technology.

Lou, P., L. Quan, Z. Zude and W. Huaiqing (2011). Agile Supply Chain Management over the Internet of Things. Management and Service Science (MASS), 2011 International Conference on.

Lymberopoulos, D. and A. Savvides (2005). XYZ: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Applications. International Conference on Information Processing in Sensor Networks (IPSN '05).

Martinez, K. and P. Basford (2011). Robust wireless sensor network performance analysis. 2011 IEEE SENSORS Proceedings.

Martinez, K., P. Basford, J. Ellul and R. Spanton (2009). Gumsense - a high power low power sensor node. 6th European Conference on Wireless Sensor Networks.

Martinez, K., P. J. Basford, D. De Jager and J. K. Hart (2012). A wireless sensor network system deployment for detecting stick slip motion in glaciers. IET International conference on Wireless Sensor Systems 2012.

Martinez, K., J. Hart, O. Bragg, A. Black, S. Bader, P. Basford, G. Bragg and A. Fabre (2016). "An Environment IoT Sensor Network for Monitoring the Environment." AGU Fall Meeting Abstracts.

Martinez, K., J. K. Hart, P. J. Basford, G. M. Bragg and T. Ward (2013). "Using Internet of Things technologies for wireless sensor networks." AGU Fall Meeting Abstracts.

Martinez, K., J. K. Hart, P. J. Basford, G. M. Bragg, T. Ward and D. S. Young (2017). "A geophone wireless sensor network for investigating glacier stick-slip motion." Computers & Geosciences **105**: 103-112.

Martinez, K., J. K. Hart and R. Ong (2004). "Environmental sensor networks." Computer **37**(8): 50-56.

Martinez, K., R. Ong and J. Hart (2004). Glacsweb: a sensor network for hostile environments. Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on.

Masinter, L. (1998). RFC 2324: Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0), IETF.

Mathews, J., M. Barnes and D. K. Arvind (2009). Low Power Free Space Optical Communication in Wireless Sensor Networks. Digital System Design, Architectures, Methods and Tools, 2009. DSD '09. 12th Euromicro Conference on.

Maxim Integrated (2015). DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal Datasheet.

McLntire, D., K. Ho, B. Yip, A. Singh, W. Wu and W. J. Kaiser (2006). The low power energy aware processing (LEAP) embedded networked sensor system. Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on.

MEMSIC Inc MICAz Wireless Measurement System.

Montenegro, G., N. Kushalnagar, J. Hui and D. Culler (2007). RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks, IETF.

Moteiv Corporation (2006). Tmote Sky Low Power Wireless Sensor Module.

Mulligan, G. (2007). The 6LoWPAN architecture. Proceedings of the 4th workshop on Embedded networked sensors. Cork, Ireland, ACM**:** 78-82.

Munaut, S. (2012). "osmo-lea6t-gps." from http://openbsc.osmocom.org/trac/wiki/osmo-lea6t-gps.

Musaloiu-E., R., C.-J. M. Liang and A. Terzis (2008). Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks. Proceedings of the 7th international conference on Information processing in sensor networks, IEEE Computer Society**:** 421-432.

Nanostack Project. (2013a). "NanoStack 6LoWPAN." Retrieved June 2013, from https://sourceforge.net/projects/nanostack/.

NanoStack Project. (2013b, 29/04/2013). "NanoStack 6lowpan." Retrieved 26/01/2016, 2016, from http://sourceforge.net/projects/nanostack/.

Natural Resources Canada. (2016). "Precise Point Positioning." from https://webapp.geod.nrcan.gc.ca/geod/tools-outils/ppp.php?locale=en.

NERC. (2014). "NE/L012405/1, Using Internet of Things technology to aid in Earth and Environmental Science Research." from http://gotw.nerc.ac.uk/list_full.asp?pcode=NE%2FL012405%2F1.

Neves, P. A. C. d. S. and J. J. P. C. Rodrigues (2010). "Internet protocol over wireless sensor networks, from myth to reality." Journal of Communications **5**(3): 189-196.

Nieminen, J., T. Savolainen, M. Isomaki, B. Patil, Z. SHelby and C. Gomez (2015). RFC 7668: IPv6 over BLUETOOTH(R) Low Energy, IETF.

Oikonomou, G. (2015). "Re: Mountains, Z1s, SubGhz ContikiMAC...". Personal Communication (E-Mail). Received 16 October 2015.

Olivarez-Giles, N. (2015). "Amazon's Dash Button Is Not a Hoax, It's Phase One." Retrieved January 2017, from https://blogs.wsj.com/personal-technology/2015/03/31/amazons-dash-button-is-not-a-hoax-its-phase-one/.

OpenWrt Project. (2004). "OpenWrt." from https://openwrt.org/.

OpenWSN Project. (2012). "Firmware/FW-16 request: FreeRTOS Support." from https://openwsn.atlassian.net/browse/FW-16.

Österlind, F. and A. Dunkels (2008). "Approaching the maximum 802.15. 4 multi-hop throughput."

PC Engines. (2008). "PC Engines alix3d1 product file." from http://www.pcengines.ch/alix3d1.htm.

Phillips. (2015). "Personal Wireless Lighting." Retrieved December 2015, from http://www.philips.co.uk/a-w/site-owner.html.

Pierce, F. J. and T. V. Elliott (2008). "Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington." Computers and Electronics in Agriculture **61**(1): 32-43.

Pinola, C. (2012). Evaluating the performance of synchronous and asynchronous media access control protocols in the contiki operating system, WORCESTER POLYTECHNIC INSTITUTE.

Pister, K. and L. Doherty (2008). TSMP: Time synchronized mesh protocol. Parallel and Distributed Computing and Systems, Orlando, Florida, USA ACTA.

Polastre, J., J. Hill and D. Culler (2004). Versatile low power media access for wireless sensor networks. Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM.

Polastre, J., R. Szewcyzk, C. Sharp and D. Culler (2004). The mote revolution: Low power wireless sensor network devices. 16th Symposium on High Performance Chips (HotChips).

Postel, J. (1981). RFC 791: INTERNET PROTOCOL, IETF.

Prayati, A., C. Antonopoulos, T. Stoyanova, C. Koulamas and G. Papadopoulos (2010). "A modeling approach on the TelosB WSN platform power consumption." Journal of Systems and Software **83**(8): 1355-1363.

Real Time Engineers Ltd. (2003). "FreeRTOS." Retrieved February 2016, from http://www.freertos.org/.

Real Time Engineers Ltd. (2012). "Official FreeRTOS Ports." Retrieved November 2014, from http://www.freertos.org/RTOS_ports.html.

RIOT. (2013). "The friendly Operating System for the Internet of Things RIOT." Retrieved September 2016, from http://www.riot-os.org/.

Rukpakavong, W., I. Phillips and L. Guan (2011). Neighbour discovery for transmit power adjustment in IEEE 802.15. 4 using RSSI. New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on, IEEE.

Sarma, S., D. L. Brock and K. ashton (2001). The Networked Physical World: Proposals for Engineering the Next Generation of Computing, Commerce & Automatic-Identification, Auto-ID Center.

Savolainen, T., J. Soininen and B. Silverajan (2013). "IPv6 Addressing Strategies for IoT." Sensors Journal, IEEE **13**(10): 3511-3519.

Savvides, A. and M. B. Srivastava (2002). A Distributed Computation Platform for Wireless Embedded Sensing.

Schwieger, K., G. Fettweis and V. S. M. Nachrichtensysteme (2004). "Multi-hop transmission: benefits and deficits." 2. Fachgesprach der GI/ITG Fachgruppe KuVS" Drahtlose Sensornetze" 26./27. Februar 2004 Universitat Karlsruhe (TH): 13.

Shammaa, S. and P. Verma (2011). Interconnection of IEEE 802.15.4 islands through free space optical communication links. Signal Processing and Communication Systems (ICSPCS), 2011 5th International Conference on.

Shelby, Z., S. Chakrabarti, E. Nordmark and C. Bormann (2012). RFC 6775: Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), IETF.

Shellby, Z., K. Harke and C. Bormann (2014). RFC 7252, The Constrained Application Protocol (CoAP), IETF.

Sheltami, T., M. Musaddiq and E. Shakshuki (2016). "Data compression techniques in Wireless Sensor Networks." Future Generation Computer Systems **64**: 151-162.

Shnayder, V., M. Hempstead, B.-r. Chen, G. W. Allen and M. Welsh (2004). Simulating the power consumption of large-scale sensor network applications. Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM.

SixXS. (2002). "Main :: SixXS - IPv6 Deployment & Tunnel Broker." from http://www.sixxs.net/main.

Smeets, C., W. Boot, A. Hubbard, R. Pettersson, F. Wilhelms, M. R. Van Den Broeke and R. S. Van De Wal (2012). "A wireless subglacial probe for deep ice applications." Journal of glaciology **58**(211): 841-848.

Song, W.-Z., R. Huang, M. Xu, A. Ma, B. Shirazi and R. LaHusen (2009). Air-dropped sensor network for real-time high-fidelity volcano monitoring. Proceedings of the 7th international conference on Mobile systems, applications, and services. Kraków, Poland, ACM**:** 305-318.

Stuntebeck, E. P., D. Pompili and T. Melodia (2006). Wireless underground sensor networks using commodity terrestrial motes. Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on.

Suping, P. and W. Birui (2013). "The Soil Moisture Content Monitoring and Irrigation System Based on IOT [J]." Journal of Agricultural Mechanization Research **7**: 106-109.

Szewczyk, R., E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring and D. Estrin (2004). "Habitat monitoring with sensor networks." Communications of the Acm **47**(6): 34-40.

Talzi, I., A. Hasler, S. Gruber and C. Tschudin (2007). Permasense: investigating permafrost with a wsn in the swiss alps. Proceedings of the 4th workshop on Embedded networked sensors, ACM.

Texas Instruments (2012). MSP430F261x MSP430F241x MIXED SIGNAL MICROCONTROLLER Datasheet.

Texas Instruments (2013a). CC112x/CC120x RX Sniff Mode (SWRA428A, Application Report).

Texas Instruments (2013b). CC112x/CC120x RX Sniff Mode Project Collateral SWRA428. CC112x_RX_Sniff_Mode.xlsx.

Texas Instruments (2013c). CC112X/CC1175 Low-Power High Performance Sub-1 GHz RF Transceivers/Transmitter User's Guide.

Texas Instruments (2015a). CC1120 High-Performance RF Transceiver for Narrowband Systems (SWRS112H Datasheet).

Texas Instruments (2015b). CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE 802.15. 4, 6LoWPAN, and ZigBee® Applications.

Texas Instruments (2016a). CC1101 Low-Power Sub-1 GHz RF Transceiver Datasheet.

Texas Instruments (2016b). CC1310 SimpleLink™ Ultra-Low-Power Sub-1 GHz Wireless MCU Datasheet (SWRS181C).

Thread Group. (n.d.). "Technology." Retrieved June 2016, from http://threadgroup.org/technology/ourtechnology.

TinyOS. (2011). "FAQ - TinyOS." Retrieved 20/11/2014, 2014, from http://tinyos.stanford.edu/tinyos-wiki/index.php/FAQ.

TinyOS. (2013). "TinyOS." Retrieved 26/01/2016, 2016, from http://www.tinyos.net/.

Tongjuan, L., L. Jun and L. Bingwu (2010). Design of intelligent warehouse measure and control system based on Zigbee WSN. Mechatronics and Automation (ICMA), 2010 International Conference on.

Tozlu, S., M. Senel, M. Wei and A. Keshavarzian (2012). "Wi-Fi enabled sensors for internet of things: A practical approach." Communications Magazine, IEEE **50**(6): 134-143.

Tu, D., S. Liu, W. Xie and Y. Zhang (2010). A Fire Monitoring System in ZigBee Wireless Network. Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2010 International Conference on.

Tubaishat, M. and S. Madria (2003). "Sensor networks: an overview." IEEE Potentials **22**(2): 20-23.

Uckelmann, D., M. Harrison and F. Michahelles (2011). An Architectural Approach Towards the Future Internet of Things. Berlin, Springer-Verlag Berlin.

Vallejo, M., J. Recas and J. L. Ayala (2015). "Proactive and Reactive Transmission Power Control for Energy-Efficient On-Body Communications." Sensors **15**(3): 5914-5934.

Vilajosana, X., P. Tuset, T. Watteyne and K. Pister (2015). OpenMote: Open-Source Prototyping Platform for the Industrial IoT. Ad Hoc Networks: 7th International Conference, AdHocHets 2015, San Remo, Italy, September 1-2, 2015. Proceedings. N. Mitton, M. E. Kantarci, A. Gallais and S. Papavassiliou. Cham, Springer International Publishing**:** 211-222.

Violet. (2010). "Nabaztag, the first Internet Connected Rabbit." Retrieved December 2015, from https://web.archive.org/web/20100209070638/http://www.violet.net/_nabaztag-the-first-rabbit-connected-to-the-internet.html.

Vučinić, M., B. Tourancheau and A. Duda (2013). Performance comparison of the RPL and LOADng routing protocols in a home automation scenario. Wireless Communications and Networking Conference (WCNC), 2013 IEEE, IEEE.

Wang, A., W. R. Heinzelman and A. P. Chandrakasan (1999). Energy-scalable protocols for battery-operated microsensor networks. 1999 IEEE Workshop on Signal Processing Systems. SiPS 99. Design and Implementation (Cat. No.99TH8461).

Ward, T. (2017). Delay Tolerance for Constrained IPv6 Networks Doctroal, University of Southampton.

Watteyne, T., X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser and K. Pister (2012). "OpenWSN: a standards-based low-power wireless development environment." Transactions on Emerging Telecommunications Technologies **23**(5): 480-493.

Wei, Y., J. Heidemann and D. Estrin (2002). An energy-efficient MAC protocol for wireless sensor networks. INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.

Werner-Allen, G., J. Johnson, M. Ruiz, J. Lees and M. Welsh (2005). Monitoring volcanic eruptions with a wireless sensor network.

Winter, T., P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur and R. Alexander (2012). RFC 6550: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, IETF.

Xiao, S., V. Sivaraman and A. Burdett (2008). Adapting radio transmit power in wireless body area sensor networks. Proceedings of the ICST 3rd international conference on Body area networks, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Xu, L., L. Rongxing, L. Xiaohui, S. Xuemin, C. Jiming and L. Xiaodong (2011). "Smart community: an internet of things application." Communications Magazine, IEEE **49**(11): 68-75.

Xu, L. D., W. He and S. Li (2014). "Internet of Things in Industries: A Survey." IEEE Transactions on Industrial Informatics **10**(4): 2233-2243.

Yibo, C., K. M. Hou, H. Zhou, H. l. Shi, X. Liu, X. Diao, H. Ding, J. J. Li and C. d. Vaulx (2011). 6LoWPAN Stacks: A Survey. 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing.

Yuasa (2005). NP SERIES VALVE REGULATED LEAD-ACID BATTERIES SHORTFORM BROCHURE.

Zanella, A., N. Bui, A. Castellani, L. Vangelista and M. Zorzi (2014). "Internet of Things for Smart Cities." IEEE Internet of Things Journal **1**(1): 22-32.

Zhang, J., Z. Huang and X. Liu (2005). "Acoustic communication in wireless sensor networks."

Zhiyong, S., L. Kui, Y. Shiping and O. Qingbo (2010). Design and implementation of the mobile internet of things based on td-scdma network. Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on.

Zhu, W., C. Dai and P. Huang (2012). "Environmental control system based on IOT for nursery pig house." Transactions of the Chinese Society of Agricultural Engineering **28**(11): 177-182.

Zimmermann, A., J. Sa Silva, J. B. M. Sobral and F. Boavida (2008). 6GLAD: IPv6 Global to Link-layer ADdress Translation for 6LoWPAN Overhead Reducing. Next Generation Internet Networks, 2008. NGI 2008.

Zolertia (2010). Z1 Datasheet.

Zolertia (2016). Zolertia RE-Mote Revision B ZOL-RM0x-B Datasheet.

Zuehlke, D. (2010). "SmartFactory—Towards a factory-of-things." Annual Reviews in Control **34**(1): 129-138.

Appendix 1. Publications

This appendix includes publications that have been authored during the course of the candidature for this PhD.

- Bragg, G. M., Martinez, K., Basford, P. J., & Hart, J. K. (2016). *868MHz 6LoWPAN with ContikiMAC for an internet of things environmental sensor network*. Paper presented at the SAI Computing Conference 2016, London.

- Martinez, K., Hart, J. K., Basford, P. J., Bragg, G. M., Ward, T., & Young, D. S. (2017). *A geophone wireless sensor network for investigating glacier stick-slip motion*. Computers & Geosciences, 105, 103-112.

- Fabre, A., Martinez, K., Bragg, G., Basford, P., Hart, J., Bader, S., & Bragg, O. (2016). *Deploying a 6LoWPAN, CoAP, low power, wireless sensor network: Poster Abstract.* Paper presented at the 14th ACM Conference on Embedded Network Sensor Systems.

- Bragg, O. M., Basford, P. J., Black, A. R., Bragg, G. M., Hart, J. K., & Martinez, K. (2016). *BRITAIN'S HIGHEST BOG: CAN WE UNLOCK ITS SECRETS?* Paper presented at the 15th International Peat Congress, Kuching, Malaysia.

- Martinez, K., Hart, J. K., Bragg, O., Black, A., Bader, S., Basford, P. J., Bragg, G. M., Fabre, A. (2016) *An Environment IoT Sensor Network for Monitoring the Environment.* AGU Fall Meeting Abstracts 2016.

- Martinez, K., Hart, J. K., Basford, P. J., Bragg, G. M., & Ward, T. (2013). *Using Internet of Things technologies for wireless sensor networks*. AGU Fall Meeting Abstracts 2013.

868MHz 6LoWPAN with ContikiMAC for an Internet of Things Environmental Sensor Network

G. M. Bragg, K. Martinez, P. J. Basford and J. K. Hart
University of Southampton
Southampton, UK
{g.bragg, km, pjb}@ecs.soton.ac.uk, jhart@soton.ac.uk

***Abstract*— When deploying an Internet of Things Environmental Sensor Network (ESN), the communications range of nodes becomes a critical factor when attempting to cover a large geographic area. The 2.4 GHz radios that are commonly used for Wireless Sensor Networks do not have sufficient range for ESN applications. We investigate the performance of an 868MHz CC1120-based sensor node that incorporates a Zolertia Z1 and runs the Contiki operating system with multi-hop 6LoWPAN networking using the ContikiMAC radio duty cycling protocol. Comparisons with the commonly-used CC2420 2.4GHz radio, in terms of latency and throughput, show that the CC1120 can offer significant performance benefits for certain deployment scenarios. Brief details of an ongoing deployment are presented.**

Keywords—6LoWPAN; ESN; WSN; IoT; Contiki; ContikiMAC; Sub-GHz; 868 MHz

I. INTRODUCTION

Environmental Sensor Networks (ESN) [1] have emerged from the Wireless Sensor Network (WSN) area with a specific set of applications and hence technical challenges. Among the difficulties faced, the issue of integrating different hardware on the low power radio network is one aspect that needs addressing. Several Internet of Things (IoT) WSNs exploit the utility of an IoT operating system, such as Contiki [2], to provide standards-compliant networking in a multitasking environment with energy-efficient radio duty cycling protocols [3]. Many contemporary IoT WSNs use short-range (up to ~300 m) low-power 2.4 GHz radios that support 802.15.4 with a multi-hop capable 6LoWPAN networking stack [4,5]. This allows for end-to-end IPv6 connectivity to the sensor nodes, and the standardisation of protocols allows different hardware designs to interoperate in one network.

In contrast, recent ESNs tend to use custom communications protocols with a wide range of sub-gigahertz radios to provide communications over longer ranges (up to several kilometres) in harsh environments. The Glacsweb project has used 151 MHz radios for communications with sub-glacial sensor probes and 868 MHz radios for surface node communications [6]. Transmit-only 30 MHz radios have also been used for sub-glacial monitoring [7]. The Permasense project has made extensive use of 868 MHz radios for monitoring in the Alps [8-10] and transmit-only 433 MHz radios have been used for wildfire detection and alerting [11].

This research was funded by the Natural Environmental Research Council and the Engineering and Physical Sciences Research Council.

Of these different frequency bands the 868 MHz band, or the 915 MHz band in North America, shows potential for use in IoT ESNs as it has good propagation characteristics that enable communications over several kilometres and is allowed for by the 802.15.4 standard [12]. Prior to 2012 the 868 MHz band was limited to 20 kb/s with simple modulation which was not consistent with the 915 MHz band. The 802.15.4G amendment introduced the 50 kb/s and 200kb/s data rates for both bands [13]. Low-cost radios that support 802.15.4G, such as the CC1120 [14], are also now readily available but not yet fully supported by IoT operating systems.

As part of the Mountain Sensing project (http://mountainsensing.org/), a platform consisting of a Zolertia Z1 sensor mote and a CC1120 sub-gigahertz radio running the Contiki operating system, the MS1, was produced. In this paper, we calculate the theoretical behaviour of this platform operating at 50 kb/s with a multi-hop 6LoWPAN networking stack and the ContikiMAC duty cycling protocol, test these expectations and present brief details of an ongoing deployment.

II. EXPECTED PERFORMANCE

The theoretical MAC-layer throughput can be estimated using the method used in [15] for establishing the throughput of nonfragmented packets for a 250 kb/s 2.4 GHz 802.15.4 radio in a lightly loaded network. Using a value of unity for $P_{inactive}$, the probability that the channel is available for communications, provides a means to establish the potential peak throughput which is useful for comparative purposes.

Table I shows the parameters for a CC1120 operating at 50 kb/s and results in a theoretical throughput of 24.45 kb/s for a payload of 81 bytes, the largest payload achievable in Contiki before a packet is fragmented. The multi-hop MAC-layer throughput can be estimated by dividing the single-hop throughput by the number of hops [15, 16].

TABLE I. Packet transmission time parameters for the CC1120 operating at 50 kb/s with a $P_{inactive}$ of one. Parameters have the same definitions as in [15]

Parameter	Symbols	Time (s)
CSMA-CA	250	0.0050
TX Packet	1064	0.0213
ACK Turnaround	35	0.0007
TX ACK	96	0.0019
LIFS	40	0.0008
Sum	**1485**	**0.0297**

A. Latency

The latency of a communications link is also an important metric in assessing its performance. The round-trip latency can be modelled using (1) where n_{BT} and n_{BR} are the number of bytes of payload to be transmitted and received respectively. n_{PT} and n_{PR} are the number of packets that the data will be split into to be transmitted and received respectively. t_{BS} is the amount of time taken to load, transmit, and read each byte. t_{PO} is the constant overhead associated with the transmission of each packet and includes radio state transition time, ACK handling time, 802.15.4 header handling time, microcontroller interrupt transition time and the time taken to transmit the PHY header. t_{CA} is the channel access time.

$$t_{Latency} = ((n_{BT} + n_{BR}) \times t_{BS}) + ((n_{PT} + n_{PR}) \times t_{PO}) + (2 \times t_{CA}) \quad (1)$$

The increased latency caused by using an RDC can be calculated with and added to $t_{Latency}$. For ContikiMAC with the CC1120, the overhead can be calculated with (2) where n_{RTX} and n_{RRX} are the average numbers of retransmissions for transmitted and received packets respectively. t_{TO} is the time taken for the radio to transition into the transmit state. t_i is the ContikiMAC inter-packet interval. n_{OB} is the number of bytes of per-packet overhead that have to be transmitted and includes sync and preamble bytes. n_{HB} is the number header bytes that are loaded, transmitted and read for each packet. R_{BR} is the transmit bitrate of the radio. n_{BT1} and n_{BR1} are the number of bytes that need to be retransmitted for transmission and reception respectively. For non-fragmented packets, where n_{PT} and n_{PR} are both equal to one, n_{BT1} and n_{BR1} will be the entire payload whereas for fragmented transmissions they are the number of bytes in the first packet. For direct communications between a border router and sensor node n_{RRX} may be zero as it is common for the border router not to be duty cycling the radio to improve performance.

$$t_{RDCOH} = (n_{RTX} + n_{RRX}) \times \left(t_{TO} + t_i + \frac{8(n_{OB} + n_{HB})}{R_{BR}}\right) + \frac{8((n_{BT1} \times n_{RTX}) + (n_{BR1} \times n_{RRX}))}{R_{BR}} \quad (2)$$

Multi-hop latency can be estimated for a network where each hop uses the same type of sensor node by multiplying the result of (1) by the number of hops. For multi-hop calculations involving ContikiMac, the latency of the first hop will need to be calculated separately from the latency of the subsequent hops as n_{RRX} is zero for the first hop only.

The latency of a connection can be determined using the Ping6 command from a Linux computer. Table II shows the parameters used to calculate the expected latencies to a node one hop away where no duty cycling protocol is used and to a node one hop away where the ContikiMAC duty cycling protocol is used. The parameters used to calculate the expected latency for a two-hop scenario where the ContikiMAC duty cycling protocol is used are also presented.

B. Energy

The average receive energy can be estimated using the amount of time that the radio is expected to be in receive and the typical receive current from the radio's data sheet. For the default ContikiMAC settings of an 8 Hertz channel check rate and two channel clear assessments per cycle, the radio will be turned on 16 times each second. For the CC1120, the total on-time is 10.08 ms in every second giving a duty cycle of approximately 1%. In the high-performance mode, the CC1120 draws 22 mA at 3.3 V giving a power consumption of 0.73 mW. In its low power mode, the CC1120 draws 17mA for a power of 0.56 mW. For comparison, the CC2420, a commonly used 2.4 GHz radio, has a receive duty cycle of less than 1% [3] and draws 0.12 mW [17].

TABLE II. Values used to calculate the expected latency of the CC1120 without an RDC and with the ContikiMAC duty cycling protocol. The parameters used to calculate the two-hop ContikiMAC latency.

Parameter	**CC1120**			
	No RDC	***ContikiMAC (Average)***	***ContikiMAC (Minimum)***	***ContikiMAC (Two-hop)***
n_{BT}	94	94	94	94
n_{BR}	93	93	93	93
n_{PT}	2	2	2	2
n_{PR}	2	2	2	2
t_{BS} (ms)	0.172	0.172	0.172	0.172
t_{PO} (ms)	7.765	7.765	7.765	7.765
t_{CA} (ms)	0	34	34	34
t_{RDCOH} (ms)	0	87.480	0	303.040
n_{RTX}	-	4.7	4.7	4.7
n*rtx*	-	0	0	4.7[a]
Latency (ms)	**63.27**	**218.75**	**131.27**	**656.58**

[a.] nrtx is zero for the first hop so tRDCOH must be calculated separately for each hop

The maximum retransmission energy overhead can be calculated in a similar manner by multiplying the amount of time it takes to transmit a full-sized packet by the TX power and the average number of times that the packet is transmitted beyond the first. For the CC1120, a full-sized packet will take 21.60 ms to transmit at a total energy cost of 3.21 mJ at 14 dBM. This gives an average retransmission overhead of 15.09 mJ per packet. In comparison, the CC2420 takes 4.32 ms to transmit the same packet giving a transmit energy of 0.25mJ per packet. As expected, the transmission overhead is significantly greater for the CC1120 due to its increased transmit power and time spent transmitting.

III. METHODS & RESULTS

A. CC1120 Contiki Driver

At the time this work was carried out Contiki did not include a driver for the CC1120. As such a driver for the CC1120 was developed. Compatibility with the ContikiMAC duty cycling protocol was included and Table III shows the settings required for ContikiMAC to function with the CC1120. Not all settings that required modification were exposed as changeable parameters so minor modifications were required to the ContikiMAC source files. Table IV details the definitions that required modification and the values required.

TABLE III. ContikiMAC timing constaints for the CC1120 operating at 50 kb/s. Each parameter has the same definition as in [3] and was calculated with reference to [14].

Parameter	Description	Time (ms)
t_a	ACK turnaround time.	0.7
t_d	ACK detection time.	0.8
t_i	Inter-packet interval.	4.5
t_c	Successive CCA interval.	5.1
t_r	Time for a stable RSSI.	0.5[b]
t_s	Shortest packet TX time.	6.5
t_l	Longest packet TX time.	21

b. Calculated with reference to [18]

While the driver attempts to respect the 802.15.4 standard, limitations in the speed of processing interrupts and transitioning from pre-emptive execution to co-operative execution on the MS1 mean that it is not possible to start transmission of an ACK within the 120 µs required by the standard. Fig 1 shows that shortest time that could be achieved with the MS1 platform was 700 ms.

B. Throughput

The single-hop unidirectional MAC-layer throughput between two nodes can be determined by measuring how long it takes to transmit a set number of full-size nonfragmented packets at the fastest rate that does not result in packet loss. This was achieved by repeatedly transmitting UDP packets with a payload of 81 bytes and observing a transmission indication pin on an oscilloscope. A throughput of 19.78 kb/s for the CC1120 was determined with this method.

TABLE IV. Internal ContikiMAC parameters that need to be modified for sub-gigahertz operation. These settings are found in contikimac.c.

Internal Setting	Related Constraint	Value
SHORTEST_PACKET_SIZE	t_s	36 B
CCA_CHECK_TIME	t_r	0.63ms
CCA_SLEEP_TIME	t_c	4.8ms
INTER_PACKET_INTERVAL	t_i	0ms
LISTEN_TIME_AFTER_PACKET_DETECTED	T_{awake}	50ms

This is 4.67 kb/s below the expected value but the inter-packet interval observed during testing was 5.8ms, 5ms greater than the value used to calculate the expected throughput. Recalculating the expected throughput using this value for LIFS results in an expected throughput of 20.84 kb/s, which is similar to the achieved throughput.

The Zolertia Z1 includes a CC2420 radio which can be used, although it was not needed in the deployment. The same test was carried out with the CC2420 and the throughput was found to be 87.28 kb/s. This is 33.52 kb/s below the throughput expected from [15]. Like the CC1120 tests, the observed inter-packet interval was more than five milliseconds greater than expected. This discrepancy is likely to be due to process overheads within Contiki.

C. Latency

The average and minimum round-trip latencies can be determined by repeatedly sending a Ping6 ICMP request from a Linux host to a node one hop from the border router once per

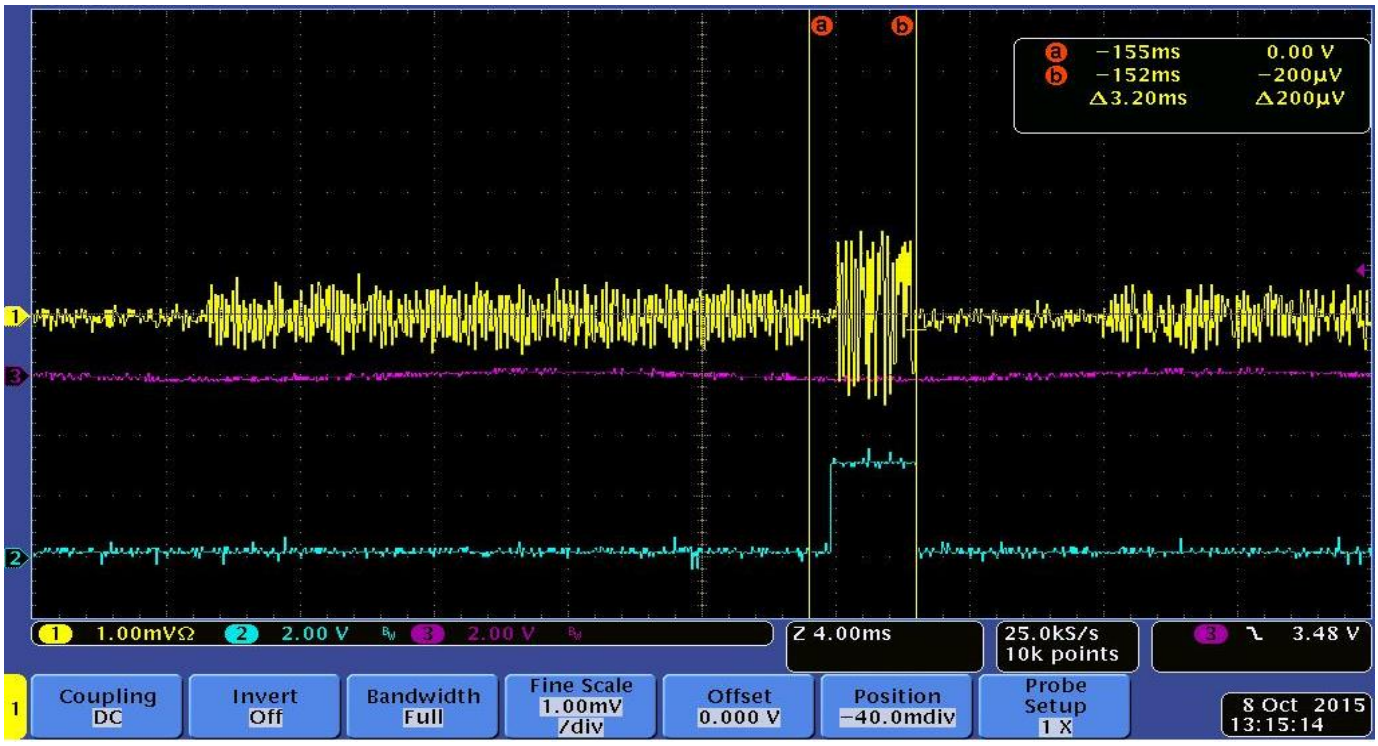


Fig. 1. CC1120 ACK turnaround time. The yellow trace shows the presence of RF signals while the blue trace shows when the node sending the ACK, the greater magnitude yellow trace, is transmitting.

second over an extended period of time. The test setup consisted of an Instant Contiki virtual machine connected to the border router by a 6-Slip interface over a USB-to-serial connection. Any pings sent to the destination node will include the latency of this connection in their results. The latency of this connection was measured by pinging the border router and found to be 21 ms.

Table V shows the results of running 43200 pings for three different scenarios for both radios. The results for the single-hop CC1120 scenarios are greater than the expected values by a similar amount that is not proportional to the latency. This implies that the discrepancy is systematic and likely to be due to the time taken to process the ICMP packets within Contiki. The result for the two-hop scenario with ContikiMAC is also greater than expected and like the single hop scenarios, this is likely to be due to processing overheads in Contiki.

The results for the CC2420 demonstrate that a significant proportion of the latency exhibited on 6LoWPAN connections is not related to the throughput of the radio as the results for the CC1120, operating at 50 kb/s, are significantly greater than one fifth of the results for the CC2420, operating at 250 kb/s.

IV. DEPLOYMENT

A deployment consisting of two dedicated routing nodes and six sensor nodes was carried out in the Cairngorm Mountains in Scotland. The nodes were deployed in two clusters in a one kilometre square area as shown by Fig. 2. The border router was situated in an estate office 3.5 km North West of Router 1 where there is permanent power and Internet connectivity. Table VI details the function of each node and the types connected sensors.

Line-of-sight RF propagation modelling for a CC2420-based 2.4 GHz deployment was carried out using Viewscheds in ArcMap. This showed that a deployment with comparable coverage would require at least 25 routing nodes with a total of 18 hops to the first closest node and 25 hops to the farthest node. Table VII shows the performance characteristics for the deployed network and the modelled 2.4 GHz deployment.

TABLE VI. Ping6 results from 43,200 echo requests for six different scenarios. The minimum and average results include the latency of the 6-Slip connection. The delta from the expected values is presented where appropriate.

Test	Minimum Latency (ms)	Average Latency (ms)	Standard Deviation (ms)	Δ from Calculated Values	
				Min (ms)	Avg (ms)
CC1120 (No RDC)	102.60	105.02	15.51	18.33	-
CC1120 (ContikiMAC)	169.50	258.33	83.42	17.23	18.58
CC2420 (No RDC)	41.86	43.00	1.73	-	-
CC2420 (ContikiMAC)	51.04	209.77	198.35	-	-
CC1120 2-hop (ContikiMAC)	-	740.13	324.23	-	62.55
CC2420 2-hop (ContikiMAC)	-	431.67	508.90	-	-

TABLE V. List of deployed nodes with their name, function and details of connected sensors.

Node	Function	Sensors
Estate	Border Router	-
Router 1	Routing Node	-
Router 2	Routing Node	-
Router 3	Routing/Sensing	Temperature spider & soil moisture
Turf	Sensing	Temperature & strain gauge chain
Lochan	Sensing	Water level & rain gauge
Hummock	Sensing	Temperature & strain gauge chain
Peat	Sensing	Temperature spider & soil moisture
Stream	Sensing	Water level & rain gauge

V. CONCLUSIONS

Our ongoing deployment demonstrates that the combination of 802.15.4G-compliant 868 MHz radio with multi-hop 6LoWPAN networking offers an effective alternative to 2.4 GHz-based networks for ESNs where nodes are spread over a large geographic area. We have demonstrated that low-power radio communications can successfully facilitate single-hop IPv6-based networking at ranges in excess of 3.5 km, over ten times what is possible with low-power 2.4 GHz radios. The deployment could support a wider geographic spread of sensor nodes than it does currently. However, it does show that despite having a lower single-hop throughput and greater single-hop latency than a network consisting of CC2420-based nodes, our CC1120-based nodes can provide a greater network throughput, lower total latency and lower energy profile in some deployment scenarios.

While ContikiMAC can operate successfully with the CC1120 and provides a comparably low duty cycle to that achievable with the CC2420, the retransmission energy overhead associated with the duty cycling protocol is substantial because the transmission energy consumption is significantly greater than the receive energy consumption. Further work is required to determine whether ContikiMAC is the most efficient duty cycling protocol for sub-gigahertz ESNs.

ACKNOWLEDGMENT

The authors would like to thank Wildland Ltd. for permitting the use of the Glen Feshie Estate for this research. We would also like to thank our partners in the Mountain Sensing Project at the Department of Geography at the University of Dundee for providing logistical support.

TABLE VII. MAC-layer throughput and post duty cycling latency to the sensor nodes closest to and farthest from the border router for. Data is shown for both the actual CC1120 –based deployment and the modelled CC2420-based deployment. The multi-hop throughput is calculated from the single-hop throughputs determined in III.

Radio	Number of hops to Nearest Node	Throughput to Closest Node (kb/s)	Average Latency to Closest Node (ms)	Number of hops to Farthest Node	Throughput to Farthest node (kb/s)	Average Latency to Farthest node (ms)
CC2420 (2.4 GHz)	18	5.13	4203.97	25	3.49	5757.27
CC1120 (868 MHz)	1	19.78	258.33	3	6.59	1221.93

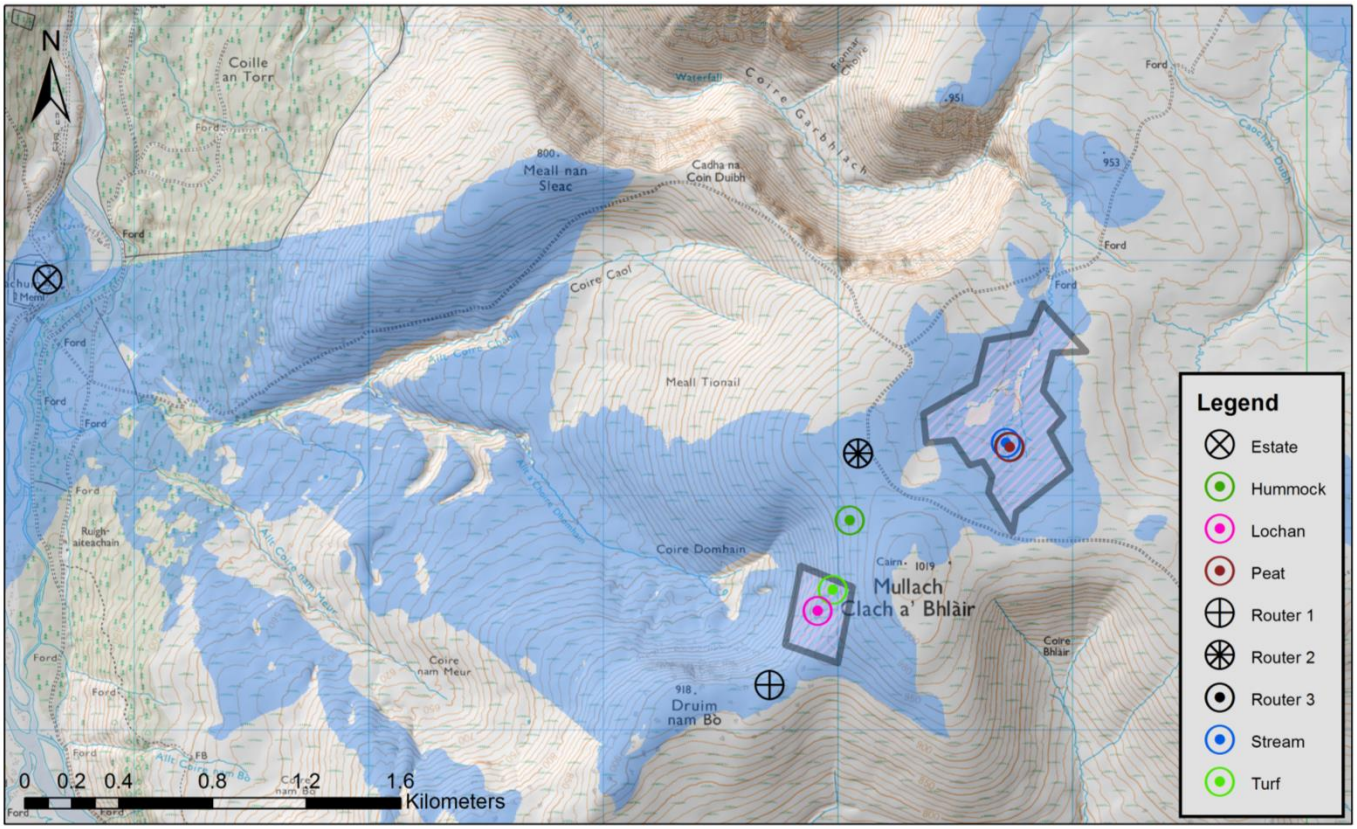


Fig. 2. Modelled coverage for the 868 MHz ESN deployment in the Cairngorm Mountains in Scotland. The outlined areas indicate the study areas. Estimated radio coverage is shown by the shaded (blue) areas. At this scale, Router 3 is co-sited with the Peat sensor node. DTM data from [19] and Crown Copyright Ordnance Survey.

REFERENCES

[1] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," *Computer*, vol. 37, pp. 50-56, 2004.

[2] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, 2004, pp. 455-462.

[3] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Swedish Institute of Computer Science, Sweden, Technical Report2011.

[4] Belcredi, Gonzalo; Modernell, Pablo; Sosa, Nicolas; Steinfeld, Leonardo; Silveira, Fernando, "An implementation of a home energy management platform for Smart Grid," in Innovative Smart Grid Technologies Latin America (ISGT LATAM), 2015 IEEE PES , vol., no., pp.270-274, 5-7 Oct. 2015

[5] D. Tu, S. Liu, W. Xie, and Y. Zhang, "A Fire Monitoring System in ZigBee Wireless Network," in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2010 International Conference on*, 2010, pp. 48-51.

[6] K. Martinez, P. J. Basford, D. De Jager, and J. K. Hart, "Poster Abstract: Using a heterogeneous sensor network to monitor glacial movement," presented at the 10th European Conference on Wireless Sensor Networks, 2013.

[7] C. Smeets, W. Boot, A. Hubbard, R. Pettersson, F. Wilhelms, M. R. Van Den Broeke, *et al.*, "A wireless subglacial probe for deep ice applications," *Journal of glaciology*, vol. 58, pp. 841-848, 2012.

[8] J. Beutel, S. Gruber, A. Hasler, R. Lim, A. Meier, C. Plessl, *et al.*, "PermaDAQ: A scientific instrument for precision sensing and data recovery in environmental extremes," in *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*, 2009, pp. 265-276.

[9] L. Girard, J. Beutel, S. Gruber, J. Hunziker, R. Lim, and S. Weber, "A custom acoustic emission monitoring system for harsh environments: application to freezing-induced damage in alpine rock walls," Geosci. Instrum. Method. Data Syst., vol. 1, pp. 155-167, 2012.

[10] A. Hasler, I. Talzi, J. Beutel, C. Tschudin, and S. Gruber, "Wireless sensor networks in permafrost research-concept, requirements, implementation and challenges," in 9th Int'l Conf. on Permafrost (NICOP), 2008.

[11] M. T. Lazarescu, "Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 3, pp. 45-54, 2013.

[12] IEEE, "IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," in *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, ed, 2011, pp. 1-314.

[13] IEEE, "IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks," in *IEEE Std 802.15.4g-2012 (Amendment to IEEE Std 802.15.4-2011)*, ed, 2012, pp. 1-252.

[14] Texas Instruments (2015, July). "*CC1120 High-Performance RF Transceiver for Narrowband Systems*" [Online]. Available: http://www.ti.com/lit/ds/symlink/cc1120.pdf [January 15, 2016].

[15] D. P. C. Jain, Reena, and D. Gautam, "MAXIMUM THROUGHPUT OF 10 NODES FOR IEEE 802.15.4 WPAN (WIRELESS PERSONAL AREA NETWORKS)," *International Journal of Engineering Science and Technology*, 2013.

[16] F. Österlind and A. Dunkels, "Approaching the maximum 802.15. 4 multi-hop throughput," 2008.

[17] Texas Instruments (2014). "*CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*" [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf [Januaru 15, 2016].

[18] Texas Instruments (2013). "*CC112X/CC1175 Low-Power High Performance Sub-1 GHz RF Transceivers/Transmitter User's Guide*" [Online]. Available: http://www.ti.com/lit/ug/swru295e/swru295e.pdf [January 15, 2016].

[19] Intermap Technologies, "NEXTMap Britain: Digital terrain mapping of the UK," NERC Earth Observation Data Centre, 2007. Available: http://badc.nerc.ac.uk/view/neodc.nerc.ac.uk__ATOM__dataent_1165833444211836 [January 15, 2016].

Computers & Geosciences 105 (2017) 103–112

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

A geophone wireless sensor network for investigating glacier stick-slip motion

Kirk Martinez[a,*], Jane K. Hart[b], Philip J. Basford[a], Graeme M. Bragg[a], Tyler Ward[a], David S. Young[a]

[a] Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
[b] Geography and Environment, University of Southampton, Southampton SO17 1BJ, UK

ARTICLE INFO

Keywords:
Environmental sensor network
Geophones
Glacier velocity

ABSTRACT

We have developed an innovative passive borehole geophone system, as part of a wireless environmental sensor network to investigate glacier stick-slip motion. The new geophone nodes use an ARM Cortex-M3 processor with a low power design capable of running on battery power while embedded in the ice. Only data from seismic events was stored, held temporarily on a micro-SD card until they were retrieved by systems on the glacier surface which are connected to the internet. The sampling rates, detection and filtering levels were determined from a field trial using a standard commercial passive seismic system. The new system was installed on the Skalafellsjökull glacier in Iceland and provided encouraging results. The results showed that there was a relationship between surface melt water production and seismic event (ice quakes), and these occurred on a pattern related to the glacier surface melt-water controlled velocity changes (stick-slip motion). Three types of seismic events were identified, which were interpreted to reflect a pattern of till deformation (Type A), basal sliding (Type B) and hydraulic transience (Type C) associated with stick-slip motion.

1. Introduction

The motion of glaciers is highly dependent on the behaviour of meltwater (generated at the glacier surface by atmospheric melting) which can influence the rate at which glaciers move by creep (Duval, 1977), reduce friction to allow basal sliding (Weertman, 1957; Iken et al., 1983), and deform underlying sediments (Boulton and Jones, 1979). Recent studies of continuous measurements of glacier velocities by GPS have indicated that ice motion is commonly episodic and it has been proposed that this reflects stick-slip motion (Bahr and Rundle, 1996; Fischer and Clarke, 1997; Tsai and Ekstrom, 2007; Wiens et al., 2008). Such a process would generate microseismic events (ice quakes) at the glacier bed, which could be measured by seismometers (Weaver and Malone, 1979; Anandakrishnan and Bentley, 1993; Metaxian et al., 2003; Smith, 2006). However, other sources of ice quakes within the glacial environment include ice calving (Qamar, 1988; O'Neel and Pfeffer, 2007), crevassing (Neave and Savage, 1970; Deichmann et al., 2000) and basal fracture (Walter et al., 2008).

Wireless sensor networks which are designed to be deployed for earth-science research have brought low power networking to remote areas (Chong and Kumar, 2003; Martinez et al., 2004; Hart and Martinez, 2006; Gehrke and Liu, 2006; Oliveira and Rodrigues, 2011; Huang et al., 2015). These environmental sensor networks have enabled a wider range of areas to be monitored for fundamental science and hazard warnings (Szewczyk et al., 2004; Delin et al., 2005; Werner-Allen et al., 2005; Hasler et al., 2008; Xu et al., 2014).

Most current commercial passive seismic systems require large power supplies and do not provide "live" data. Surface based deployments also require regular manual re-levelling, due to surface melt. In contrast, we required a long-term, low power automatic system housed in a borehole in order to avoid re-levelling, lessen the effects of noise from the glacier surface and insure a direct contact with the ice. We have developed a low power borehole geophone as part of a wireless sensor network, which can be used alongside GPS, subglacial wireless probes (Martinez et al., 2004), temperature and time lapse camera data (Young et al., 2015) to monitor a range of glacial processes. One advantage of the sensor network is its ability to send data back to a server in the UK daily, which provides researchers with a "live" feed via an internet connection. Due to the potentially high levels of data produced from continuous recording, we used an event detection system, so that the system only stored and communicated data related to the ice quakes (events). This new system, which is the first of its kind, consists of a small, borehole based, low power, event detection system providing a "live" data stream. The design has the potential to

* Corresponding author.
E-mail address: km@ecs.soton.ac.uk (K. Martinez).

http://dx.doi.org/10.1016/j.cageo.2017.05.005
Received 15 March 2016; Received in revised form 1 April 2017; Accepted 8 May 2017
Available online 10 May 2017

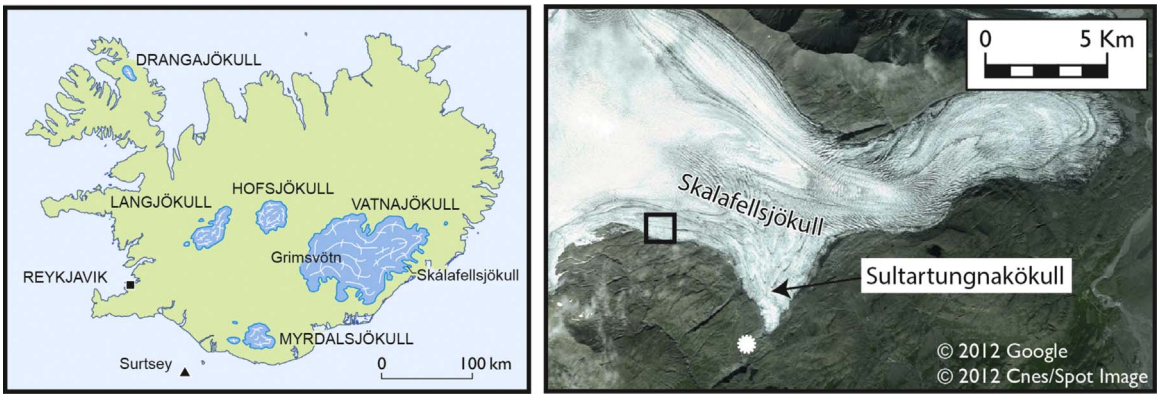


Fig. 1. Location of study area at Skálafellsjökull, b) Image of study area, square indicates main Glacsweb study site, star indicates camera location.

allow geophone sensing over longer time periods while providing researchers with frequent updates and an understanding of the state of the hardware. It could be used in other seismic deployments where long-term monitoring of short-term events is required.

The aims of this paper are:

- To discuss the design and development of the geophone system and its integration into a heterogeneous environmental sensor network.
- To report the findings from a field trial using a commercial passive seismic recording device on the glacier surface. This was undertaken to detect basal ice quakes and also investigate the required sampling rates, detection and filtering for our custom system.
- To present analysis of data generated from the new system to study the timing and nature of microseismicity associated with daily stick-slip motion at an Icelandic glacier.

2. The field site

Skalafellsjökull, Iceland (Fig. 1a) is an outlet glacier of the Vatnajökull icecap which rests on Upper Tertiary grey basalts. This glacier has an area of approximately 100 km^2 and is 25 km in length (Sigurðsson, 1998). The study site was located on the glacier at an elevation of 792 m a.s.l., where the ice was flat and crevasse free. The subglacial meltwater in this area emerged 3 km away at the southern part of the glacier (known as the Sultartungnajökull tongue, Fig. 1b).

The Glacsweb sensor network was deployed at Skalafellsjökull, Iceland (Fig. 1a) (2008–2013) and provided the ideal infrastructure for this research. This consisted of multiple heterogeneous nodes which have been developed during several years of continuous deployments (Martinez et al., 2009, 2012; Hart et al., 2006). Fig. 2 illustrates the design of the wireless sensor network system in 2012/3. A set of sensor nodes on/in the glacier used appropriate radio frequencies (868 MHz surface, 173 MHz ice) to communicate to a base station that uses either Wi-Fi or GPRS to send the data to a server hosted in the Amazon Web Services cloud (Martinez and Basford, 2011). As well as acting as routing nodes the gateways include a meteorological station, GPS, cameras and other diagnostic sensors. There were also four standalone dGPS units recording ice velocity 2012/3 and a time-lapse camera monitoring river discharge (Young et al., 2015).

Ground penetrating radar (GPR) surveys and borehole measurements have shown that the glacier at the study site ranges from 0 to 200 m in thickness. The glacier rests on a fine grained till, with a series of active till thrust sheets approximately 5 m thick, moving at 3 m per year throughout the year. The water content (calculated from GPR) of the glacier is very low (0.5%), but surface meltwater moves rapidly though englacial crevasses and moulins to the glacier bed (Hart et al., 2015). Data from the wireless Glacsweb probes show the water pressure in the till is high during the summer, but fluctuates during the winter depending on meltwater inputs.

Weather data were obtained from the base station and, during periods of mechanical failure, from a transfer function applied to data from the neighbouring Icelandic meteorological station at Höfn. Daily surface melt was calculated by the degree day algorithm (PDD) (Braithwaite, 1985; Hock, 2003), using degree day factors for Satujökull, Iceland (Johannesson et al., 1995), 5.6 mm d^{-1} $°C^{-1}$ for snow and 7.7 mm d^{-1} $°C^{-1}$ for ice. Albedo was calculated from the MODIS data, using the threshold between ice and snow to be 0.45, on a 30×30 m grid ASTER DEM.

3. Field trial using a commercial system in 2011

Six commercial geophones were installed on the glacier surface. Each station consisted of a 4.5 Hz 3-component geophone with pre-amplifier, powered by lead-acid battery and two 20 W solar panels. Data were recorded by ISSI (Integrated Seismic Systems International) SAQS (Stand Alone Quake Systems) systems units capable of sampling at up to 24 kHz. The units recorded 24 bit data per channel at a sample rate of 1 kHz. The systems were placed 90 m apart from so the spacing

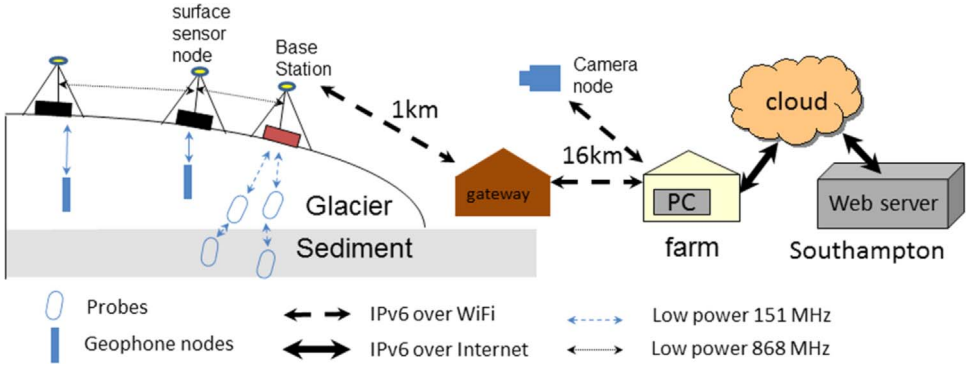


Fig. 2. Sensor network deployed in Iceland. Showing the geophone nodes integrated into the radio network.

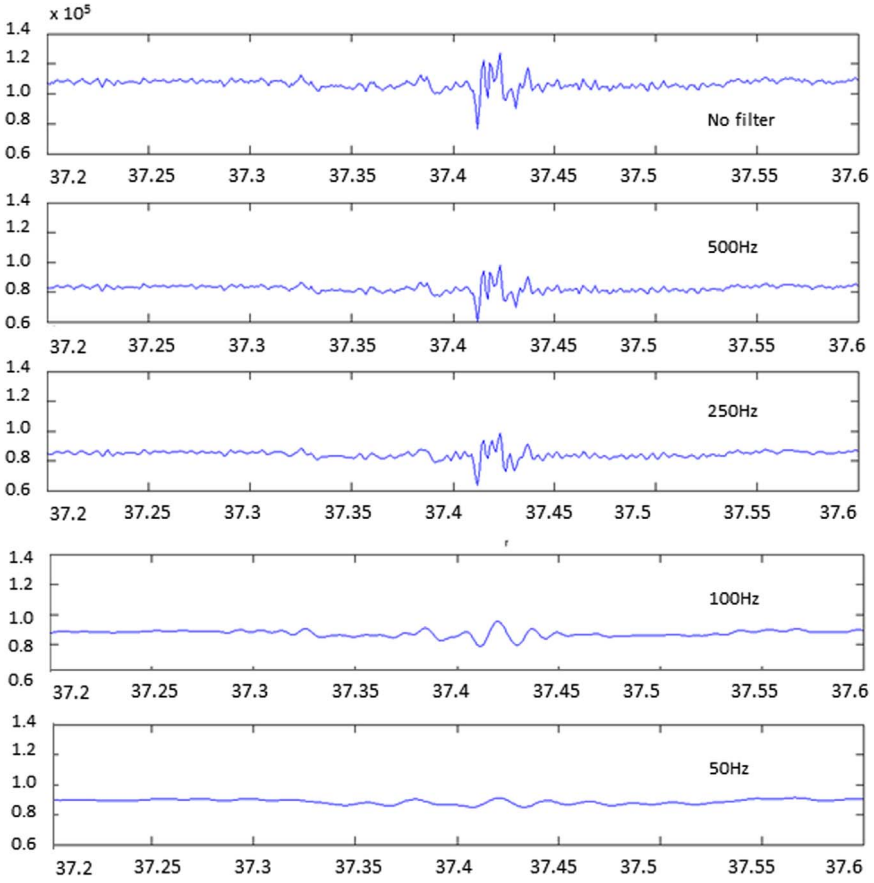


Fig. 3. Results of applying low pass filters to z component from the ISSI SAQS system.

between them was approximately equivalent to the ice thickness in the deployment region. The geophones were mounted on white plywood boards 0.5 m square which were dug down into the ice approximately 0.3 m and placed level and covered with ice chips. They required re-levelling and re-alignment with true north each day (Hart et al., 2011). Recording was carried out over 8 days, with some periods where one or two stations were not active but overall data was collected during 76% of the time available for the field trial.

Basal signals are typically marked by an initial P wave followed by the S wave. The approximate velocities of these waves in ice are well known (3600 m s^{-1}, 1800 m s^{-1}) (Röthlisberger, 1972), and the time difference between the arrivals can be used to calculate the ice quake source (Smith, 2006).

3.1. Sampling parameters and filtering

Our new geophone system required a sampling rate sufficient to capture the events (basal ice quakes) but not so high that it created unnecessary data volumes. Filtering can be used to separate the signal from the background in a data sample from the SAQS system. Fig. 3 shows the impact of low pass filtering at different frequencies (500, 250, 100 and 50 Hz) along with the unfiltered signal of the Z component of one of the units. The 500 and 250 Hz low pass filters still preserve the event, but it is reduced at 100 and 50 Hz. This shows that the new geophone nodes could use a sampling rate of 500 Hz with a low pass filter at 250 Hz (to satisfy the Nyquist criterion) without losing too much information. In order to detect an event in the SAQS system, we used a technique (Walter et al., 2008) which compares the root-mean-square (rms) values of a short term average (STA) and long term average (LTA). When the ratio STA/LTA is greater than a threshold an event is recorded. This threshold is adjusted through observation in order to pick out the P wave using SDX (Seismic Data eXplorer) software (http://doree.esc.liv.ac.uk:8080/sdx). It was found that a 10 Hz and 100 Hz band pass filter, with a threshold ratio around 10 and STA and LTA periods of 0.005 and 0.1 s respectively picks out the P wave reasonably well. The P wave pick times are generally located part way through the P wave up kick. Trials showed that lower thresholds and adjustments in the LTA/STA created more false events.

3.2. Basal events

Basal events are assumed to occur at ice depth or further from stations, and be detectable by several stations. There were at least 5 basal events that occurred on at least 4 stations. These occurred within the virtually continuous recording period from 17.00 h day of the year (DOY) 215–12.00 h DOY 219. Fig. 4 shows an example of the event 16:43:37 on DOY 218. The average difference in time between the P and S waves during this event was 0.03 s (s.d. 0.009), and for the five basal events 0.039 s (s.d. 0.014). The difference between the P and S waves gives an estimated distance of 72–205 m which is similar to depth recorded from the GPR survey (up to 200 m). This provides evidence that they were basal events.

Fig. 5 shows the air temperature, rainfall, recording time and seismic events during the fieldwork period. The seismic events occur during days of relatively high temperature (and low rainfall), over 8 °C, and very soon after peaks in temperature (9 min to 3 h 28 min). If these events represent stick-slip events then they show that the water melted from the glacier surface travels quickly to the base of the glacier where it lubricates the bed allowing it to rapidly slide.

3.3. Outcomes

The results of the 2011 field trial were to a) demonstrate that basal events occurred and could be detected by a passive seismic system; and

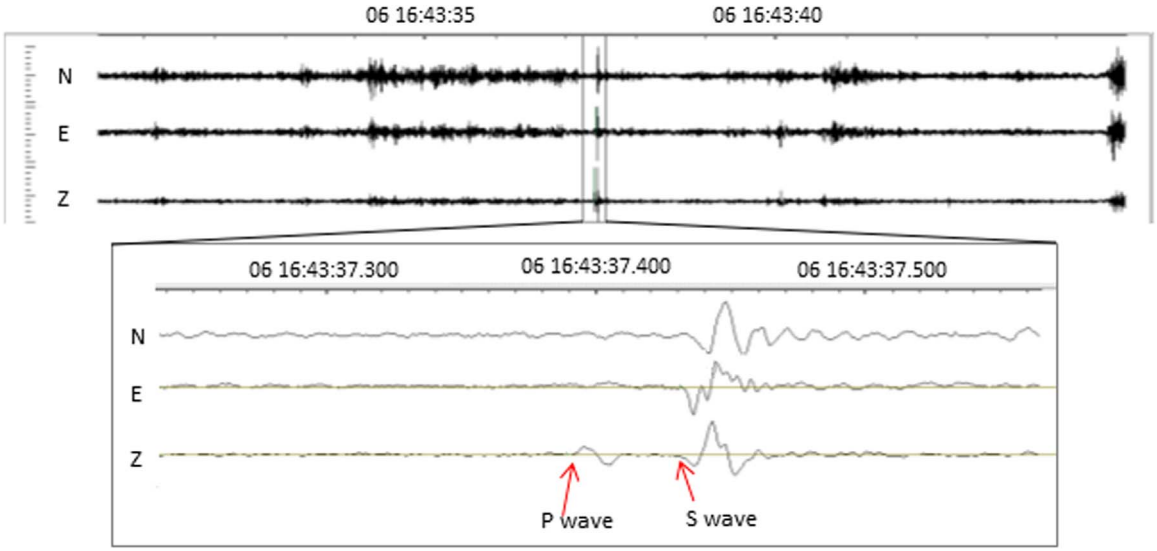


Fig. 4. Event 16:43:37 on DOY 218 SDX (Seismic data eXplorer).

b) to determine the appropriate parameters for the design of the Glacsweb geophones. A threshold ratio of 10, LTA of 0.1 s, STA of 0.005 s, a total record time of 1 s and pre-trigger record time of around 0.3 s was used as the basis for the design of the Glacsweb geophones.

4. Design of the geophone nodes

The geophone nodes were developed in 2012 as an integral part of the Glacsweb sensor network (Fig. 2). They consisted of a custom PCB using an ARM® Cortex®-M3 processor (Energy Micro EFM32G880F128), amplifiers and 2 GByte micro-SD card storage (Fig. 6). The 32-bit M3 processor, running at 28 MHz was powerful enough to carry out simple signal processing which may be required for the decision to store the data. The package included other desirable features including: 12-bit analogue to digital converters (ADC), six serial interfaces (UART) and very low power modes (0.9 μA sleep). The nodes were linked via an RS485 cable link directly to a seismic surface node (SSN) which stored and relayed the data. These surface nodes were used partly because of the relatively large volumes of data but also so that the geophone nodes could eventually use a short range radio link to them on the surface. They SSNs were based on a similar design with a micro-SD card for storing the data, so that in the event of the ice nodes being destroyed the data would be preserved. The serial cabled link is also more energy efficient than radio due to its relatively high data rate and low error rate.

As a borehole has to be drilled to insert the geophone node it was also simple to position a surface node above it and avoid surface runs of cable to the central base station. The cable link also allowed power to be sent to the geophone node from the SSN which has a larger 12AH battery. However, the cable is likely to break in a long deployment so the design also included a backup 173 MHz radio link. This frequency had been found to work well in glaciers (Hart et al., 2015) although it was not deployed due to a lack of testing. Data retrieved by on the surface node is then transferred via the 868 MHz radio network to the base station. This surface network was a custom design based on the Glacsweb packet structure used in the sub-glacial sensor nodes. A file

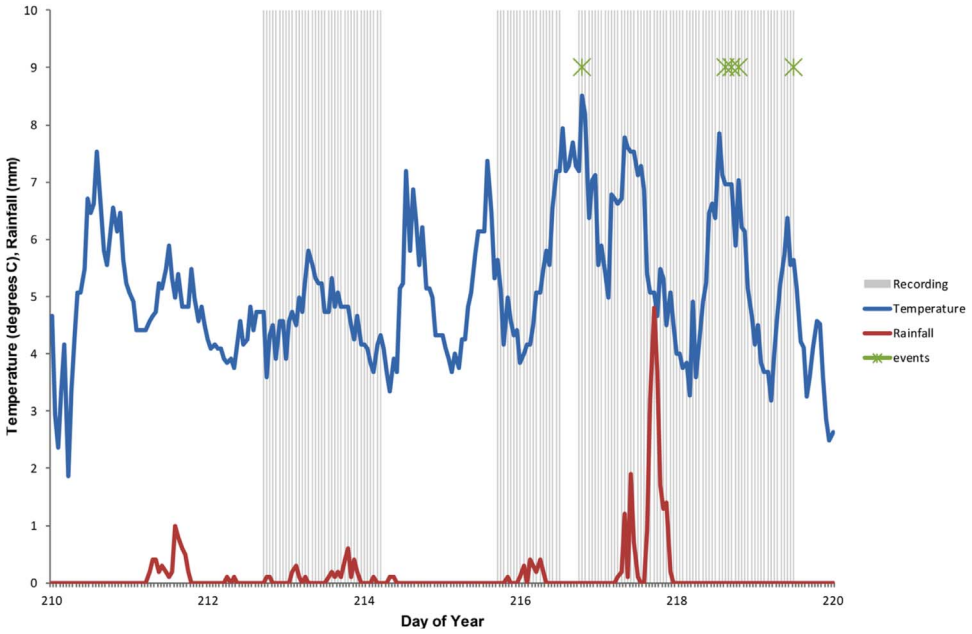


Fig. 5. Air temperature, rainfall, basal seismic events (green stars) when 3 or more stations were recording (grey) during the 2011 field trial. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

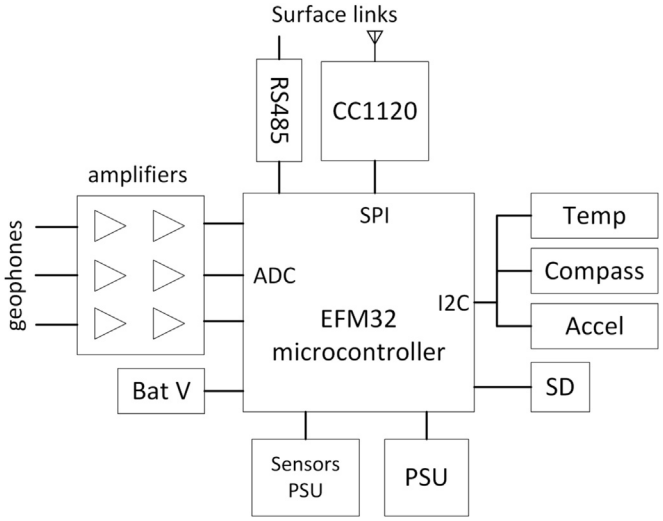


Fig. 6. Schematic of the geophone node electronics.

transfer protocol based on XMODEM was added in order to retrieve the files over radio links without any errors. The system was designed so that the data reaches a safer place with each transfer, from the node to the surface, to the base station then the server. This could give researchers new data every day as long as the surface networks were intact.

The geophone nodes continually sample the output of three orthogonal geophones and store the data to microSD only when an event is detected. We used 28 Hz Geospace GS-20DH geophones encapsulated in resin inside a 70 mm diameter polycarbonate tube

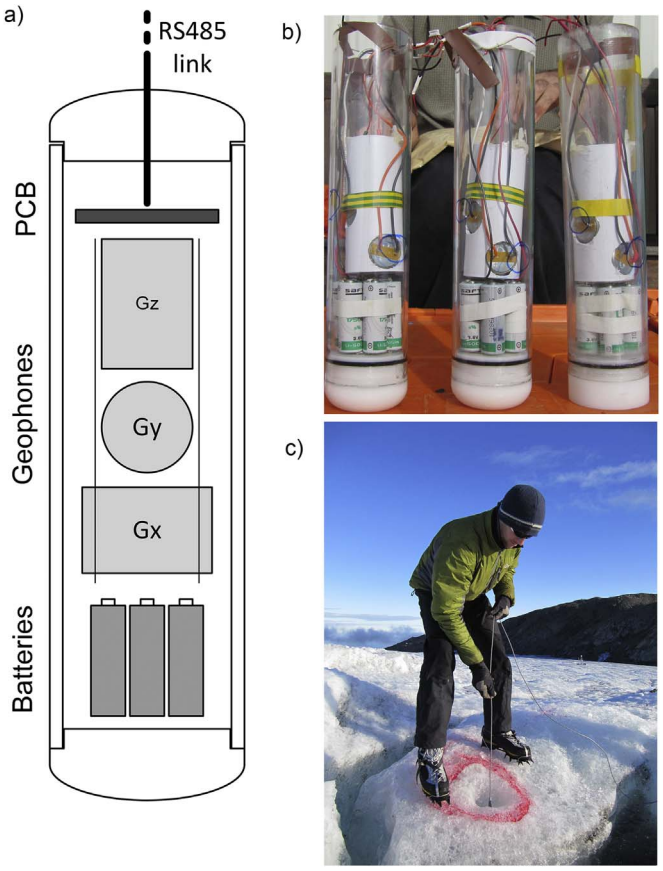


Fig. 7. Glacsweb low power geophone: a) diagram showing the component layout, b) photograph of nodes before potting, c) deploying a geophone node.

(Fig. 7a and b). These geophones were chosen because they had been used successfully in glacier borehole experiments with a commercial logger on the surface (Walter et al., 2008). Our nodes also measure their orientation, with a 3D accelerometer, digital compass and temperature. The nodes contained seven 14.5 AH 3.6 V Lithium Thionyl Chloride AA-size batteries for use when the surface power feed was unavailable. Fig. 7c shows one being lowered into a bore-hole.

The sampling parameters were chosen taking into account the experience with the commercial system. An instrumentation amplifier (INA321E) was used with a gain of 25 dB amplifier to provide sufficient signal. Then a bandpass filter of 0.5–234 Hz. The sampling rate of 512 Hz was chosen to preserve signals up to 256 Hz. The 12 bit ADCs where oversampled to provide 16 bits per sample.

In order to detect events the geophones have to be continuously sampled. However, continuous data storage from three sensors per node would lead to storage and transmission problems while providing little useful signal. To preserve battery life and avoid storing unwanted data a low power technique was devised. Continuous analogue to digital conversion was used, with the data being written to RAM using Direct Memory Access from the ADC while the rest of the hardware is in sleep mode. If the ADC comparator circuit detects an impulse in the signal it sets a flag which causes the processor to write the data buffer to the SD card (Martinez et al., 2012) when it is woken up. This was the simplest technique to implement for this prototype, however a filter-based algorithm (Song et al., 2009) could be implemented in hardware in the future to reduce noise sensitivity. If the node detected an event in the one second sample it recorded the data to a file on the micro-SD card. The binary file format created used a 28 byte header (time stamps) and each of the three channels in 16 bit words, for a total of 3100 bytes. The micro-SD card had a capacity of 2 GBytes and could therefore store 692,000 samples/events, which was far more than expected in one year. The surface nodes also have a 2 GByte card to store all data, which provides a backup which can be retrieved if the radio systems fail. The node uses 300 μA while sampling, with bursts of 3 mA when writing to the SD card.

5. Results from the WSN Geophones 2012

Four geophone nodes were installed within boreholes, to avoid surface seismic noise. Boreholes were drilled in the glacier with a Kärcher HDS1000DE jet wash system. The boreholes were located on a diamond grid, separated by the depth of the glacier. The nodes sat in the base of the 30 m deep boreholes which became embedded into the ice by freezing and creep. Due to the prototype nature of the enclosures and electronics only one geophone (ID 62) functioned for more than one day. It is possible that the cables and sealing failed in many of the nodes, which are too deep to recover once they are embedded. However, node 62 operated for 25 days in 2012 from DOY 270–295. During this time 180 events were recorded.

5.1. Data analysis

We first investigate the general relationship between the number of events per day and the meteorological conditions, and then look in more detail at the individual traces. The signal was similar on all three planes. Without multiple data sources and synchronisation it was impossible to use common detection to locate the ice quake in three dimensions. However it was possible to derive the dominant frequency of the events by fast Fourier transform analysis and carry-out event type classification. Using these two techniques we were able to interpret the styles of microseismicity recorded.

There was a generally positive relationship between events per day and glacier surface melt (Fig. 8a), and we can divide the results into three time periods based on geophone activity (Fig. 8a). High seismic activity (DOY 270–279), is associated with high and generally positive air temperatures at the base station. The day with the highest melt has

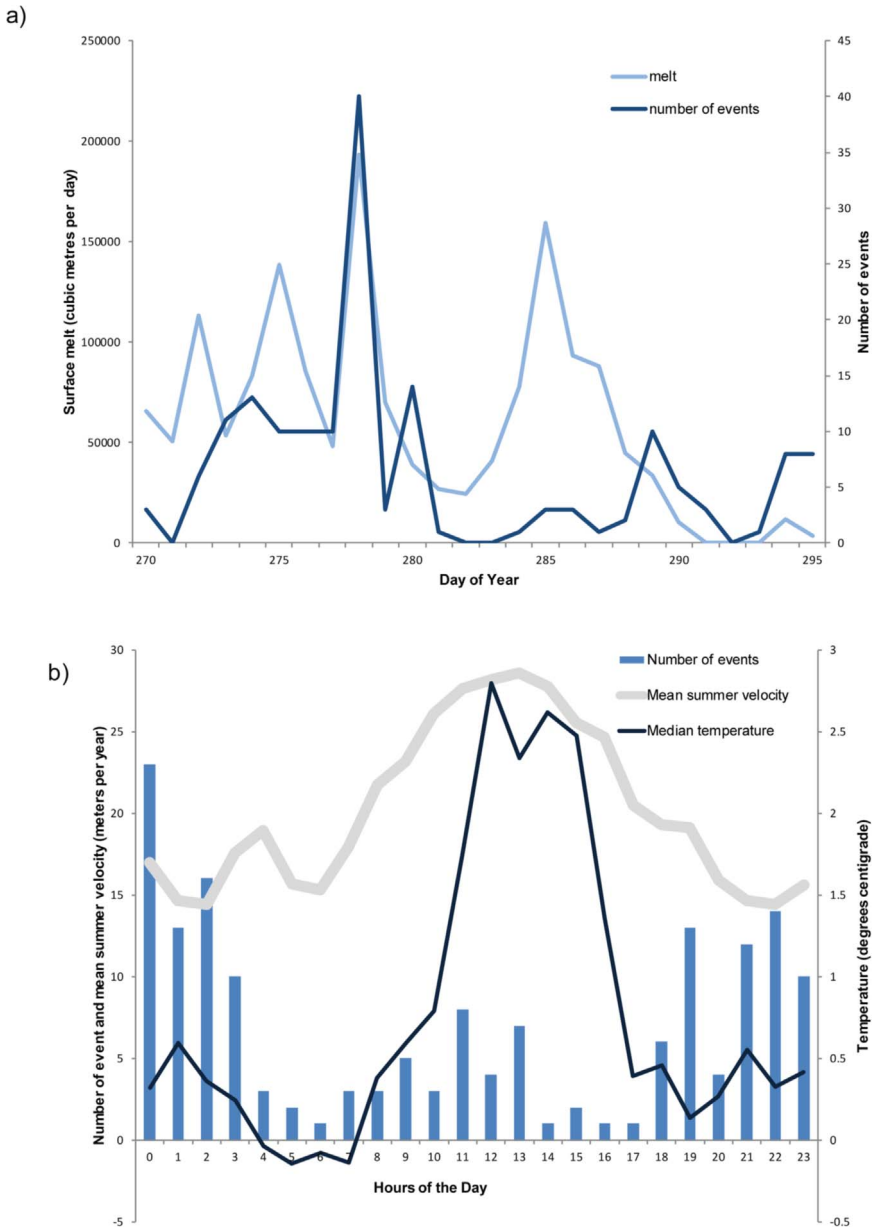


Fig. 8. a) Relationship between number of seismic events (dark line) and glacier surface melt (calculated from PDD) (light line); b) diurnal timing of seismic events.

the most events. This is followed by a period of low seismic activity (DOY 280–288), when air temperatures are lower and are below zero at night. Here the maximum events also occur on the day with the most melt. Finally, there is an intermediate seismic activity (DOY 289–295) during the winter when temperatures are almost always below zero, but on a few days' temperatures at the base station rose above zero (DOY289, 290, 294 and 295). Day 294 had the highest positive temperatures during this period and had the highest number of seismic events.

There was also a relationship between diurnal temperature patterns and events. At the site, the temperatures were lowest during the night (18:00–08:00) and warmest during the afternoon (12:00–16:00). The majority of seismic events occurred during the temperature rise and peak and during the night (Fig. 8b). The mean hourly summer ice velocity is also shown (from the available period DOY 218–269). Glacier velocities are highest during the temperature rise and peak and lowest during the night.

5.2. Classification of event types

Fast Fourier transform analysis was carried out on each trace, in order to find the dominant frequencies. In additional, each trace was manually analysed and classified. This enabled us to characterize the type of events into three forms (Table 1, Figs. 9 and 10):

- Type A – consisting of a low frequency wave (Fig. 9a). This typically has a dominant frequency of 14 Hz (with smaller peaks at 4 and 6 Hz).
- Type B – consisting of a distinct wave form which we interpret to reflect the P and S wave (Fig. 9b). This also has a dominant frequency of 4 Hz (and smaller peaks at 6 and 8 Hz).
- Type C – consisting of a cigar shaped envelope (Fig. 9c), with a dominant frequency of 6 Hz (and also 8–16 Hz)

Fig. 11 shows the diurnal occurrence of each event type during the three distinct time periods. During the high seismic activity period all three seismic types are found and 17% of events occurred during the

Table 1
Seismic event types.

	Type A	Type B	Type C
Onset	Emergent	Emergent	Impulsive
Amplitude	Monochromatic waveform	Monochromatic waveform, impulsive P and S wave and distinct coda	Episodic events, often forms cigar-shaped envelope
Spectra		Multiple pulses, lasting between 0.08 and 0.2 s with intervals 0.12 s	
Dominant frequency	14 Hz	4 Hz	6 Hz
Frequency range	3–40 Hz	3–22 Hz	3–38 Hz
% low frequency (3–18 Hz)	93%	95%	84%
% high frequency (19–40 Hz)	7%	6%	16%
Duration	> 1 s	Mean 0.18 s (s.d. 0.04 s)	Mean 0.13 s (s.d. 0.06 s)
Interpretation	Deformation	Sliding	Hydraulic

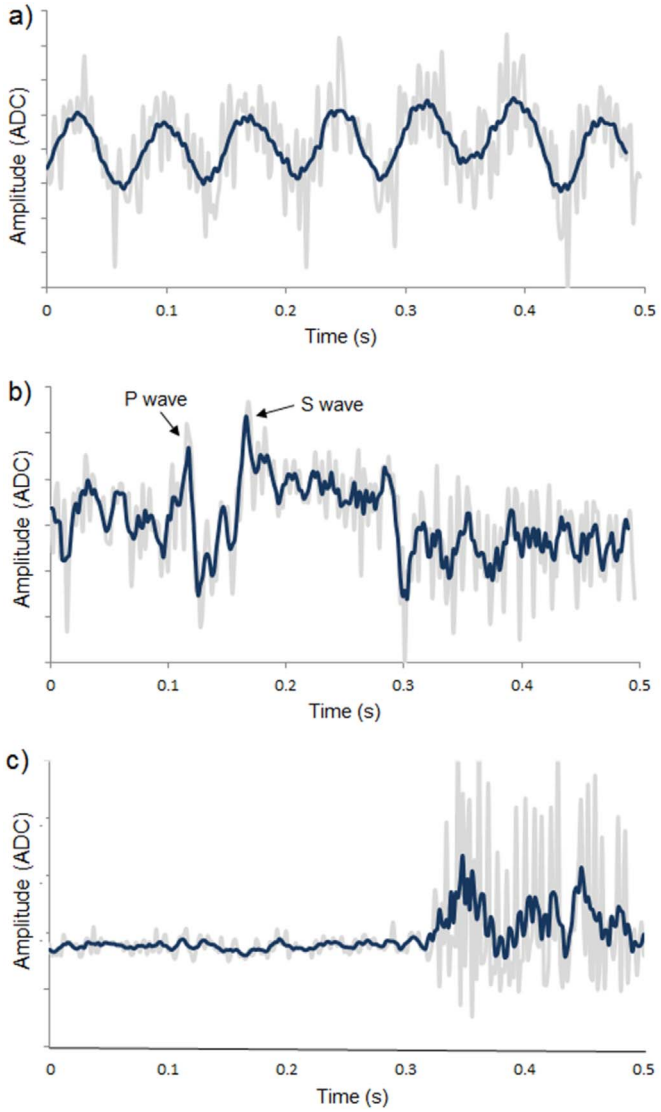


Fig. 9. Examples of different seismic event types (light line is unfiltered signal, dark line filtered to show the dominant frequency found from the FFT analysis): a) Type A, b) Type B, c) Type C.

temperature/velocity rise, 23% during the temperature/velocity maximum, and 60% after the temperature/velocity peak. Type A and C events occurred most during the temperature/velocity fall (73% and 100% respectively), whilst Type B occurred mostly during the temperature/velocity rise (33%).

During the low seismic activity period only Type A occurs, and 38% of events occurred during the temperature/velocity rise, 8% during the maximum, and 54% after the peak.

During the intermediate seismic activity (winter) period all three seismic styles are present, 76% of the events occurred when air temperatures were negative, and 80% of these were Type A events (16% Type C and 4% Type B) of which 86% occurred during the temperature/velocity decline.

5.3. Interpretation of the event types

It has been suggested that different microseismic sources can be identified by distinct wave forms, but that basal signals may be masked by other signals (in particular crevassing) (Pomeroy et al., 2013; Helmstetter et al., 2015). However, it has been shown that crevassing is typically characterized by high frequency (20–35 Hz) events (with short duration (< 1 s) impulsive onsets), whilst basal events are characterized by low frequency (6–15 Hz) events [46].

Our results show that 91% of the dominant frequencies were between 3–18 Hz, showing a prevalence of low frequency events. Combined with the evidence for basal events from the 2011 field trial, we conclude that our 3 event types reflect subglacial events.

Type A – These are the most common event type and occurred throughout the whole period, even when air temperatures were negative. They are most commonly found during the night. The characteristics of these signals (Table 1) were similar to those associated with subglacial sediment (till) deformation during stick-slip motion (Anandakrishnan and Bentley, 1993) especially during basal recoupling after a stick-slip event (West et al., 2010).

We have shown previously shown that at Skálafellsjökull subglacial deformation takes place throughout the year (Hart et al., 2015), and so suggest that Type A also reflects subglacial deformation. This process occurs as the glacier begins to accelerate in the morning, but this is more frequent when the glacier recouples with the bed as the glacier decelerates in the night.

A typical value for basal stick-slip is 1 mm in 0.01 s slip (Zoet et al., 2013). We measured 3 m a year till movement (with GPR) at Skálafellsjökull. If we assume this represents stick-slip this would require an average of 8.2 events per day. If we suppose our 3 seismic times represent summer, early spring/late autumn (when night temperatures are below zero) and winter respectively, we can calculate value of the annual average events per day. For Type A this would be 5 events per day (and for all events 7 events per day), which is remarkably similar to the predicted figure.

Type B – These are the second most common event, but have a far

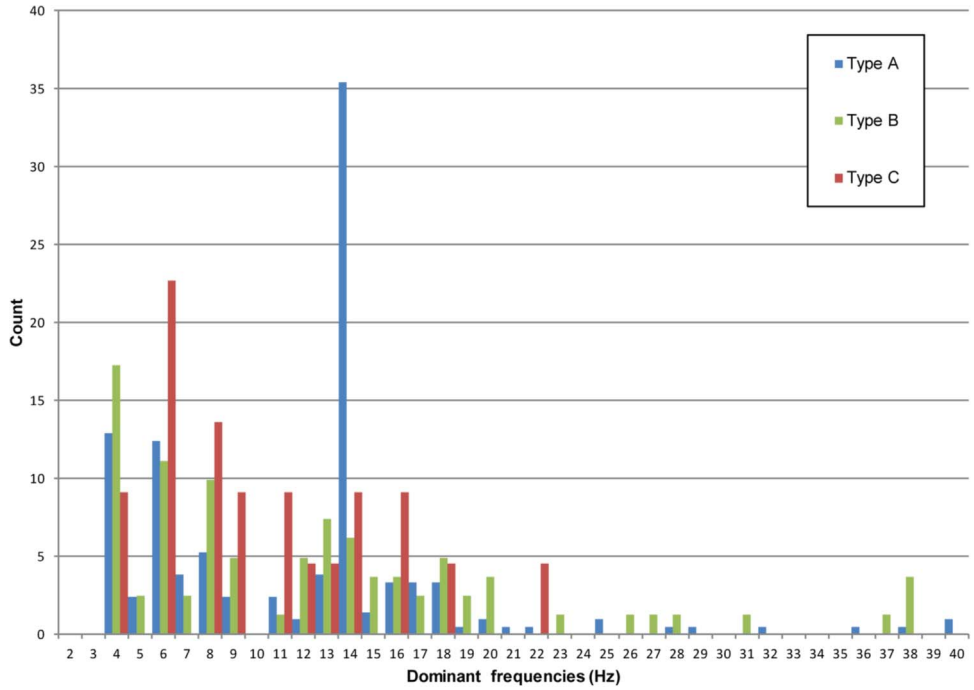


Fig. 10. Histogram of the top three dominant frequencies for each trace, separated into the three event types.

more restricted time range. During the high seismic activity period these events occurred either just before and/or during the temperature/velocity maxima. During the winter, these events only occurred during positive temperature days and these all occurred during the maxima in temperature/velocity.

This type has virtually identical wave forms with a distinct P and S waves with an average separation of 0.05 s (s.d. 0.02). The difference between the P and S waves gives an average depth of 180 m, which is within the range of the ice depth in the study area. We suggest the Type B events reflect a large main basal sliding event.

Type C – These are the least frequent of the event types. They only occur in the night (after the temperature/velocity maxima). Several researchers have suggested that low frequency signals with extended codas and weakly developed emergent onsets or no seismic phases (P and S waves), are typical of water sourced seismicity (Lawrence and Qamar, 1979; Stuart et al., 2005; West et al., 2010). Hydraulic events, possible disturbances of the water flow (associated with melting), that can generate a 'water hammer' effect in plumbing systems could cause the observed harmonic signal (Wolf and Davies, 1986; Kavanaugh and Clarke, 2001).

At Skálafellsjökull these events may reflect hydraulic disturbances associated with changing drainage pathways created as a result of the stick-slip events.

5.4. Evaluation of the system

The commercial system installed on the glacier surface in 2011 provided good evidence for basal events. However, this system required daily manual re-levelling and high levels of power supply. Thus this system could not be used for long-term automatic recordings. Due to noise from the glacier surface, rain, crevassing and melting, more processing was needed to interpret the data. The borehole system installed in 2012 automatically recorded the signals containing possible events. This system was less affected by noise and so only 180 events were recorded in 25 days which were shown to be non-random. The low power system design means it can be left running for long periods without solar generation. Improvements to the enclosures and software reliability would then allow long-term monitoring. The prototype radio network was successful and showed that the data volumes were not too high for a real deployment. Further development on the robustness of the system could allow more accurate isolation of the source of the basal signals (where multiple geophones are triggered simultaneously) and allow the investigation of the seasonal changes, particularly winter high discharge events (Young et al., 2015). An improved time synchronisation system is also needed which can operate over the radio network, this could be helped by including a GPS unit in each surface node.

6. Conclusion

This study has demonstrated the successful design and development of a borehole geophone as part of an environmental wireless sensor network. A field trial with commercial surface geophones served as a baseline and helped to determine the capture parameters. The design makes it feasible to obtain long term geophone event capture, with the data being transmitted back to a data server.

Our basal signals showed (as noted by other researchers, Smith, 2006) a predictable, periodic recurrence of signals which suggest similar source events, occurring regularly at a similar location from the release of accumulated stress. The results showed that there was a relationship between surface melt production and the number of microseismic events. These events occurred more frequently at specific times during the day. Three types of seismic events were identified. The most common was Type A, typified by an emergent, monochromatic waveform, which occurred mostly during the glacier velocity decrease, and was interpreted as till deformation. Secondly, Type B, an emergent monochromatic waveform, with impulsive P and S waves (with an average separation of 0.05 s), and only occurred during the glacier velocity rise and peak. This was interpreted as resulting from the main basal sliding event. Lastly, Type C, with a waveform comprising episodic events, often with a cigar-shaped envelope, occurring typically 11 h after the velocity peak. This is interpreted as reflecting hydraulic transience as the drainage system returns to 'normal'.

Future work to improve the reliability of the nodes would allow more locations to be sampled simultaneously to locate events. This would also require extra synchronisation, which could be implemented with small GPS units on each surface node. Testing the 173 MHz radio links to the surface could show that the backup cables are unnecessary

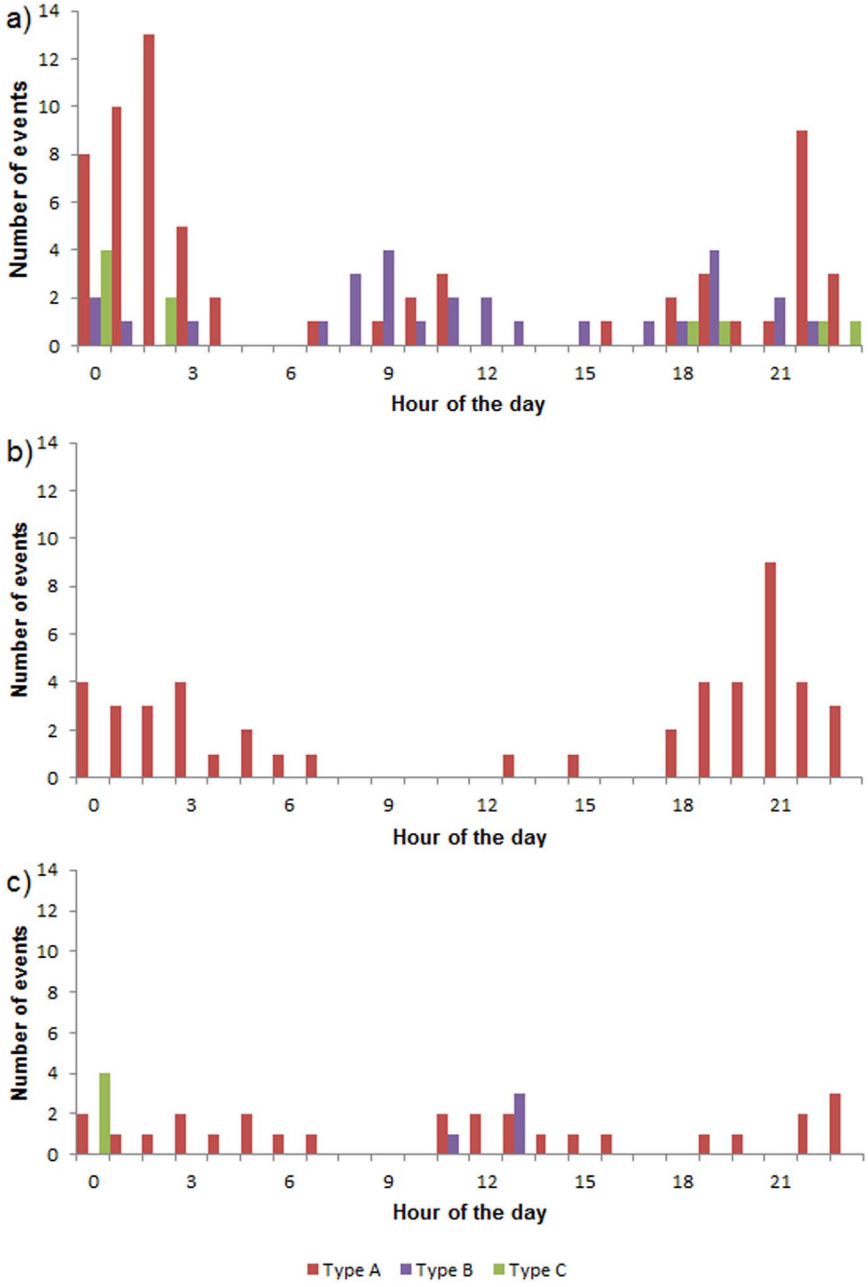


Fig. 11. Diurnal timings of the different seismic activity periods: a) high seismic activity (DOY 270-270), b) low seismic activity (DOY 280–288), c) intermediate seismic activity (DOY 289–295).

if extra batteries are enclosed in the nodes. The use of new techniques and technologies in this design has been shown to provide a breakthrough in the capabilities of long term continuous monitoring for geoscience.

Acknowledgements

The authors would like to thank the Glacsweb Iceland 2011 and 2012 teams for help with data collection and thanks to Laura Edwards, Matthew Burke and Andrew Turner for data processing, Dirk De Jager for PCB design and Mark Dover for Cartography, and Alex Brisbourne and Victoria Lane from SEIS UK for advice and field and data processing assistance. This research was funded by the ESPRC (Grant EP/C511050/1), the Leverhulme Trust (Grant number F/00 180/AK), National Geographic Society Science and Exploration Europe GEFNE45-12 and ARM. The NERC Geophysical Equipment Facility provided loans of the Seismology and GPR equipment.

References

Anandakrishnan, S., Bentley, C.R., 1993. Micro-earthquakes beneath Ice Streams B and C, West Antarctica: observations and implications. J. Glaciol. 39, 455–462.
Bahr, D.B., Rundle, J.B., 1996. Stick-slip statistical mechanics at the bed of a glacier. Geophys. Res. Lett. 23 (16), 2073–2076.
Braithwaite, R.J., 1985. Calculation of degree-days for glacier-climate research. Z. Gletsch. Glazialgeol. 20, 1–8.
Boulton, G.S., Jones, A.S., 1979. Stability of temperate ice caps and ice sheets resting on beds of deformable sediment. J. Glaciol. 24, 29–44.
Chong, C.Y., Kumar, S.P., 2003. Sensor networks: evolution, opportunities, and challenges. Proc. IEEE 91 (8), 1247–1256.
Deichmann, N., Ansorge, J., Scherbaum, F., Aschwanden, A., Bernardi, F., Gudmundsson, G.H., 2000. Evidence for deep icequakes in an Alpine glacier. Ann. Glaciol. 31 (1), 85–90.
Delin, K.A., Jackson, S.P., Johnson, D.W., Burleigh, S.C., Woodrow, R.R.J., Mcauley, M., Dohm, J.M., Ip, F., Ferré, T.P.A., Rucker, D.F., Baker, V.R., 2005. Environmental studies with the Sensor web: principles and practice. Sensors 5, 103–117.

Duval, P., 1977. The role of water content on the creep rate of polycrystalline ice. IAHS 118, 29–33.
Fischer, U.H., Clarke, G.K.C., 1997. Stick-slip sliding behaviour at the base of a glacier. Ann. Glaciol. 24, 390–396.
Gehrke, J., Liu, L., 2006. Sensor-network applications. IEEE Internet Comput. 10, 16–17.
Hart, J.K., Martinez, K., 2006. Environmental Sensor Networks: a revolution in the earth system science? Earth-Sci. Rev. 78 (3), 177–191.
Hart, J., Martinez, K., Edwards, L., 2011. Investigating glacier stick-slip motion using a wireless Sensor Network. Report for NERC GEF. p. 10.
Hart, J.K., Martinez, K., Ong, R., Riddoch, A., Rose, K.C., Padhy, P., 2006. A wireless multi-sensor subglacial probe: design and preliminary results. J. Glaciol. 52 (178), 389–397.
Hart, J.K., Rose, K.C., Clayton, A.I., Martinez, K., 2015. Englacial and subglacial water flow at Skálafellsjökull, Iceland derived from ground penetrating radar, in situ Glacsweb probe and borehole water level measurements. Earth Surf. Process. Landf. 40 (15), 2071–2083.
Helmstetter, A., Nicolas, B., Comon, P., Gay, M., 2015. Basal icequakes recorded beneath an Alpine glacier (Glacier d'Argentière, Mont Blanc, France): evidence for stick-slip motion? J. Geophys. Res. Earth Surf. 120, 379–401.
Hock, R., 2003. Temperature index temperature modelling in mountain areas. J. Hydrol. 282, 104–115.
Hasler, A., Talzi, I., Beutel, J., Tschudin, C., Gruber, S., 2008. Wireless sensor networks in permafrost research: Concept, requirements, implementation, and challenges. In: Proceedings of the 9th International Conference on Permafrost, Fairbanks, US, 669–674.
Huang, Y., Martinez, J.-F., Sendra, J., López, L., 2015. Resilient wireless sensor networks using topology control: a review. Sensors 15, 24735–24770.
Iken, A., Röthlisberger, H., Flotron, A., Haeberli, W., 1983. The uplift of Unteraargletscher at the beginning of the melt season — a consequence of water storage at the bed? J. Glaciol. 29, 28–47.
Johannesson, T., Sigurdsson, O., Laumann, T., Kennett, M., 1995. Degree-day glacier mass-balance modelling with applications to glaciers in Iceland, Norway and Greenland. J. Glaciol. 41 (138), 345–358.
Kavanaugh, J.L., Clarke, G.K.C., 2001. Abrupt glacier motion and reorganization of basal shear stress following the establishment of a connected drainage system. J. Glaciol. 47, 472–480.
Lawrence, W.S.T., Qamar, A., 1979. Hydraulic transients: a seismic source in volcanoes and glaciers. Science 203, 654–656.
Martinez, K., Basford, P., 2011. Robust wireless sensor network performance analysis. IEEE Sens. Limerick.
Martinez, K., Hart, J.K., Ong, R., 2004. Environmental Sensor Networks. IEEE Comput. 37 (8), 50–56.
Martinez, K., Hart, J.K., Ong, R., 2009. Deploying a Wireless Sensor Network in Iceland, lecture notes in computer science. Proceedings of the Geosensor Networks. 5659, 131–137.
Martinez, K., Basford, P.J., De Jager, D., Hart, J.K., 2012. A wireless sensor network system deployment for detecting stick slip motion in glaciers. In, IET International conference on Wireless Sensor Systems, London, GB, 18–19 Jun, p. 3, 14-16.
Metaxian, J.-P., Araujo, S., Mora, M., Lesage, P., 2003. Seismicity related to the glacier of Cotopaxi Volcano Equador. Geophys. Res. Lett. 30 (9), 1483.
Neave, K.G., Savage, J.C., 1970. Icequakes on the Athabasca Glacier. J. Geophys. Res. 75 (8), 1351–1362.
Oliveira, L.M.L., Rodrigues, J.J.P.C., 2011. Wireless Sensor Networks: a survey on environmental monitoring. J. Commun. 6, 143–151.
O'Neel, S., Pfeffer, W.T., 2007. Source mechanics for monochromatic icequakes produced during iceberg calving at Columbia Glacier, AK. Geophys. Res. Lett. 34 (22).
Pomeroy, J., Brisbourne, A., Evans, J., Graham, D., 2013. The search for seismic signatures of movement at the glacier bed in a polythermal valley glacier. Ann. Glaciol. 54 (64), 149–156.
Qamar, A., 1988. Calving icebergs: a source of low-frequency seismic signals from Columbia Glacier, Alaska. J. Geophys. Res. 93 (B6), 6615–6623.
Rothlisberger, H., 1972. Seismic exploration in cold regions. Cold Reg. Sci. Monogr. II-A 2a, Hanover, CRREL, 139.
Sigurðsson, O., 1998. Glacier variations in Iceland 1930–1995. Jökull 45, 3–25.
Smith, A.M., 2006. Microearthquakes and subglacial conditions. Geophys. Res. Lett. 33 (24).
Song, W.-Z., Huang, R., Xu, M., Ma, A., Shirazi, B., LaHusen, R., 2009. Air-dropped sensor network for real-time high-fidelity volcano monitoring. In: Proceedings of the 7th International Conference on Mobile systems, applications, and services, Wroclaw, Poland.
Tsai, V.C., Ekstrom, G., 2007. Analysis of glacial earthquakes. J. Geophys. Res. 112, F03014.
Stuart, G., Murray, T., Brisbourne, A., Styles, P., Toon, S., 2005. Seismic emissions from a surging glacier: bakaninbreen. Svalbard, Ann. Glaciol. 42, 151–157.
Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., Estrin, D., 2004. Habitat monitoring with sensor networks. Commun. ACM 47 (6), 34–40.
Walter, F., Deichmann, N., Funk, M., 2008. Basal icequakes during changing subglacial water pressures beneath Gornergletscher, Switzerland. J. Glaciol. 54 (186), 511–521.
Weaver, C.S., Malone, S.D., 1979. Seismic evidence for discrete glacier motion at the rock-ice interface. J. Glaciol. 23 (89).
Weertman, J., 1957. On the sliding of glaciers. J. Glaciol. 3, 33–38.
Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J.M., Welsh, M., 2005. Monitoring volcanic eruptions with a wireless sensor network. In: Proceedings of the European Workshop on Sensor Networks (EWSN'05), #1568945184.
West, M.E., Larsen, C.F., Truffer, M., O'Neel, S., LeBlanc, L., 2010. Glacier microseismicity. Geology 38, 319–322.
Wiens, D.A., Anandakrishnan, S., Winberry, J.P., King, M.A., 2008. Simultaneous teleseismic and geodetic observations of the stick–slip motion of an Antarctic ice stream. Nature 453 (7196), 770–774.
Wolf, L.W., Davies, J.N., 1986. Glacier-generated earthquakes from Prince William Sound, Alaska. Bull. Seismol. Soc. Am. 76, 367–379.
Xu, G., Shen, W., Wang, X., 2014. Applications of wireless sensor networks in marine environment monitoring: A survey. Sensors, 14, 16932–16952.
Young, D.S., Hart, J.K., Martinez, K., 2015. Image analysis techniques to estimate river discharge using time-lapse cameras in remote locations. Comput. Geosci. 76, 1–10.
Zoet, L.K., Alley, R.B., Anandakrishnan, S., Christianson, K., 2013. Accelerated subglacial erosion in response to stick-slip motion. Geology 41, 159–162.

Deploying a 6LoWPAN, CoAP, low power, wireless sensor network

Arthur Fabre, Kirk Martinez, Graeme M. Bragg, Philip J. Basford
Electronics and Computer Science
University of Southampton
{af1g12, km, g.bragg, pjb}@ecs.soton.ac.uk

Jane Hart
Geography and Environment
University of Southampton
j.k.hart@soton.ac.uk

Sebastian Bader
Electronics Design
Mid Sweden University
sebastian.bader@miun.se

Olivia M. Bragg
Environment
University of Dundee
o.m.bragg@dundee.ac.uk

ABSTRACT

In order to integrate equipment from different vendors, wireless sensor networks need to become more standardized. Using IP as the basis of low power radio networks, together with application layer standards designed for this purpose is one way forward. This research focuses on implementing and deploying a system using Contiki, 6LoWPAN over an 868 MHz radio network, together with CoAP as a standard application layer protocol. A system was deployed in the Cairngorm mountains in Scotland as an environmental sensor network, measuring streams, temperature profiles in peat and periglacial features. It was found that RPL provided an effective routing algorithm, and that the use of UDP packets with CoAP proved to be an energy efficient application layer. This combination of technologies can be very effective in large area sensor networks.

CCS Concepts

•**Computer systems organization** → **Sensor networks; Embedded software;**

Keywords

6LoWPAN, CoAP, low power, sensor network, deployment

1. INTRODUCTION

While wireless sensor networks have advanced in many areas, issues of standardization and heterogeneity are ongoing challenges. One possible solution to networking standardization is to use 6LoWPAN, which brings IPv6 compatibility and tools, with 802.15.4 physical and MAC layers. The next issue is using a standard application layer suitable for data transfer over low bandwidth radio links. A good candidate for this is CoAP[3], which only uses UDP and replicates a REST HTTP-like interface optimized for low data rates through the use of binary requests. Data must also be encoded in an efficient, extensible, cross-platform manner, which can be achieved using Protocol Buffers[2]. We have implemented a complete environmental sensor network system and deployed it in the Cairngorm mountains of Scotland in order to test the feasibility and effectiveness of such a standards compliant but low energy design.

SenSys '16 November 14-16, 2016, Stanford, CA, USA

ACM ISBN 978-1-4503-4263-6/16/11.

DOI: http://dx.doi.org/10.1145/2994551.2996707

2. NODES

The sensor nodes consist of a main processor board, and a carrier board which can be connected to a number of smart sensors, as shown in Figure 1. The processor board features a microcontroller; flash for storing samples and the configuration. The carrier board provides a sub-GHz radio, real time clock, RS485 driver, and power supplies.

The first generation of nodes used a Zolertia Z1 (MSP430) as the main processor board, and were powered from 12V lead-acid batteries. The MSP430 proved to be RAM constrained, consequently a second generation of nodes was designed using an ARM Cortex-M3. This generation maintained pin-compatibility with the first generation, allowing for a heterogeneous network. Power is delivered either through a separate power controller or from a 12V lead acid battery. The separate power controller features an ARM Cortex-M0+, that handles the maximum power point tracking of a 5W solar panel, and LiPo battery charging. It also provides power to the smart sensors.

3. SMART SENSORS

The smart sensors all implement a common interface and protocol, using RS485, and feature a unique identifier. The nodes are configured with the identifiers of connected sensors, and receive a binary payload (typically a Protocol Buffer), that is stored as is. Its contents are decoded server-side. Consequently, new sensors can be created and deployed with no changes to node firmware or hardware. Typical sensors use an AVR to aggregate data from multiple sensors, such as an array of thermometers or accelerometers. The same protocol is used to communicate with the power controller.

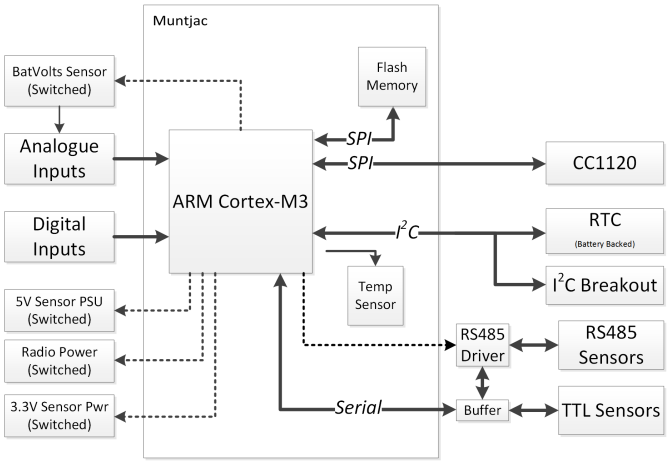


Figure 1: Block diagram of the sensor nodes

4. SOFTWARE

The node firmware is based on Contiki-OS, as it supports 6LoWPAN, RPL and CoAP (through Erbium). Drivers were written to support the CC1120, flash, RTC and RS485 peripherals. A common abstraction layer was developed, allowing multiple platforms to be supported by the firmware. A sampling process uses the real time clock to ensure that nodes always begin sampling at predictable times. In order to maximize sample storage, Coffee file system headers were reduced by removing micro log related entries, allowing a sample to fit in a single logical page (128 bytes). This allows up to 16,384 samples to be stored, with a typical size of 100 bytes.

5. NETWORK

Our previous research[1] led us to use an 868 MHz radio based on the CC1120 transceiver. This provides the long range links required (up to 3km) at an acceptable data rate (50 kbit/s) and is a commonly used frequency for environmental sensing. The nodes form a mesh network using RPL, with a border-router providing public IPv6 connectivity.

The network is built up from a number of layers:

- 802.15.4: Physical & MAC layer.
- ContikiMAC: Provides low energy radio duty cycling, allowing the radio to receive incoming packets at any time, whilst significantly reducing the power usage.
- 6LoWPAN: Encapsulates IPv6 traffic.
- IPv6 / Routing Protocol for Low power and Lossy Networks (RPL): Network layer: Sets up the routing for the network, enabling nodes to discover neighbors, and select the best available route to the border router.
- UDP: Transport layer. Provides low latency, stateless data transport. Reliability is not provided, in contrast with TCP, as it is offloaded to the application layer.
- CoAP: Application layer. Provides a stable, discoverable API to the nodes.
- Protocol Buffers: Provide an extensible encoding for samples and the node configuration.

The gateway micro-PC running the border-router GETs and DELETEs available samples from the nodes using a CoAP API. The Protocol Buffer encoded samples are then pushed over a satellite link to a MySQL database, where they are decoded. GETing from the nodes instead of having the nodes POST, allows centralized network scheduling thereby alleviating the strain of multiple nodes sending samples simultaneously through the network. Erbium also lacked support for blockwise POST requests to clients at the start of the project, limiting POSTable samples to 64 bytes.

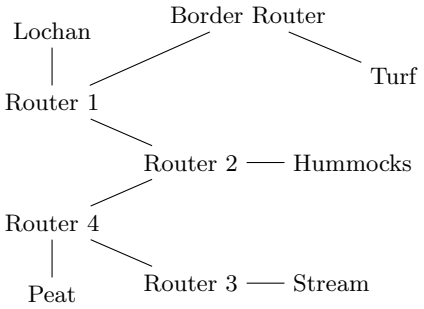


Figure 2: Default RPL routes

6. DEPLOYMENT

Nine second generation nodes are currently deployed in the Cairngorms, replacing the original eight[1]. Their default routes can be seen in Figure 2. Nodes sample every 20 minutes, and data is fetched from them hourly.

6LoWPAN and RPL have allowed a mix of hardware to be deployed, the border router is MSP430 based, while the remaining nodes are second generation ones.

The use of CoAP has allowed us to deploy nodes with heterogeneous firmwares, using undefined resources returns an error code indicating the resource is not present. The Protocol Buffer encoding also allows the introduction of new fields, for example a power board ID was added to the configuration to support second generation nodes, with no ill effects on previously deployed nodes.

End to end IPv6 connectivity eases system maintenance, allowing nodes to be configured, monitored and rebooted from computers in the lab. Standard network tools such as ping can also be used.

7. CONCLUSION

The deployment has shown that using 6LoWPAN together with CoAP brings many benefits compared to the previous approaches using a selection of WSN algorithms. The low throughput of the 868 MHz radio (approx 20 kbit/s) was not a hindrance to using IP protocols and IETF standards, while providing long range, heterogeneity and ease of deployment.

8. ACKNOWLEDGMENTS

Thanks to Wildland Ltd for access to the site and logistical support.

9. REFERENCES

[1] G. Bragg, K. Martinez, P. Basford, and J. Hart. 868MHz 6lowpan with contikimac for an internet of things environmental sensor network. 2016.
[2] Google. Protocol buffers, 2016.
[3] Z. Shelby, K. Hartke, and C. Bormann. The constrained application protocol (coap). Technical report, 2014.

BRITAIN'S HIGHEST BOG: CAN WE UNLOCK ITS SECRETS?

Olivia M. Bragg[1], Philip J. Basford[2], Andrew R. Black[1], Graeme M. Bragg[2], Jane K. Hart[2] and Kirk Martinez[2]

[1]University of Dundee and [2]University of Southampton, United Kingdom

SUMMARY

The Glenfeshie Mòine Mhór (Great Moss) is Britain's highest bog, the largest bog in the Cairngorm Mountains (Scotland) and a water source area for the River Spey. The area was managed primarily for sport hunting for about two centuries, but deer numbers have been heavily reduced in the last decade to allow regeneration of natural woodland and the return to more natural condition of all ecosystems including peatland. However, it may not be realistic to expect spontaneous improvement in peatland condition and ecosystem services provision in the harsh environment of the Mòine Mhór, which retains snow cover for more than half the year and differs floristically from lower-altitude bogs. To understand whether and where management intervention may be required, we need first to understand how the system works at scales ranging from microform to macrotope, and from sub-catchment to whole-system level. Multi-disciplinary condition and process studies (involving various collaborators) are in progress, with a current emphasis on streamflow generation and fluvial carbon loads. This presentation develops two sub-themes. First, ground survey and GIS analysis are used to address the questions: what are the special features of this bog; what is the nature and extent of degradation; and what are the implications for water delivered to the outflow streams? Secondly, a striking feature is the bare peat patches which were favourite resting places for deer on warm, dry summer days. The occurrence of seasonally extreme surface conditions seems a likely factor in preventing their recolonisation by bog plants now. Information about these conditions that cannot readily be accessed through direct observation, originating from temperature sensors and delivered at 60-minute intervals *via* a low power internet link, is explored in this context. Finally, we discuss aspects of the suitability of our investigation methods for remote and intermittently accessible field sites such as the Mòine Mhór.

KEY WORDS: bare peat, Cairngorms, environmental sensor network, Internet of Things, vegetation

INTRODUCTION

The Glenfeshie Mòine Mhór ('Great Moss' in the Gaelic language) occupies a plateau of siliceous Dalradian country rock (Thomas *et al*. 2004), about 3 km across, at the south-western corner of the Cairngorm Mountains in Scotland. It has been described as the largest expanse of peatland in the Cairngorms National Park and the highest ('raised') bog in Britain (900–950 m a.s.l.). Located within the privately owned Glenfeshie Estate, the centre of the plateau (57º 01′ 58″ N, 03° 48′ 51″ W) lies approximately 600 m above and 5.5 km distant from the Estate office at Carnachuin on the left bank of the River Feshie (Figure 1). It is usually snow-covered for half of the year (November to May), with snow patches often persisting through June. Access at any time of year is constrained by the logistics of fording the notoriously flashy River Feshie and most of the plateau is in mobile phone shadow. The site has considerable interest for nature conservation, being included within the Cairngorms Site of Special Scientific Interest (SSSI) (SNH 2016) and Special Area of Conservation (SAC) (JNCC 2016).

Glenfeshie was managed for sport hunting - primarily stalking of Red Deer (*Cervus elaphus*) - from the early 19th century, and deer stocking levels of 35–40 km^{-2} were usual until 2006. Since then the density of deer has been substantially reduced, to ~1 km^{-2}, to promote the return of native woodland and more natural ecosystems in general. After two centuries of heavy use by deer in summer, the most striking features of the Mòine Mhór are the expanses of bare peat dissected by an extensive system of erosion channels feeding into headwaters of the Feshie, which is in turn a tributary of the River Spey. It is not immediately clear how severely ecosystem services such as water quality maintenance, carbon cycling and support of wildlife have been impaired, or whether it is realistic to expect spontaneous improvements in condition of the peatland without further management intervention.

Here we report some early results from ongoing studies that aim to inform future management of the Mòine Mhór. At macro scale we make first estimates of the extent of peatland on the plateau and how much of it still retains vegetation cover, in the context of the implications for river water quality. At micro scale we investigate the environment of the bare peat surfaces with a view to assessing their suitability for recolonisation by plants. For this site, the unpredictability of access poses a major risk to successful implementation of field-based investigations, making efficient use of fieldwork time a priority and remote observation techniques highly attractive. We conclude with some comments about the effectiveness of our methods in this context.

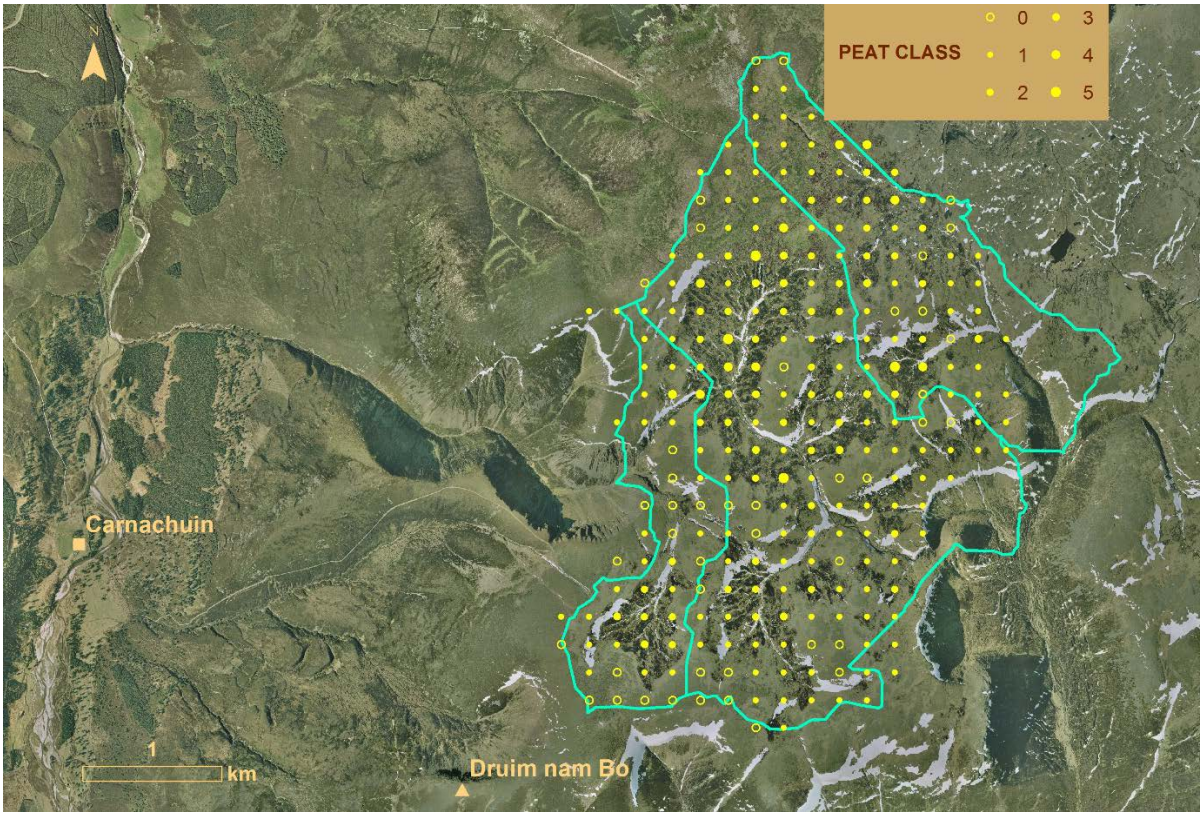


Figure 1. Aerial view of the Mòine Mhór plateau and River Feshie (flowing south–north at left side of image). Cyan polygons divide the plateau between three stream catchments, namely (from west to east) Allt Garbhlach, Coachan Dubh and Allt Sgairnich. Locations of survey quadrats are indicated by yellow points whose sizes reflect peat thickness (key to classes: 0 = no peat, 1 = 1–30 cm, 2 = 31–60 cm, 3 = 61–90 cm, 4 = 91–120 cm, 5 = 121–150 cm). The locations of Carnachuin and Druim nam Bo are also shown. Aerial imagery: Ordnance Survey, Licence No. 100030994. Catchment boundaries derived from OS Terrain 5 DTM [ASC geospatial data], Scale 1:10000, Tiles: nn89ne, nn89se, nn99sw, nn99nw, Updated: 24 March 2015, Ordnance Survey (GB), Using: EDINA Digimap Ordnance Survey Service, <http://digimap.edina.ac.uk>, Downloaded: 2015-08-04 21:44:16.33.

METHODS

A 200 m × 200 m grid of waypoints (defined in six-figure British National Grid co-ordinates, i.e. to 1 m precision) covering the approximate area of the Mòine Mhór plateau and catchment was generated in Microsoft Excel and uploaded to a hand-held GPS with integral camera (Garmin GPSMAP® 62sc). In the field, an observer navigated to each waypoint in turn. At each location a 0.5 × 0.5 m quadrat was laid on the ground, a geotagged vertical photograph of the quadrat was captured using the GPS and, if the observer had botanical knowledge (219/279 quadrats), a list of plant species occurring within the quadrat was recorded. A soil auger or metal probe was then used to determine the thickness of peat (defined as soft organic sediment ≥ 1 cm thick) at the centre of the quadrat. The field data were transcribed into an Excel spreadsheet, then converted to an ArcGIS feature class to which the geotagged quadrat photographs were spatially joined. This facilitated viewing of individual quadrat photographs in conjunction with a base map or aerial photograph, i.e. in their spatial context within the site. The quadrat photographs could then be reviewed and attributed with additional information as required. For present purposes the pertinent attributes are peat thickness and whether more than 50 % of the quadrat was occupied by bare peat.

Catchment boundaries were generated from Ordnance Survey Terrain 5 DTM (digital terrain model) data using the Spatial Analyst 'Hydrology' tools in ArcGIS 10.2.2. After removing small imperfections such as isolated 'sinks' from the data using the 'Fill' tool, flow direction and flow accumulation rasters were generated. Pour point features were placed near the lowest point of each focus stream, then located accurately as pixels in the flow accumulation raster using the 'Snap Pour Point' tool. The generated pour point raster was used in conjunction with the flow direction raster to derive catchment (Watershed) rasters which were then converted to polygons.

Peat temperatures were measured on an expanse of bare peat in the southern (upstream) part of the Allt Garbhlach catchment (Figure 1) using ten temperature sensors (1-Wire DS18B20 digital thermometer; Maxim Integrated Products, San Jose, CA, USA). Five of the sensors (Cluster 106B) were installed horizontally, supported by a strip of plastic mesh, at 10 cm depth intervals in the wall of a 5 cm × 5 cm pit excavated in a small degraded peat hagg

using a Malcolm peat sampler (Cuttle & Malcolm 1979), after which the excavated peat was carefully re-packed around them. The remaining sensors (Cluster 106A) were pushed vertically into the ground (to a depth of approximately 1 cm) at intervals of ~20 cm along a transect starting in, and running perpendicular to, a shallow runoff channel on the bare peat surface. The two sensor clusters form part of an environmental sensor network (ESN; Hart & Martinez 2006) that covers the south-western part of the Mòine Mhór plateau. Each sensor cluster is wired to a MSP430-based node running the Contiki operating system (Dunkels *et al*. 2004). The captured data are stored in flash memory, encoded using protocol buffers to save space. They are fetched hourly *via* an 868 MHz low-power (20 mW) radio network (G.M. Bragg *et al*. in press) which uses 6LowPAN and CoAP. Once on the micro-PC (border router) at Carnachuin, they are sent *via* the Estate's internet connection to the database server where they are unpacked and loaded into a database for previewing and export for analysis.

RESULTS

In total, quadrats were recorded at 279 GPS waypoints. Of these, 184 (66 %) were on vegetated peat, 51 (18 %) were on bare peat and 44 (16 %) had no peat. Where present, the thickness of the peat layer ranged from > 0 to 143 cm (mean 30 cm) (Table 1). The average peat thickness across all quadrats was 25 cm. Rather than accommodating a single expanse of peatland, the Mòine Mhór plateau turned out to consist of a series of knolls topped with *Racomitrium lanuginosum* heath on extremely shallow mineral soils and flanked by *Nardus stricta* grassland. It is probably above the altitude limit for *Calluna vulgaris* but other dwarf shrubs (e.g. *Empetrum* and *Vaccinium* species) were found, as well as *Rubus chamaemorus* growing on isolated *Sphagnum fuscum* cushions. Peatland is more or less confined to basins between the knolls which together form the headwater catchments of three streams, namely Allt Garbhlach (1.64 km^2), Coachan Dubh (5.98 km^2) and Allt Sgairnich (2.88 km^2) (Figure 1). The total number of quadrats recorded within the three stream basins was 237. When assigned to their respective catchments, the data for the 237 quadrats indicated that the relative extents of non-peatland (12–14 %) and vegetated peat (65–68 %) were similar for Coachan Dubh and Allt Sgairnich, whereas 31 % of the Allt Garbhlach catchment had no peat layer and only 55 % was covered by vegetated peat. Whilst 20–21 % of the Allt Garbhlach and Allt Sgairnich peatland had no plant cover, 26 % of the Coachan Dubh peat layer lacked vegetation. Overall, the fraction of bare peat cover *per* catchment was 14 % for Allt Garbhlach, 18 % for Allt Sgairnich and 23 % for Coachan Dubh (Table 1). These figures might provide a first basis for prioritisation of different parts of the plateau for any management intervention driven by peat erosion / water quality considerations.

Example segments of the peat temperature data delivered by Sensor Clusters 106A and 106B are shown graphically in the upper two panels of Figure 2; the lower two panels show concurrent data from an independently operating Campbell CR1000 automatic weather station (Campbell Scientific, Shepshed, UK) located on the Druim nam Bo ridge (Figure 1) that delivers data by conventional telemetry. The data period chosen runs for 21 days from 00:00 hrs on 07 November 2015, and illustrates the transition from autumn to winter conditions on the plateau. A further point of interest is that the first two Atlantic storms to be named (Abigail and Barney) by British Isles meteorologists passed over the UK during this time, and the example record ends as the third (Clodagh) was approaching (Met Office 2016). Up to and including 12 November, peat surface temperature was above 0 °C and followed the more or less diurnal fluctuations in air temperature. The coincident drop in barometric pressure and peak in air temperature that day can be linked to Storm Abigail, which delivered the heaviest rainfall of the period as well as the first dusting of snow on the summits (visible in webcam images of Druim nam Bo captured from Carnachuin around 10:00 hrs on the 13th). Thereafter, the temperature of surface peat fell to ~ 0 °C and the diurnal fluctuations ceased, presumably due to presence of snow and/or ice. The peak in air temperature that preceded Storm Barney during the 15th/16th was mirrored (only) at the three surface peat sensors located highest in the microtopography, suggesting that concave parts of the bare peat surface remained frozen. The next warming event

Table 1. Summary characteristics of the peat layer of the Mòine Mhór plateau and catchments, as indicated by the quadrat data (see text for details).

		Allt Garbhlach	Coachan Dubh	Allt Sgairnich	All quadrats
catchment	area (m^2)	1643277	5984334	2878267	
	number of quadrats	36	145	56	279
	% without peat	31	12	14	16
	% with vegetated peat	55	65	68	66
	% with bare peat	14	23	18	18
peat	mean thickness (cm ± SD)	29 ± 32	30 ± 30	32 ± 34	30 ± 31
	maximum thickness (cm)	105	143	128	143
	% of peat area bare	20	26	21	22

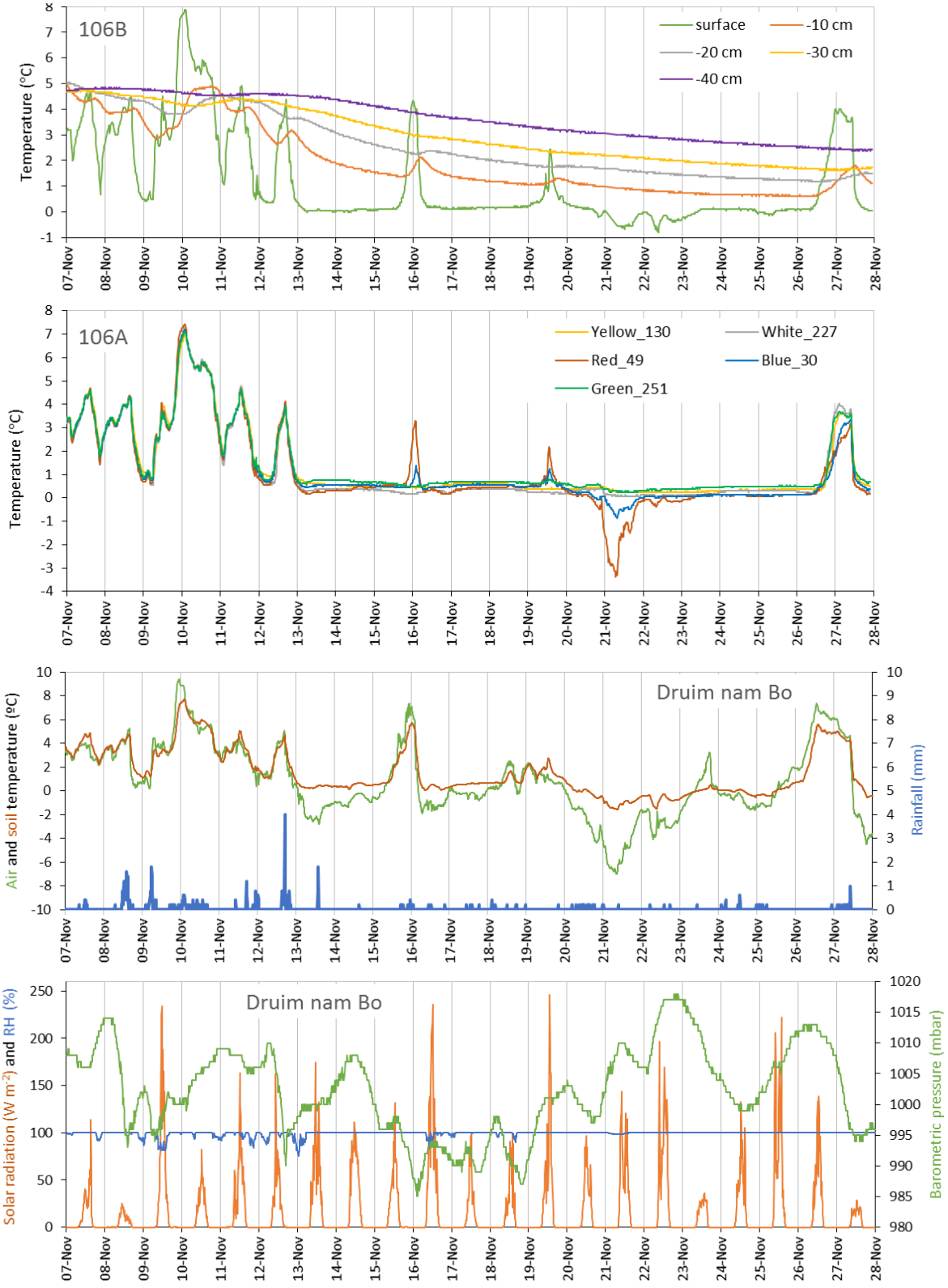


Figure 2. Environmental data for the 21-day period 07–27 November 2015 (the x-axes start at 00:00 hrs on 07 November). The data are peat temperatures from Sensor Clusters 106A and 106B (upper two panes; interval between consecutive records 20 minutes) and information gathered by the (telemetric) Wildland automatic weather station (Campbell CR1000) on Druim nam Bo (lower two panes; interval between consecutive records 15 minutes; RH = relative humidity). Notable weather events during this period were the UK's first two named storms, Abigail and Barney, which tracked over northern Scotland on 12–13 and 17–18 November, respectively, followed by Clodagh on 29 November (Met Office 2016). The sensors in Cluster 106A were positioned in surface peat along a transect (described in the text) in the following order: White_227 (lowest, in channel) - Yellow_130 - Green_251 - Blue_30 - Red_49 (highest, on the side of a peat hagg). The 'surface' sensor of Cluster 106B was located near the centre of a small degraded hagg.

detected on the peat may have been due to insolation on the 19th, as it hardly shows in the air temperature record. This time, four peat-surface sensors responded, perhaps indicating a lower thaw line than previously. Certainly, it appears that the sensor locations were differentially exposed to frosting in the early hours of the 21st as only three sensors indicated cooling. Conversely, when the next warm air mass arrived on the 26th, all surface sensors responded in a sequence that can be interpreted in terms of meltwater moving through the channel and snow at other levels requiring time to melt. The '106B' data indicate that subsurface peat cooled quite steadily from the 12th onwards, although the sensors at 10 cm and 20 cm depth showed some responses (with time lags) to warming events at the surface.

DISCUSSION AND CONCLUSIONS

The use of geotagged photographs was effective in expediting the peat/vegetation survey under circumstances of limited field time, species-poor plant communities and multiple observers with different skill levels; although there were instances where it was not easy to distinguish between graminaceous species in photographs that were not accompanied by field species lists. Nonetheless, the use of geotagging improves on the approach of e.g. Gilbert & Butt (2009) by removing the need for meticulous logging of photograph locations in the field. The intended even coverage of the stream catchments with sampling points was largely but not entirely achieved, firstly because some grid locations were too steep for observers to access safely, and secondly because the data and GIS resources needed to definitively generate the catchment boundaries were not in place before the seasonally determined time window for fieldwork. Now that the photographs have been captured there is potential to re-use them in interpretation of remotely sensed imagery of the site, which would effectively increase the fraction of ground sampled from $< 6.25 \times 10^{-4}$ % (0.25 m^2 *per* 40,000 m^2) to 100 %. The above consideration of just a segment of the peatland data captured *via* the ESN confirms the immense potential of this approach for remote data collection across inaccessible field sites in telemetry shadow under hostile conditions. It is difficult to imagine how such data could otherwise have been gathered on the Mòine Mhór through such a meteorologically eventful period. The environmental sensor network is ideal for investigations in locations like this where there is very limited mobile phone signal, as it allows the network to 'hop' from node to node to reach the field site and provides a 'live' data feed.

ACKNOWLEDGEMENTS

We thank Sangita Pandit Karki for recording part of the GPS grid; and Sebastian Bader, Emma Bryder, Arthur Fabre, Daniel Playle and Tyler Ward for their wide-ranging contributions to establishing and maintaining the Glenfeshie ESN. Sue Jones (Wildland Limited) and Kevin Down (ESRIUK) assisted with GIS resources and analysis; and Alan Long and Craig Phillips provided technical support for fieldwork. Development of the Glenfeshie ESN was funded by NERC 'Technology Proof of Concept' research award NE/L012405/1 "Using Internet of Things technology to aid in Earth and Environmental Science Research". We particularly thank Wildland Limited for permitting and supporting our research, and Wildland's Director of Conservation Thomas MacDonnell and Glenfeshie Estate staff for their valued assistance with the practicalities of site access.

REFERENCES

1. Bragg, G.M., Martinez, K., Basford, P.J. & Hart, J.K. (in press) 868MHz 6LoWPAN with ContikiMAC for an Internet of Things environmental sensor network. *Proceedings of the SAI Computing Conference 2016*, July 13–15, London, UK, 6 pp.
2. Cuttle, S.P. & Malcolm, D.C. (1979) A corer for taking undisturbed peat samples. *Plant and Soil*, 51, 297–300.
3. Dunkels, A., Gronvall, B. & Voigt, T. (2004) Contiki - a lightweight and flexible operating system for tiny networked sensors. *Proceedings of the 29th IEEE International Conference on Local Computer Networks*, Tampa, Florida, USA, 455–462.
4. Gilbert, J.A. & Butt, K.R. (2009) Evaluation of digital photography as a tool for field monitoring in potentially inhospitable environments. *Mires and Peat*, 5(05), 1–6.
5. Hart, J.K. & Martinez, K. (2006) Environmental Sensor Networks: A revolution in the Earth System Science? *Earth Science Review*s, 78, 177–191.
6. JNCC (2016) *Cairngorms*. Web page, Joint Nature Conservation Committee, Peterborough and Aberdeen, UK. Online at: http://jncc.defra.gov.uk/protectedsites/sacselection/sac.asp?EUCode=UK0016412, accessed 14 Mar 2016.
7. Met Office (2016) *UK Storm Centre*. Web page, The Meteorological Office, Exeter, UK. Online at: http://www.metoffice.gov.uk/uk-storm-centre, accessed 14 Mar 2016.
8. SNH (2016) *Site Details for Cairngorms*. Web page, Scottish National Heritage, Inverness, UK. Online at: http://gateway.snh.gov.uk/sitelink/siteinfo.jsp?pa_code=288, accessed 14 Mar 2016.
9. Thomas, C.W., Gillespie, M.R., Jordan, C. & Hall, A.M. (2004) *Geological Structure and Landscape of the Cairngorm Mountains*. Scottish Natural Heritage Commissioned Report No. 064 [ROAME No. F00AC103]. SNH, Edinburgh, UK, 128 pp.

(Martinez, Hart et al. 2016)

An Environment IoT Sensor Network for Monitoring the Environment

Martinez, K.; Hart, J. K.; Bragg, O.; Black, A.; Bader, S.; Basford, P. J.; Bragg, G. M.; Fabre, A.

The Internet of Things is a term which has emerged to describe the increase of Internet connectivity of everyday objects. While wireless sensor networks have developed highly energy efficient designs they need a step-change in their interoperability and usability to become more commonly used in Earth Science. IoT techniques can bring many of these advances while reusing some of the technologies developed for low power sensing. Here we concentrate on developing effective use of internet protocols throughout a low power sensor network. This includes 6LowPAN to provide a mesh IPv6 network, 40mW 868 MHz CC1120 radio transceivers to save power but provide kilometre range, a CC2538 ARM® Cortex®-M3 as main processor and CoAP to provide a binary HTTP-like interface to the nodes. We discuss in detail a system we deployed to monitor periglacial, peat and fluvial processes in the Scottish Highlands. The system linked initial nodes 3km away further up the mountain 2km away and used a CoAP GET sequence from a base station in the valley to gather the data. The IPv6 addressing and tunnelling allowed direct connectivity to desktops in Southampton. This provides insights into how the combination of low power techniques and emerging internet standards will bring advantages in interoperability, heterogeneity, usability and maintainability.

(Martinez, Hart et al. 2013)

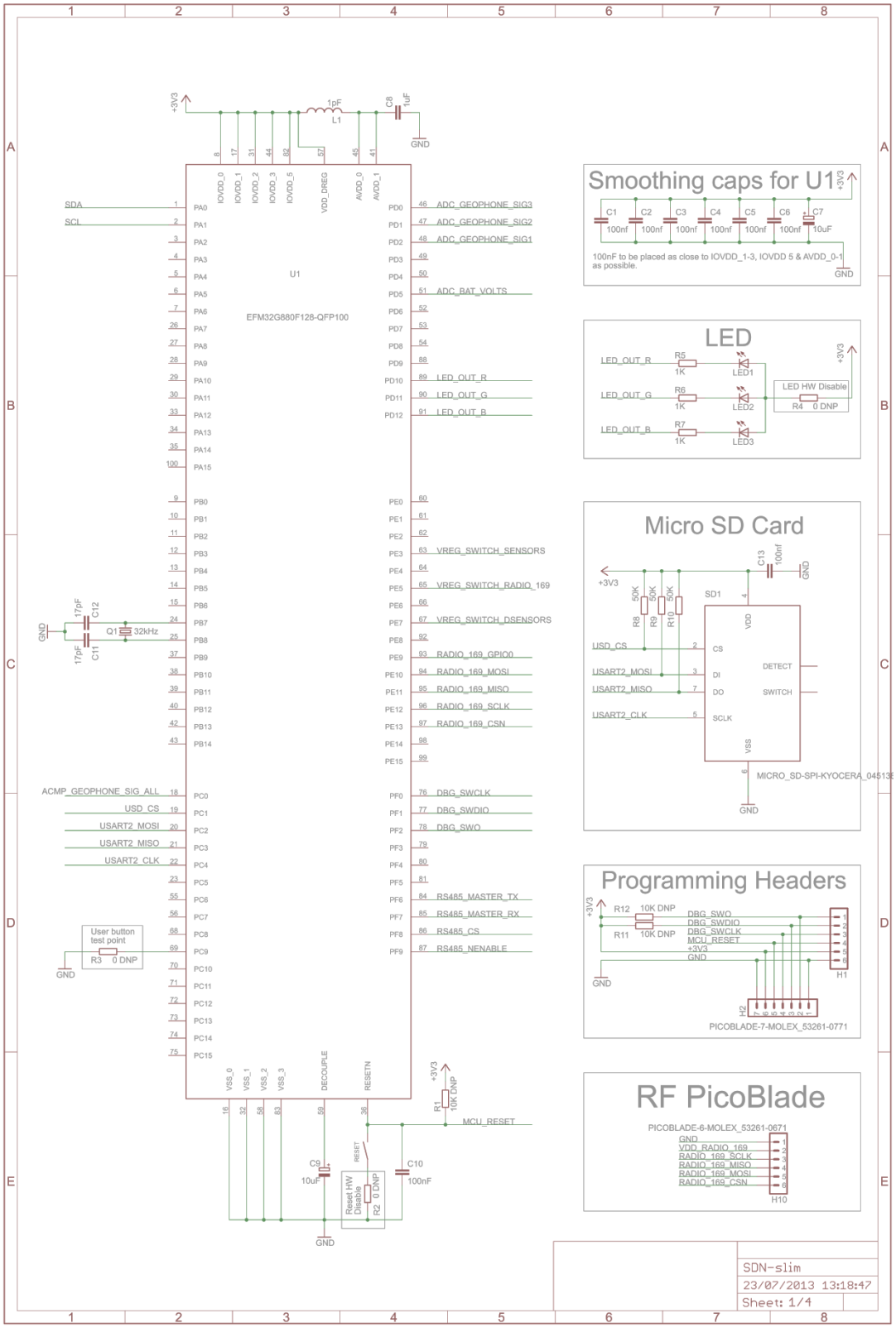
Using Internet of Things technologies for wireless sensor networks

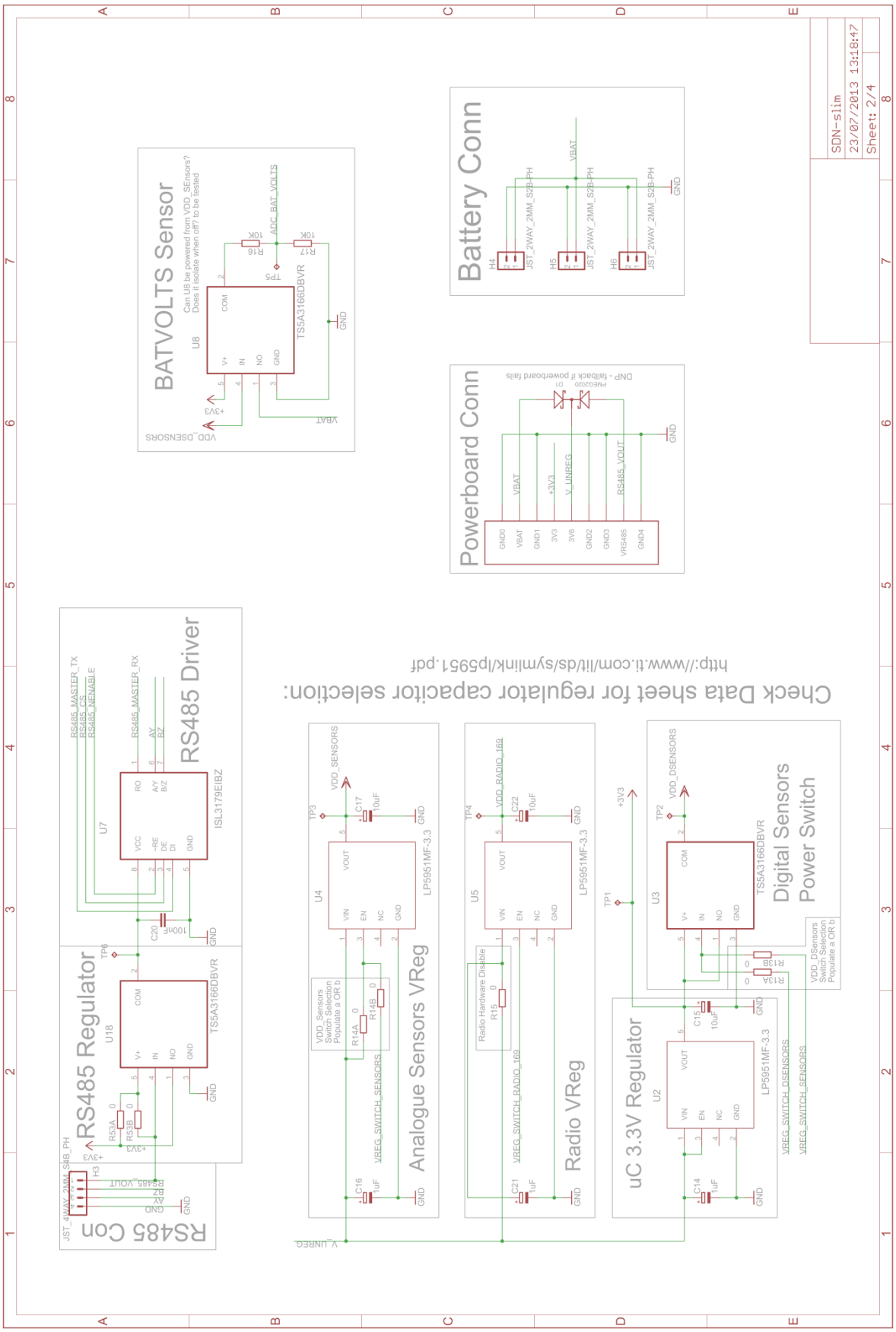
Martinez, K.; Hart, J. K.; Basford, P. J.; Bragg, G. M.; Ward, T.

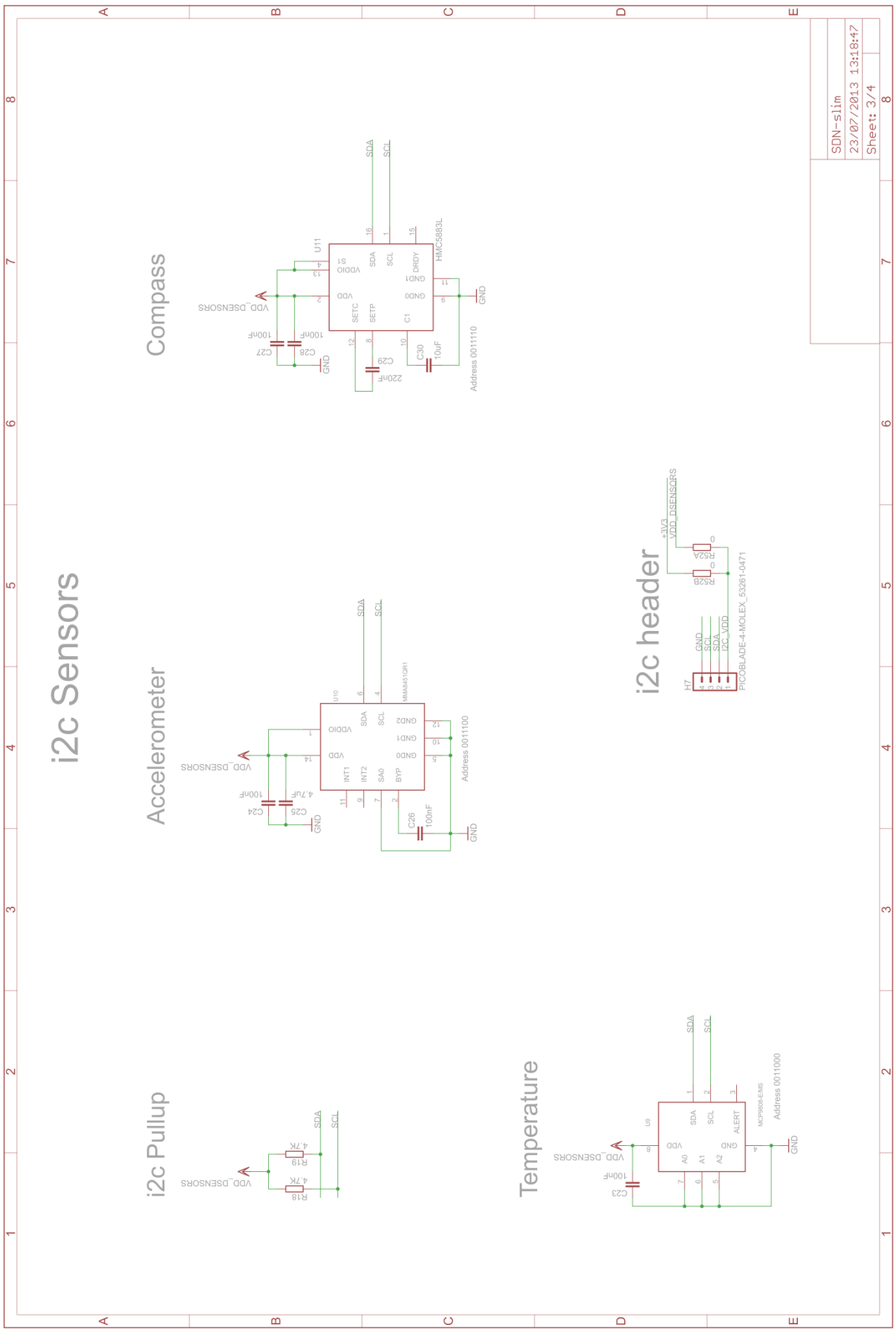
Numerous authors have envisioned the future internet where anything will be connected: the Internet of Things (IoT). The idea is an extrapolation of the spread of networked devices such as phones, tablets etc. Each device is expected to have its own Internet address and thus be easy to access. The key building blocks of any IoT system are networking, hardware platforms and node software - so they are similar to wireless sensor network requirements. Most existing IoT demonstrators and applications have been gadget-style objects where power and connectivity problems are not too restricting. Environmental sensor networks can benefit from using some of the technologies involved in IoT development. However it is expected that tuning the networking and power management will be necessary to make them as effcient as state of the art wireless sensor networks. Some IoT assumptions such as always-connected nodes and full IP capability need to be considered. This paper will illustrate the advantages and disadvantages of IoT techniques for environment sensing drawing on a range of employment scenarios. We also describe a glacial 'Internet of things' project, which aims to monitor glacial processes. In particular we describe the IoT developments in a deployment in Iceland to examine glacier seismicity, velocity and provide camera images.

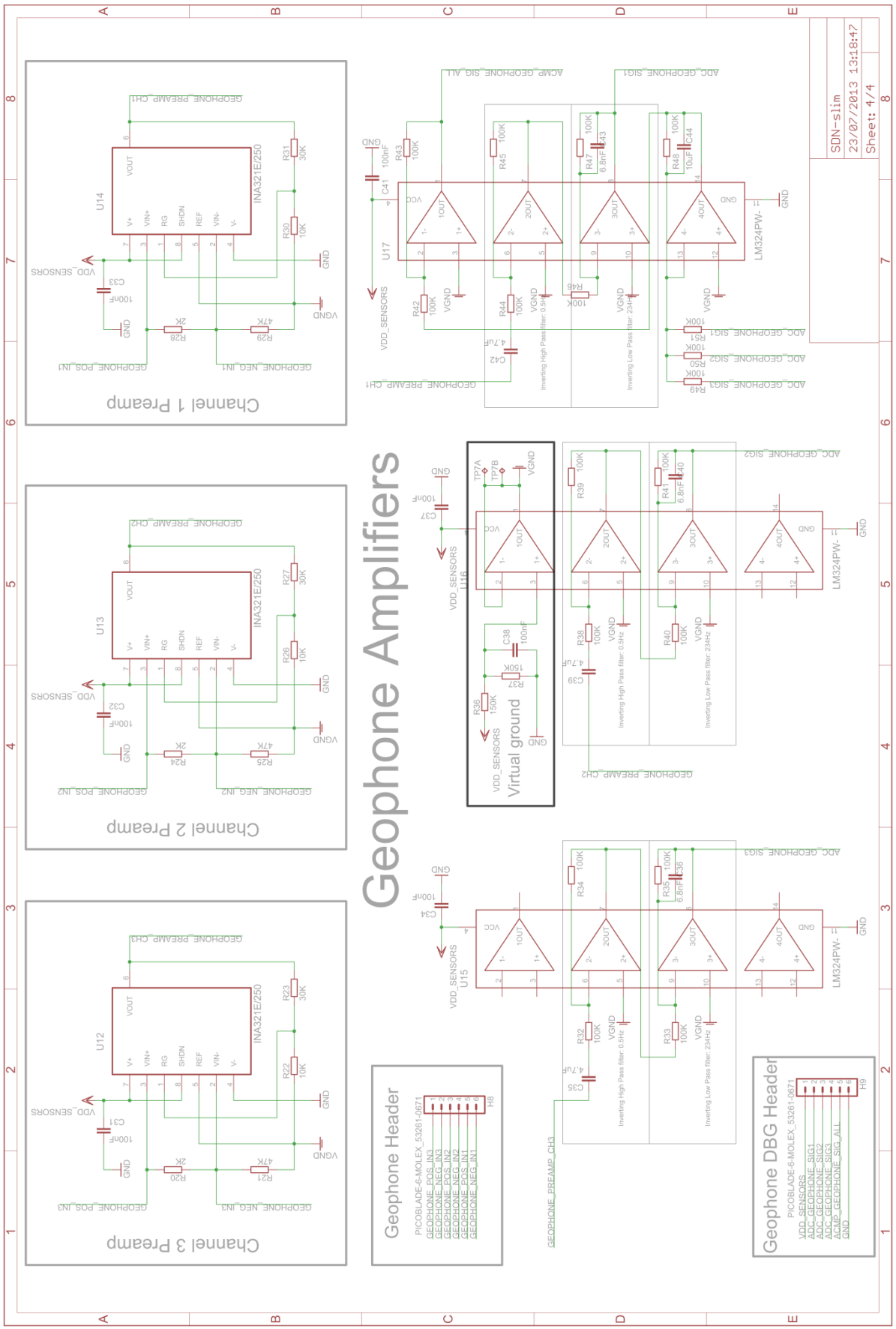
Appendix 2. SSDN Schematics

A 2.1. SSDN Schematics

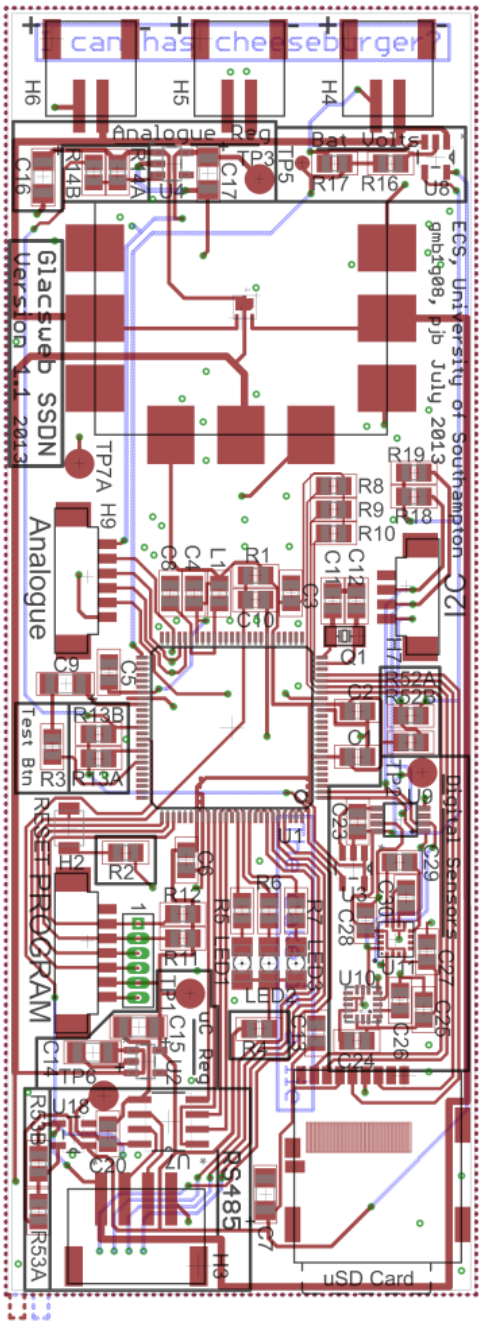



SDN-slim
23/07/2013 13:18:47
Sheet: 2/4
BATVOLTS Sensor
Can U8 be powered from VDD_SEnsors?
Does it isolate when off? to be tested
U8
TS5A3166DBVR
ADC_BAT_VOLTS
VDD_DSENSORS
VBAT
Battery Conn
JST_2WAY_2MM_S2B-PH
Powerboard Conn
DNP - fallback if powerboard fails
PMEG2020
VBAT
+3V3
V_UNREG
RS485_VOUT
GND
Check Data sheet for regulator capacitor selection:
http://www.ti.com/lit/ds/symlink/lp5951.pdf
RS485 Driver
ISL3179EIBZ
RS485_MASTER_TX
RS485_CS
RS485_ENABLE
RS485_MASTER_RX
RS485 Regulator
RS485 Con
Analogue Sensors VReg
LP5951MF-3.3
VDD_SENSORS
VREG_SWITCH_SENSORS
Radio VReg
VDD_RADIO_169
VREG_SWITCH_RADIO_169
Radio Hardware Disable
uC 3.3V Regulator
Digital Sensors Power Switch
VDD_DSensors Switch Selection Populate a OR b
VREG_SWITCH_DSENSORS

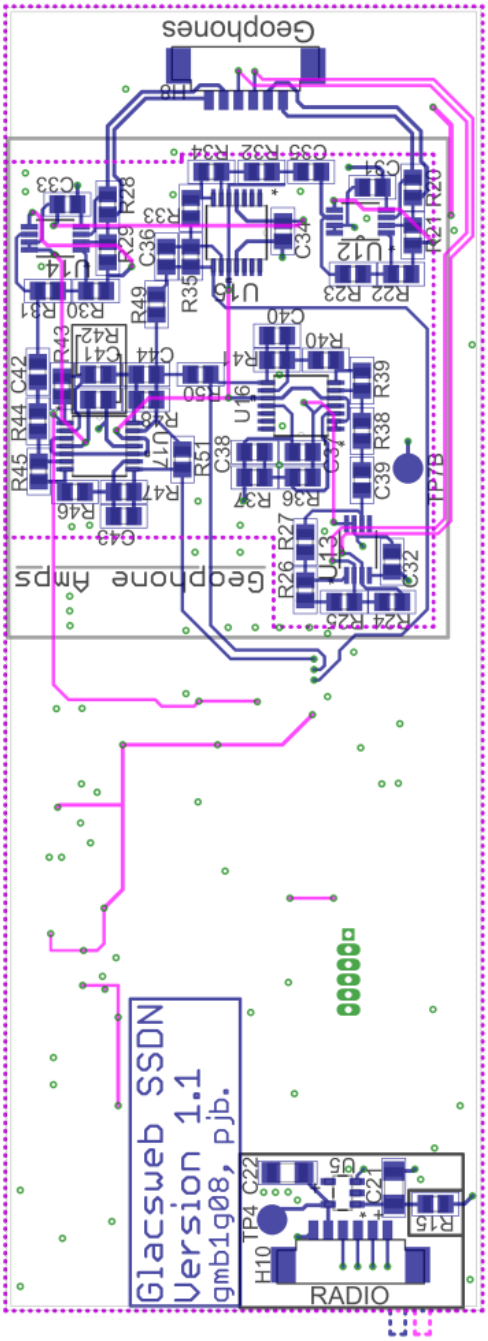

i2c Sensors
Compass
Accelerometer
i2c header
i2c Pullup
Temperature
SDN-slim
23/07/2013 13:18:47
Sheet: 3/4


Geophone Amplifiers
SDN-slim
23/07/2013 13:18:47
Sheet: 4/4
Channel 1 Preamp
Channel 2 Preamp
Channel 3 Preamp
Virtual ground
Geophone Header
Geophone DBG Header
INA321E/250
LM324PW-
VDD_SENSORS
GEOPHONE_PREAMP_CH1
GEOPHONE_PREAMP_CH2
GEOPHONE_PREAMP_CH3
ADC_GEOPHONE_SIG1
ADC_GEOPHONE_SIG2
ADC_GEOPHONE_SIG3
ACMP_GEOPHONE_SIG_ALL
GEOPHONE_PREAMP_CH3
PICOBLADE-6-MOLEX_53261-0671

A 2.2. SSDN PCB Layout

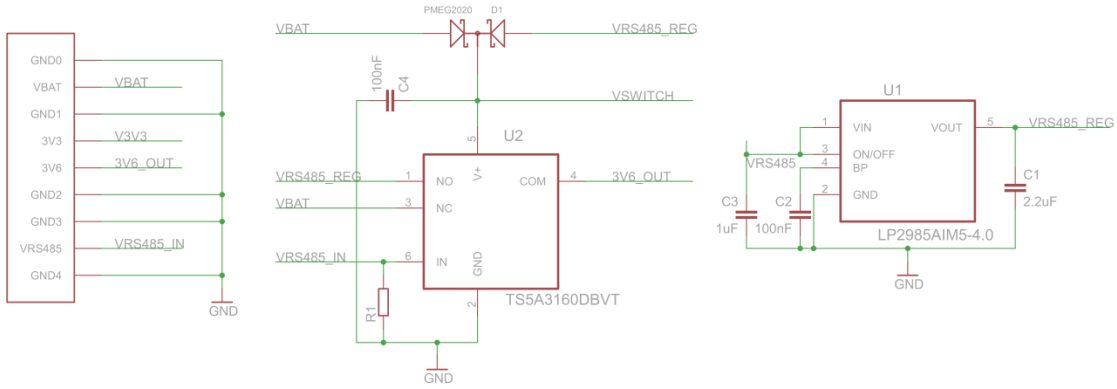


SSDN PCB top and inner-top layers

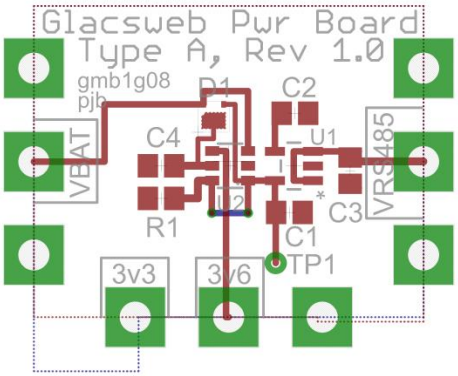


SSDN PCB bottom and inner-bottom layers

A 2.3. SSDN Power Board Schematic



A 2.4. SSDN Power Board Layout



Appendix 3. CC1120 Driver & Platform Code

The CC1120 driver code consists of three C header files, the main driver code, one implementation-specific C source file and a platform-specific C source file. The following table details these files. The driver has been extended to support the CC1200 and this requires two additional files.

Filename	**Description**	**Platform Specific?**
cc1120.c[2] cc1120.h[3]	Main platform-agnostic CC1120 driver code.	N
cc1120-config.h[4]	Header for implementation-specific CC1120 configuration.	N
cc1120-arch.h[5]	Header for platform-specific CC1120 functions	N
cc1120-const.h[6]	CC1120 constants. Includes register memory addresses, states, etc.	N
cc1120-config.c[7]	Configures the CC1120 to match a configuration exported from Texas Instruments SmartRF Studio. This file can be generated with Smart RF using a custom export template.	N
cc1200-const.h[8]	CC1200 constants. Includes register memory addresses, states, etc.	N
cc1200-config.c[9]	Configures the CC1200 to match a configuration exported from Texas Instruments SmartRF Studio. This file can be generated with Smart RF using a custom export template	N
cc1120-arch-z1.c[10]	CC1120 functions specific to the Zolertia-Z1 implementation used for testing and deployment.	Y
cc1120-zoul-arch.c[11]	CC1120 functions specific to the CC2538-based platform used for the Mountain Sensing project	Y

[2] https://github.com/feshie/contiki/blob/mountain-sensing-master/dev/cc1120/cc1120.c

[3] https://github.com/feshie/contiki/blob/mountain-sensing-master/dev/cc1120/cc1120.h

[4] https://github.com/feshie/contiki/blob/mountain-sensing-master/dev/cc1120/cc1120-config.h

[5] https://github.com/feshie/contiki/blob/mountain-sensing-master/dev/cc1120/cc1120-arch.h

[6] https://github.com/feshie/contiki/blob/mountain-sensing-master/dev/cc1120/cc1120-const.h

[7] https://github.com/feshie/contiki/blob/mountain-sensing-master/dev/cc1120/cc1120-config.c

[8] https://github.com/feshie/contiki/blob/mountain-sensing-master/dev/cc1120/cc1200-const.h

[9] https://github.com/feshie/contiki/blob/mountain-sensing-master/dev/cc1120/cc1200-config.c

[10] https://github.com/feshie/contiki/blob/mountain-sensing-master/platform/z1-feshie/cc1120-arch-z1.c

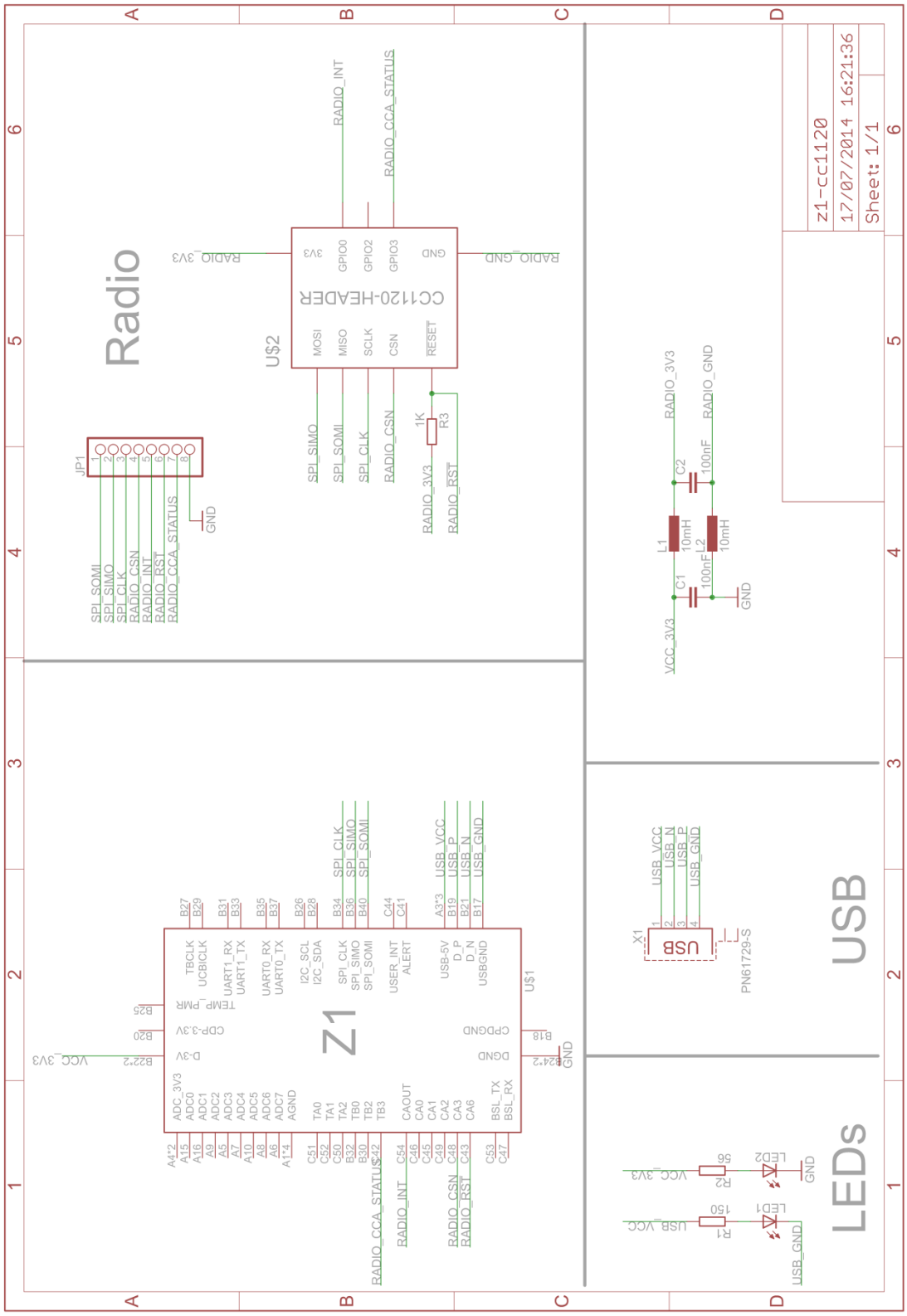
[11] https://github.com/feshie/contiki/blob/mountain-sensing-master/platform/zoul/dev/cc1120-zoul-arch.c

There are also several parameters that need to be defined in the platform's configuration headers to ensure correct operation of the driver. These are detailed in the table below.

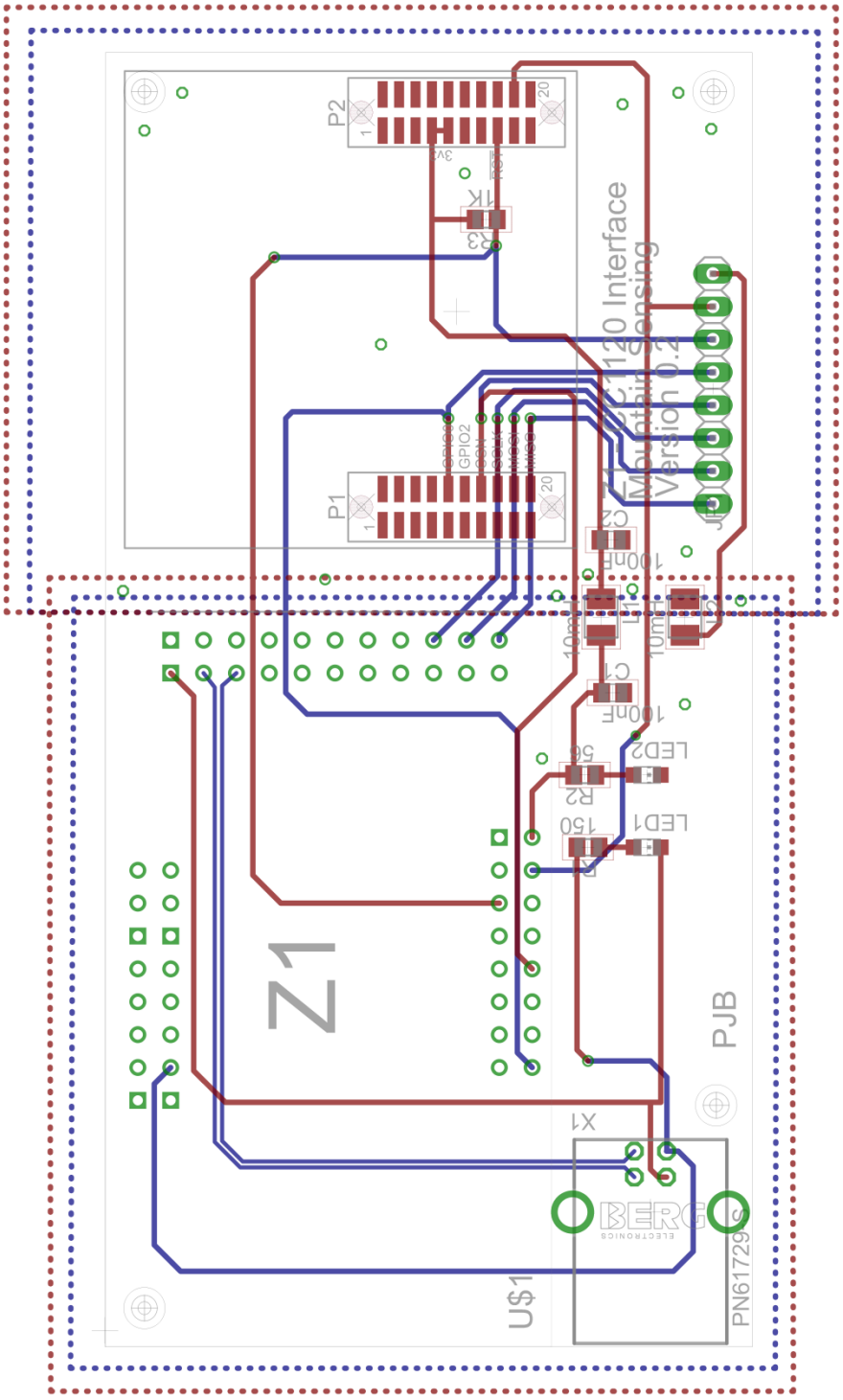
Defined Parameter	**Type**	**Description**
RF_CHANNEL	Int, 0 to 50	The channel number of operation. Sets the operating frequency with reference to ETSI or FCC frequency allocations. Channel 42 is equal to 868.250MHz.
CC1120_CS_THRESHOLD	8-bit int	The carrier sense threshold for determining whether the channel is clear. 0x9C corresponds to a threshold of -100dBm.
CC1120LEDS	Bool	Sets whether LEDs should be used to indicate CC1120 radio activity.
CC1120_LBT_TIMEOUT	Int	When using the in-built LBT function for channel access, this sets the amount of time that the driver will wait before aborting the transmission.
CC1120_INTER_PACKET_INTERVAL	Int	Sets the amount of time between packets.
CC1120_EN_TIMEOUT	Int	Used by the platform-specific code to set how long the driver will wait for the CC1120 to indicate that it is ready after an enable.
CC1120_FHSS_ETSI_50 ***OR*** CC1120_FHSS_FCC_50	Bool	Sets the operating frequency range. ETSI uses the 863MHz band, FCC uses the 915MHz band.
CC1120_OFF_STATE	Int	Sets the state that the CC1120 enters when "off".
CC1120_GPIO0_FUNC	Int	Sets the function of the GPIO0 pin. Should be set to MCU Wakeup when using interrupts.
CC1120_GPIO2_FUNC	Int	Sets the function of the GPIO2 pin. (Optional)
CC1120_GPIO3_FUNC	Int	Sets the function of the GPIO3 pin. (Optional)

Appendix 4. CC1120 Interface Board Schematics

A 4.1. Z1 CC1120 Interface Board Schematic



A 4.2. Z1 CC1120 Interface Board PCB Layout



Appendix 5. Modified UDP Client & Server Code

A 5.1. Modified UDP Server Code

```
/*
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the Institute nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * This file is part of the Contiki operating system.
 *
 */

#include "contiki.h"
#include "contiki-lib.h"
#include "contiki-net.h"

#include <string.h>
#include <stdlib.h>

#define DEBUG DEBUG_PRINT
#include "net/uip-debug.h"

#define UIP_IP_BUF ((struct uip_ip_hdr *)&uip_buf[UIP_LLH_LEN])
```

```
#define MAX_PAYLOAD_LEN 1000

static struct uip_udp_conn *server_conn;

PROCESS(udp_server_process, "UDP server process");
AUTOSTART_PROCESSES(&resolv_process,&udp_server_process);
/*---------------------------------------------------------------------------*/
static void
tcpip_handler(void)
{
 static uint16_t seqr, seqp = 0;
 static int seq_id, len;
 char buf[MAX_PAYLOAD_LEN];

 if(uip_newdata()) {
 ((char *)uip_appdata)[4] = 0;
 seqr = strtol((char *)uip_appdata, NULL, 16);
 if(seqr != (seqp + 1)) {
 printf("\t MISSING PACKETS: %d", (seqr - (seqp + 1)));
 PRINTF("\n");
 }

 seqp = seqr;

 /* Restore server connection to allow data from any node */
 memset(&server_conn->ripaddr, 0, sizeof(server_conn->ripaddr));
 }
}
/*---------------------------------------------------------------------------*/
static void
print_local_addresses(void)
{
 int i;
 uint8_t state;

 PRINTF("Server IPv6 addresses: ");
 for(i = 0; i < UIP_DS6_ADDR_NB; i++) {
 state = uip_ds6_if.addr_list[i].state;
 if(uip_ds6_if.addr_list[i].isused &&
 (state == ADDR_TENTATIVE || state == ADDR_PREFERRED)) {
 PRINT6ADDR(&uip_ds6_if.addr_list[i].ipaddr);
 PRINTF("\n");
 }
 }
```

```
}
/*---------------------------------------------------------------------------*/
PROCESS_THREAD(udp_server_process, ev, data)
{
#if UIP_CONF_ROUTER
 uip_ipaddr_t ipaddr;
#endif /* UIP_CONF_ROUTER */

 PROCESS_BEGIN();
 PRINTF("UDP server started\n");

#if RESOLV_CONF_SUPPORTS_MDNS
 resolv_set_hostname("contiki-udp-server");
#endif

#if UIP_CONF_ROUTER
 uip_ip6addr(&ipaddr, 0xaaaa, 0, 0, 0, 0, 0, 0, 0);
 uip_ds6_set_addr_iid(&ipaddr, &uip_lladdr);
 uip_ds6_addr_add(&ipaddr, 0, ADDR_AUTOCONF);
#endif /* UIP_CONF_ROUTER */

 print_local_addresses();

 server_conn = udp_new(NULL, UIP_HTONS(3001), NULL);
 udp_bind(server_conn, UIP_HTONS(3000));

 while(1) {
 PROCESS_YIELD();
 if(ev == tcpip_event) {
 tcpip_handler();
 }
 }

 PROCESS_END();
}
/*---------------------------------------------------------------------------*/
```

A 5.2. Modified UDP Client Code

```
/*
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the Institute nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * This file is part of the Contiki operating system.
 *
 */

#include "contiki.h"
#include "contiki-lib.h"
#include "contiki-net.h"
#include "net/resolv.h"

#include <string.h>
#include <stdbool.h>

#define DEBUG DEBUG_PRINT
#include "net/uip-debug.h"

#define SEND_INTERVAL 0
#define MAX_PAYLOAD_LEN 500
```

```
static struct uip_udp_conn *client_conn;
/*---------------------------------------------------------------------------*/
PROCESS(udp_client_process, "UDP client process");
AUTOSTART_PROCESSES(&resolv_process,&udp_client_process);
/*---------------------------------------------------------------------------*/
static void
tcpip_handler(void)
{


}
/*---------------------------------------------------------------------------*/
static char buf[MAX_PAYLOAD_LEN];
static uint16_t len;
static void
timeout_handler(void)
{
 static uint16_t seq_id;
 char buf2[5];

 /* 70-byte payload. */
 sprintf(buf2, "%04x", ++seq_id);
 buf[0] = buf2[0];
 buf[1] = buf2[1];
 buf[2] = buf2[2];
 buf[3] = buf2[3];

 //printf(buf);
 //printf("\n\r");
 uip_udp_packet_send(client_conn, buf, len);
}
/*---------------------------------------------------------------------------*/
static void
print_local_addresses(void)
{
 int i;
 uint8_t state;

 PRINTF("Client IPv6 addresses: ");
 for(i = 0; i < UIP_DS6_ADDR_NB; i++) {
 state = uip_ds6_if.addr_list[i].state;
 if(uip_ds6_if.addr_list[i].isused &&
 (state == ADDR_TENTATIVE || state == ADDR_PREFERRED)) {
 PRINT6ADDR(&uip_ds6_if.addr_list[i].ipaddr);
 PRINTF("\n");
```

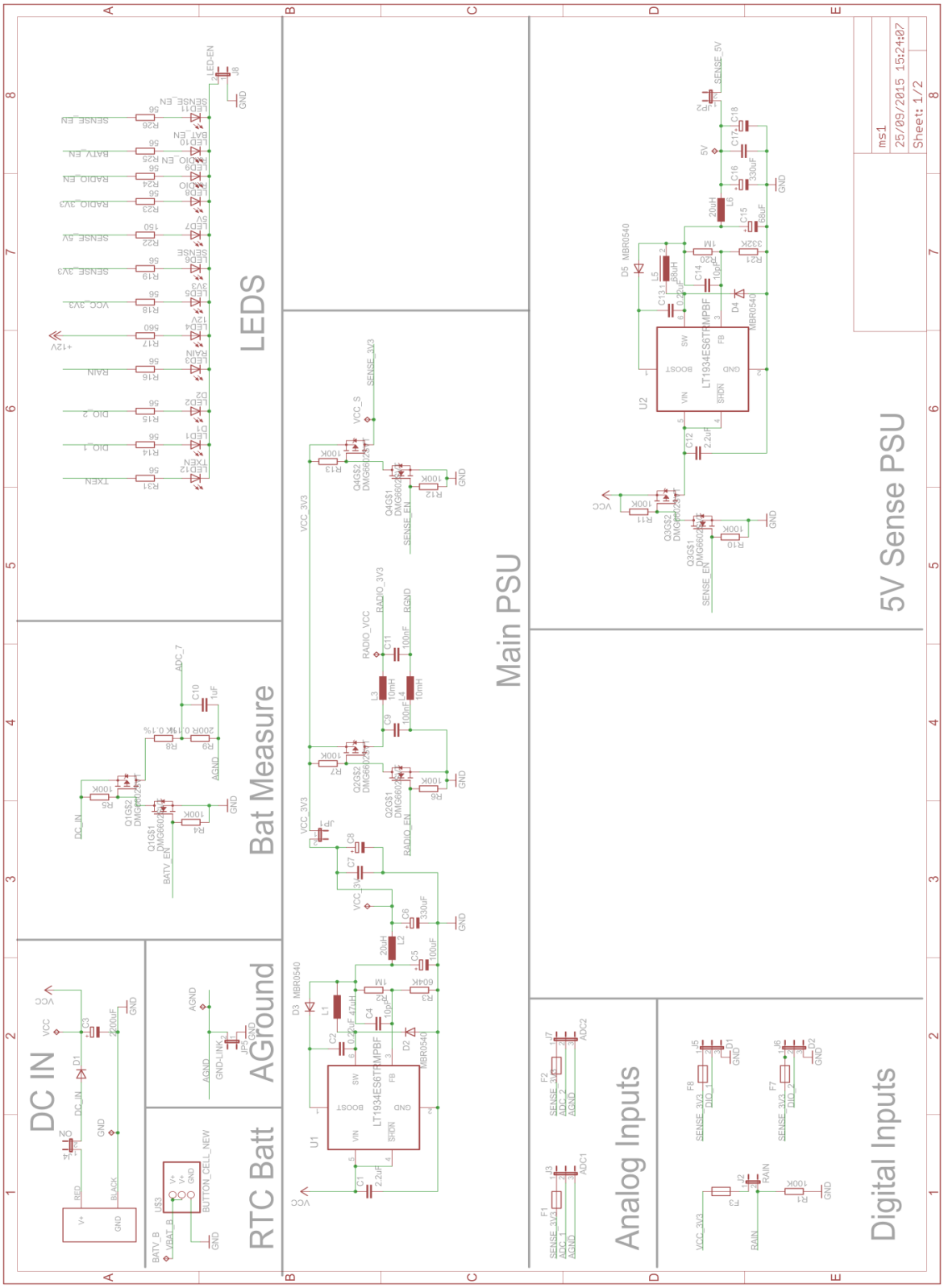
```
 }
 }
}
/*---------------------------------------------------------------------------*/
#if UIP_CONF_ROUTER
static void
set_global_address(void)
{
 uip_ipaddr_t ipaddr;

 uip_ip6addr(&ipaddr, 0xaaaa, 0, 0, 0, 0, 0, 0);
 uip_ds6_set_addr_iid(&ipaddr, &uip_lladdr);
 uip_ds6_addr_add(&ipaddr, 0, ADDR_AUTOCONF);
}
#endif /* UIP_CONF_ROUTER */
/*---------------------------------------------------------------------------*/
static resolv_status_t
set_connection_address(uip_ipaddr_t *ipaddr)
{
#ifndef UDP_CONNECTION_ADDR
#if RESOLV_CONF_SUPPORTS_MDNS
#define UDP_CONNECTION_ADDR contiki-udp-server.local
#elif UIP_CONF_ROUTER
#define UDP_CONNECTION_ADDR aaaa:0:0:0:0212:7404:0004:0404
#else
#define UDP_CONNECTION_ADDR fe80:0:0:0:6466:6666:6666:6666
#endif
#endif /* !UDP_CONNECTION_ADDR */

#define _QUOTEME(x) #x
#define QUOTEME(x) _QUOTEME(x)

 resolv_status_t status = RESOLV_STATUS_ERROR;

 if(uiplib_ipaddrconv(QUOTEME(UDP_CONNECTION_ADDR), ipaddr) == 0) {
 uip_ipaddr_t *resolved_addr = NULL;
 status = resolv_lookup(QUOTEME(UDP_CONNECTION_ADDR),&resolved_addr);
 if(status == RESOLV_STATUS_UNCACHED || status == RESOLV_STATUS_EXPIRED) {
 PRINTF("Attempting to look up %s\n",QUOTEME(UDP_CONNECTION_ADDR));
 resolv_query(QUOTEME(UDP_CONNECTION_ADDR));
 status = RESOLV_STATUS_RESOLVING;
 } else if(status == RESOLV_STATUS_CACHED && resolved_addr != NULL) {
 PRINTF("Lookup of \"%s\" succeded!\n",QUOTEME(UDP_CONNECTION_ADDR));
 } else if(status == RESOLV_STATUS_RESOLVING) {
 PRINTF("Still looking up \"%s\"...\n",QUOTEME(UDP_CONNECTION_ADDR));
```

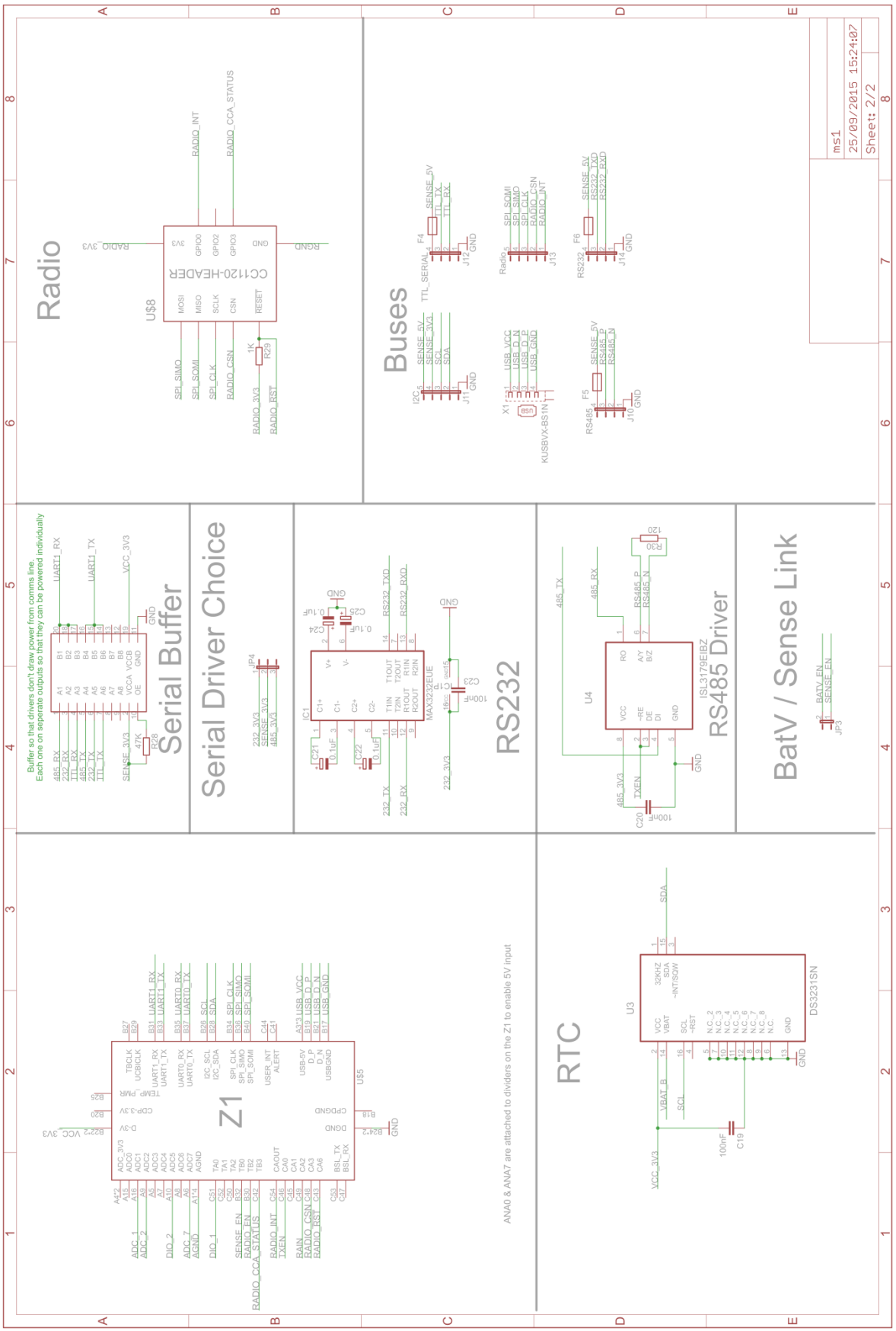
```
 } else {
 PRINTF("Lookup of \"%s\" failed. status =
%d\n",QUOTEME(UDP_CONNECTION_ADDR),status);
 }
 if(resolved_addr)
 uip_ipaddr_copy(ipaddr, resolved_addr);
 } else {
 status = RESOLV_STATUS_CACHED;
 }

 return status;
}
/*---------------------------------------------------------------------------*/
PROCESS_THREAD(udp_client_process, ev, data)
{
 static struct etimer et;
 uip_ipaddr_t ipaddr;

 PROCESS_BEGIN();
 PRINTF("UDP client process started\n");

#if UIP_CONF_ROUTER
 set_global_address();
#endif

 print_local_addresses();

 static resolv_status_t status = RESOLV_STATUS_UNCACHED;
 while(status != RESOLV_STATUS_CACHED) {
 status = set_connection_address(&ipaddr);

 if(status == RESOLV_STATUS_RESOLVING) {
 PROCESS_WAIT_EVENT_UNTIL(ev == resolv_event_found);
 } else if(status != RESOLV_STATUS_CACHED) {
 PRINTF("Can't get connection address.\n");
 PROCESS_YIELD();
 }
 }

 sprintf(buf, "%04x bytes from client. 71 Bytes of data total: 1234567690. abcd
efgh i", len);
 len = strlen(buf);

 /* new connection with remote host */
 client_conn = udp_new(&ipaddr, UIP_HTONS(3000), NULL);
```

```
 udp_bind(client_conn, UIP_HTONS(3001));

 PRINTF("Created a connection with the server ");
 PRINT6ADDR(&client_conn->ripaddr);
 PRINTF(" local/remote port %u/%u\n",
 UIP_HTONS(client_conn->lport), UIP_HTONS(client_conn->rport));

 etimer_set(&et, SEND_INTERVAL);
 while(1) {
 PROCESS_YIELD();
 if(etimer_expired(&et)) {
 timeout_handler();
 etimer_restart(&et);
 }
 }

 PROCESS_END();
}
/*---------------------------------------------------------------------------*/
```

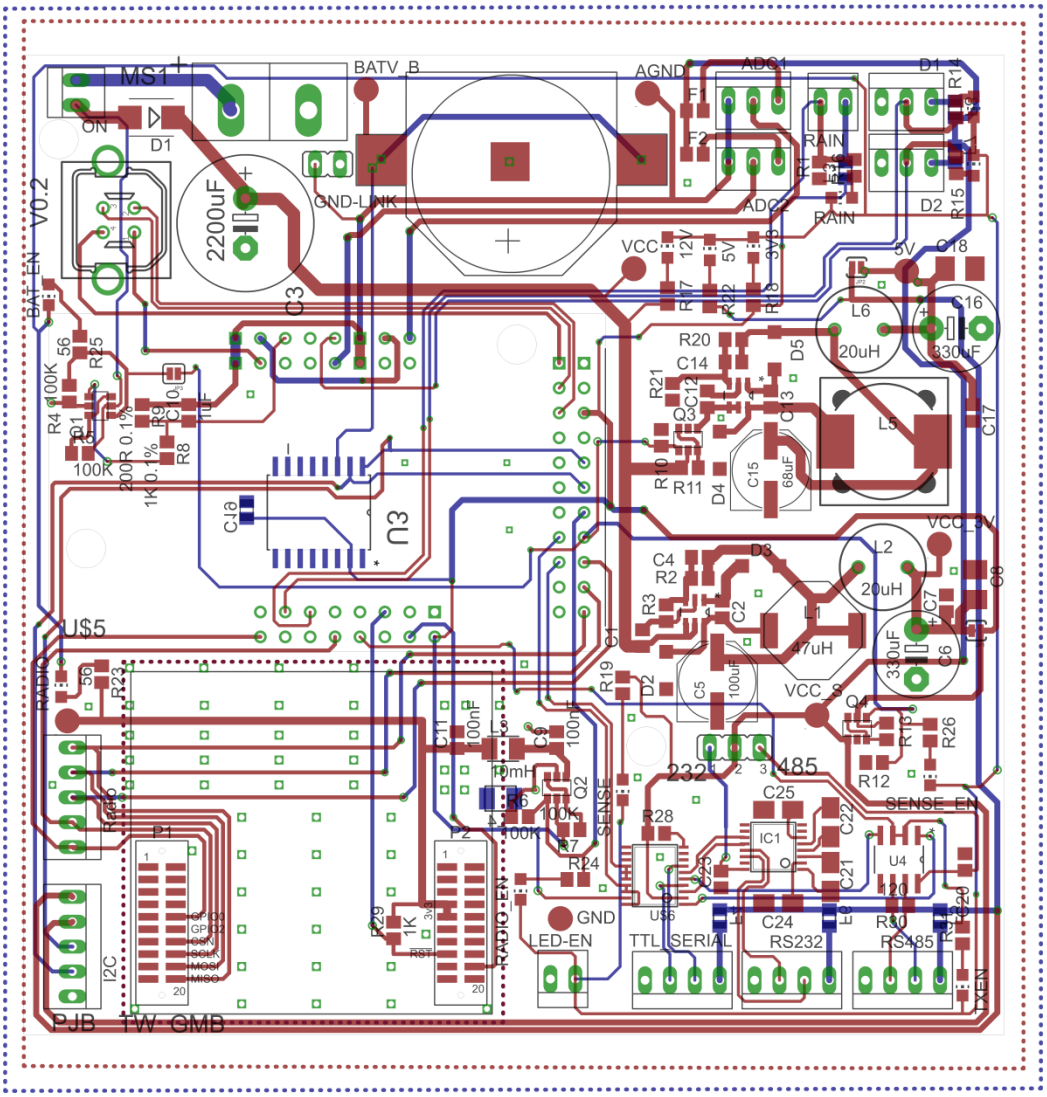
Appendix 6. MS1 Sensor Node Schematics

A 6.1. MS1 Schematics




Radio
Buses
Serial Buffer
Serial Driver Choice
RS232
RS485 Driver
BatV / Sense Link
RTC
Z1
ms1
25/09/2015 15:24:07
Sheet: 2/2
Buffer so that drivers don't draw power from comms line.
Each one on seperate outputs so that they can be powered individually
ANA0 & ANA7 are attached to dividers on the Z1 to enable 5V input
CC1120-HEADER
ISL3179EIBZ
MAX3232EUE
DS3231SN

A 6.2. MS1 PCB Layout



Appendix 7. Mountain Sensing Code

All of the code developed for the Mountain Sensing project is available on the project's Github respository at https://github.com/feshie/contiki.

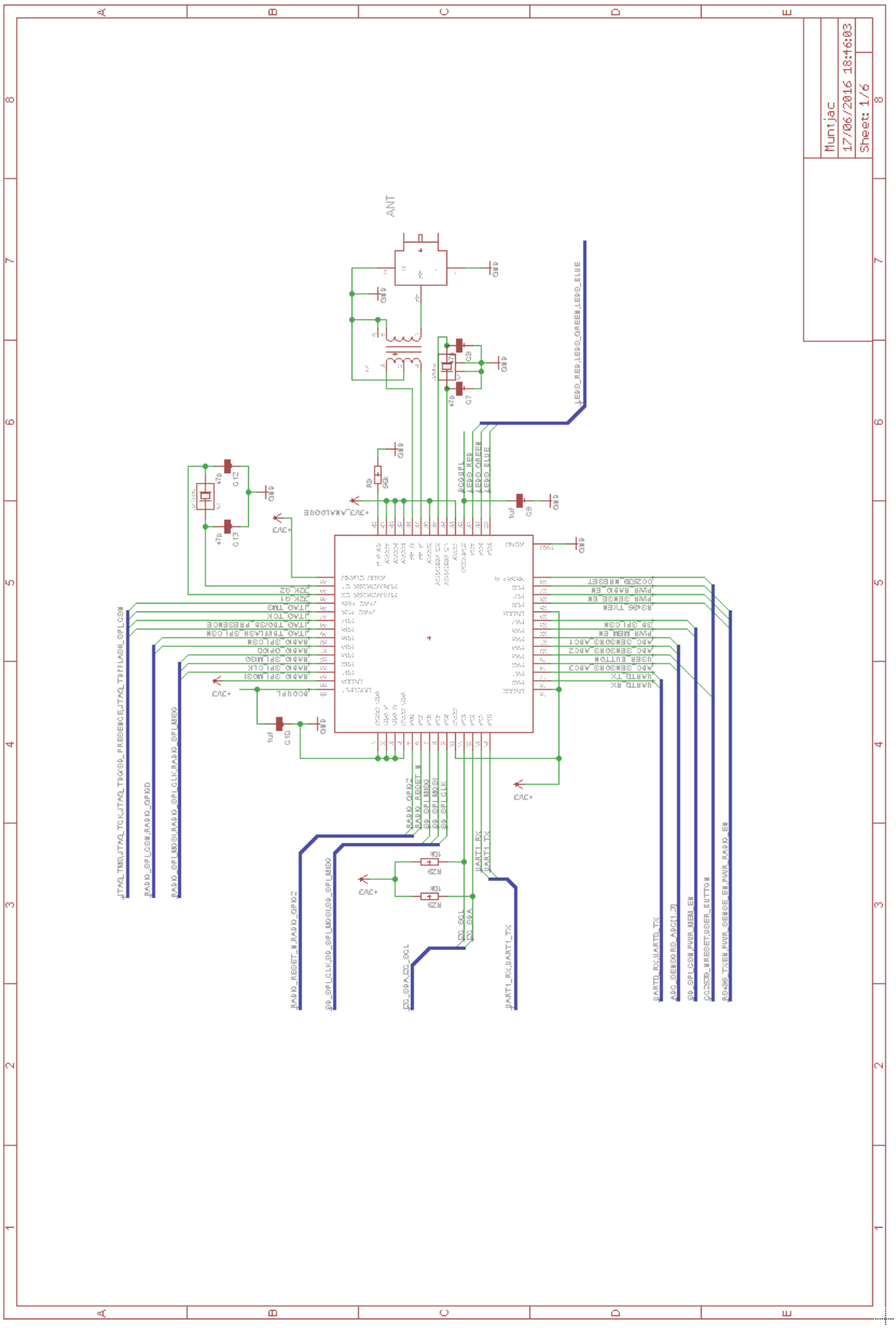
M0+ Partial Port:
A port to a platform based on an FRDM-KL25Z development board connected to a CC1120.

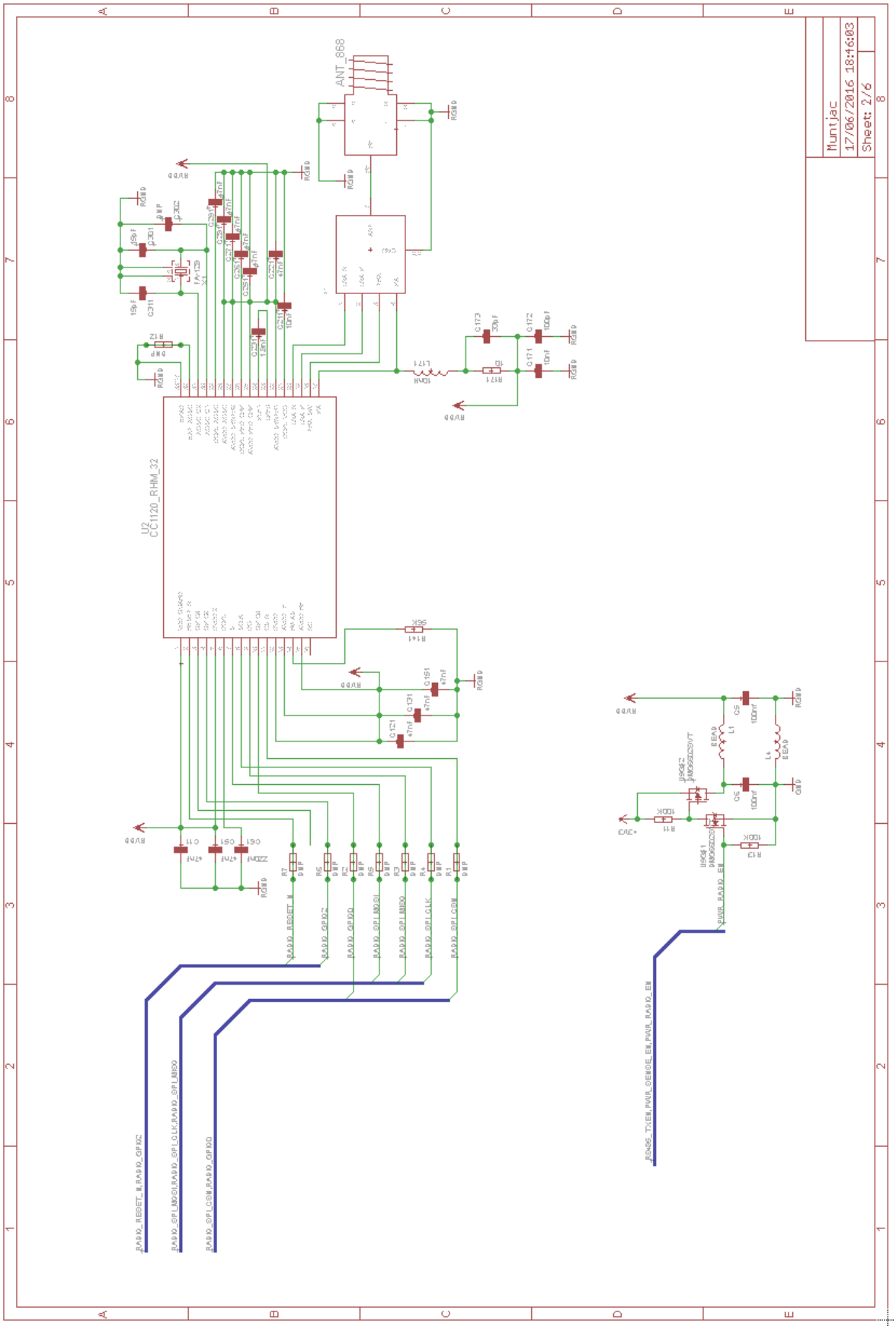
CPU Port:
https://github.com/feshie/contiki/tree/mountain-sensing-master/cpu/arm/mkl25z

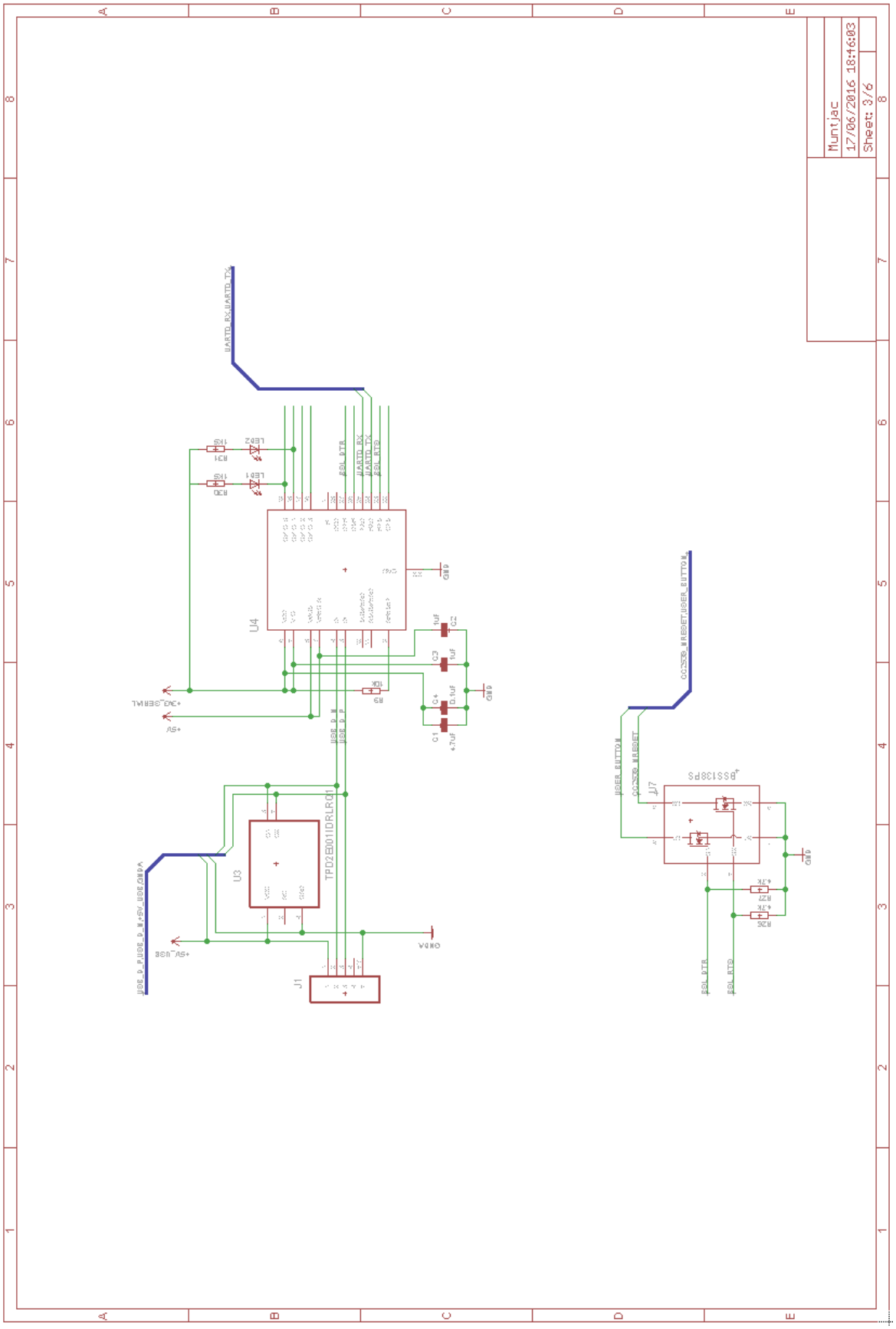
Platform:
https://github.com/feshie/contiki/tree/mountain-sensing-master/platform/frdm-kl25z-rf

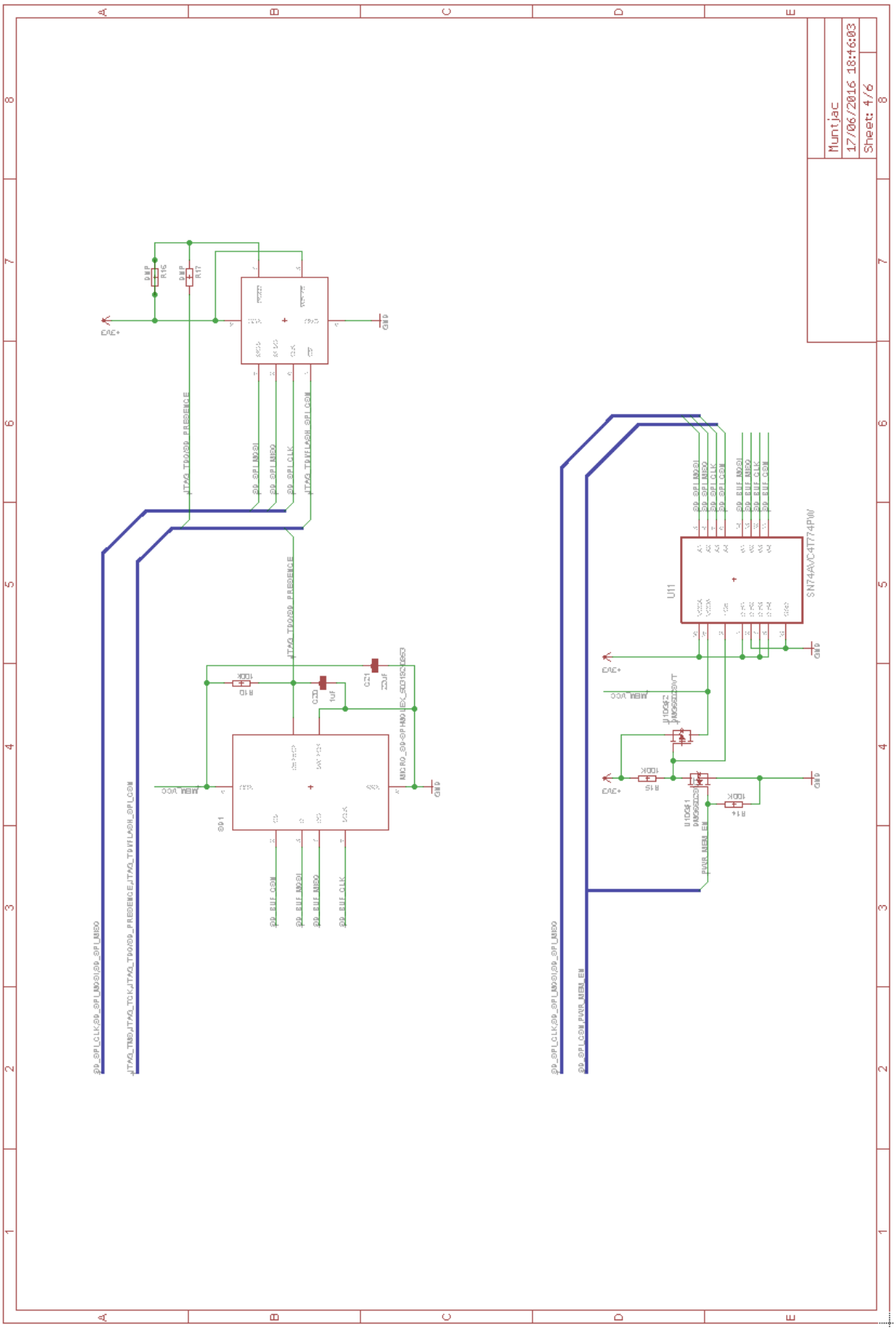
Appendix 8. Muntjac & MS2 Schematics

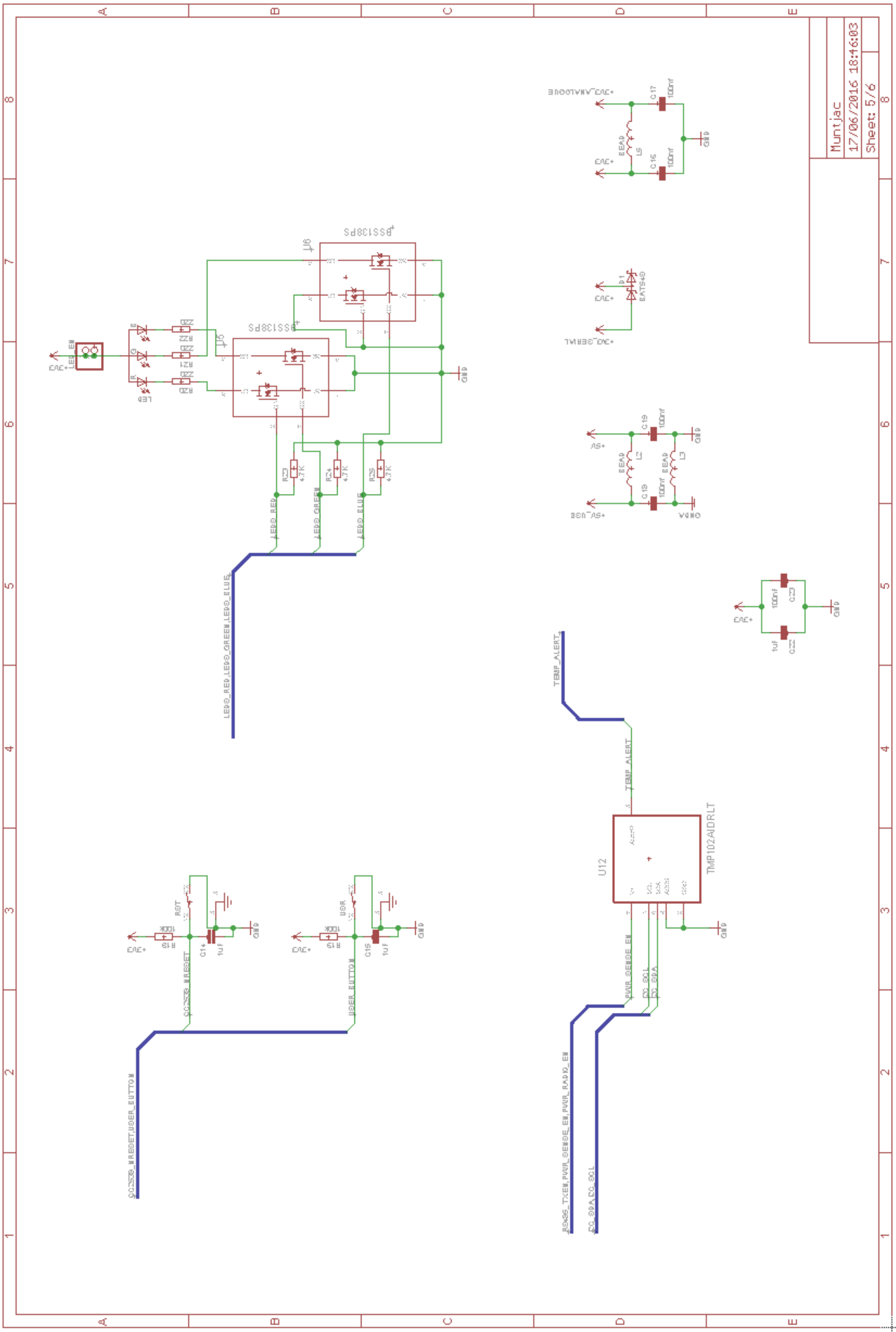
A 8.1. Muntjac Schematics

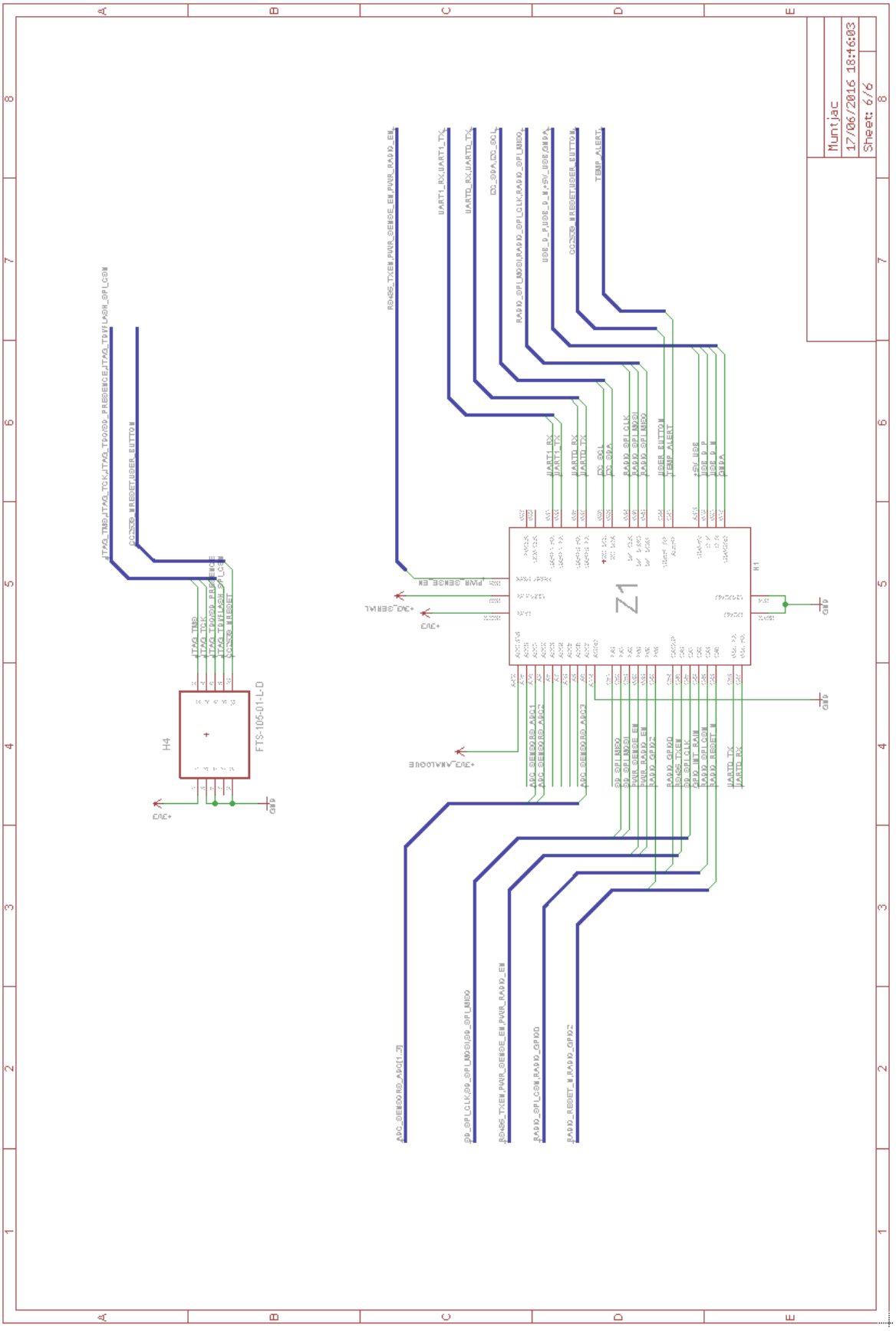



Muntjac
17/06/2016 18:46:03
Sheet: 2/6
ANT_868
U2
CC1120_RHM_32
RADIO_RESET_N
RADIO_GPIO2
RADIO_SPI_MOSI
RADIO_SPI_MISO
RADIO_SPI_CLK
RADIO_SPI_CSN
RADIO_GPIO0
PWR_RADIO_EN
PGASE_TXEN,PWR_GENSE_EN,PWR_RADIO_EN


Muntjac
17/06/2016 18:46:03
Sheet: 3/6
U4
U3
TPD2EB001DRLRQ1
J1
J7
BSS138PS
GND
+5V_USB

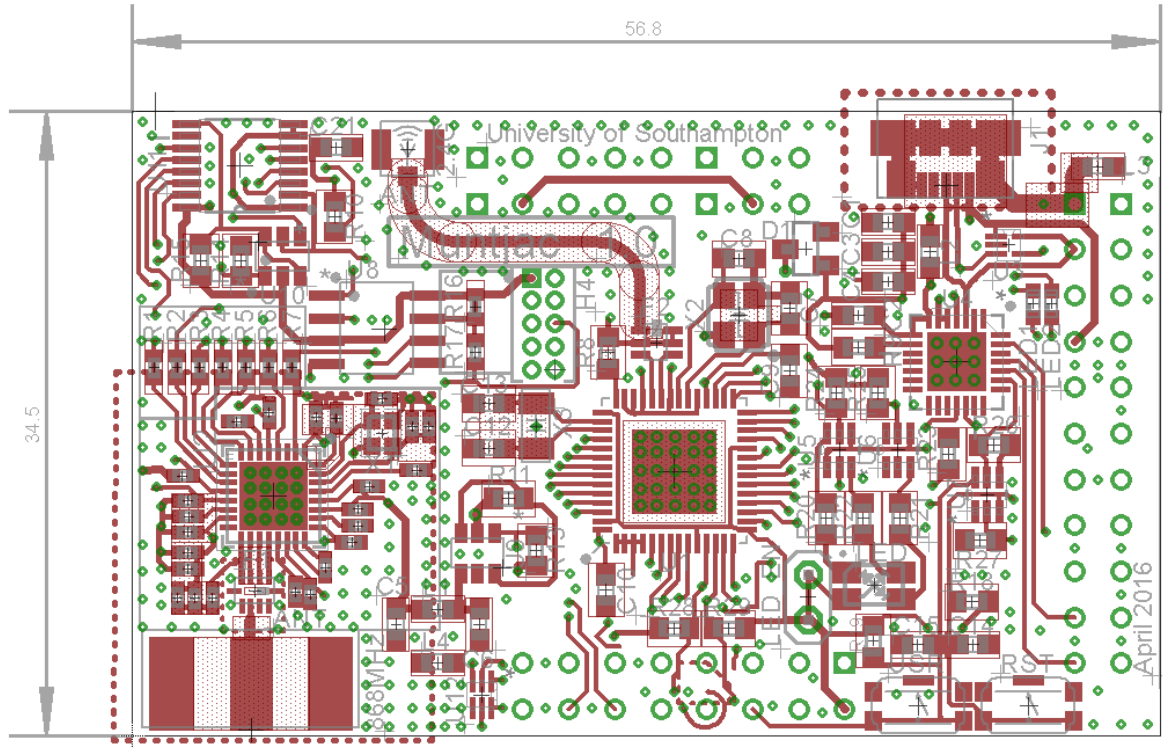

Muntjac
17/06/2016 18:46:03
Sheet: 4/6
SN74AVC4T774PW0
U11
JTAG_TDO/SD_PRESENCE
JTAG_TMS/JTAG_TCK/JTAG_TDO/SD_PRESENCE/JTAG_TDI/FLASH_SPI_CSN
SD_SPI_CLK/SD_SPI_MOSI/SD_SPI_MISO
SD_SPI_CSN/PWR_MEM_EN


Muntjac
17/06/2016 18:46:03
Sheet: 5/6
U12
TMP102AIDRLT
TEMP_ALERT
BSS138PS
GND

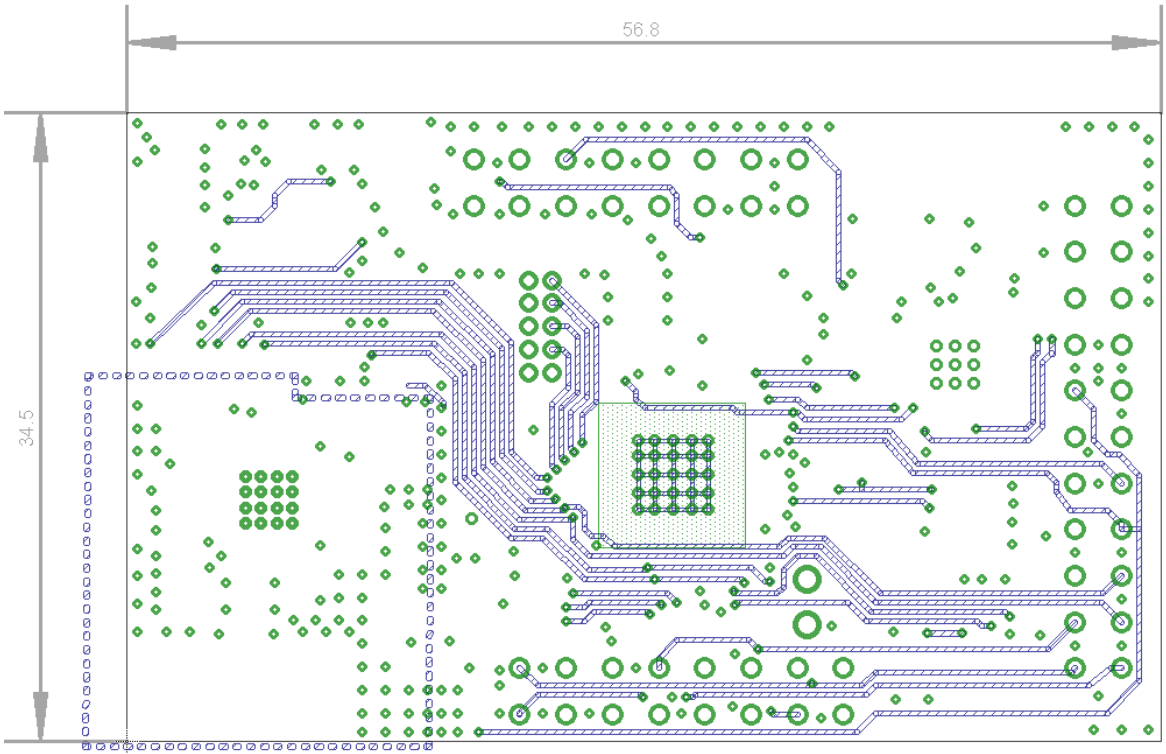

Muntjac
17/06/2016 18:46:03
Sheet: 6/6
Z1
H4
FTS-105-01-L-D

A 8.2. Muntjac PCB Layout

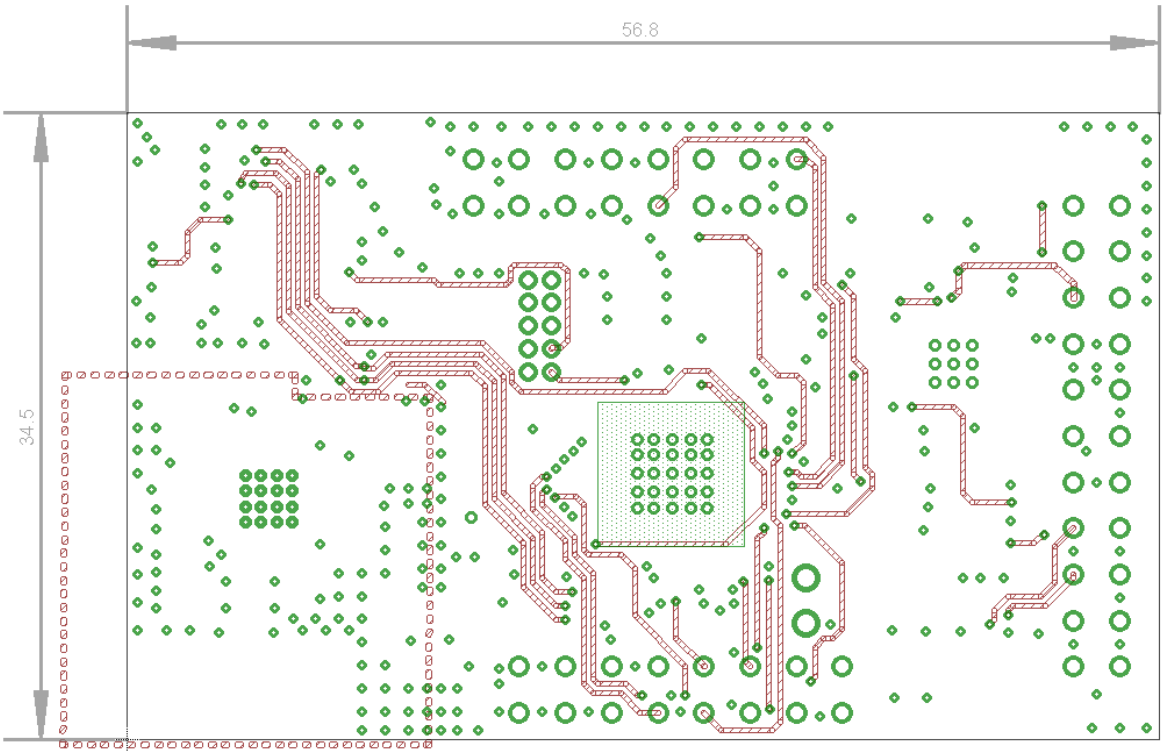
Muntjac Top layer



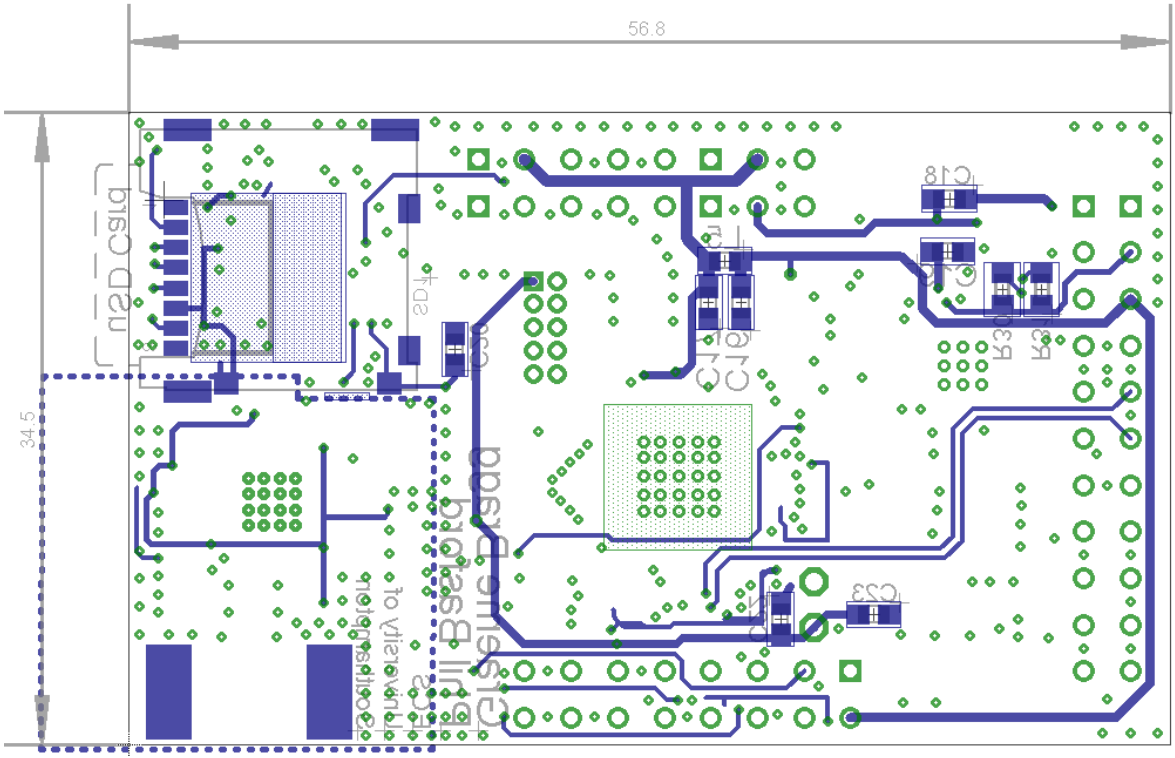
Muntjac Layer 2



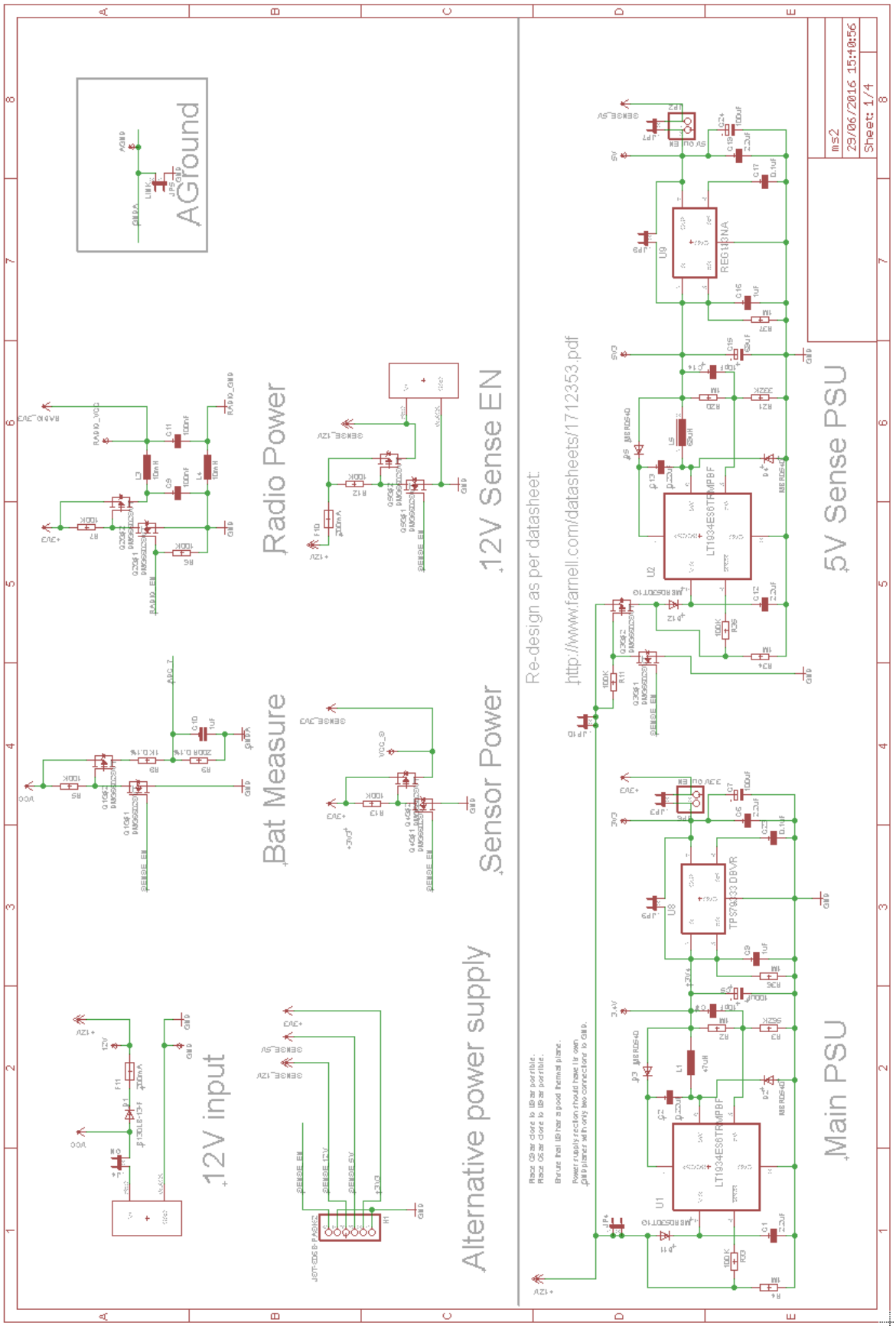
Muntjac Layer 3

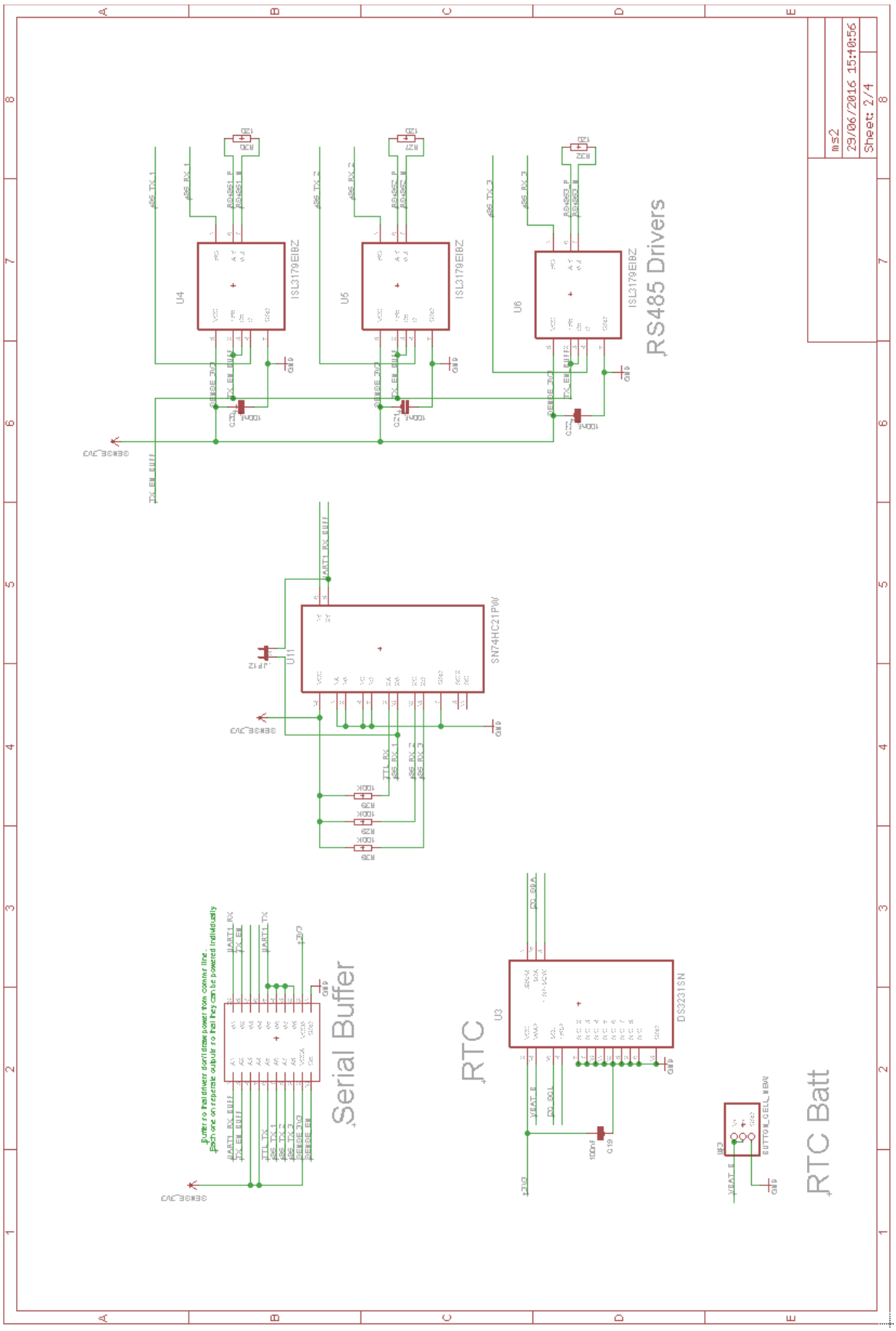


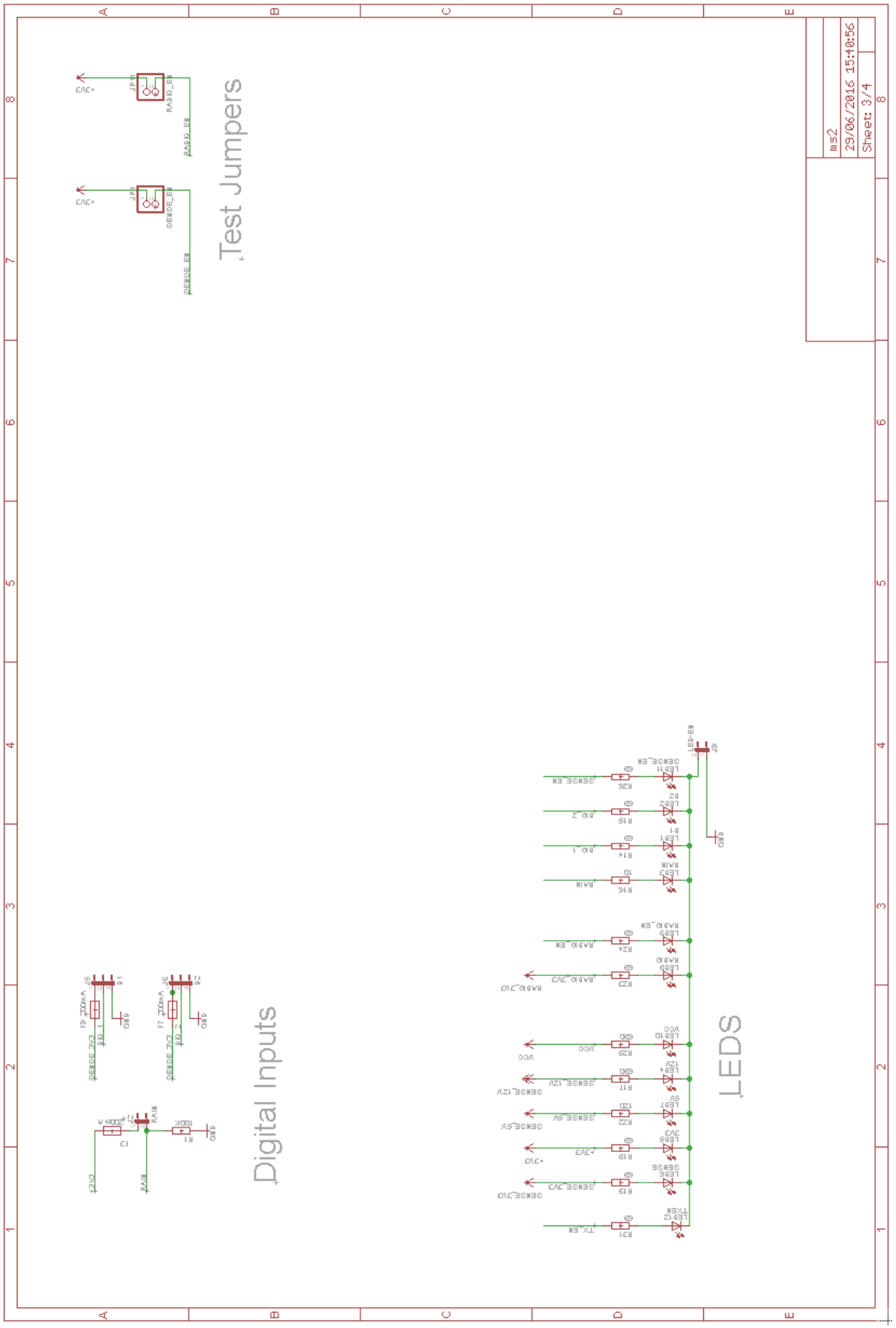
Muntjac Bottom Layer

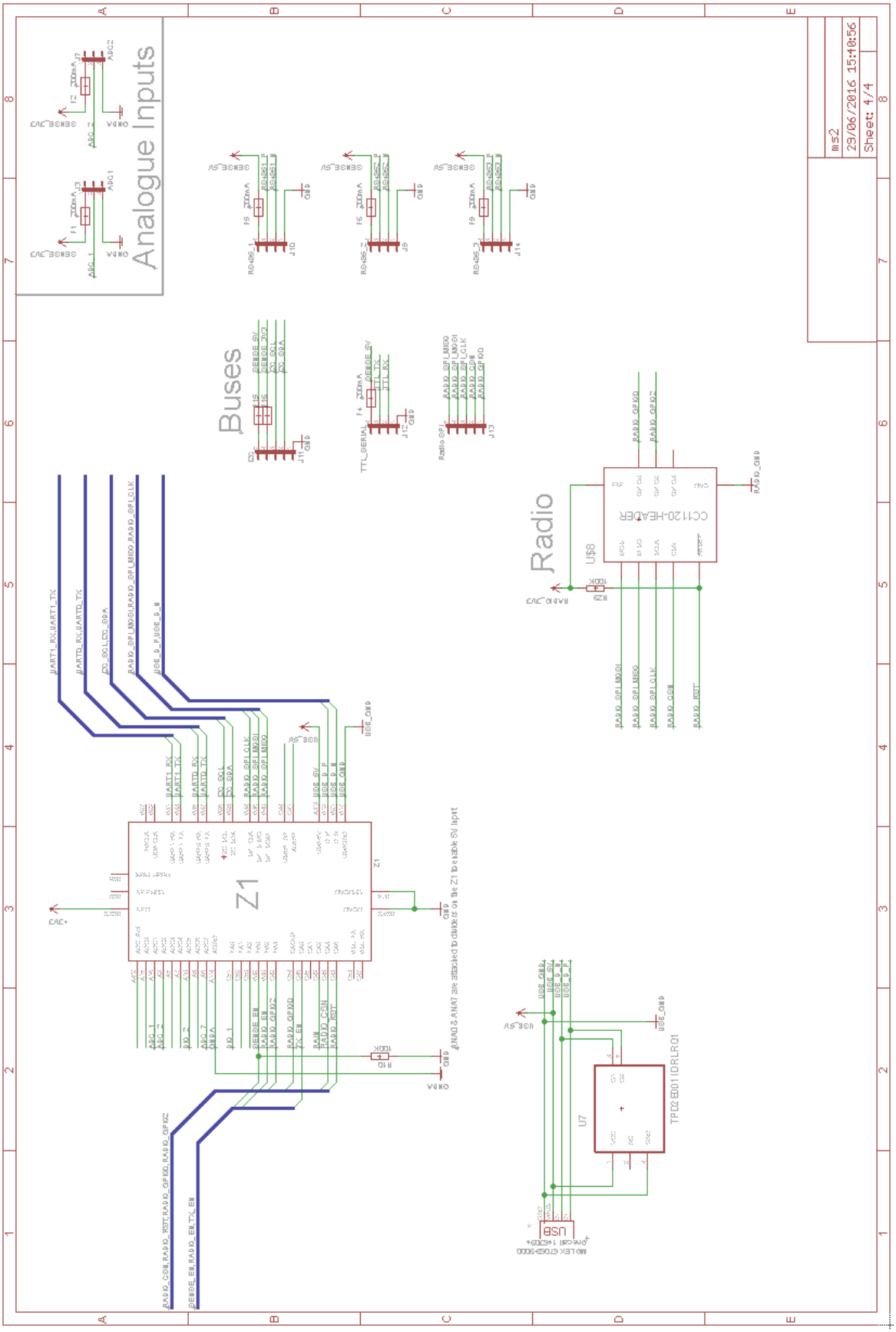


A 8.3. MS2 Schematics



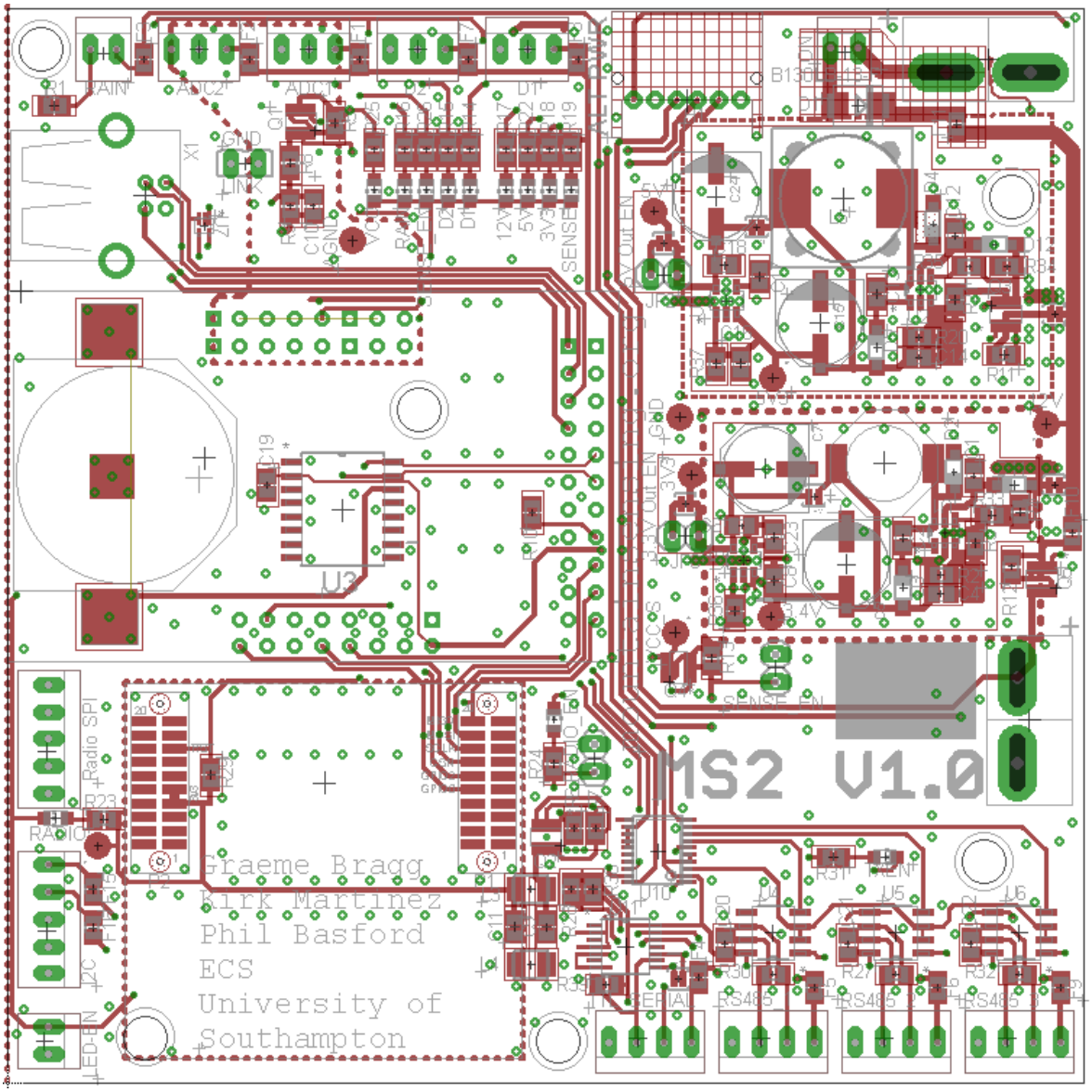

RS485 Drivers
Serial Buffer
RTC
RTC Batt
ms2
29/06/2016 15:40:56
Sheet: 2/4


Test Jumpers
Digital Inputs
LEDS
ms2
29/06/2016 15:10:56
Sheet: 3/4


Analogue Inputs
Buses
Radio
Z1
CC1120-HEADER
TPD2EB010RLR01
USB
ms2
29/06/2016 15:40:56
Sheet: 4/4

A 8.4. MS2 PCB Layout

MS2 Top Layer



MS2 Bottom Layer