Application of novel Internet of Things technology to assess Urban Heat Island Effect – a case study of University of Cape Town

Prepared by: Samuel D. Alexander

Supervised by: Dr. John Okedi

Report submitted in partial fulfilment of the requirements for the degree of MSc Civil Engineering

Department of Civil Engineering

University of Cape Town, Private Bag Rondebosch, 7700

South Africa

Acknowledgements

Completing this research dissertation has been an extraordinary experience, and I feel truly fortunate to have been afforded this opportunity. It goes without saying that I would not be where I am today without the support of those around me.

To my mother, Carol, and my father, Wayne, thank you for paving the way for me to step into the world of postgraduate academia. From the outset, you have both stood by my side and remained unwavering in your support. Not only could I draw on your expertise and guidance when tackling a master's degree, but I could also confide in you when times were tough. At every step along this journey, you both have always made me feel that you were proud of me. Thank you, and I love you.

To my brother, Mathew, you have always been a rock-solid support in my life. Thank you to you and Ilhaam for always being a keen ear to my ever-changing ideas, rants, and woes. Having you in my corner throughout this process has made me ever more grateful for you. Thank you for believing in me. I love you.

To my supervisor, Dr. John Okedi, thank you for being so generous with your expertise and knowledge. Taking on this topic came with new concepts and steep learning curves, forcing me out of my comfort zone. Through all of this, you were always encouraging and supportive. Under your tutelage, I have grown as an engineer, an academic, and most importantly, as a person. Thank you for backing me. I could not have asked for a better supervisor.

KU Leuven granted me the opportunity to complete the coursework component of my degree in Belgium. This was my first time traveling to Europe and it completely changed my life. I would like to express my gratitude to the Bio-Science Engineering faculty for taking me in, as well as the UCT exchange programme coordinators for facilitating the process.

Research is about the friends you make along the way! To my good friend and IoT predecessor, Roberto, thank you for showing me the ropes and for your patience. To Nadiya and Meghan, we started this journey together, and I couldn't have come this far without the support of you both. To Samir, thank you for your stellar work in helping me pull the final document together. To Zaid, thank you for assisting with the deployment and upkeep of the sensor nodes. To Pranesh, thank you for your input during the development of the linear regression model. Aadil and Yashir, thank you for your invaluable help with the formatting.

A special thank you must go to my dear friend James, for being so generous with his time and knowledge of microelectronics, databases, and web development. A true hero! To Ameen, for taking the time to teach and guide me as I stepped into the space of microelectronics, and to Curwin for producing top-quality circuit boards.

Lastly, I would like to thank Eugene, Johan, and Garth for providing mentorship and advice throughout my master's, as well as for always keeping me on my toes!

Abstract

This study investigated the feasibility of employing Internet of Things (IoT) technology as an alternative data collection method for studying the Urban Heat Island Effect (UHIE). Urban Heat Islands (UHIs) are localised and typically built-up areas, that experience significantly higher temperatures than the surrounding undeveloped areas. This temperature difference is primarily due to increased heat absorption and reduced cooling from construction materials like concrete and asphalt, as well as the removal of shaded green spaces.

An IoT Wireless Sensor Network (WSN) comprising 14 sensor nodes were implemented using readily available, 'off-the-shelf' products in South Africa, resulting in a competitive build cost of R1523.14 per node. The sensor nodes were deployed at the University of Cape Town (UCT) campus in both shaded green spaces and unshaded paved areas to monitor temperature and humidity differences. Over the course of 116 days, from 7 September to 31 December 2023, the IoT WSN provided real-time temperature and humidity data, yielding 84 148 transmissions with only a 0.1% transmission error rate. The data was stored and managed using the MongoDB database.

The investigation found that urban shaded green spaces were consistently cooler than unshaded paved areas; peak temperatures on the warmest days of each month reduced by 4°C on 28 September and by 2°C on 19 October, 15 November, and 27 December. This study demonstrates that IoT technology is highly capable of monitoring UHIE whilst remaining economically feasible to deploy.

Plagiarism declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Name and Surname: Samuel D. Alexander
Student Number: ALXSAM005
Date: 9th March 2025
Signature:

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API	Application Programming Interface
ASM	Aginova Sentinel Micro
BPSK	Binary Phase-Shifting Keys
CPS	Campus Protection Services
CPU	Central Processing Unit
CSV	Comma Separated Values
CSS	Chirp Spread Spectrum
DAC	Digital to Analogue Converter
DBMS	Database Management System
DL	Downlink
ETSI	European Telecommunications Standards Institute
FDMA	Frequency Division Access
GPS	Global Position System
GSM	Global System for Mobile Communication
GW	Gateway
HI	Heat Index
H_2SO_4	Sulphuric Acid
I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
I/O	Input/Output
IoT	Internet of Things
ISM	Industrial, Scientific, and Medical
LAN	Local Area Network
LiPo	Lithium Polymer
LoRa	Long Range
LoRaWAN	Long Range Wireless Area Network
LTE	Long Term Evolution
LST	Land Surface Temperature
MAC	Media Access Control
MAC*	Message Authentication Code
MAP	Mean Annual Precipitation
MCU	Microcontroller Unit
MQTT	Message Queuing Telemetry Transport
MSB	Most Significant Bit
MPU	Micro Processing Unit
NFC	Near Field Communication
NiCd	Nickel Cadmium
NiMH	Nickel-Metal Hydride
NS	Network Server
OFDMA	Orthogonal Frequency Division Access

OTAA	Over the Air Activation
Pb	Lead
PbO_2	Lead Dioxide
PCB	Printed Circuit Board
PIC	Peripheral Interface Controller
PWSN	Periodic Wireless Sensor Network
PV	Photovoltaic
QPSK	Quadrature Phase Shifting Keying
RAM	Random Access Memory
RFID	Radio Frequency Identification
ROM	Read Only Memory
RSSI	Received Signal Strength Indicator
RTC	Real Time Clock
SF	Spreading Factor
SMA	Sub Miniature version A
SMS	Short Message Service
SQL	Structured Query Language
SUHII	Surface Urban Heat Island Intensity
TDOA	Time Difference of Arrival
TER	Transmission Error Rate
TTN	The Things Network
UCT	University of Cape Town
UHIE	Urban Heat Island Effect
UHI	Urban Heat Island
UL	Uplink
WAN	Wide Area Network
WDT	Watch Dog Timer
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

List of Symbols

°C	Degrees Celsius
°C/h	Degrees Celsius per Hour
mAh	Milliampere Hours
kWh	Kilowatt Hours
Wh/L	Watt-hours/litre
bps	Bits per Second
MB	Megabytes
GHz	Gigahertz
h	Hour
MHz	Megahertz
V	Voltage
Ω	Ohm Resistance
R	South African Rand
$	United States of America Dollar

1. Introduction

1.1 Background

Urban Heat Islands (UHIs) are epitomised by the large-scale paving of green spaces to make way for growing built-up spaces due to rapid urbanisation. A consequence of rapid urbanisation is the paving of green areas which play an important role in regulating local climates. Paved areas in urban environments are the primary cause of the Urban Heat Island Effect (UHIE) (Levermore, et al, 2018). These urban hot spots have an array of impacts ranging from negative health implications, increased energy consumption, climate abnormalities and poor air quality (Kim & Guldmann, 2014). Data on the UHIE is particularly limited in the context of South Africa, and as such, this research investigated an alternative method to collect data. Accurate hydrological data is a major limiting factor in the African context. The lack of hydrological data available in southern Africa can be attributed to inadequate hardware and software to run hydrological models, deficiency of qualified personnel, and a shortage of financial resources (Meque et al., 2021). With the rapid changes in climate, extreme heat events occur more frequently, and a decline in precipitation is noted in southern Africa. With limited hydrological monitoring and warning systems, communities in these areas have been left vulnerable to the consequences of climate change. Thus, the drive to find alternative methods to collect hydrological data capable of measuring UHIs is of utmost importance.

The inception of the Internet of Things (IoT) came as a result of the significant advances achieved during the development of the internet and World Wide Web. According to Hassan et al. (2017) '*things'* refer to all physical objects, people and animals. Objects are not restricted to electronic or 'smart' devices. IoT incorporates sensor nodes and actuators to modify ubiquitous objects into 'smart' objects through the usage of embedded computer systems. This allows objects to possess virtual personalities and identities, enabling them to interact with their surrounding environment (Hassan et al., 2017). Collectively the objects form a network, where wireless communication technology is used to communicate with one another. Many of the hardware components are available "off the shelf" in South Africa, making IoT technology feasible and affordable to use. Given the need for hydrological data and scalable data collecting methods, testing whether IoT technology is suitable in the South African context is important. IoT networks send live data, making the hydrological data more accessible and 'user-friendly' than conventional methods (Kaynak et al., 2025). As urban areas expand, more UHIs will be created, thus, having access to ample reliable data is crucial to finding solutions to minimising the UHIE.

1.2 Problem statement

As green spaces are increasingly paved, UHIs are formed and consequently have negative impacts on the quality of life in urban areas. Data on this phenomenon is limited, as conventional hydrological data collection methods are not suitable or scalable enough to provide adequate

information on the matter. Current methods are too labour intensive and tedious to collect data on a large scale, thus making it inefficient as well as costly. Therefore, it is proposed that the feasibility of using IoT technology to collect and live stream hydrological data, in a South African context, be investigated.

1.3 Research tasks

- Build an IoT network comprising of sensor nodes and a gateway, capable of collecting and sending live data to The Things Network (TTN).
- Measure the consistency of data rate transmissions and accuracy of the data collected by the network.
- Improve power efficiency of nodes and explore energy harvesting alternatives.
- Determine whether it is feasible and cost effective to use IoT technology to collect hydrological data in South Africa.
- Monitor the UHIE and determine the impact on the surrounding areas (University of Cape Town (UCT) as a case study).

1.4 Scope and limitations

The scope of this study is comprised of deploying a sensor network to monitor the micro-climate at the study site. A comparative study was also undertaken to determine the types of hardware used in terms of availability, cost and performance. The site of this study was limited to the UCT campus with a data collection period of four months. Two sets of sensor nodes were employed, one for shaded spaces and the other for unshaded spaces.

2. Literature review

2.1 The Urban Heat Island Effect (UHIE)

UHIs are characterised by the increased temperatures of urban areas compared to suburban and rural areas. The dissimilarity between the temperatures of urban, compared to suburban and rural areas are caused by the phenomena of "hyperthymisation". Hyperthymisation is the result of paving green areas, reducing water bodies, and ineffectively dissipating heat generated and stored by large buildings (Chen et al., 2022). An increase in solar radiation absorption is attributed to the use of construction materials with high thermal conductivity properties. This attributes significantly to the formation of UHIs, and as urbanisation continues to increase, the UHIE will continue to worsen. Warmer surfaces have a significant impact on albedo, morphology, hydrology and overall thermal aerodynamic related issues (Chen et al., 2022; Mohammad Harmay et al., 2021). Increased air temperature is a prime source for an increased amount of extreme precipitation events. For every 1°C of increasing air temperature, there is a subsequent 7% gain of water holding capacity, which significantly increases water vapour in the air; additionally warmer air temperatures increase the rate of evaporation and surface drying (Zaki et al., 2019). It is predicted that the global average air temperature will rise anywhere between 1.4 – 5.8 °C (Houghton et al., 2001). In light of South Africa's recent issues with devastating precipitation events (the 2022 Durban floods as an example) and a history of water scarcity (The Cape Town drought in 2016), mitigating the UHIE is of paramount importance. Previous studies on Surface Urban Heat Island Intensity (SUHII) show that there is a positive, linear and non-linear correlation between daytime SUHII and the Mean Annual Precipitation (MAP) of the region (Yang & Yao, 2022; Zhao et al., 2014). Another study investigated the phenomenon in humid regions versus semi-arid regions. The smallest SUHIIs are shown in areas that are considered semi-arid in contrast to greater SUHIIs in humid areas, further showing the relationship between SUHII and MAP (Li et al., 2019; Yang & Yao, 2022). Urban centres in drier climates are aerodynamically 'rougher' than surrounding rural or open areas which in turn helps dissipate heat better, while the inverse occurs across urban centres in humid areas (Y. Li et al., 2019). UHIs are a combination of atmospheric temperature and surface temperature. Urban atmospheric heat island data is collected via the use of weather stations or hydrological sensor nodes. The latter, being urban surface heat island, is collected and interpreted from thermal sensing imagery. With the rapid increase of urbanisation, the impact of UHIs have become far more significant. UHIs negatively impacting citizen health and sustainable development, increasing the importance of researching the phenomenon. Despite UHIs being discovered over a century ago, issues surrounding data accuracy and obstacles surrounding scalability have not been adequately dealt with (Chen et al., 2022).

2.2 IoT technology

Since the inception of the concept of IoT, the need to fully realise its potential has risen dramatically. The onset of the fourth industrial advancement/revolution and the explosive progress of online/remote technology has made IoT a cornerstone concept gaining large

traction. In alignment with the twenty-first century's technological advancements, it became widely acknowledged that deploying Wireless Sensor Networks (WSNs) would be a critical component of global efficiency increasing. The European Research Centre provides the following definition for the IoT:

> "A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual things have identities, physical attributes, and virtual personalities and use intelligence, and are seamlessly integrated into the information network" (Noura et al., n.d.).

Considering the above definition for IoT, it is essential to acknowledge that there are variations of this term. Nonetheless, all definitions broadly characterise IoT as a network of all things and people that communicate wirelessly and use modern computing technology to enable every object with this capability. The term *things* encompasses all material entities, including humans and animals. These physical things include a wide array of inanimate objects from chairs to fabrics and even food, in addition to, electronic gadgets and "smart" technologies. The list of things subsumed under IoT, is interminable. However, using sensor technologies and actuators, IoT transforms these ordinary objects into 'smart' objects. Utilising embedded computer systems, this allows for items to have so-called "virtual personalities" and identities, engaging with their local environment (Hassan et al., 2017). All items are part of a network in which they employ wireless communication technologies to communicate.

IoT encompasses the expansion of (i) Cyber physical systems, focusing on integrating the virtual-online and physical worlds; (ii) Pervasive computing systems, focusing on equipping all physical entities with processing power and the capability to wirelessly interact with the internet; and (iii) It aspires to become so seamlessly and unnoticeably incorporated into our daily lives and activities, which is known as ubiquitous computing. Further, it enables machine-to-machine communication, minimising the requirements of human intervention and reducing inefficiencies. Overall, all items in our world will become more receptive and responsive to their surroundings, making modifications and predictions on their own depending on the information and data collected (Madakam et al., 2015).

2.3 IoT building blocks

An IoT system's architecture comprises of five distinct types of functional layers (Jabraeil et al., 2020). These layers are referred to as functional blocks and are explored briefly in the subsequent subsections, they include: Sensing Layer, Communication layer, Cloud Layer, Management Layer and Services and Applications Layer.

2.3.1 Sensing layer

The primary component of the sensing layer, as the name suggests, are the physical sensors. Through the application of sensing devices, this layer is designed to monitor the surrounding physical environment, identify and locate intelligent or 'smart' objects and collect data. A key characteristic of the sensing layer is that it is able to be heterogenous and capable of performing

multiple different sensing functions simultaneously, by accommodating an array of different actuators and sensors (Al-Fuqaha et al., 2015).

2.3.2 Communication layer

The communication layer is tasked with facilitating the interaction between the other layers of an IoT network. Within the communication layer, collected data is sent to cloud or application services for storage and analysis. Hardware components such as routers and gateways form the communication layer (Jabraeil Jamali et al., 2020). Various protocols are used to link IoT end-devices (also referred to as sensor nodes) to upper layers, such as, Message Queuing Telemetry Transportation (MQTT), in combination with technologies, such as, Wi-Fi, Bluetooth, Zigbee, Radio Frequency Identification (RFID), 3G and Global System for Mobile communications platform (GSM) (Yang et al., 2011); the protocols are covered in more detail in section 2.6.

2.3.3 Cloud layer

The cloud layer is also referred to as the ‘processing unit’, this layer receives all the data collected by the sensing layer and sent by the communication layer. The cloud layer is the point at which the data is housed and analysed; data servers are commonly used to execute these functions. As technology and research into cloud computing improves, cloud computing continually becomes more decentralised, resulting in higher computing ability (Varghese & Buyya, 2018).

2.3.4 Management layer

The management layer allows for users to control and monitor the rest of the layers and their components. Importantly, the management layer, enables programmers to manage end-devices without having to consider individual hardware requirements and specifications (Al-Fuqaha et al., 2015).

2.3.5 Services and applications layer

The final layer, service and applications layer, is where the user interacts with all the layers. Here, in the services and applications layer, the data is visualised and made available for analysis (Jabraeil Jamali et al., 2020).

2.4 Architecture

As the IoT is predicated on the communication between 'things' and the pursuit of maximum efficiency, it is essential to build a system that permits this. In this context, architecture refers to how the chain of command of physical components are organised, how they work, and the information they are intended to collect. The architecture of an IoT system comprises of "physical components, sensors, cloud services, developers, actuators, communication layers, users, business layers, and IoT protocols" (Jabraeil Jamali et al., 2020). The simplest architecture design consists of three layers: the perception layer, which contains the physical sensing nodes, the network layer, linking the sensors ("smart" devices) and application layer (Sethi & Sarangi, 2017). Using sensors and actuators, the perception layer collects data and

information from its surrounding environment, this layer's purpose is to additionally filter required data before transmission. The network layer is responsible for defining "routes" along which data will be transmitted to the ‘topmost' layer. Data is "stored and integrated" at the application layer. The application layer allows the user to adjust the user interface based on the service (Ghapar et al., 2018). The simple three-layer system (Application, Network and Perception) is shown in Figure 2-1.

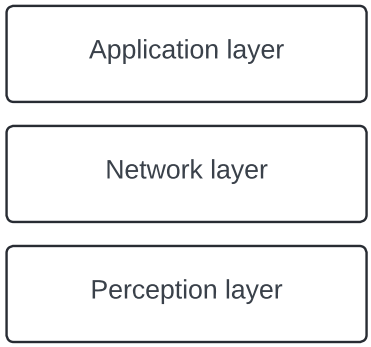


Figure 2-1: Conventional IoT architecture (Jabraeil Jamali et al., 2020)

It was during the early stages of research into IoT networks where the simple three layer, conventional IoT architecture, saw its inception. Although these three layers encompass the essence of an IoT network, it is widely understood that it does not sufficiently describe the finer details, and for this reason, a five-layer architecture has been developed as a means to expand on the simple three-layer architecture (Banu, 2017). The new five-layer architecture is shown in Figure 2-2.

Referring to Figure 2-2, the perception and application layers remain the same, with the addition of the transport, processing, and business layers serving new functions compared to the previous three-layer architecture from Figure 2-1. The second layer, transport, is solely designed to move sensor data from the perception layer to the processing layer, and it operates as a two-way connection, meaning that communication can also be sent to devices in the perception layer.

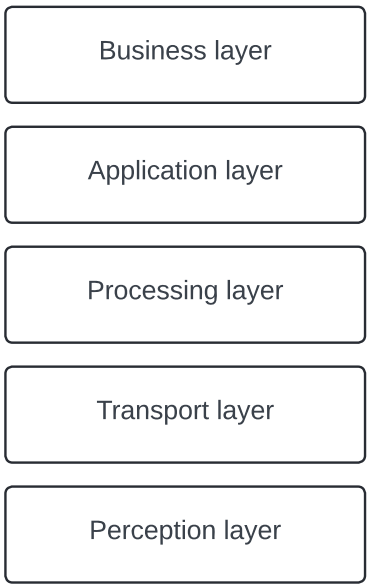


Figure 2-2: Five-layer IoT architecture (Jabraeil Jamali et al., 2020)

This two-way communication is facilitated by network technologies, such as, wireless, Third Generation (3G), Local Area Network (LAN), Bluetooth, RFID, and Near Field Communication (NFC) (Kumar & Mallick, 2018). The third layer being the processing layer, also referred to as the Middleware layer, employs all the characteristics mentioned under the 'cloud sensing' IoT building block as described in Section 2.3.3. The function of the processing layer is to store and analyse data by utilising databases and cloud computing. In addition, the processing layer also provides services to the lower level layers, if required (Kumar & Mallick, 2018). The final layer, the business layer, manages the whole system, as well as applications, business models and user privacy (Jabraeil Jamali et al., 2020; Sethi & Sarangi, 2017).

2.5 Wireless Sensor Network Technology

Wireless Sensor Network (WSN) technology is commonly acknowledged as a crucial component for the success of IoT. WSN technology is recognised as an innovative new technological development that has the potential to transform how we engage with the physical world. As sensor technology gains traction sensor precision, efficiency, adaptability, and affordability, is expected to improve exponentially (Jabraeil Jamali et al., 2020). WSN technology allows one to track vast and expansive regions with greater reliability and precision. Analysing urban cooling systems in the past has been hampered by the fact that it was simply too time-consuming to cover such expansive areas. Previous conventional methods to data collection were less precise, as the only solution implemented was to interpolate over great distances which yielded inaccurate results. WSN provides the opportunity to utilise many small devices, known as sensor nodes, with the capacity to track environmental changes, process data, and communicate this data to other devices and platforms. Typically, these sensor nodes are "low-power radios, many intelligent sensors, and integrated CPUs" (Al-Qurabat & Abdulzahra, 2020). Sensor nodes gather information and transmit it to sink nodes (also referred to as gateways). The gateway is the location from which data is sent to the internet. Al-Qurabat

& Abdulzahra (2020) and Rhee et al. (2004) identified four categories of WSN applications as follows:

(i) The first application, *event-driven* sensor nodes, are directed to detect the presence of a predetermined event, and proceeds to transmit a warning to the gateway, notifying it of the occurrence (Al-Qurabat & Abdulzahra, 2020). Event-driven sensor nodes transmit data only when a predetermined threshold is met, two examples of use cases for this WSN application are fire alarms and security systems (Rhee et al., 2004).

(ii) The second application, *query-driven*, is when the gateway transmits a "query" to the sensor node. In this case, an investigation is conducted by the sensor node and sends a response to the gateway (Al-Qurabat & Abdulzahra, 2020).

(iii) The third application, *store-and-forward,* is when sensor nodes gather data, and in certain instances, process it. The information is stored locally (on the sensor node) and sent when requested, or, in range of a network. An example of store-and-forward, can be seen when a sensor is instructed to collect data on cargo while the ship is in transit but does not transmit it until the ship has docked (Rhee et al., 2004).

(iv) The fourth and final application is the *Periodic Wireless Sensor Network* (PWSN), which is utilised in this study. The PWSN application instructs sensor nodes to periodically collect data or take readings, making this an optimal method for collecting large quantities of data, as was the case in this study (Al-Qurabat & Abdulzahra, 2020; Rhee et al., 2004).

Due to the continuous data acquisition, power consumption is a crucial component of this process. It is essential that IoT devices be as energy efficient as achievable, as they must last for the duration of the study. In the case of the UHI inquiry at the UCT campus, where access to the site is not an obstacle, manually replacing batteries, if necessary, is not a challenge. However, in less secure study areas or more remote environments, manual replacement of batteries could be detrimental and affect the quality of the data, especially, in a large-scale study that spans a vast geographical area, over an extended period of time. Therefore, some practical prerequisites for efficient WSNs are that it has a low data rate and contains substantial battery life while also possessing a small form factor-device that is pervasive for the conditions in which they are deployed, while still being relatively inexpensive.

2.6 Low-Powered Wide Area Network

IoT continues to expand, with new devices constantly connecting to it. Previously, devices utilised "short-range radio protocols", such as, Bluetooth, WiFi, cellular data, and ZigBee (Mekki et al., 2019), which came with its own set of limitations and disadvantages. The first disadvantage of short-range radio protocols is the limited distance over which devices can send data. A further disadvantage is their high energy usage and the cost of large-scale implementation; WiFi appears inexpensive, but it heavily relies on a stable source of power because it draws a great deal of energy. These limitations led to Low-Powered Wide Area

Network (LPWAN) technologies being developed which comes with the potential to inexpensively connect devices to one another on a massive scale (Pop et al., 2017). Figure 2-3 depicts the relationship between the data rate and range of transmission of popular communication protocols.

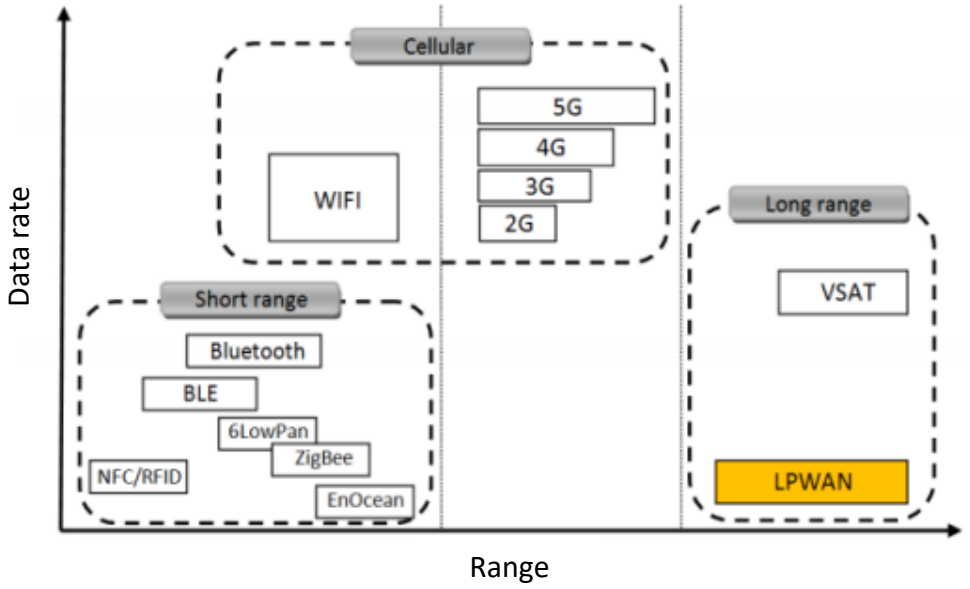


Figure 2-3: Data rate vs Range (Mekki et al., 2019)

LPWANs sacrifice complexity for energy efficiency, enabling them to operate on battery power over extended periods of time, typically 10 years or longer. Wireless Local Area Network (WLAN) and LPWAN are distinguished by the convolution of the tasks they are intended to accomplish. Eliminating complex tasks has enabled LPWAN to improve energy consumption (through a decrease of energy consumption), as well as making the manufacturing of associated devices inexpensive and decreasing costs associated with long-distance transmissions (Pop et al., 2017). Significantly, these improvements allow sensors to be deployed in remote and otherwise hazardous locations without the need to replace the power source/battery. However, the trade-off when utilising LPWAN technology is it has a significantly lower latency and transfer rate. LPWANS aim for a range of 1-5 km in urban areas and 10-40 km in more rural regions (Mekki et al., 2019). The most prevalent LPWAN technologies are SigFox, NB-IOT and LoRaWAN (Mekki et al., 2018). These technologies are explored in the subsequent subsections. LPWAN networks are usually designed to connect the following types of devices:

- One directional (device-to-gateway) communication
- Low data transfer rate devices, often less than 10 bytes per hour.
- Low power utilisation, typically, these devices are powered by batteries, with a pre-determined battery life.

2.6.1 Sigfox

Sigfox provides end-to-end technologies. Using Sigfox-exclusive base stations with software-defined radios that connect to back-end servers via a network based on "intellectual property". Devices connected to these stations use Binary Phase-Shifting Keys (BPSK) on an ultra-narrow frequency band. Sigfox utilises unrestricted Industrial, Scientific, and Medical (ISM) frequencies. The use of 'ultra-narrow bandwidth' produces very little noise (minimises interference and unwanted frequencies), allowing for optimal energy efficiency with high receiver sensitivity and an inexpensive antenna model. These upsides come with a drawback, as 100 bits per second (bps) is the highest throughput that can be attained (Mekki et al., 2019). Sigfox was originally only able to uplink (communication from end-device/sensor node to a network), but since has been modified and expanded to accommodate downlink (communication from network to an end-device/sensor node). The uplink is capable of transmitting 140 messages per day, with a limited message size of 12 bytes. The downlink can only transmit four messages per day, which is far fewer than the uplink's capacity. The difference in uplink and downlink messages has the disadvantage that not every uplink message can be accounted for; this makes it difficult to guarantee consistency and reliability (Mekki et al., 2019).

2.6.2 Narrowband IoT

Narrowband IoT (NB-IoT), is a LPWAN technology operating within licensed frequency bands, co-existing with Long Term Evolution (LTE) or GSM networks. With a frequency bandwidth of 200 Kilohertz (KHz), NB-IoT supports standalone, guard band, and in-band operations. In standalone mode, NB-IoT, can utilise GSM frequency bands, while guard band mode leverages unused blocks in LTE carriers, and in-band mode uses resource blocks within LTE carriers. The communication protocol for NB-IoT, derived from LTE, is streamlined to suit IoT applications, minimising unnecessary features and battery usage. NB-IoT can support over 100 000 devices per cell, by employing a type of modulation referred to as Quadrature Phase Shift Keying (QPSK), Frequency Division Multiple Access (FDMA) for uplink, and Orthogonal FDMA (OFDMA) for downlink, with a maximum data rate of 200 kilobits per second (kbps) downlink and 20 kbps uplink. Each message can carry up to 1600 bytes of data, and devices can achieve a battery life of up to 10 years with an average transmission of 200 bytes per day (Mekki et al., 2019).

2.6.3 LoRa and LoRaWAN

Long Range (LoRa) is a physical layer technology that utilises a technique called Chirp Spread Spectrum (CSS) to handle signals in the sub-GHz ISM band. CSS, a method that incorporates the spreading a narrow band over a wider channel, is used to achieve two-way communication. CSS technology affordably achieves low noise levels, better security, reliability, and resilience (Mekki et al., 2019). CSS utilises six spreading (SF) factors, SF7 to SF12, to manage the relationship between data rates and transmission distance. The distance that data can be sent is proportional to the spreading factor, whereas the size of the data is inversely proportional. All LoRa gateways communicate simultaneously, with the payload of each message being no more than 243 bytes.

In 2015, LoRa alliance created the LoRa Wireless Area Network (LoRaWAN) communication protocol. LoRaWAN is a Media Access Control (MAC) protocol designed to function based on LoRa modulation which is defined by the 'star of stars' topology. The utilisation of the star of stars topology and protocol ensures that all LoRa-based stations or gateways within range receive a message simultaneously (redundant reception). This increases the likelihood that a message will be successfully received. This method is effective but costly, because the more gateways that are deployed in the area, the better. Due to the fact that numerous base stations receive identical messages, a backend system is required to filter out data redundancy, and evaluate the security of messages. The data carried by the message is then conveyed to back-end-devices, which is communicated to the application layer. Simultaneous data messages also aid with device localisation (Bellouch et al., 2021; Mekki et al., 2019). The message exchange in the LoRaWAN star of stars topology operates on a single-hop link between end-devices and gateways. This single-hop link means that data is communicated directly to gateways, unlike a mesh topology, from end-devices to gateways with no data being passed through intermediate nodes in between (Leonardi et al., 2020). Figure 2-4 illustrates the LoRaWAN star of stars topology, showing the bi-directional communicational relationship between end-devices and the network server (NS), transmitting through gateways.

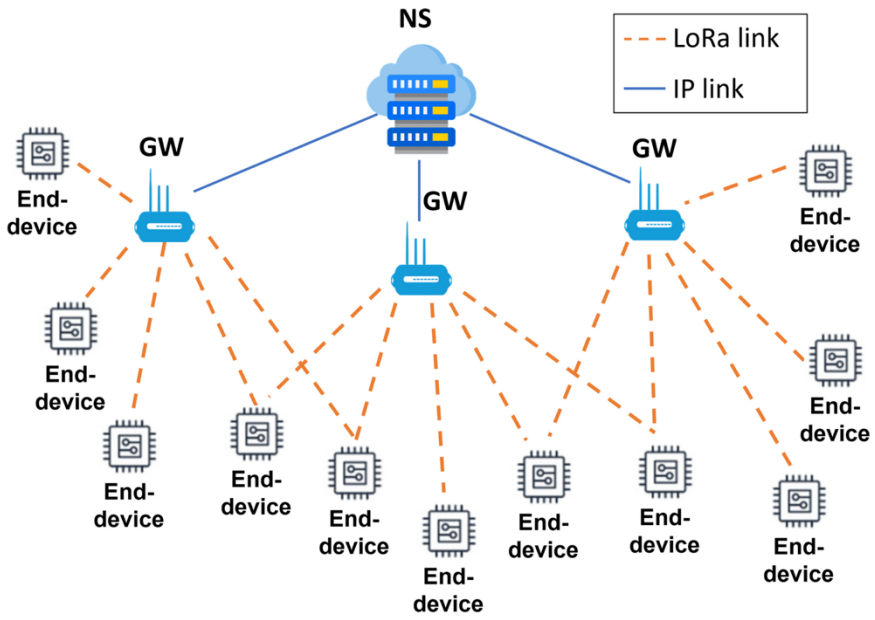


Figure 2-4: LoRaWAN star of stars topology (Leonardi et al., 2023)

Gateways (GW) connect to network servers via internet protocol and transmit messages from the server to end-devices via downlinks and from end-devices to the server via uplinks. A key point to note is that multiple of the same uplinks can be sent through multiple in-range gateways (the aforementioned redundant reception) to the network server; the same does not apply to downlink messages, where a single message is sent via a single gateway to a lone end-device (Leonardi et al., 2023).

2.6.3.1 LoraWAN structure

LoraWAN is divided into classes, each class is designed to be suited for different tasks.

- The first class (class A) is ideal for scenarios requiring optimal power efficiency. This works particularly well with applications that do not require large downlink coupled with uplink messages. The default class is designed to immediately send an uplink message following a downlink message (Pop et al., 2017). Downlink windows allow for control commands to be communicated during available windows, further optimising power efficiency and current draw. Class A hardware is capable of executing sleep modes (Mekki et al., 2019; *What Is LoRaWAN® Specification - LoRa Alliance®*, n.d.).
- Second class (class B) devices are able to switch on or 'wake-up' at predetermined intervals to receive scheduled downlink messages (Pop et al., 2017). Nodes function as a time synchronised beacon, this allows for a network to be notified that the end-devices are 'listening'. Class B is generally more costly and energy consuming than class A (Mekki et al., 2019; *What Is LoRaWAN® Specification - LoRa Alliance®*, n.d.).
- Third class devices (class C) are always 'listening'. These devices are on perpetually, allowing them to listen continuously. Due to this design, class C is the most expensive and require a steady power supply i.e. mains-powered devices (Pop et al., 2017).

2.6.3.2 Achieving long range

LPWAN's primary objective is to send data over as great a distance as possible, while being inexpensive by reducing current draw and maintenance costs. A comparison of LPWAN technologies, the methods and techniques they employ are described in table 2-1. The use of a 'sub-1GHz band' rather than the 2.4 GHz band, allows for LPWAN to have reduced signal attenuation and multipath fading since transmissions over lower frequencies are less susceptible to being affected. The sub-1 GHz frequency is also far less crowded than the 2.4 GHz range. This is due to the widespread usage of the 2.4 GHz band by wireless technologies such as WiFi and Bluetooth (Raza et al., 2017).

Another common technique used by LoRa as mentioned in Section 2.6.3 is called Spread Spectrum. The Spread Spectrum technique splits narrowband over a wider spectrum, resulting in a less efficient use of the spectrum and requires decoding on the receiver end. The lesser efficiency is rectified by using several orthogonal sequences. Raza et al. (2017) describes the benefit of using the Spread Spectrum technique and decoding on the receiver end: "As long as multiple end-devices use different channels and/or orthogonal sequences, all can be decoded concurrently, resulting in a higher overall network capacity". Table 2-1 shows a tabulated comparison of the three LPWANs explored in this section.

Table 2-1 LPWAN technology comparisons (Mekki et al., 2019)

	Sigfox	**LoRaWAN**	**NB-IoT**
Modulation	BPSK	CSS	QPSK
Frequency	Unlicensed ISM bands (868 MHz in Europe)	Unlicensed ISM bands (868 MHz in Europe)	Licensed LTE frequency Bands
Bandwidth	100 Hz	250 kHz and 125 kHz	200 kHz
Maximum Data Rate	100 bps	50 kbps	200 kbps
Bi-directional	Limited/Half-duplex	Yes/Half-duplex	Yes/Half-duplex
Maximum Messages per day	140 (UL), 4 (DL)	Unlimited	Unlimited
Maximum payload length	12 bytes (UL), 8 bytes (DL)	243 bytes	1600 bytes
Range	10 km (urban), 40 km (rural)	5 km (urban), 20 km (rural)	1 km (urban), 10 km
Interference immunity	Very High	Very high	Low
Authentication and encryption	Not supported	Yes (AES 128b)	Yes (LTE encryption)
Adaptive data rate	No	Yes	No
Handover	End-devices do not join a	End-devices do not join a single base station	End-devices join a single
Localisation	Yes (RSSI)	Yes (TDOA)	No (under specification)
Allow private network	No	Yes	NO
Standardisation	Sigfox company is collaborations with ETSI on the standardisation of Sigfox-based network	LoRa-Alliance	3GPP

2.7 Processing, storing, and visualisation of data

IoT sensor networks are increasing in size and coverage daily. With the increased amount of data being gathered by end-devices, methods for storing this data have become an important point of development. Traditional databases have major drawbacks when managing IoT data, hence the rise in popularity of NoSQL databases (Amghar et al., 2018). Databases are structured collections of data. Data transactions and problems are controlled by Database Management Systems (DBMS). Traditional databases are described as being relational and use Structured Query Language (SQL), whereas, more modern data bases, NoSQL, are non-relational, this being the primary difference between the two (Rautmare & Bhalerao, 2016). Traditional databases are capable of storing and handling massive volumes of data but are less adept when it comes to scalability compared to NoSQL databases, which is crucial in the context of the fast-growing number of IoT devices (Michel, 2017). NoSQL databases are also more adept to supplying real time streaming of data, which is continually becoming a key requirement in application layers of IoT networks (Amghar et al., 2018; Onojakpor, 2023).

Although Rautmare & Bhalerao (2016) found that both SQL and NoSQL performed adequately, the general consensus is that NoSQL databases are better suited for the growth of IoT technologies.

Many NoSQL databases exist with varying capabilities. Amghar et al. (2018) gathered information on the five most prominent NoSQL databases: Redis, Cassandra, MongoDb, Couchbase, and Neo4j in context to IoT data management requirements. This information was used to determine which of these were best suited for IoT applications. Five criteria were used to compare the databases based on the requirements of IoT applications: scalability (R1), real time processing (R2), security (R3), spatial data handling (R4) and data aggregation (R5). The results are shown in Table 2-2, where '++' is equated to high, '+' to middle and '-' to low.

Table 2-2 NoSQL database comparison table (Amghar et al., 2018)

NoSQL database	R1	R2	R3	R4	R5
Redis	-	+	+	+	+
Cassandra	++	+	+	-	+
MongoDB	+	+	+	++	++
Couchbase	++	++	+	++	+
Neo4j	+	+	+	++	+

Upon analysis of the results coming from the comparison criteria, Amghar et al. (2018) found that three best performing databases for IoT applications were Couchbase, MongoDB and Neo4j. All NoSQL databases are well suited for scalability and real time processing is needed, but Couchbase proved to be adept in most areas. MongoDB had been observed to be the most capable in the area of data aggregation and processing. Due to the large amount of data being collected by the WSN in this research, MongoDB was used as it proved to have the strongest data aggregation capability.

2.7.1 Sensor node

Sensor nodes are micro-electromechanical systems that monitor the physical changes of their surrounding environment (temperature, humidity, pressure emissions, etc.). The changes are detected and converted into signals to be transmitted for 'surveillance' purposes (Hamami & Nassereddine, 2020). A WSN sensor node comprises of a Micro Controller Unit (MCU), a sensor, a LoRa transmission module, and a battery power supply (independent of the local electrical grid) (Cerchecci et al., 2018). Sensor nodes are referred to as end-devices in the network and application layers. A typical sensor node architecture, shown in Figure 2-5 describes the way in which the aforementioned MCU components (i.e. a sensor, a LoRa transmission module, and a power supply component) interact with one another (Nakas et al., 2020). An important aspect of IoT WSN is coverage and scalability. In order to achieve these two aspects, coverage and scalability, the sensor nodes must be designed and built in an affordable manner, as well as being light weight and easily deployable (Lazarescu, 2013).

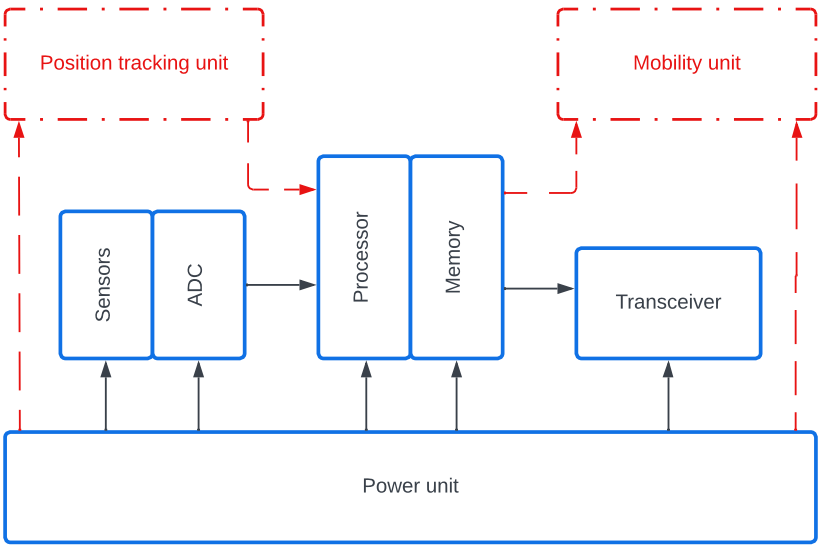


Figure 2-5: Typical sensor node design (Nakas et al., 2020)

2.7.2 Micro Controller Unit and Micro Processing Unit

MCUs are small Integrated Circuits (IC) designed to manage or control other components within an electronic system. This is commonly carried out by a Microprocessor Unit (MPU) (Keim, 2019). Microcontrollers are optimised for embedded systems and are engineered to execute designated tasks and can be seamlessly incorporated into a wide array of applications. These applications span from industrial machinery and warehouse inventory systems to wearable gadgets, household appliances, and beyond (Bhavsar, 2018). Developing complex micro-electronic components are continually becoming cheaper, and with this the distinction between a MCU and MPU also becomes less clear. However, the primary difference between the two components is that a MCU is designed to stand alone with a direct connection to sensors and peripherals whereas a MPU is designed to maximise the computing ability of a chip (Lutkevich, 2019). MCUs make use of Central Processing Units (CPU), which essentially performs the same function as a MPU. For an IoT sensor node, using a MCU over a MPU is often more desirable (Teel, 2022).

MCUs are the sum of several parts and core elements, with the first being the CPU. The CPU processes data which is then used to guide the function of the MCU. These processes encompass a range of tasks, including arithmetic computations, Input/Output (I/O) operations, logic functions, and the transmission of directives to various elements within an embedded system (Lutkevich, 2019). The second core component of an MCU is its memory. The MCU's memory is responsible for storing program code and data. Memory blocks are used by CPUs to process the stored data to execute certain tasks. Memory blocks in MCUs are referred to as 'arrays', which are split into cells that store data and can be called/accessed by utilising unique identifiers containing its location (address) and its position within an array (Arduino et al., 2023). Two types of memory exist and are referred to as being volatile or non-volatile. Volatile memory is also known as temporary memory as it stores data during runtime but once the

device is switched off, the data is lost forever. Volatile data is useful as it fetches and stores data at high speeds (Arduino et al., 2023; Renesas, n.d.). On the other hand, non-volatile data, also known as permanent memory, is constant, regardless of whether the MCU is switched on or not. Non-volatile memory is used to store information pertaining to start-up programs or constant values ("MCU Basic Structure/Operation, Renesas," n.d.). Examples of non-volatile and volatile memory storage are random-access memory (RAM) and read-only memory (ROM), respectively. The third core element or component are Input/Output (I/O) peripherals. I/O ports receive data from sensors or actuators in binary form. The CPU processes the data to determine what instructions are required to be communicated via the output ports to perform processes or functions outside of the MCU (Arduino et al., 2023). In this sense, I/O peripherals enable the CPU to interact with the surrounding environment. The fourth and fifth core components within an MCU are the Analogue to Digital Converter (ADC), and the Digital to Analogue Converter (DAC) (Yelamarthi et al., 2017). ADC convert analogue signals incoming from sensor readings of the external environment to a digital representation to be processed by a digital computer (Oshana, 2006). As the name suggests, DAC perform the inverse function of ADC. DACs are crucial in MCUs as they convert digital to analogue data (an electrical voltage) which is used to control external electrical components such as a temperature sensor (Stanbury et al., 2017). The sixth component is referred to as the system bus. The system bus are the cables and connectors that provide a route for data to travel between the CPU and memory of the MCU. The system bus also facilitates the communication of control signals between important components within and surrounding the MCU (Rouse, 2015).

2.7.3 Sensors and actuators

In most, if not all IoT projects, sensors and actuators make up the first and last steps of the overall process. The basis of sensing and actuation is the science of transduction, which in short, is the process of converting energy from one form into another (Misra et al., 2021). Sensors and actuators act as transducers, components or devices designed to facilitate transduction. Bröring et al. (2011) defined sensors as "devices that convert a physical, chemical or biological parameter into electrical signals". Sensors are designed to measure, monitor, quantify, or respond to stimuli or changes within the environment that they have been deployed. Sensors convert these stimuli into electrical pulses/signals. It is key to note that sensors are not intended to influence the environment in which they are placed (Misra et al., 2021). The difference between sensors and actuators is important to note as both are transducers. Actuators, unlike their sensor counterpart, converts electrical signals into mechanical work, which is intended to influence the surrounding environment (*Difference between Sensors and Actuators - Javatpoint*, 2021; Misra et al., 2021).

Sensor technology has come a long way and is continually becoming smaller and more affordable, with higher functioning complexity and more power efficiency. Given these improvements, sensor technology is being used in an array of applications ranging from (but not limited to) disaster management, environmental monitoring, agriculture, early warning systems, safety and security, and even population health (Bröring et al., 2011). Sensors can be classified based off three considerations: power requirements, output, and the measured

property. The power requirements of a sensor are dependent on whether the sensor is passive or active. Passive sensors require an external power supply as they use an external voltage to interact with the observed properties of the surrounding environment. The voltage drop across a passive element is converted into a sensing output (GeeksforGeeks, 2022; Misra et al., 2021). A common example of a passive transducer is a temperature sensor that uses the properties of a thermistor to generate temperature readings. The resistance of these components is inversely proportional to temperature, this relationship is used to compute temperature readings (Bentley, 1984).

Active sensors on the other hand do not require an external power supply to generate an output. Active sensors interact directly with the surrounding environment and convert stimuli into electrical signals internally as the sensing output(Misra et al., 2021). Photodiodes are a well-known type of active sensor, made from semiconductors, to detect light and create a photocurrent, or light-induced current, as a result of the photoelectric effect (Bolic, 2023). The output of the sensor is a key factor when determining whether additional circuitry components are required before the output is communicated to the CPU. Typically, modern day CPUs are digital which means integrating sensors with a digital output is seamless. This seamless integration however is not the case with sensors that produce an analogue output, where an ADC is required before the output can be processed (Misra et al., 2021). The measured property is a major determinant for the number of sensor nodes that are required for a task. If the property being measured presents low spatial variability, fewer sensors are required to monitor a given area.

2.7.4 Sensor node power supply

IoT WSN technology over the years has become a popular choice for collection and monitoring remote areas of large scale. As the use of IoT WSNs increase globally, a central focus is on developing methods of power supply capable of ensuring stable, consistent and continuous operation of sensor nodes (Raj & Steingart, 2018). Prasad & Chawda (2018) said, “Power management is a key challenge in the field of IoT devices using batteries for a longer time since various factors impact the battery lifetime and understanding their role helps in optimising our solutions.”

2.7.4.1 Batteries (energy storage)

Technology generally improves with the miniaturisation of components. Smaller components often decrease defects, reduces power requirements and are cheaper to produce, while increasing scalability. This however is not the case for battery technology. The primary barrier to miniaturising batteries is that its performance depends on the quantity of material available within a given system (Raj & Steingart, 2018). An array of batteries are available for use and suitable for IoT applications. Popular choices are Lead-Acid batteries, Nickel-Metal Hydride batteries (NiMH), Lithium-Ion batteries, Solid-State batteries, Alkaline batteries, Zinc-Air batteries and Flow batteries. By cross referencing literature, the most common and viable battery options for IoT applications are lead-Acid, NiMH, Lithium-ion, and Alkaline batteries.

Lead-Acid batteries are one of the older forms of battery technology available. Lead-acid batteries make use of chemical reactions between two cathodes, lead dioxide (PbO_2) and lead (Pb), that are immersed in an electrolyte solution, sulphuric acid (H_2SO_4). These batteries are affordable and reliable, but deteriorate in performance when left to discharge for too long. More modern battery technologies also provide a higher energy density (Hasan et al., 2023).

In applications where recharging is required, NiMH batteries have become the more favoured alternative to Nickel Cadmium (NiCd) batteries. Part of the reasoning for this change is that NiMH batteries are far less harmful for the environment (Somov & Giaffreda, 2015). Compared to its NiCd relative, NiMH also possesses a greater energy density and has a less prolific 'memory effect' (when a batteries ability to completely discharge deteriorates). Whilst outperforming its NiCd predecessor, NiMH batteries do have a higher discharge rate and self-discharge rate and are less energy dense than Lithium-ion alternatives (Hasan et al., 2023). Lithium-ion batteries came into fruition as a solution to the deficiencies presented by its predecessors, namely, NiMH and lead-acid batteries, with relatively low energy densities and little promise for future potential (Li et al., 2018).

Lithium-ion batteries are a popular choice for an array of applications including portable electronics and electric vehicles. The reasoning for its popularity is its higher energy density, lower self-discharge rate and a lesser 'memory effect' when compared to alternative rechargeable batteries. A key factor explaining Lithium-ion battery performance is that electrodes are not required to be immersed in an electrolyte to generate electricity. This aids in limiting battery deterioration and adds to the total number of charge and discharge cycles of the battery (Hasan et al., 2023). Lithium-ion cells have increased dramatically in energy density from 55 watt-hours/litre (Wh/L) in 2008 to 295 Wh/L by 2015, the cost of these batteries have also fallen significantly from $1000/kWh in 2010 to $ 145/kWh in 2016 (Raj & Steingart, 2018). With many variations of lithium-ion cells, one presents itself with the most potential, the Lithium-Polymer (LiPo) battery. LiPo batteries mitigate against safety hazards and leakage by eliminating the need for a fluid electrolyte. The absence of a fluid electrolyte also makes LiPo batteries more adaptable for a wider range of applications, due to being more lightweight and compact, making it more commercially viable than conventional lithium-ions. Solid-state batteries offer a potential solution to the use of liquid electrolytes whilst also addressing many of the shortcomings of the current Lithium-ion battery, by implementing a solid-state ceramic electrolyte (Hasan et al., 2023). Alkaline batteries are by far the most used battery in households. This makes the alkaline battery more popular as they are low-cost batteries and work well in low-current-draw applications. Practical draw backs of Alkaline batteries are that they do not work well in cold conditions and struggle to power high-current-draw applications (Young, 2008). The most significant drawback for using Alkaline batteries is that they are primary cells, which suggest they are not rechargeable. Nonetheless, alkaline batteries are suitable for IoT applications that require infrequent battery replacements, long downtime periods and are aiming to be low-cost (Hasan et al., 2023).

2.7.4.2 Harvesting and generating power

Sensor nodes and end-devices are often not able to meet the requirements of a prolonged operation with only a battery unit as a power supply. Thus, investigating opportunities linked to harvesting energy has emerged as a promising alternative or additional solution. Energy-harvesting is the practice of converting 'ambient energy' into electrical energy (Sudevalayam & Kulkarni, 2011).

2.7.4.2.1 Energy-harvesting architectures

Energy-harvesting can be divided into two overarching categories given the requirements of the application; harvest-use architecture mode and harvest-store-use-architecture. The former, harvest-use architecture, is when the energy-harvesting system supplies the sensor node directly. As a result, the power requirement of the sensor node must always be comfortably below the energy supply of the harvesting system, if the energy supply is inadequate, the node will be un-operational (Sudevalayam & Kulkarni, 2011). Kansal et al. (2007) refer to this energy-harvesting system design as 'energy-neutral operation'. A node is said to be in energy-neutral operation when the required power supply can be supported perennially. Moreover, the harvest-store-use architecture is where a storage unit/component is used in tandem with the energy-harvesting apparatus. In cases when more than the required energy is being generated, the storage unit will store the excess. This storage capability is particularly useful when ambient energy is not available for harvesting or when the sensor requires a stint of increased power supply to maximise performance; this energy can be drawn from the storage units This architecture is well suited for harvesting 'uncontrollable but predictable' energy sources such as solar (Sudevalayam & Kulkarni, 2011). Single or double stage storage components can be used with a secondary storage facility to cover for when the primary storage is depleted (Jiang et al., 2005). This method maintains energy neutral operation as a minimum but enables the sensor node to exploit the available ambient energy to improve its capabilities and power management, which effectively maximises its performance life (Kansal et al., 2007).

2.7.4.2.2 Solar as a source of ambient energy available for harvesting

Many sources of harvestable energy exist and are viable, these are (but not limited to) wind energy, solar energy, vibration energy, acoustic energy, and thermal energy. Solar energy in particular has garnered major interest as it shows the potential to be a viable renewable source of energy to be applied to an array of applications (Hao et al., 2022). Much research has gone into improving the efficiency and performance of solar Photovoltaic (PV) systems; these systems convert sunlight into electricity using the photovoltaic effect. Irradiance from the sun supplies 100 mW/cm^2, with energy conversion efficiency improving from 15-20% in 2004 to 26.7% by 2017 as the PV technology advanced (Akorede, 2022; Raj & Steingart, 2018). Energy sources can be divided into four categories: uncontrollable but predictable, uncontrollable and unpredictable, fully controllable, and partially controllable. Energy sources that cannot be made to provide energy or a harvesting window when required are deemed uncontrollable, the inverse is true for controllable energy. Controllable energy sources allow the user to generate energy on command. If the availability and abundance of the energy source can be determined to a high degree of accuracy and consistency, it is deemed to be predictable. Conversely, if the

availability of the energy source cannot be consistently determined, it is deemed to be unpredictable (Kansal et al., 2007). Solar energy is uncontrollable but what makes it so useful is its abundance and predictability, allowing for future proof design opportunities. The abundance and predictability of the energy source is particularly important when designing an IoT WSN being deployed in inaccessible remote locations for extended periods of time.

2.7.4.3 Energy saving techniques

While using high quality energy storage components as well as utilising energy-harvesting techniques is crucial for achieving WSN longevity, further optimisation can be found in exploring ways to minimise the current draw of the node itself (Somov & Giaffreda, 2015). Energy consumption of a battery powered IoT sensor node can be broken down into the different operations being executed by the various components within the system. These operations are radio-frequency communication, processing of data (e.g. transducer data, self-checks, Real-Time Clocks (RTC)), sensing, onboard safety components and energy required to power critical components during power down modes (Lazarescu, 2013). Over and above the required energy consumption, there are four significant reasons for energy leakage in a WSN node: (i) the first being unnecessary energy consumption whilst the node is in idle mode, when the node is not collecting or sending data but rather waiting to do so or listening for transmissions; (ii) the next reason is packet collision during transmission, as these packets are required to be re-sent; (iii) the third being receiving transmissions that are not addressed to it; and (iv) the fourth and final major reason is the transmitting of control packets (Sendra et al., 2011). Energy-saving/efficient techniques were developed based on the understanding of the primary causes for energy consumption combined with the identified energy leakage culprits. Four classifications of energy-efficient techniques can be used to mitigate against the aforementioned energy leakage and consumption challenges:

- Data driven approach
- Duty cycling
- Energy-efficient routing protocol
- Mobility of nodes or gateways

The data driven approach aims to minimise the quantity of the data being sensed, which then reduces the amount of data required to be processed and delivered to the gateway. Duty cycling focuses on implementing sleep cycles for the transceiver module and topology control to manage energy usage. A well-designed routing protocol reduces the energy consumed by end-to-end communication. Finally, mobility of nodes and gateways manage the distance of transmission between them by assigning more energy intensive tasks to mobile nodes (Khriji et al., 2018).

2.8 Application of IoT to monitor the UHIE and cooling effect

Urbanisation has major impacts on the local climate. This is due to the consistent paving of land cover and replacement of natural greenery with artificial building materials, resulting in the formation of UHIs. As urbanisation continues to increase steadily, engineers, urban

planners, and researchers alike are becoming more interested in developing ways in which UHIs can be monitored and predicted, using IoT WSN systems. This section covers previous investigations in a similar context to this research.

In Depok, Jawa Barat, Indonesia, an IoT WSN was developed to collect and live stream temperature data. The data from the IoT network was used to train an artificial neural network designed to predict the future impact of UHIs. Data from the network was also streamed to a user interface where the temperature data could be observed at any point on a website, providing real time information on temperature and UHIs. Hardware used in this sensor node design was an ESP8266 MCU coupled with a DS18b20 temperature sensor (Hidayat & Soekirno, 2021).

The deployment and operational challenges of a temperature sensor network was also experienced in a study in Birmingham, United Kingdom. These challenges were highlighted in a study testing technical applications to develop infrastructure for a 'smart city' through utilisation of Aginova Sentinel Micro (ASM) sensors to measure UHIs (Chapman et al., 2014). Positioned strategically throughout the city, these ASM sensors were powered by an AA 3.6V Lithium-Thionyl Chloride battery. Sensor nodes were equipped with weatherproof enclosures and wireless communication cards which leveraged existing WiFi networks for data transmissions to significantly reduce deployment costs. Integration with nearby WiFi networks (e.g. schools) had presented obstacles, mostly stemming from issues related to integrating modern sensor technology with old WiFi infrastructure, impacting communication and battery-related issues. Despite these challenges, the study underscored the potential of the ASM sensor network in supporting smart city initiatives and IoT systems, contingent upon overcoming initial hurdles and ensuring reliable internet connections and power supply. This highlights the importance of addressing connectivity and maintenance concerns to fully leverage the benefits of IoT technology in urban temperature monitoring and management.

Barthwal & Sharma (2021) developed a location aware IoT sensing system designed to observe and analyse the severity and intensity of the UHI effect in different geographical regions. The temperature data collected was used to predict the surface temperature within these regions. The hardware used in the sensor node design was the temperature-humidity sensor (DHT 22), Arudino Uno R3 board, 5V battery 4500 mAh. A Raspberry Pi 3 Model B and LTE stick was utilized as the gateway.

In the city of Hyderabad, India, researchers proposed the implementation of a novel IoT sensor network to provide real time data of the local climate. The aim of this study in Hyderabad was to collect data using heterogeneous sensor data via the same IoT network. The study ran for 15 days in 2018 and yielded successful results. Sensor nodes used a Peripheral Interface Controller (PIC) MCU to power the sensors and transmit the data to a Raspberry Pi gateway. Data was uploaded to the cloud and transmitted the data in the form of a Short Message Service (SMS) using GSM. The transducer that was used to acquire temperature and humidity data, was a DHT11 sensor (Kishorebabu & Sravanthi, 2020).

Das et al. (2018) built an operational IoT WSN prototype designed to monitor surrounding environmental parameters. An Atmega328 MCU interacted directly with sensors to collect temperature, humidity, rain, sunlight intensity and air pollution data. Unlike many other IoT sensor networks that utilize a central gateway and radio communication, these sensor nodes were coupled with an ESP8266 WiFi module. The WiFi module allows for direct communication with online servers without the need of a gateway to act as a communication bridge, however, this method comes with higher costs as it requires more energy to power each sensor node.

Finally, research conducted in Minna, Nigeria, by Ajao et al. (2017), used the IoT concept to develop a sensor network to monitor temperature and humidity data over an extended period of time. In the case of this project, the aim was not to measure UHIs per se, but to develop a WSN capable of monitoring climates to aid the agricultural industry. Even so, the framework for developing this IoT WSN followed the same design steps and principles as other IoT networks that intended to collect temperature and humidity data. Noteworthy, Das et al. (2018) and Ajao et al. (2017) used the ATmega328 in tandem with an ESP8266 WiFi module.

3. Methodology

This section describes the method implemented for this project. The overall objective of this study was to assess UHI and the cooling effect of shaded green spaces using IoT network. The study included comparing the ambient air temperature and humidity data collected by the sensors of shaded to unshaded spaces. The methodology involved identifying project requirements, site selection, hardware development, software/code adaptation, LoRaWAN network connection, data collection, storage, and analysis.

3.1 Research Framework

The research framework for this project included systematic observation and designed experiments as shown in Figure 3-1.

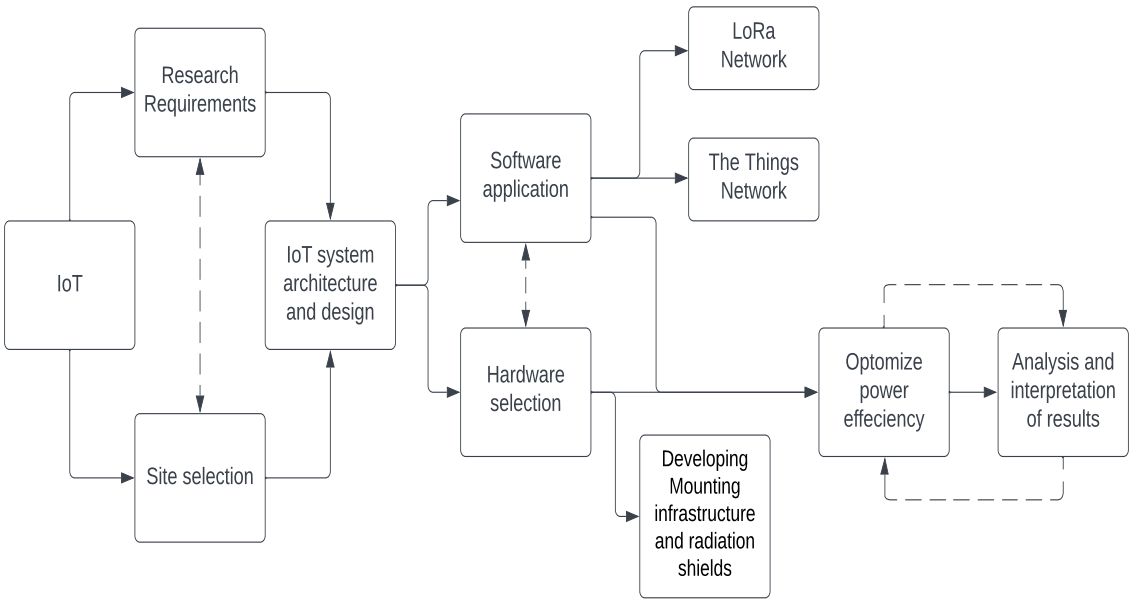


Figure 3-1: Research Framework

3.2 Site selection

This section details the process outlining the areas where data was collected. The study assessed temperature and humidity disparities between shaded green and unshaded paved spaces to determine how cooling can mitigate UHIE. The site had to be easily accessible to enable troubleshooting and servicing of sensor nodes. The security of end-devices was also a major consideration. The underpinning main requirements for the site selection was accessibility and security of the study area. The locality for UCT is shown in Figure 3-2.

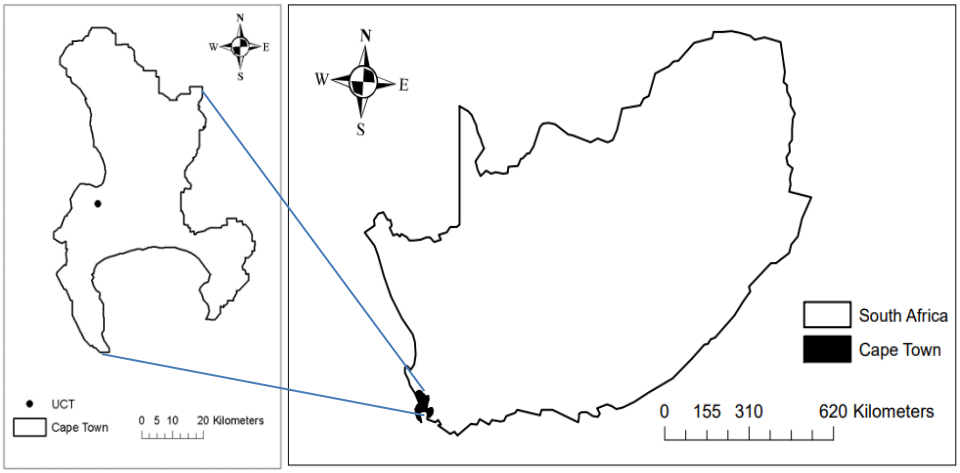


Figure 3-2: UCT locality map

The UCT is divided into three campuses, Upper, Middle and Lower campus. Each location varies slightly with land cover. Upper campus has the highest density of buildings and paved areas, but generally all sites have paved areas with some green spaces. Sensor nodes were placed in positions to collect ambient temperature and humidity data of built-up spaces, and green spaces, to draw comparisons between the two. An important variable to note is the difference in elevation between each campus was negligible. The data collection period was from September 2023 to the end of December 2023. This period allowed for the monitoring of temperature and humidity trends as the weather conditions progressed from ‘cooler’ more ‘wet’ months to the warmer ‘dryer’ months of the year.

Figure 3-3 also shows the final locations selected for each sensor node. The yellow symbols represent shaded sensor nodes and the red represent the unshaded. In total, 14 sensor nodes were deployed, with the bulk, eight, of these being positioned on upper campus, four on middle campus, and two on lower campus. The majority of the sensors were placed on upper campus due to the extensive land cover with a variety of unshaded and shaded spaces, such as paved or tree covered areas. In addition, upper campus covers the largest area in terms of space, followed by middle and then lower campus. Lastly, the LoRa gateway was stationed on upper campus atop the New Engineering Building (NEB), minimising the range of transmission for the majority of sensor nodes.

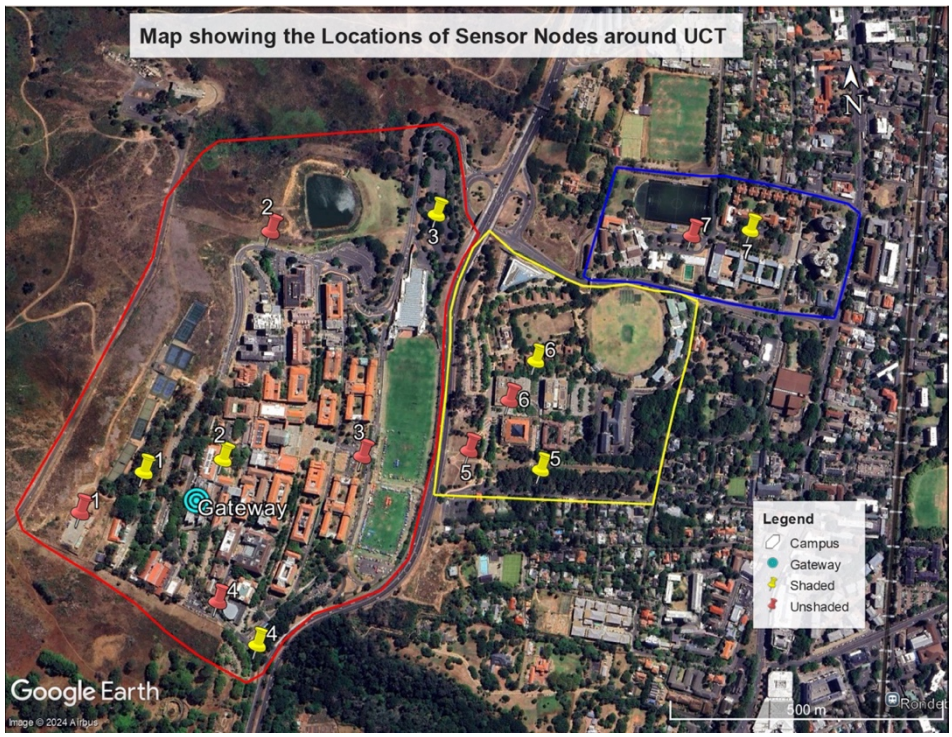


Figure 3-3: Sensor node locality map

3.3 System architecture

The architectural design focused on the requirements outlined by the objectives in Section 1.2. Primarily, the system was required to transmit data and information, wirelessly, collected by sensor nodes via a local gateway, enabling users to interact with the data through a web page (Pretorius & Mthethwa, 2019). The best suited architecture for this application is also the most simple with a total of three layers, perception, network and application. The methodology was broken down into three major stages, following the development of each of the layers (perception, network and application). It is important to note that although the methodology is categorised by these three layers (shown in Figure 2-1), many of the processes had to be completed simultaneously, interchangeably, and iteratively to ensure that all layers operated harmoniously.

3.3.1 Perception layer

The first stage of developing the IoT sensor network was the perception layer. The perception layer encompasses all the end-devices tasked with monitoring and collecting temperature and humidity data of its local environment. The full node design and development was completed in this stage. The data collected in the perception layer was sent using the LoRa breakout board on the MCU and transmitted to the network layer.

3.3.2 Network layer

The data from the perception layer was sent via a local gateway (acting as a communication bridge) to the network layer, using LoRaWAN technology. In the network layer, devices were registered with the LoRa network and connected to TTN. Payloads of uplinked data from the

perception layer were decoded and processed before being distributed to servers. TTN services were used as the primary tool.

3.3.3 Application layer

The application layer is where the user interacted with the data. The application layer of the system was managed predominantly by Mongo DB, Vercel front-end development (handles the user interface) and Application Programming Interface (API) to represent the data on a live streaming map. Data from the network layer was sent to the application layer via an automated 'Web-hook' to the Vercel front-end.

3.4 Hardware selection and application (start of perception layer)

The hardware selection focused on the requirements of the IoT sensor node network for this study. Hardware components had to be locally accessible and "off the shelf", this included determining the feasibility of using IoT as a hydrological application in the South African context. Additionally, a major focus was on designing an inexpensive IoT sensor network.

3.4.1 Gateway

End-devices in the perception layer of this setup was built around MCUs utilising LoRa radio transceivers. This meant the end-devices were not able to interact or transmit data via WiFi, given the dissonance between the two frequency bandwidths (end-devices operating at 863 MHz where WiFi operates between 2.4 – 5 GHz). In order to connect end-devices to WiFi, required for enabling data transmission to TTN, a LoRa gateway was used to bridge this gap in communication. The 'SenseCAP Gateway – LoRaWAN' developed by Seeed Studio was selected for the study. The gateway came with the option to either transmit data using ethernet or 4G (sim card), making it suitable for sites in remote locations. Coverage of the gateway goes up to 10 km in line of site and 2 km in urban spaces (*SenseCAP Gateway - LoRaWAN EU868, US915, AU915, AS923 - Seeed Studio IIoT Solutions*, n.d.)

3.4.2 Node Design

Design focused on building pervasive nodes in the context to the environments, as well as being durable in the outdoors. For cost effectiveness and accessibility, only 'off the shelf' materials and products were used. Figure 3-4 shows the final version of the unshaded sensor node that was deployed.

Figure 3-4: Completed node assembly

3.4.2.1 Micro controller unit selection

The MCU is the central electrical component in the node design of which all external components are dependent upon. Selection of the MCU came down to which boards were most suited for integrating with the required sensor, user accessibility, LoRaWAN technology compatibility, power consumption and cost. Two boards were identified as strong fits for the study, those being the Maduino and LoPy 4 shown in Figure 3-5 and 3-6, respectively. The Maduino is an MCU based on the Arduino Pro mini design powered by the ATmega328P chip. To enable the Arduino Pro Mini to communicate with LoRa, a separate breakout board with the LoRa radio component would have been required to be integrated into the circuit. Maduino offers a product with an RFM96 868MHz module already embedded into the MCU board, thus preferable as it provided savings on monetary costs and time. The Maduino is compatible with the Arduino IDE and the C++ programming language (*Maduino ATmega328 + RFM96 868MHz - Micro Robotics*, n.d.).

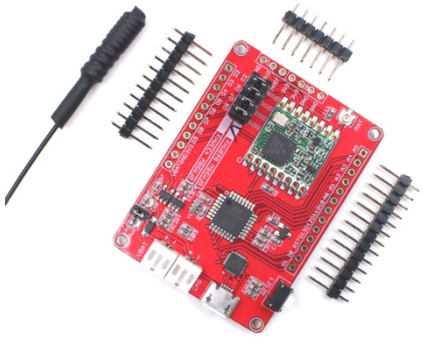


Figure 3-5: Maduino ATmega328 (*Maduino ATmega328 + RFM96 868MHz - Micro Robotics*, n.d.)

LoPy 4 is a Micropython compatible device developed by Pycom. Powered by the Espressif ESP32 chipset, as well as possessing a dual WiFi radio system. Pycom dubs the LoPy 4 a "compact quadruple network" board, as it is compatible with LoRa, Sigfox, WiFi, and Bluetooth. The LoPy 4 has an input voltage of 3.3V and 5V (*LoPy4 - Pycom - Quadruple Bearer MicroPython Enabled Dev Board*, n.d.).

Figure 3-6: Pycom LoPy 4 (*LoPy4 - Pycom - Quadruple Bearer MicroPython Enabled Dev Board*, n.d.)

The LoPy 4 MCU provides more communication protocol options, as well as coming with extensive capabilities. However, the board itself was considerably more expensive than the Maduino and required a custom-made Printed Circuit Board (PCB). In addition, there were

doubts surrounding whether Pycom would continue producing the LoPy 4 board. Thus, it was ultimately decided that the Maduino alternative would be used for this study.

3.4.2.2 Radio

To transmit temperature and humidity data over substantial distances, the RFM96 868 MHz LoRa radio was selected. As stated in Section, 3.4.2.1, that the LoRa radio module had already been integrated into the board, this made the pair of components a strong match given the requirements of the project. Combined with the patented LoRa modulation technique, the radio was designed to be used in long range sensing applications and still remain inexpensive (Hoperf Electronic, n.d.; *Maduino ATmega328 + RFM96 868MHz - Micro Robotics*, n.d.).

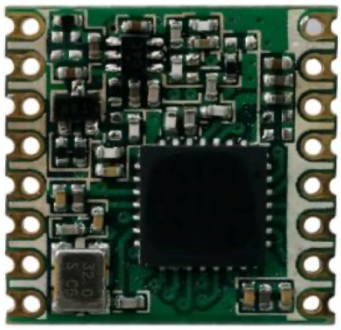


Figure 3-7: Low power long range transceiver module (Hoperf Electronic, n.d.)

3.4.2.3 Antenna

To increase the range of transmission, a LoRa antenna was fitted to the node. The selected antenna has 5 dBi of gain and 50 Ω impedance, making it ideal to use for frequencies between 850 MHz and 950 MHz. The Sub Miniature version A (SMA) connection made it easy to fit and remove from the Maduino MCU, reducing the effort required to maintain or replace components in nodes (*GitHub - Microrobotics/YN-868MHZ-5DBI_datsheet: Yetnorson YN-868MHZ-5DBI Antenna Datasheet*, n.d.).

3.4.2.4 Sensor: SHT 31

All sensor nodes used the SHT 31 weather proofed temperature and humidity sensor, shown in Figure 3-8. Integrated capacitive humidity and band-gap temperature sensing components enabled the sensor to produce accurate and reliable results (humidity accuracy of ± 2%, temperature accuracy of ± 0.2 °C), low current draw and anti-interference capabilities. The SHT 31 is compatible with most 3.3V/5V MCUs and incorporates I2C communication protocol, allowing for easy data reading.

Colour		Label	Description
Red		VCC	+
Black		GND	-
Green		SDA	Data line
Yellow		SCL	Clock line

Figure 3-8: SHT 31 sensor

The sensor and MCU have to communicate with one another to co-ordinate when and how the data is collected and transferred between the two devices. The selected method was the I2C communication protocol. The primary benefit of using I2C devices in this application was that it allowed for back-and-forth communication between MCU and the sensor, whilst only using two pins and effectively only two wires. As the sensor nodes in this study were built to be simple and compact this was highly useful. A secondary benefit, for future progression, of using this design is that the I2C protocol facilitates the assignment of unique addresses to each device. The allocation of unique addresses enables both controller and peripheral devices to alternate communication over a single line. Consequently, an Arduino board can effectively communicate with numerous devices or other boards, whilst still utilising only two pins of the microcontroller (*Inter-Integrated Circuit (I2C) Protocol | Arduino Documentation*, n.d.).

All Arduino MCUs are compatible with I2C communication protocol, making it simple to use in this design. In the case of the Maduino board the Serial Data Line (SDA) and Serial Clock (SCL) pins are Analogue 4 (A4) and Analogue (A5) respectively. Figure 3-9 shows an Arduino MCU I2C being utilised to integrate with the SHT 31 sensor.

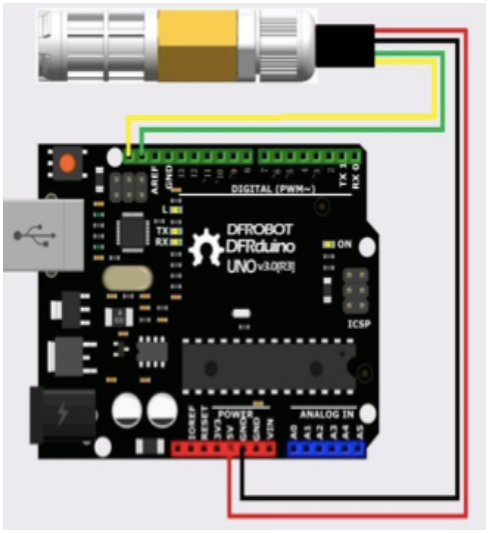


Figure 3-9: Sensor and MCU diagram (*SHT31 Weather-Proof Temperature & Humidity Sensor - DFRobot*, n.d.)

3.4.2.5 Circuit breakout board

Although the LoRa radio and the MCU had already been integrated, specific pin connections between the radio, MCU, and the SHT 31 sensor was required. Connections between Digital pins 4,5 and 6 and data I/O pins 0,1,2 along with analogue pins 4 and 5, as well as the MCU's GND and PWR pins being used to interface and power the SHT 31 sensor. To facilitate these connections, a circuit breakout board was designed and soldered according to the circuit design/schematic shown in Figure 3-10. The addition of a circuit breakout board ensured stable pin connections and mitigated the risk of the MCU and accompanied components from short circuiting. This circuit breakout board design approach is also the most suitable option when building a multiple sensor node system as reproducing these breakout boards are simple and cost effective. A further benefit was the "plug-and-play" nature of this design, with MCUs being

easily fitted and removed from female pin sockets on the breakout board. Figure 3-10 shows the schematic of the developed circuit breakout board.

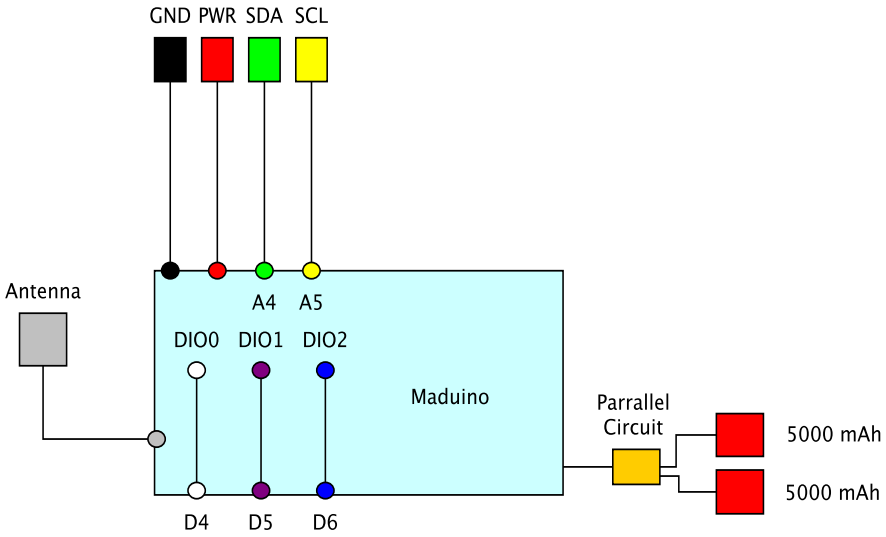


Figure 3-10: Circuit breakout board schematic

3.4.2.6 Packaging temperature and humidity data

Using the float **temp = sht.getTemperature()** and float **hum = sht.getHumidity()** functions from the SHT31 library, the SHT 31 sensor collects two values of data, temperature and humidity respectively, as float values. Initially, the approach was to concatenate the two float values and transmit them as a single data packet. However, this proved unfeasible because the concatenated values created a packet that was too large for efficient transmission. As LoRaWAN is designed for long-range, low-power transmissions, it was crucial to keep data as compact as possible. The second and successful approach, was to convert the data from a float to a far less memory intensive format. Float values were converted into 16-bit integer values multiplied by 100 which ensured that two decimal places were retained. Next, the 16-bit integers were packed into an unsigned 8-bit integer byte array of four elements.

The **LMIC_setTxData2** function was then called to prepare the data for transmission. This function is part of the LMIC library and was used to queue data for transmission. The parameters were:

- **1**: Port number. i.e., the port on the LoRaWAN network to which the data was to be sent.
- **data**: The byte array containing the data to be transmitted.
- **sizeof(data)**: The size of the data array in bytes.
- **0**: This parameter was not used in this call.

```
// Get temp and humidity as float
        float temp = sht.getTemperature();
        float hum = sht.getHumidity();
        Serial.println(temp);
        Serial.println(hum);

        // Convert float to int (multiplied by 100 to preserve two decimal places)
        int16_t tempInt = temp * 100;
        int16_t humInt = hum * 100;


        // Pack the integer values into a byte array
        uint8_t data[4];
        data[0] = tempInt >> 8;
        data[1] = tempInt;
        data[2] = humInt >> 8;
        data[3] = humInt;

        // Prepare upstream data transmission at the next possible time
        LMIC_setTxData2(1, data, sizeof(data), 0);

        Serial.println(F("Packet queued"));
    }
    // Next TX is scheduled after TX_COMPLETE event.
}
```

Figure 3-11 Packaging temperature and humidity data code

3.4.2.7 Preliminary testing: phase one

Two testing phases were implemented throughout the sensor and network building process. This was to test/stress-test the network at major milestones in its inception for flaws or errors. This section describes the first of those tests. Testing phase one focused on trouble shooting the sensor node's ability to (i) connect to TTN, (ii) collect a temperature and humidity value simultaneously and (iii) successfully transmit this data to TTN. Only once it was determined that the sensor node was capable of achieving all three of these tasks consistently, did the rest of the design continue.

3.4.3 Power considerations

The proposed IoT sensor network required a power supply strategy to adequately power sensors for the duration of the data collection period. This was a crucial step in the design and build of the sensor network, particularly when trying to keep costs low without compromising the performance or maintenance of the network. Two design approaches were taken, for the shaded sensor nodes, batteries were used as the sole energy source; and in the unshaded sensors nodes, batteries were used in combination with solar panels (described in more detail in section 3.4.3.3).

3.4.3.1 Battery

When selecting the battery, the most important selection criteria was the current capacity to cost ratio. The second, was the product availability in South Africa. The third, was how well the battery could be integrated into the sensor node build. Current most common battery technologies include Alkaline, Lithium and NiMH, but Lithium batteries were identified to be the most suited

for this project. Lithium batteries have a greater current capacity and lower discharge rates when being compared to Alkaline batteries. The search for a suitable battery option was narrowed down to four batteries, with varying battery capacities and costs.

Table 3-1 shows the cost break down for each of the four batteries and their capacity ratings. All batteries operate at a voltage rating of 3.7 V. Notably, the 6800 mAh battery presented the lowest costs, however, these ultimately were not used given the unreliability of the battery stipulated by multiple customer reviews of this product. It was decided that the 5000 mAh batteries would be used for the project. Although the 2000 mAh would have been marginally cheaper, using batteries of this capacity would have required the circuit board to accommodate for at least four batteries as compared to two when using the 5000 mAh battery, in parallel. Using the 2000 mAh would have also posed challenges related to space within the electrical housing unit box.

Table 3-1 Battery costing

mAh	2000	3100	5000	6800
Price	R188.00	R319.00	R124.00	R149.00
Price per unit	R47.00	R79.75	R124.00	R74.50
Price per sensor (10 000 mAh)	R235.00	R257.26	R248.00	R109.56
Price per sensor (20 000 mAh)	R470.00	R514.52	R496.00	R219.12

3.4.3.2 Sleep mode implementation (Arduino)

Sensor nodes were programmed to collect and transmit a data reading once every 15 minutes. This provided an opportunity to save battery life in between scheduled transmissions. Equations 3.1 and 3.2 were used to calculate an estimation of how much time the MCU would not be required to collect data (when in idle mode or sleep mode) and operate at a much lower functional level. The values may vary in practice as the board was often seen to spend far less time being awake to take and transmit a reading. The calculation highlighted the potential for energy saving by implementing a sleep mode. Arduino provides various sleep mode options in the "lowpower" library. The Maduino MCU ATmega328 chip has six sleep mode options i.e., powerDown – lowest current consumption, powerSave, powerStandby, powerExtStandby, adcNoiseReduction, and idle – highest current consumption (Khatri, 2019):

$$TC = (SM + AM)/DS \tag{3.1}$$

$$TSM = TC \times SM \tag{3.2}$$

where: TC = Total Cycles in a day (95); SM = Sleep mode duration (900 s); Active mode (10 s); DS = length of one day (86400 s); TSM = Total duration of sleep mode in one day (85450 s)

Sensor nodes were required to enter a sleep mode immediately after confirming a successful transmission was received by TTN, and to be woken up at the end of the 15 minute cycle. To avoid having to use additional external components, any option that required an external interrupt to pull the MCU out of the sleep mode, was disregarded. The "Power Down" sleep mode option was selected as it stopped all internal clocks, which most processes relied on to maintain synchronicity in execution. By halting these clocks, these processes were temporarily disabled, allowing for energy conservation. The stopping of clocks, however, did not apply to asynchronous modules, such as the "Watch Dog Timer" (WDT). The primary function of the WDT was to monitor the MCU to ensure that all programs were running correctly. In the case of the sleep mode, the WDT monitored the time the MCU was in a sleep mode and woke it up at the end of the scheduled duration. A drawback of this method was that the maximum duration the board could enter a sleep mode for, was eight seconds. To overcome this issue, a *for loop* (a programming control structure) condition was written into the code that would only be satisfied once the sleep mode portion of the code had been executed a total of 113 times. This resulted in the MCU being in a power down state for a little over 15 minutes. Figure 3-12 shows the code for the sleep mode implementation.

```
// Confirming that the board has connected to the Things Network
if (joined) {

  //Confirming that the data package has been transmitted successfully
    if (sent) {

      Serial.println("Entering Sleep Mode");
    //15 minute low power
      for( int i=0; i< 113; i++) {
      LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);

      }

      sent = false;
      // wake up and start sending data again

    }
  }
}
```

Figure 3-12 Sleep mode code

To measure how much the sleep mode decreased the current draw, a circuit breakout board was built. The board was placed in series between the LiPo battery and the Maduino MCU. The breakout circuit itself placed a one ohm resistor in series with the battery and the MCU with two pins in parallel with the resistor. The breakout circuit allowed for the nodes of a multi-meter to measure the current draw/voltage drop across the resistor. The purpose of measuring the current draw and voltage drop was for two reasons, the first being to test whether the board was entering a low power sleep mode and, the second was to determine how much lower the current draw would be during these sleep mode cycles. These measurements also provided a theoretical baseline for power consumption and helped to develop a maintenance plan for the shaded group of sensors. As the resistor was one ohm, using Ohms law it was deduced that the voltage measured

would have a 1:1 ratio with the operating current being equal to the operating voltage. Once the current draw of the board had been measured while it was active, as well as, being in a low power mode, the battery life was calculated using the theoretical sleep mode (TSM) duration over 24 hours, calculated in Section 3.4.2.3. Table 3-2 shows the current draw readings of the board.

Table 3-2: Current draw measurements

Cycle no.	Active (mA)	Low power (mA)
1	12,4	8,6
2	12,5	8,5
3	12,5	8,6
4	12,5	8,6
5	12,6	8,6
6	12,5	8,5
7	12,5	8,7
8	12,6	8,6
9	12,4	8,6
10	12,5	8,5
Average (mA)	**12,5**	**8,6**

The average operating current for the board whilst in active mode was 12.5 mA and 8.6 mA during low power mode. With the assumptions made in section 3.4.3.2, these values were used to calculate the battery life duration. The calculation, utilising Equations 3.3-3.6, was as follows:

$$TTC = \frac{1}{360}\ h\ active + \frac{1}{4}\ h\ low = \frac{1}{360} + \frac{90}{360} = 91/360\ \ h \tag{3.3}$$

$$TD = \left(12.5\ mA\ \times \frac{1}{360}\ h\ active\right) + \left(8.6\ mA\ \times \frac{1}{4}\ h\ low\right) = 0.03472\ mAh + 2.15\ mAh = 2.18\ mAh \tag{3.4}$$

where: TTC = Total time per cycle; TD = Total current draw per cycle

Next the average current draw per hour had been calculated as follows:

$$ACD = \frac{2.18\ mAh}{\frac{91}{360}\ hours} = 8.645\ mA \tag{3.5}$$

where: ACD = Average current draw

Using the average hourly current draw, a theoretical battery life was calculated in hours and converted into days:

$$BL = \frac{BC}{ACD} = \frac{5000\ mAh}{8.645\ mA} = 578.3\ h = 24.1\ days \quad (3.6)$$

where: BL = Battery life; BC = Battery capacity

The calculation produced a theoretical battery duration of 24.1 days.

3.4.3.3 Charging module and solar panel (unshaded sensor node design)

Given the requirements and the nature of the research, half of the sensors were deployed into areas with ample solar radiation on the site. This provided an opportunity to investigate an alternative method to extend battery lifespan without having to increase battery capacity. The unshaded power supply aimed to utilise the available solar radiation for energy to overcome the current draw of the sensor and the discharge rate of the battery. To achieve this, a 6V 4.5W solar panel was added to the sensor node. To monitor the battery capacity and manage the current supply to the battery and to the MCU, a TP4056 charging module was built into the circuit. The charging module would monitor the batteries voltage level and supply the battery with current from the solar panel as required. When the battery was adequately recharged, the excess current received from the solar panel was used to power the MCU directly. This configuration meant that in conducive weather conditions, the batteries would only be required to power the sensor during dark nighttime periods. The charging module and solar panel configuration produced a peak current of 750 mA, far surpassing the total draw of the system. The schematic of the charging module and solar panel connected to the MCU is shown in Figure 3-13.

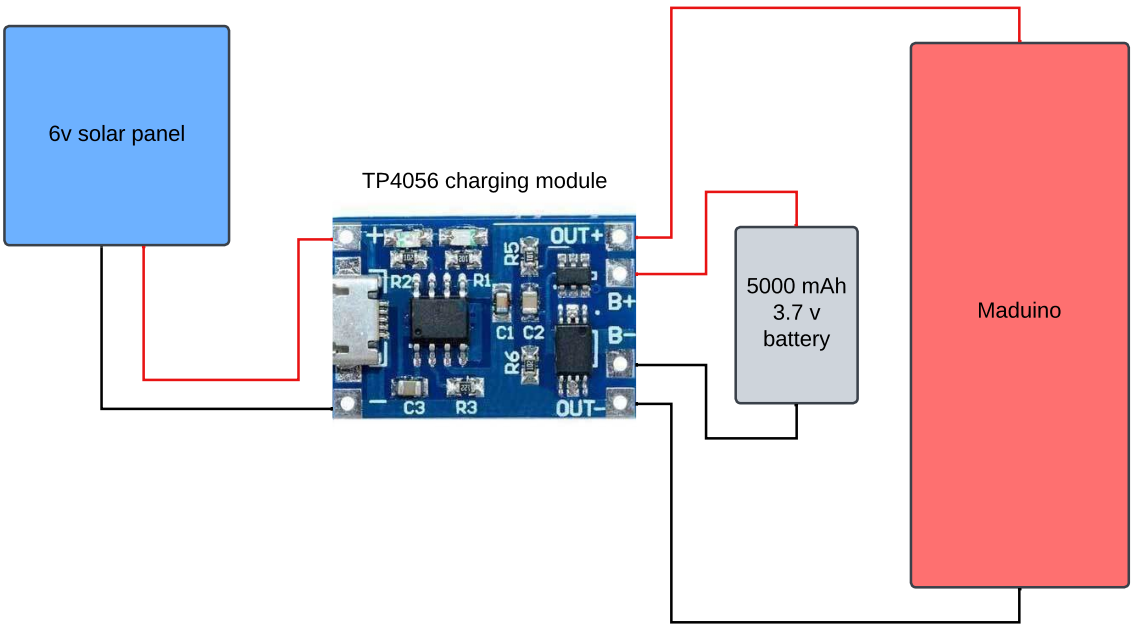


Figure 3-13: Charging module and solar panel schematic

3.4.4 Node design and assembly

Once the electrical and software portion of the node was completed, the overall design and assembly of the final node took place. As these sensors were placed outdoors being exposed to the sun, warm and cold weather, wind, and rain; this step in the process crucially focused on protecting these components from changing weather conditions and climates. Additionally, the node design had to consider the potential for vandalism and tampering. This section details the steps taken to build the final node to be deployed on site.

3.4.4.1 Radiation shield design

The sensors were set up to collect ambient air temperature data of its local environment. To ensure data was not skewed by the sensor being over exposed to solar radiation, solar radiation shields were built to house the SHT 31 sensor. Shields comprised of five inverted plastic plates, held in place by three steel rods. The placement of these rods was designed to be the mounting points onto the node bracket. The radiation shield is shown in Figure 3-14.

3.4.4.2 Bracket and node structure

Sensor nodes were required to be placed at 2.4 m above the ground at all locations, for both shaded and unshaded instances. In the case of shaded sensors, permission was granted to drill the brackets into tree trunks to hold the sensor nodes in place. For the unshaded sensors, damage to buildings and property was a concern, thus lampposts were used within the site. The “bent arm” design allowed for the bracket to be fixed at two points either using screws for the tree trunk application or stainless-steel hose clamps to secure the bracket around the diameter of lampposts. The material used for the bracket was aluminium, a lightweight rust resistant material, further ensuring no damage would be imparted to the lampposts. Given the lightweight nature of the material, securing the nodes were simple and safety issues were considered and mitigated. Further details of this process and being granted permission to place these nodes on UCT campus can be found in Appendix A. The sensor node bracket and solar radiation shield is shown in Figure 3-14.

Figure 3-14: Bracket and solar radiation shield

3.4.4.3 Solar panel bracket

With the addition of a solar panel to the sensor, a secondary bracket had to be made to secure the panel. Initially, mounting the panel to the top of the radiation shield was explored. However, this was decided against as the orientation that the panel would have been mounted would effectively have acted as a wind "sail", causing instability to the sensor node in windy conditions which may have resulted in damage. The final solution required a sturdy light weight design. By 3D printing the bracket, a light weight, sturdy and inexpensive bracket could be produced. A weatherproof adhesive was used to secure the panel to the printed bracket. Figure 3-15 shows the 3D printed solar panel bracket model.

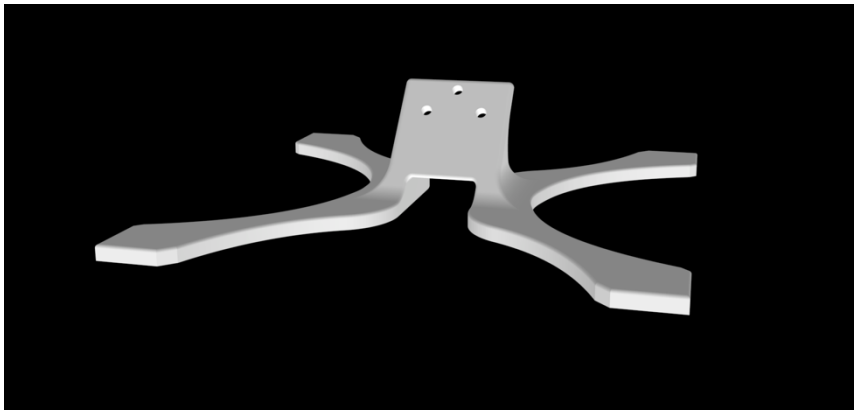


Figure 3-15: 3D printed solar panel bracket model

3.4.4.4 Housing and water proofing

Once the brackets and radiation shields were completed, the electrical housing unit had to be secured to the bracket. The securing of the electrical housing unit to the bracket was an important stage of building the node as it housed the MCU, TP4056 charging module, battery breakout circuits and batteries. All these components were not weatherproof or water resistant, and exposure to weather elements may have resulted in the functional failure of the sensor node. An electrical weatherproof box was used to house the components. Points at which wires and the antenna entered the box, tightly fitted grommets were placed to firmly secure the antenna and wires. Once these entry points were secured, silicone was used to ensure the housing unit was adequately waterproofed. A further security measure was to use an opaque box as the housing unit, to discourage those passing by from tampering with the device.

3.5 Software adaption and application

The following sections describe the steps taken to build and develop the network and application layers; these two layers were mostly made up of software applications. Network communication and connecting to the LoRa gateway, TTN, Mongo DB, and the live stream webpage are covered in the following sections.

3.5.1 Network communication

When establishing connection between the end-device and TTN using LoRaWAN, certain procedures and steps had to be followed, such as, the end-device activation process, pin mapping and feedback code for the user. The following subsections detail these steps.

3.5.1.1 Message structure

Transmissions over the LoRa network are either uplink or downlink messages. The structure of each of these messages is what distinguishes them from one another. Uplink messages utilise the 'LoRa radio packet explicit mode', incorporating the LoRa Physical Header (PHDR) alongside a header Cyclic Redundancy Check (CRC) (PHDR_CRC). A CRC ensures the integrity of the payload, with fields such as PHDR, PHDR_CRC, and the payload CRC being added by the radio transceiver to protect it (LoRa Alliance Technical Committee, 2018).

Every downlink message is dispatched from the network server to an individual end-device, these messages were distributed by the gateway. Downlink messages employ the radio packet explicit mode, encompassing the inclusion of the LoRa physical header (PHDR) and a header CRC (PHDR_CRC). The Section 3.4.2 on payload formatting explains how 'bit shifting' was used to optimise transmission efficiency by minimising the size of each transmission.

All sensor nodes or end-devices were required to be registered with the LoRa network before data messages could be transmitted, the method in which this process was done is called, Over The Air Activation (OTAA), code shown in Figure 3-16. To join LoRaWAN, two Message Authentication Code (MAC) messages need to be exchanged between the device and the network. These messages are a join-request from the device to the server and then a join accept from the server to the device. In order to do so, an AppEUI, DevEUI and AppKey must be generated and embedded into the code of the end-device. AppEUI and DevEUI keys were generated by TTN server, whereas the AppKey is a 'secret' code personalised by the user. The name/identity for each sensor node was also given via TTN ("End Device Activation, The Things Network," n.d.).

```
// This EUI must be in little-endian format, so least-significant-byte
// first. When copying an EUI from ttnctl output, this means to reverse
// the bytes. For TTN issued EUIs the last bytes should be 0xD5, 0xB3,
// 0x70.
static const u1_t PROGMEM APPEUI[8]={ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
void os_getArtEui (u1_t* buf) { memcpy_P(buf, APPEUI, 8);}

// This should also be in little endian format, see above.
static const u1_t PROGMEM DEVEUI[8]={0x51, 0xC9, 0x05, 0xD0, 0x7E, 0xD5, 0xB3, 0x70};
void os_getDevEui (u1_t* buf) { memcpy_P(buf, DEVEUI, 8);}

// This key should be in big endian format (or, since it is not really a
// number but a block of memory, endianness does not really apply). In
// practice, a key taken from ttnctl can be copied as-is.
// The key shown here is the semtech default key.
static const u1_t PROGMEM APPKEY[16] = {0x58, 0xF7, 0x0B, 0xCB, 0xD0, 0x49, 0x26, 0xEB,
0xBE, 0xA1, 0xB6, 0x79, 0xAE, 0x54, 0xB3, 0xAD};
void os_getDevKey (u1_t* buf) {  memcpy_P(buf, APPKEY, 16);}
```

Figure 3-16 OTAA code

Pin mapping in embedded systems is the process in which physical pins of a micro controller are allocated functions or peripheral components. It is essential as it tells the software which pins on the board are required to be used. The structure '**const lmic_pinmap lmic_pins =**' was used to assign functions to certain pins. The pin mapping code used is shown in Figure 3-17.

```
// Pin mapping
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = {4,5,6},
};
```

Figure 3-17 Pin mapping code

By consulting the MCU schematic documentation, found in Appendix B, along with the requirements of LoRaWAN connection, the pin mapping/assignment was as follows:

- **NSS (Slave Select):** Pin 10
- **RX/TX Switching:** Not Used
- **Reset Pin:** Pin 9
- **Digital I/O(DIO) Signals:**
    - DIO0: Pin 4
    - DIO1: Pin 5
    - DIO2: Pin 6

The RX/TX switching function was not required for the function of the nodes in this case.

Once the end-device activation and pin mapping was completed, the end-device was ready to establish communication with the LoRa network and transmit data to TTN. However, this communication protocol had many functions and processes that had to be completed for successful transmission to take place. Some events or scenarios also required an action or a response to allow for the MCU to execute a successful transmission. To manage this, code that did not return any value, a void loop, was written, "**void onEvent (ev_t ev)".** This handled the behaviour of the device when confronted with the certain situations that required a decision. It also provided the user with feedback when a certain event milestone had been reached in the process by printing prompts on the serial monitor such as "**EV_JOINED**". The prompted serial monitor feedback proved to be a great assist when troubleshooting the software during testing. This code is shown in Figure 3-18.

```
void onEvent (ev_t ev) {
    Serial.print(os_getTime());
    Serial.print(": ");
    switch(ev) {
...


case EV_JOINED:
            Serial.println(F("EV_JOINED"));

            // Board has joined the network
            joined = true;

            // Disable link check validation (automatically enabled
            // during join, but not supported by TTN at this time).
            LMIC_setLinkCheckMode(0);
            break;
```

Figure 3-18 On event code (EV_JOINED)

When confirmation was received that the device had successfully joined the network, the boolean for 'joined' is reset from 'false' to 'true', which indicated that the board had joined and could now disable the '**LMIC_setLinkCheckMode(0)**'. The resetting of the 'joined' boolean also initiated the first 'if' statement of the code, being the first of two checks to determine whether it was time for the board to enter sleep mode. Once a successful transmission was recorded '**case EV_TXCOMPLETE**', the boolean for sent was reset from 'false' to 'true'. The code used an initial 'if' statement to check for '**(LMIC.txrxFlags & TXRX_ACK)**' which effectively acknowledged receipt of the transmission. A secondary 'if' statement prints the length of the data using the '**LMIC.dataLen**' function. By resetting the 'sent' boolean to true, this also allowed for the second 'if' statement to be satisfied in the sleep mode portion of the code. This was done to ensure that the board only entered sleep mode once it was completely certain that a transmission was sent, and not during the procedure. This code is shown in Figure 3-19.

```
case EV_TXCOMPLETE:
            Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
             // transmit complete event
              sent = true;
            if (LMIC.txrxFlags & TXRX_ACK)
              Serial.println(F("Received ack"));
            if (LMIC.dataLen) {
              Serial.println(F("Received "));
              Serial.println(LMIC.dataLen);
              Serial.println(F(" bytes of payload"));
            }
```

Figure 3-19 On event code (EV_TXCOMPLETE)

3.5.2 Payload formatter

Once the data had been received by TTN, the payload was required to be decoded for the temperature and humidity values to be extracted. Temperature and humidity values were

packaged into four separate unsigned-8 bit integer byte arrays. The first two making up the temperature reading and the second two being the humidity reading. First the ‘bytes’ had to be extracted from the uplink. Next the bits in the first byte array ‘bytes[0]’ had to be shifted eight bits to the left. Bytes ‘byte[0]’ and ‘byte[1]’ were then concatenated to form a 16-bit integer representing the temperature value. The bit shifting ensures that the Most Significant Bit (MSB) was allocated to the higher bits of the resulting 16-bit integer. The same was then done for bytes ‘byte[2]’ and ‘byte[3]’ for the humidity reading. Once the values were in integer format, they were required to be converted back into the original float value format. To do this the integers were divided by 100. Finally, the decoded temperature and humidity values were returned and then displayed in a readable form on TTN. The payload formatter code is shown in Figure 3-20.

```
function decodeUplink(input) {
    var bytes = input.bytes;
    var temperature = (bytes[0] << 8) | bytes[1];
    var humidity = (bytes[2] << 8) | bytes[3];

    // Convert the integer values back to float, divided by 100 to get the original
values
    var decodedTemp = temperature / 100;
    var decodedHum = humidity / 100;

    return {
      data: {
        temperature: decodedTemp,
        humidity: decodedHum
      }
    };
  }
```

Figure 3-20 Payload formatter

3.5.3 Webhook, database and user interface

Once the data had been decoded and displayed on TTN, it had to be sent to a ‘destination system’ for the data to be stored and accessed (*What Are Webhooks And How Do They Work*, n.d.). A server-side framework, Next.js (Node.js), was used to handle incoming data and communicate with the database. The data was directed through this framework to Mongo DB, a NoSQL database used to store all the collected sensor data. The client-side framework used was Next.js (React), which was responsible for pulling data from Mongo DB to generate the information being represented on the live feed map. The following subsection details the sequence of events once the sensor data had been received and formatted by TTN.

3.5.3.1 Webhook activation

A webhook was used in TTN to notify a specific endpoint within the Next.js (Node.js). The function of a webhook is to send a message to a designated location automatically whenever a specified event occurs, this enables data to be transferred in real time (Adepoju, 2021).

3.5.3.2 Endpoint processing

The data is received by the Next.js (Node.js) endpoint, and the temperature and humidity readings are extracted from the incoming request. Once extracted, the data was saved in the MongoDB database. Each sensor's data included additional information, over and above the temperature and humidity data, such as Global Position System (GPS) coordinates and whether the sensor node was in a shaded or unshaded space.

3.5.3.3 Data storage

Within MongoDB, the data was allocated to two main collections, shown in Figure 3-21, devices and readings. The devices collection contained the co-ordinates for each sensor node, and if it was shaded or unshaded. The readings collection contained the extracted temperature and humidity readings along with their corresponding sensor node ID's.

```
_id: "shaded-1"
location: Object
  latitude: -33.95823849710656
  longitude: 18.45840829068154
  height: 0
shade: true


_id: ObjectId('630b75863ff5ae68f92141b1')
device_id: "working"
data: Object
  temperature:  ""
  humidity:  ""
created: 1661695366243
__v: 0
```

Figure 3-21: MongoDB object collections: (top) devices collection and (bottom) readings collection

3.5.3.4 Front-end visualisation and data transformation

A Next.js React front-end fetched the data from respective endpoints. These end-points read the data from MongoDB and passed it on to the front-end for displaying. The front-end consists of three main components. The first component was a table, displaying data in a tabular format, the second was a map to represent all sensors and their latest readings, as well as an icon indicating whether it is shaded or unshaded; and the third was to visualise data in various chart formats. Data 'buckets' were provided which allows the user to filter for different dates/periods, as well as, increasing or decreasing the temporal resolution of the data being displayed, i.e. from hours to days, weeks or months. Figures 3-22, 3-23, and 3-24 show screenshots of the website dashboard, map and charts respectively.

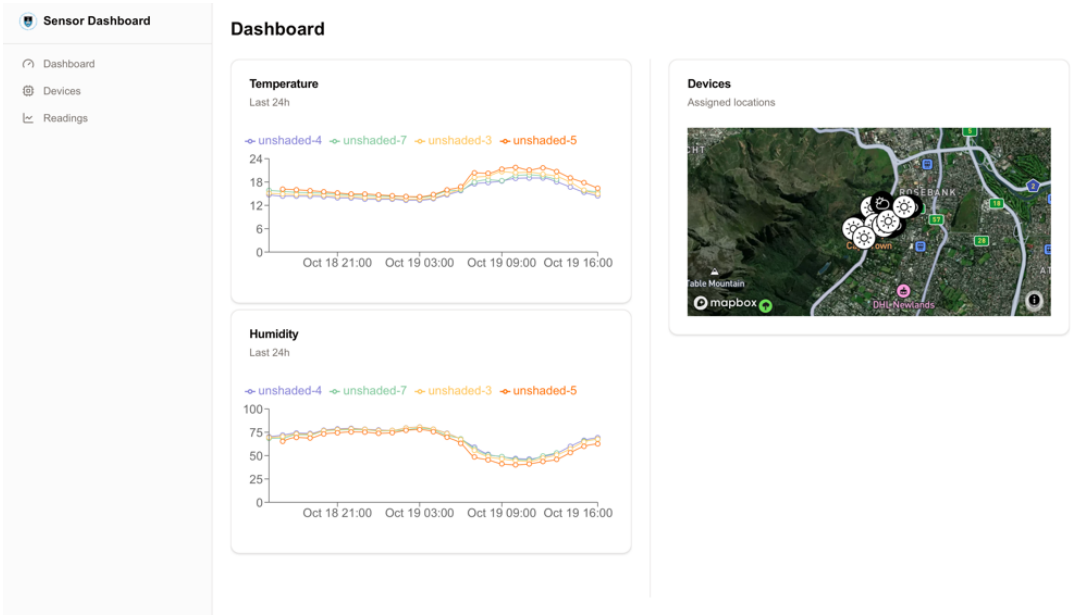


Figure 3-22: Website dashboard

Figure 3-23: Website sensor map locations

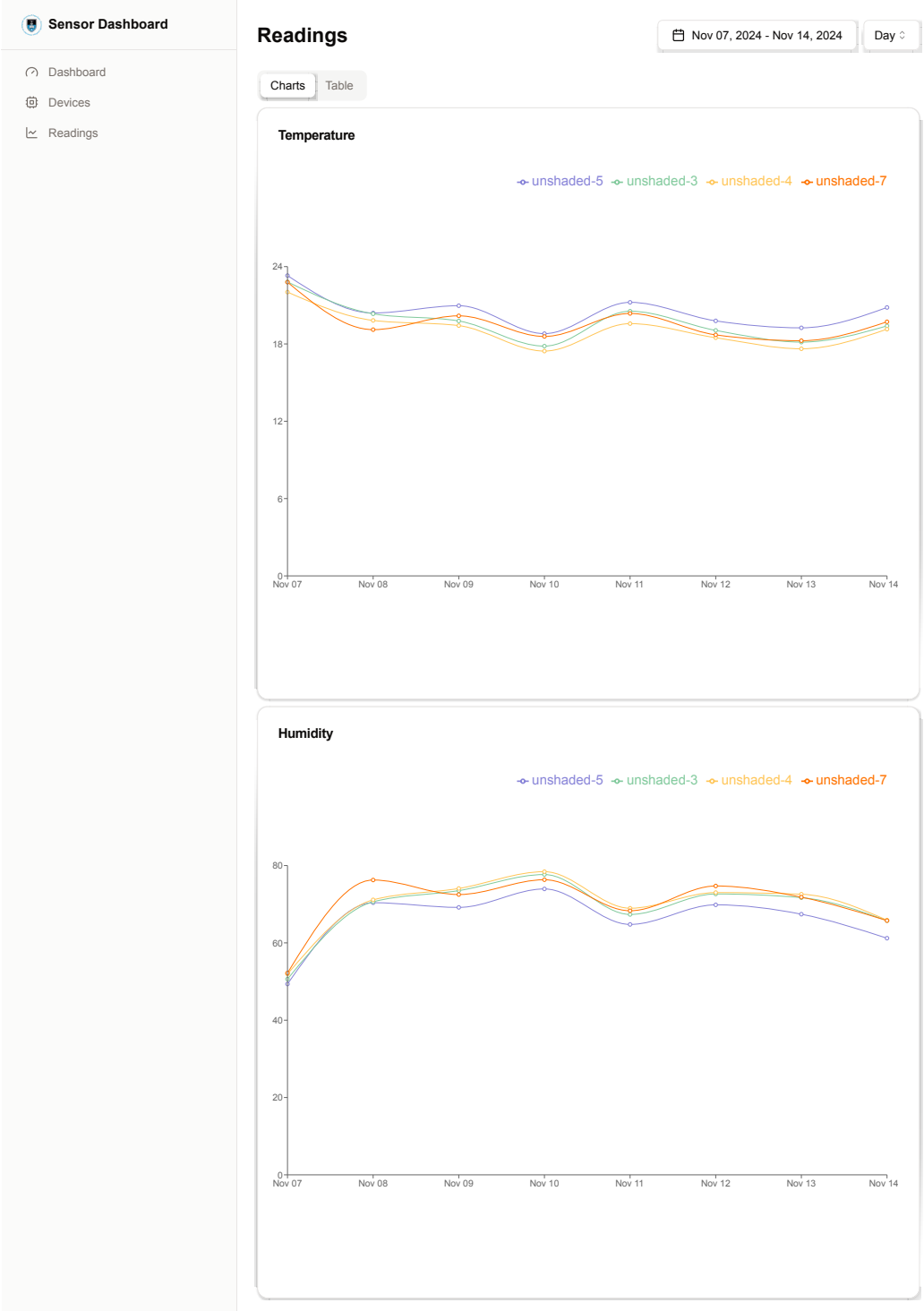


Figure 3-24: Website charts displaying temperature and humidity data with option to change temporal resolution in the top right-hand corner

On the client end, the fetched data was converted/transformed into a format suitable for rendering in the table, chart and map. This includes computing averages, converting units and formatting timestamps. The below flow chart in Figure 3-25, provides an overview of the above-mentioned process of transferring data from TTN to the database, and then to visualisation.

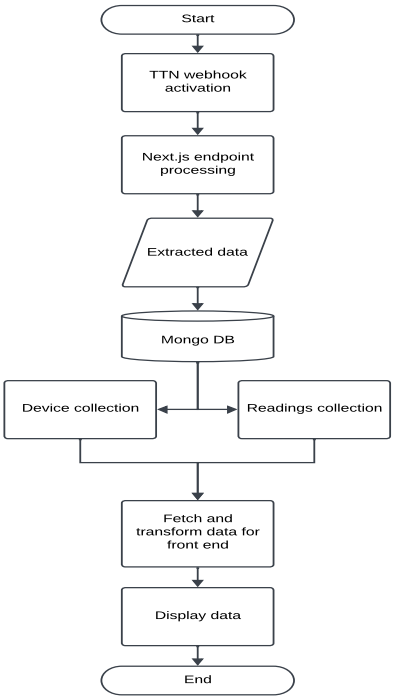


Figure 3-25: Front-end flow diagram

3.5.4 Final testing: preliminary rollout and current draw (testing phase 2)

After completing the assembly of the node, setting up network communication, and developing the user interface, the IoT sensor network was poised for deployment. To verify everything functioned properly, a final testing phase was initiated. Four sensor nodes were deployed, equipped with the shaded power supply, during a rainy period to assess waterproofing quality and monitor battery life and performance. Waterproofing proved adequate with no signs of water damage being found inside of the housing unit.

3.5.5 Data analysis and results

During and after the data collection period, data collected by the IoT sensor network was analysed. This section focused on using the collected data to generate numerical depictions of the relationship of temperature and humidity, between shaded an unshaded spaces on UCT campus.

Apart from the temperature and humidity data, the quality of the sensor network had to be determined by calculating Transmission Error Rates (TER) and the final performance of the power supply. The following subsections describe the breakdown of the data analysis and results section.

3.5.5.1 Temperature and humidity

The approach was to first analyse the temperature data independently. Temperature values and their corresponding recorded times were plotted. This was done from an overview of the total four month data collection period and step by step homed in toward a 24-hour period. By homing in, 'warming' and 'cooling' patterns and peak temperatures of each sensor node could be observed and compared to one another. Next, humidity values were plotted against temperature readings. This showed the relationship between temperature and humidity in the region. Additionally, these graphs showed humidity patterns in shaded and unshaded spaces on a micro-climate level. Statistical inference on the relationship between temperature, humidity, and shade/unshaded spaces was conducted using the programming software 'R'.

The relationship between temperature and humidity is inverse but the impact humidity has on the sensation or the 'feeling' of temperature is not. In 'warmer' temperature conditions, high humidity values result in a warmer experience of the true temperature where lower humidity can have the opposite effect, making it seem 'cooler'. To further examine the temperature results , calculations were done using the Heat Index (HI) (Anderson et al., 2013) for humidity and temperatures in shaded regions.

The Heat index values in were calculated using Equation 3.7:

$$HI = -42.379 + 2.04901523 \times \mathrm{T} + 10.14333127 \times \mathrm{H} - 0.22475541 \times \mathrm{T} \times \mathrm{H} - 6.83783 \times 10\text{^}(-3) \times T\text{^}2 - 5.481717 \times 10\text{^}(-2) \times 10\text{^}2 + 1.22874 \times 10\text{^}(-3) \times T\text{^}2 \times H + 8.5282 \times 10\text{^}(-4) \times T \times H\text{^}2 - 1.99 \times 10\text{^}(-6) \times T\text{^}2 \times H\text{^}2 \quad (3.7)$$

where: HI = Heat Index; T = temperature (°F); H = humidity

Temperature readings were converter from degrees Celsius to degrees Fahrenheit before using Equation 3.7. HI value was then converted back to degrees Celcius.

3.5.5.2 IoT sensor network performance

In evaluating the study of the IoT sensor network, particular attention was given to assessing its performance across key areas: data transmission efficiency, reliability of power supply, system robustness under varying conditions, and the effectiveness of security measures implemented. Primarily, the network was designed to collect and transmit data from end-devices to be stored in an online cloud and to be made easily accessible to users. The data transmission efficiency was determined by calculating the amount of data lost due to failed transmission and false readings from the sensors. Following on from determining the TER, the power supply components were analysed and discussed. A final current draw calculation was computed based on the total duration of each battery cycle per shaded sensor node. The unshaded solar panel power supply performance was evaluated and compared to the shaded power supply performance, to determine the better and more feasible design. The robustness and security of the system was considered. Robustness was determined by looking at whether the sensor nodes were able to withstand the

varying weather conditions of the region. In cases where sensor nodes were damaged, troubleshooting was conducted to identify the potential weak point of the build. Signs of tampering and vandalism were searched for to comment on whether the sensor nodes were hindered by security concerns.

3.5.5.3 Cost analysis

At the project's inception, a primary objective was to ascertain the viability of employing IoT technology for hydrological data collection in South Africa. Central to this assessment was the determination of whether the sensor network could be constructed using readily available products and components found within the South African market, and whether these components were economically feasible. Section 4.2.4 provides a detailed breakdown of the expenses associated with building and deploying the network. Additionally, a comparative analysis was conducted between the developed sensor node, a sensor node utilized in a previous IoT study conducted by De Oliveira (2023) , and a commercially available sensor on the market.

4. Results and discussion

In total, 14 sensors were built and deployed to collect and output temperature and humidity data from shaded and unshaded spaces on UCT campus, from the 7 September 2023 – 31 December 2023. The following section displays and discusses the temperature and humidity results produced by the WSN.

4.1 Temperature

The SHT 31, an all-weather sensor was used to collect the temperature data. To protect the temperature data from being skewed, a radiation shield was developed to encase the sensor. As the data was required to reflect ambient temperature, sensors were placed as close to the ground as possible whilst maintaining an acceptable degree of security and safety of the actual device. Two groups of nodes were deployed i.e. in shaded and unshaded spaces. An overview of the temperature (°C) data can be seen in Figure 4-1.

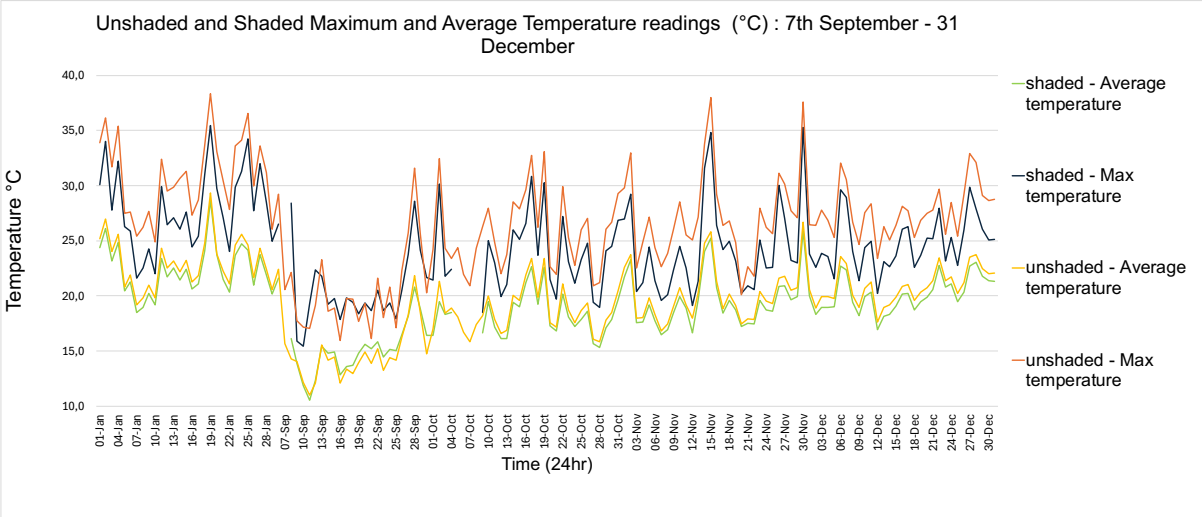


Figure 4-1: Temperature data overview for data collection period

The graph shows the maximum temperature values for the combined shaded versus unshaded spaces, for each day over the span of four months. Additionally, it shows the average temperature reading for both shaded and unshaded spaces. Immediately apparent is how the graph tends upwards toward the summer months, and how significant the gap between the max temperature peaks when comparing shaded and unshaded values. This gap is more noticeable on days where unshaded temperature values surpassed values of 30 °C. This pattern remains consistent throughout the data. The break in data from 4 – 9 October 2023 was a maintenance period for the unshaded sensors.

4.1.1 Temperature – monthly overview

Temperature values during cooler periods of the day (before sunrise and in the evening toward and beyond sunset) for shaded and unshaded regions converge to one another. In some cases, the temperature in the shaded green spaces surpass those of the unshaded.

Figure 4-2 shows the average reading for each hour for all shaded and unshaded sensors for months September through December 2023. The graph provides a clear depiction of the difference in temperature between shaded and unshaded spaces, and pattern of temperature throughout the day, between sunrise and sunset (07:00 to 19:00). The largest gap between temperature readings of the two categories of spaces can always been seen at the time of the peak temperature readings (ranging from the time 12:00 to 15:00).

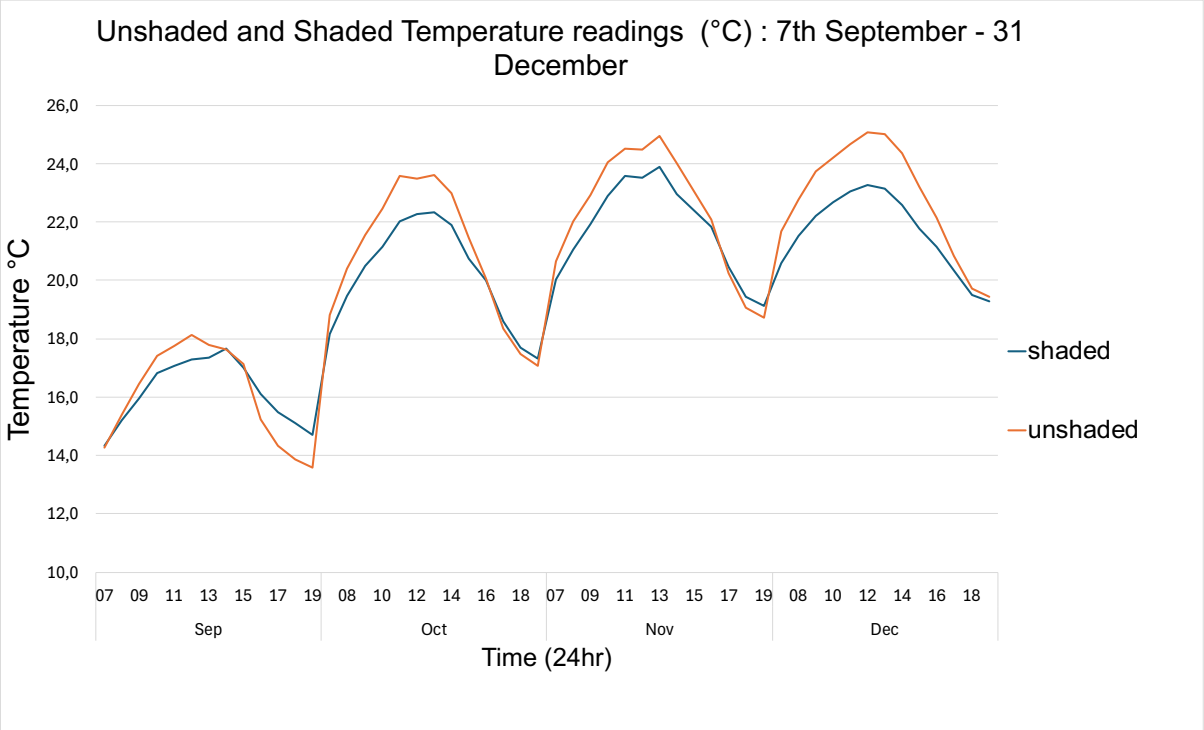


Figure 4-2: Average temperature reading per hour per month for all shaded and unshaded sensors for months September through December 2023

Tables 4-1 to 4-4 show a clear indication of the cooling effect the urban shaded green spaces had compared to the urban unshaded paved spaces. During time intervals between sunrise and sunset, shaded areas often reached its peak temperature value after the unshaded spaces, as well as, always decreasing in temperature in a more gradual manner. The rate at which temperature increased and decreased is discussed in more detail in section 4.1.2 and displayed in table 4-6.

Table 4-1 shows data for the month of September using a time interval from a general sunrise at 07:00 and sunset at 19:00. The peak temperature for the shaded spaces was on average 17.7 °C at 14:00 compared to 18.1 °C at 12:00 for the unshaded spaces.

Table 4-1: Shaded vs Unshaded temperatures per hour (September 2023)

	Shaded		Unshaded	
Hours	Average Temperature	StdDev of Temperature Data	Average Temperature	StdDev of Temperature Data
7	14,3	2,8	14,3	2,8
8	15,2	2,6	15,4	3
9	16	2,6	16,4	3
10	16,8	2,7	17,4	3,4
11	17,1	2,9	17,8	3,7
12	17,3	3,2	18,1	4
13	17,4	3,1	17,8	3,9
14	17,7	3,6	17,6	4
15	17	3,2	17,1	3,7
16	16,1	3,1	15,2	2,6
17	15,5	3,1	14,3	2,4
18	15,1	3,1	13,9	2,2
19	14,7	3,1	13,6	2,2

Table 4-2 shows data for the month of October using a time interval from sunrise at 06:00 and sunset at 19:00. The peak temperature for the shaded spaces was on average 22.3 °C at 12:00 compared to 23.6 °C at 11:00 for the unshaded spaces.

Table 4-2: Shaded vs Unshaded temperatures per hour (October 2023)

	Shaded		Unshaded	
Hours	Average Temperature	StdDev of Temperature Data	Average Temperature	StdDev of Temperature Data
6	17,1	2,3	17,4	2,2
7	18,2	2	18,8	2,3
8	19,5	2,7	20,4	2,9
9	20,5	2,8	21,5	3
10	21,1	3,1	22,4	3,4
11	22	3,5	23,6	3,7
12	22,3	3,4	23,5	3,8
13	22,3	3,4	23,6	3,9
14	21,9	3,3	23	3,7
15	20,8	3	21,5	3,4
16	20	2,8	20,1	3
17	18,6	2,2	18,3	2,1
18	17,7	2	17,5	1,8
19	17,3	1,9	17,1	1,8

Table 4-3 shows data for the month of November using a time interval from sunrise at 06:00 and sunset at 20:00. The peak temperature for shaded spaces was on average 23.9 °C compared to 24.9 °C for the unshaded spaces, both average peak values were aggregated to have been recorded at 13:00.

Table 4-3: Shaded vs Unshaded temperatures per hour (November 2023)

	Shaded		Unshaded	
Hours	**Average Temperature**	**StdDev of Temperature Data**	**Average Temperature**	**StdDev of Temperature Data**
6	18,8	2,4	19,2	2,5
7	20	2,9	20,6	2,9
8	21,1	3,3	22	3,3
9	21,9	3,5	22,9	3,6
10	22,9	3,8	24,1	4,1
11	23,6	4,3	24,5	4,2
12	23,5	4,6	24,5	4,4
13	23,9	4,6	24,9	4,6
14	23	3,8	24	3,9
15	22,4	3,1	23	3,3
16	21,8	3	22,1	3,2
17	20,5	2,6	20,2	2,6
18	19,4	2,1	19,1	2,1
19	19,1	2,4	18,7	2,4
20	18,7	2,5	18,3	2,4

Table 4-4 shows data for the month of December using a time interval from sunrise at 05:00 and sunset at 20:00. The peak temperature for the shaded spaces was on average 23.9 °C compared to 24.9 °C for the unshaded spaces, both average peak values were aggregated to have been recorded at 13:00.

Table 4-4: Shaded vs Unshaded temperatures per hour (December 2023)

	Shaded		Unshaded	
Hours	**Average Temperature**	**StdDev of Temperature Data**	**Average Temperature**	**StdDev of Temperature Data**
5	18,2	1,6	18,7	1,6
6	19,3	1,6	20,2	1,7
7	20,6	1,8	21,7	2
8	21,5	1,8	22,8	2
9	22,2	1,9	23,8	2,1
10	22,7	2,1	24,2	2,3
11	23,1	2	24,7	2,2
12	23,3	2,3	25,1	2,4

13	23,1	2,2	25	2,5
14	22,6	2,4	24,4	2,8
15	21,8	2,2	23,2	2,6
16	21,2	2,2	22,1	2,5
17	20,3	2	20,8	2,1
18	19,5	1,8	19,7	1,8
19	19,3	1,9	19,4	1,8
20	19	1,8	19,2	1,8

Notably, the standard deviation of the temperature values was highest during the warmest periods of the day. This could be attributed to the variation in data being collected at different locations. Table Mountain played a significant role in these variations as sunlight would not have been distributed evenly across the examinable site, e.g. sensors closer to the mountain would spend fewer hours in direct sunlight. Excluding the month of November, standard deviation values were consistently lower during the 'red' and 'orange' periods of the day (colour coding ranging from 'green' being 'cool' to 'red' being 'warm') in shaded regions compared to those in unshaded regions. These findings corroborate with Chen et al. (2022), that shaded spaces provide more moderate temperatures compared to those of paved and unshaded spaces. The month of September represented the period where the most cooling was observed, this is expanded in the greater detail in Section 4.1.2.

4.1.2 Temperature – 24-hour analysis

This section goes one step further and isolated the 'warmest' day observed in each month. Here the sensor node that recorded the highest temperatures is compared to the sensor nodes that were nearest, as well as sensor nodes that recorded the lowest values. The comparison allows for a more concentrated depiction of how the temperature patterns differ from one another, in unshaded and shaded spaces. In the month of September, the 'warmest' temperature recorded was on the 28th day of that month, the peak temperature reading was 31 °C, recorded at node 'unshaded-7'. Comparatively the nearest node 'shaded-7' recorded a peak value of 28 °C, 3 °C less than an unshaded node approximately 100m away with an elevation change of less than 5m between the two nodes. This is important to note as it indicates that the dissonance between the two readings of the nodes is not related to the distance between them or their differing heights above sea level. The temperature readings for the 28th of September 2023 are shown in Figure 4-3.

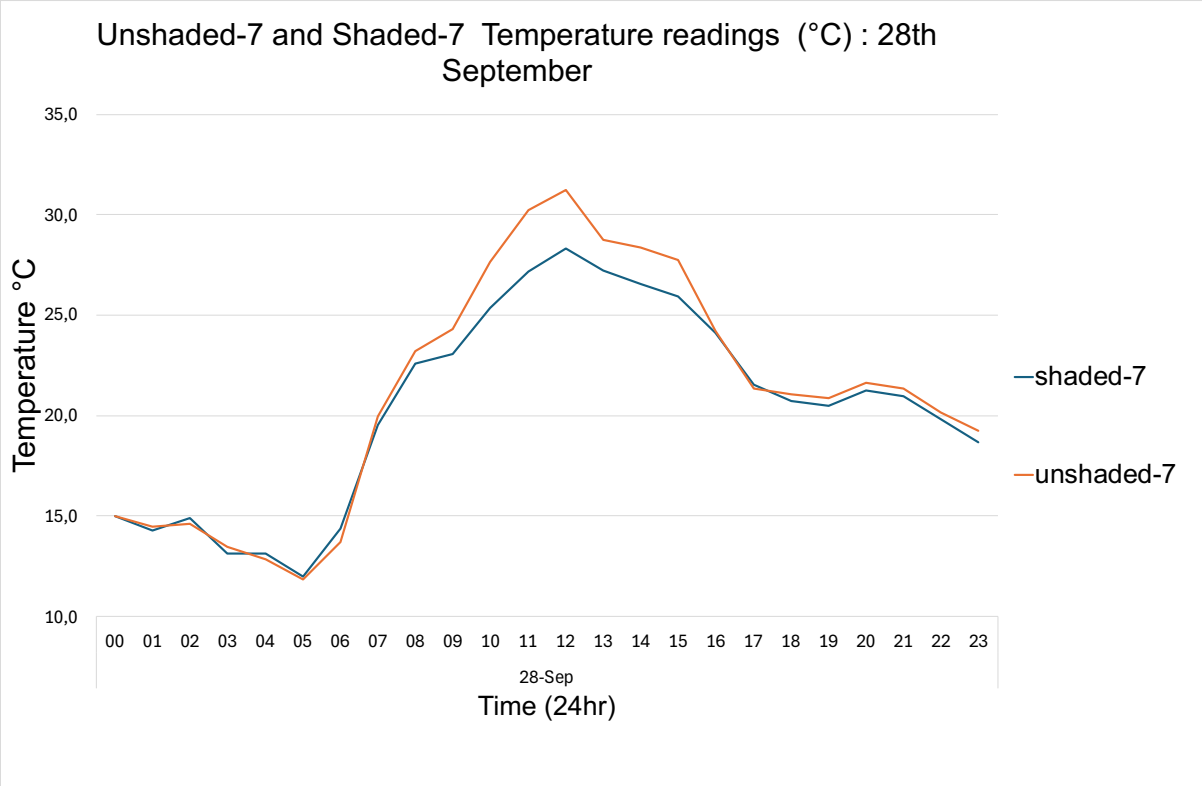


Figure 4-3: Unshaded – 7 and Shaded – 7 temperature readings (°C) (28th September)

When comparing sensor node 'unshaded-7' to 'shaded- 3', the readings from both sensors differ significantly. The shaded sensor in this case is in a more shaded and greener region, with little to no paving and very little exposure to sunlight. See the graph in Figure 4-4 below for a comparison.

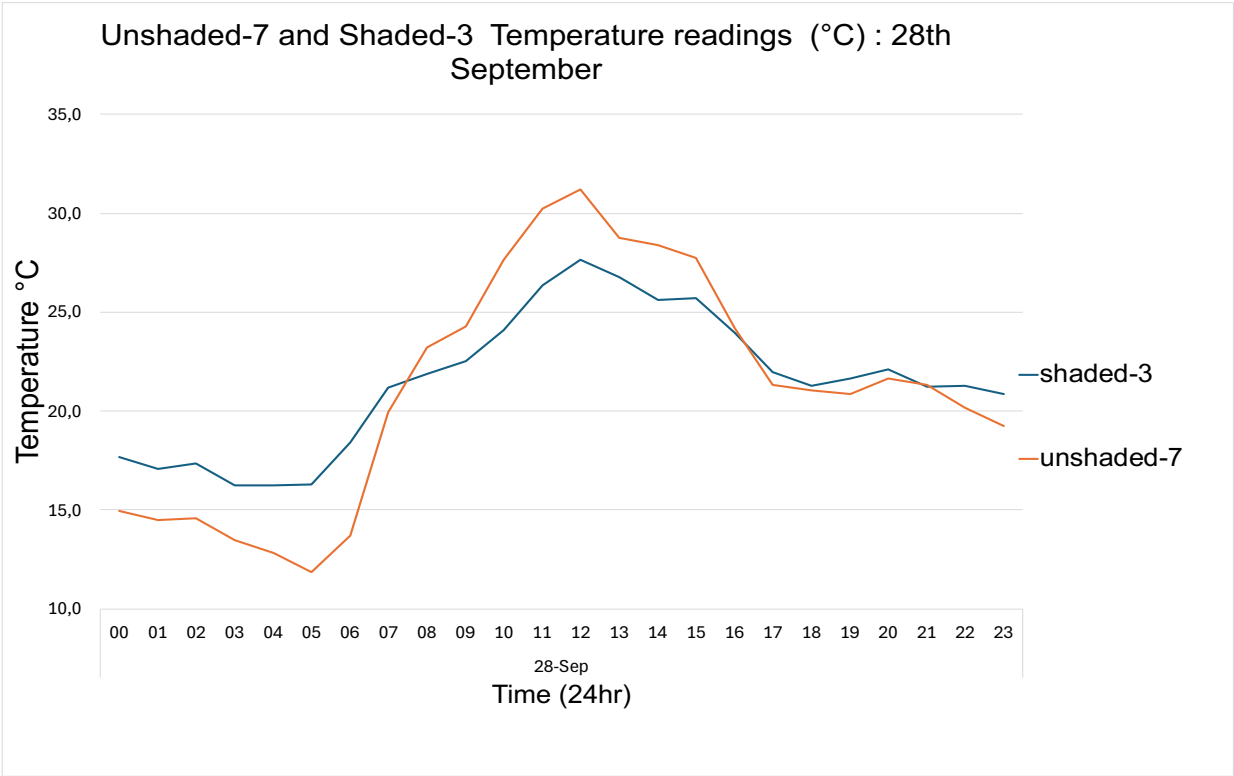


Figure 4-4: Unshaded – 7 and Shaded – 3 temperature readings (°C) (28th September)

The months October through to December present a similar pattern with a slight increase in peak temperature values in the afternoon hours of the day, as the season progressed into summer. Graphs for months October, November and December can be found in Appendix E. An overview of results for readings on the 'warmest' day of each month is shown in Table 4-5.

Table 4-5: Overview of the 'warmest' days throughout the data collection period

Date	Shaded	Sensor no.	Temp (°C)	Unshaded	Sensor no.	Temp (°C)	Difference (°C)
28th Sep	Overall		26	Overall		30	**4**
	Min (12:00):	3	28	Prox:	3	31	**3**
	Prox:	7	29	Max (12:00):	7	32	**3**
	Min:	3	28	Max:	7	32	**4**
19th Oct	Overall		29	Overall		31	**2**
	Min (12:00):	1	29	Prox:	2	32	**3**
	Prox:	5	29	Max (12:00):	5	33	**4**
	Min:	1	29	Max:	5	33	**4**
15th Nov	Overall		34	Overall		36	**2**
	Min (12:00):	3	34	Max (12:00):	3	37	**3**
27th Dec	Overall		29	Overall		31	**2**
	Min (13:00):	1	29	Prox:	2	30	**1**
	Prox:	5	29	Max (13:00):	5	33	**4**
	Min:	1	29	Max:	5	33	**4**

Table 4-5 summarises the results from the warmest days over the data collection period. 'Overall' values display the average peak temperature of all sensors, shaded and unshaded, for that day, at the time of the peak temperatures being recorded. The 'Min' values represent the coolest/lowest peak temperature reading recorded, in a shaded region. The peak recorded value in an unshaded region is represented by the 'Max' values. The last row in each section compares the minimum shaded temperature value to the maximum recorded unshaded temperature value. The final column 'Difference °C' shows the computed differences between unshaded and shaded temperature readings. Table 4-5 shows that shaded spaces are consistently cooler than unshaded spaces at times of peak temperature readings. The rate at which temperature increased and decreased in these shaded and unshaded spaces was observed and the findings are displayed in Table 4-6.

Table 4-6 Rate of heating and cooling on the 'warmest' days throughout the data collection period

Date	Rate of heating (°C/h)		Rate of cooling (°C/h)	
	Shaded	**Unshaded**	**Shaded**	**Unshaded**
28 Sept (shaded-3 vs unshaded-7)	0.6	2	0.4	2.2
19 Oct (Shaded- 1 vs Unshaded- 5)	1	1.7	1.1	1.6
15 Nov (Shaded- 3 vs Unshaded- 3)	1.3	1.6	2	2.6
27 Dec (Shaded-1 vs Unshaded-5)	1.3	1.6	0.8	1.3

Table 4-6 shows the rate at which temperatures climbed from sunrise to peak recorded temperatures, and the subsequently decreased from peak to sunset. The data shows that that the shaded areas warm up at slower rate to unshaded areas. A key finding is the difference in heat

fluctuation between the sensors. The Unshaded-7 area increased from 14.7 °C to 31.6 °C (minimum to maximum) over the course of 8 hours. Showing an increase of 2 °C/hour whereas in the shaded-3 region, a far lesser rate is observed; 19.26 °C to 23.58 °C, over a time interval of 12 hours, suggesting an increase of 0.6 °C/hour. Although, the difference in rate of heating and cooling is observed to be most drastic on the 28th September between shaded-3 and unshaded-7, this pattern persists for all shaded and unshaded spaces as Table 4-6 displays.

The same pattern seen in the buildup of temperature from sunrise to peak midday temperature, is reflected as the sun begins to set. It is noted that shaded spaces are consistently cooling at a rate slower to unshaded spaces, a pattern also reflected in Table 4-7, which shows the average rate of heating and cooling per month for shaded and unshaded regions.

Table 4-7 Rate of heating and cooling, monthly overview

Month and time intervals	Rate of heating (°C/h)		Rate of cooling (°C/h)	
	Shaded	Unshaded	Shaded	Unshaded
September (7-13-19)	0,5	0,6	0,4	0,7
October (6-13-19)	0,7	0,9	1,3	1,6
November (6-13-20)	0,7	0,8	0,7	0,9
December (5-12-20)	0,7	0,9	0,5	0,7

4.1.3 Humidity

All sensor nodes were built to transmit a humidity reading in conjunction with every temperature measurement. The humidity data typically decreased as the temperature values increased throughout the day and then proceeded to increase as those temperature values began to decrease, thus showing the inverse relationship between the two measurements. Figure 4-5 displays this finding, as it shows the temperature versus humidity over the entirety of the data collection period.

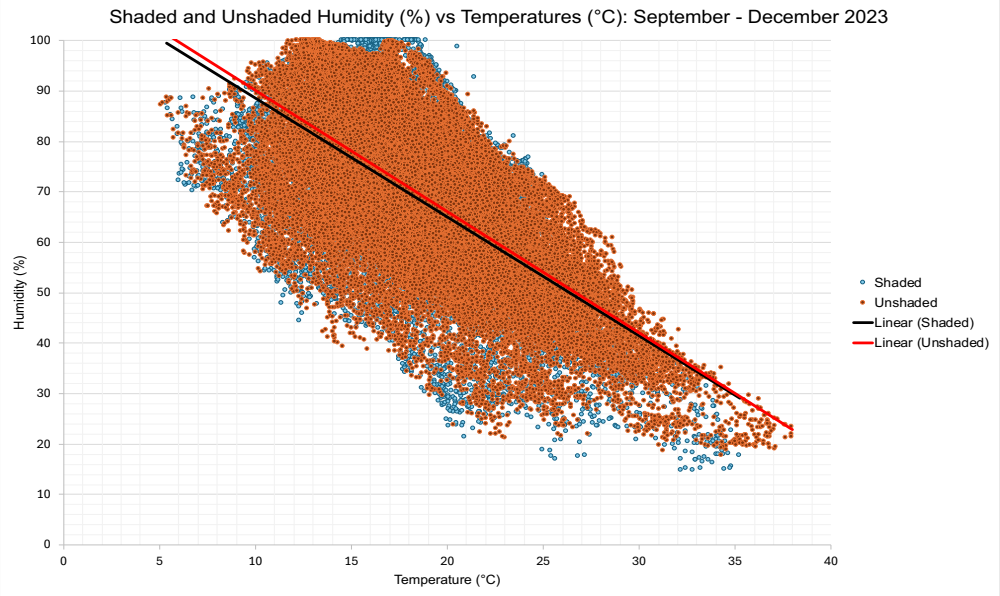


Figure 4-5: Humidity vs temperature throughout the data collection period

To better understand the collected data, a multiple linear regression analysis was conducted in the statistical programming software, R, to model the relationship between temperature and humidity. The model describes how temperature and shading impact humidity, and further explored whether the interaction between temperature and shaded/unshaded spaces altered this relationship. The R code is shown in Figure 4-6.

```
head(Corrected_and_cleaned_final_excel_sheet)
names(Corrected_and_cleaned_final_excel_sheet)

humidity_temp <- select(Corrected_and_cleaned_final_excel_sheet ,"Corrected
date",'SHADE/SHADED',
'corrected time',
data.temperature,
data.humidity )

start <- as.Date("yyyy-mm-dd")
end <- as.Date("yyyy-mm-dd")

start_time <- "hh:mm:ss"
end_time <- "hh:mm:ss"

filtered <- humidity_temp %>%
  filter(`Corrected date` >= start & `Corrected date` <= end) %>%
  filter(format(`corrected time`, "%H:%M:%S")>= start_time & format(`corrected time`,
"%H:%M:%S") <= end_time)

model <- lm(data.humidity ~ data.temperature * `SHADE/SHADED`, data = filtered)
summary(model)

ggplot(filtered, aes(x = data.temperature, y = data.humidity , color =
`SHADE/SHADED`)) +
  geom_point() +
  geom_smooth(method = "lm", aes(fill = `SHADE/SHADED`), alpha = 0.3) +
  labs(title = "Impact of Humidity on Temperature by Shading Condition",
       y = "Humidity (%)", x = "Temperature (°C)")
```

Figure 4-6: R programming multiple regression analysis code

The code in Figure 4-6 selects columns date, shading condition, time, temperature, and humidity from the inputted excel file holding the collected temperature and humidity data. It then filters all the data that falls within the specified date and time range. A multiple linear regression model is then created to analyse how the interaction between shade conditions and temperature impact humidity. Lastly, code was written to provide the option to visualize the model as a scatter plot with regression lines for both shaded and unshaded spaces.

For the months October, November, and December, consistent patterns emerged in the following areas: average humidity readings, the interaction between temperature and shade, and the overall model fit (R^2) value. In all three of these areas, September differed from the rest.

The average humidity readings from October to December show that unshaded spaces had significantly lower humidity readings than shaded spaces; while September showed that humidity in unshaded spaces were moderately higher. See table 4-8 for a summary of humidity values as a result of shade and unshaded spaces.

Table 4-8: Summary of average humidity shaded vs unshaded readings

Month (unshaded)	Estimate (difference in avg. humidity)	p value
September	0.77	0.111
October	- 5.33	< 2e-16 *
November	- 9.31	< 2e-16 *
December	- 3.1	8.21e-06 *

The summary in Table 4-8 uses shaded spaces as a reference, therefore the values in the estimate column is the approximate value to be added to the humidity in a shaded space to arrive at the humidity value expected in an unshaded space. These estimates assume temperature is constant and the shown values are solely a measure of the impact shade, or the lack thereof, has on humidity, as well as excluding periods of nighttime. The third column labelled *'p value'* is the probability that the estimated value is due to 'random' variability rather than the effect of shading. A *p value* below 0.05, was deemed as low, suggesting that estimated value was significantly impacted by shading. Significant *p values* are marked with an '*'. Although the September estimate suggests that humidity in unshaded spaces is on average higher than shaded spaces, the *p value* is greater than 0.05 suggesting that the difference in humidity in shaded or unshaded spaces is not significant. When looking at the months October to December, all the estimates are negative with very low *p values* suggesting that the humidity in unshaded spaces are significantly lower than in shaded spaces.

The next factor being highlighted is the interaction between temperature and shade, and whether this alters the relationship between temperature and humidity. In this case, the *p value* is an indication as to whether the interaction between the presence of shade has a significant or negligible impact on the resulting humidity value. See Table 4-9 for a summary.

Table 4-9: Summary of interactions between and shaded/unshaded spaces

Month (unshaded)	Estimate (interaction)	p value
September	0.019	0.391
October	0.27	< 2e-16 *
November	0.44	< 2e-16 *
December	0.18	4.41e-09 *

The approximated interaction values show that for every unit °C increase, the humidity in shaded areas can be estimated to be slightly higher in unshaded spaces, suggesting that the humidity in shaded spaces decrease at a sharper rate.

The third finding being highlighted is the R^2 values generated by the model. R^2 is a measure of how well the linear regression model fits the data, and effectively describes how much of the variance in the dependent, in this case the humidity, variable can be explained by the model. As there is more than one independent variable in this model, the adjusted R^2 value is used, which adjusts the R^2 value to account for the additional independent variable. These values can also be interpreted as percentages e.g. 0.45 = 45%. Table 4-10 provides a summary of the R^2 values outputted by the model for months September to December.

Table 4-10: Summary of adjusted R^2 values

Month (unshaded)	Adjusted R^2
September	0.45
October	0.51
November	0.56
December	0.55

In months October, November and December the R^2 value were consistently greater than 50%, where September is recorded to be 45%.

All three of these highlighted factors as per Tables 4-8 to 4-10, suggest that the model during the month of September is moderately less robust than months October to December, and that the humidity did not vary as much between shaded and unshaded spaces. An explanation for this may be the greater number of days with overcast conditions, as well as, rainfall, reducing the variance in conditions between shaded and unshaded spaces.

Graphs for months September to December, visually illustrating the multiple regressing lines can be seen as Appendix F.

Table 4-11 summarises the Heat Index calculation (Equation 3.7) results.

Table 4-11 Heat index calculation for the 'warmest' day of each month during the data collection period

Date	Shaded			Unshaded	ΔT (°C)	ΔHI T (°C)
	Humidity %	Temperature (°C)	HI Temperature (°C)	Temperature (°C)		
28 Sept	28	28	-	31	3	-
19 Oct	34	29	28	33	4	**5**

15 Nov	25	34	33	37	3	**4**
27 Dec	32	29	28	32	3	**4**

Calculations show that when temperatures surpass 27 °C, the lower humidity in shaded spaces further increase the cooling effect. To ensure comparisons were consistent, the same sensor node combinations displayed in Tables 4-5 and 4-6 were used to compute the ΔT (°C) and ΔHI T (°C) values shown in the above Table 4-11. Stand out cases were on the 'warmest' days, such as the 15th November, where the maximum unshaded temperature reading reached 37 °C and shaded spaces reached 34 °C. Humidity readings for many of these shaded areas at 34 °C were as low as 15 %, resulting in a 'feels like' temperature of 33 °C, 4 °C cooler than the unshaded spaces. The Table 4-11 above shows that shaded areas consistently 'feel' cooler by 1 °C when applying the heat index calculation. Shaded areas across the entire site show the same pattern, as represented in Table 4-12, with the average Heat Index Temperature ranging between 0.5 and 0.92 ºC less than the actual recorded value for temperatures surpassing 27 °C.

Table 4-12 Heat index summary for each month

Month	September	October	November	December
ΔHIT (°C)	0.92	0.5	0.95	0.6

4.2 Performance of the network

The IoT network was built for the use of hydrological data collection, and in this specific case to monitor UHIs and cooling spaces on UCT campus. An IoT sensor network tasked with collecting hydrological data must appease certain requirements to be deemed as a feasible solution. This section discusses how the sensor network performed regarding transmission of data, power supply, robustness and security.

4.2.1 Data transmission

Sensor nodes were programmed to transmit a message with a temperature and humidity reading accompanied by a timestamp. The data was transmitted from the sensor nodes to a local gateway, then to TTN; a webhook then pulled the data to a monitored database. Table 4-13 shows the amount uplinks with data recorded by TTN as well as data packets received by the database. These data packets or messages were 'webhooked' to Mongo DB, the server storing all the data. The table shows the number of received readings recorded by the database and how many of them were "False transmissions", messages with no temperature and/or humidity readings. The "Error in reading" column counts the total of incorrect messages transmitting "0 %" and "610.0 °C" for humidity and temperatures respectively.

Table 4-13 Overview of data transmissions

Date	Total transmissions	False transmissions	Error in reading	TER %
28/08/2022 – 20/02/2024	146437	22533	-	-
07/09/2023 – 31/12/2023	84148	85	3	0.10

Overall, the system transmitted 146 437 data messages with 22 533 "False transmissions". It is important to note that 22 359 of these transmissions were during the testing phase of the network. For the period of onsite data collection, 07/09/2023 – 01/01/2024, the total number of transmissions were 84 148 messages with 85 "False transmissions" and 3 "Error readings", resulting in a transmission error rate (TER) or data lost percentage of 0.10 %. In the case of this application where the recording intervals were relatively small, 15 minutes apart, a 0.10 % loss of data does not significantly impact the temporal resolution of the data. Appendix G is a full monthly break down of the transmission data.

4.2.2 Power Supply

For an IoT network to be deemed sustainable, it is a requirement that the sensor nodes have a reliable and long-term power supply. Power considerations are aimed at either designing the network to draw as low a current as possible or to develop a way to supply enough current and battery life. In this section the two design approaches regarding power supply, for shaded and unshaded sensors, are discussed and evaluated.

4.2.2.1 Shaded sensors

Shaded sensors were fitted with 2 x 5000 mAh LiPo batteries configured in a parallel circuit, effectively doubling the battery life cycle. The Maduino boards were programmed to enter a sleep mode after receiving confirmation of a successful transmission. The sleep mode was coded to last for 15 minutes. Shaded sensor nodes on average required a battery replacement once every 23 days (562 h). The calculation below, utilizing Equation 4.1, provides the true current draw of the deployed sensor node.

$$\text{True ACD} = \ (Battery\ capacity\ (mAh))/(Duration\ (h)\ ) \tag{4.1}$$

where: Battery capacity = 10 000 mAh; Duration = 562 h; True ACD = 17.8 mA

The calculation above shows that the sensor node draws an average of 17.8 mA current while operating. This was significantly higher than the projected 8.6 mA theoretical average current draw calculated in Section 3.3.3.2. Sensor nodes on average only managed to achieve 562 hours using two 5000 mAh batteries in parallel. The calculated duration was 578.3 hours using a single 5000 mAh LiPo battery meaning that the nodes only managed to achieve 49% of its battery life duration target. A key reason for this major discrepancy may be that the calculation did not properly account for 'spikes' in energy usage at the end of each eight second low power mode

cycle, which would have accumulated over time. In total, with this configuration, each sensor required three battery changes and services over the course of 4 months. This resulted in provisions to be made for the use of 42 x 5000 mAh LiPo batteries to power the shaded sensor nodes, 14 more than the initial estimate.

4.2.2.2 Unshaded sensors

Unshaded sensor nodes presented an opportunity to utilise solar energy as an alternate power solution. These nodes were fitted with a single 5 000 mAh LiPo battery, 6V 4.5 W solar panel, and a TP4056 charging module. These sensors had the same 'sleep mode' lines of code that was implemented in the shaded sensor nodes. The current draw of the node was calculated to be 8.6 mA and later proven to be 17.8 mA, while the solar panel configuration was able to provide up to 750 mA in ideal conditions, comfortably able to account for the additional energy losses and current draw of the node. Batteries would power the nodes during the nighttime, the solar panels successfully recharged the batteries and powered the Maduino and SHT 31 sensor during the day. Over the course of the 4-month period, no battery changes or power supply services were required for any of the unshaded sensors. Unshaded sensor nodes lasted for a total of 116 days without requiring a single battery change, 85 days longer than shaded sensor node setup.

4.2.3 Robustness and Security

For the sensor nodes to operate successfully, the nodes were designed to be able to endure the weather conditions of Cape Town. Safety and security of the devices was also an important consideration. Weather conditions such as heat and wind were concerns but the priority was to ensure that the sensor nodes were adequately waterproofed/water-resistant. Throughout the data collection period, there was rain on 27 days. None of the sensors failed due to water or weather-related damage. To protect nodes from tampering, sensors were secured to structures (trees or lampposts) at 2.4 m above ground. The higher the node was secured the safer it would be, but it was important to keep near enough to ground level in order to collect ambient temperatures and humidity readings. A further step taken was to house sensors in a fully concealed box rather than a transparent open face one. Throughout the data collection period, no significant hindrances or damages were detected from tampering. It is important to note that this is also largely due to the sensors being deployed on a secured university campus such as UCT. A drawback in robustness came from a design decision to not further secure wire pin heads. The 'plug-and-play' design approach was utilised to increase accessibility and ease of servicing and troubleshooting of sensor nodes. However, the plug-and-play method had an adverse effect, in some cases, as sensor nodes would stop transmitting, and upon investigation, a wire was found to be disconnected on two occasions. This issue would be averted with screw-ins or clipped in pin heads.

4.2.4 Cost analysis

Part of the overall investigation was to determine the cost of developing an IoT sensor node in South Africa and gauge its feasibility. To best achieve this, it was decided that all components used within the build and setup of the node were to be readily available and purchased off of 'South African shelves. This means that for an item or component to qualify for use, it would have to be procurable through a South African supplier. Table 4-14 shows a full break down of the cost to build a single deployable sensor node.

Table 4-14 divides the cost of the sensor node build into two categories: the node housing structure and the sensor node setup. The node housing structure encompasses all the materials and components used to mount, protect and weatherproof the node. The sum total cost of the node housing components is ‘Sum: 1” which came to R478,28. All items and components involved in sensing and transmitting data, harvesting and providing energy, and managing data are included in the ‘sensor node setup’ costs. The sum total cost of the sensor node setup is ‘Sum: 2” which came to R1 044,86. The combination of sum 1 and sum 2 provide the total cost of a full deployable sensor node, which had amounted to R1 523,14.

Table 4-14 Sensor node cost breakdown

Item	Quantity	Total (R)	
1	Plates	5	R74,95
2	Nuts and bolts	1	R50,00
3	Steele rod	1	R245,00
4	Aluminium bracket	1	R33,33
5	Solar panel bracket	1	R25,00
6	Electrical box	1	R50,00
Node housing structure		**Sum 1:**	**R478,28**
7	Solar panel	1	R176,00
8	TP4056 module	1	R9,50
9	Maduino	1	R329,00
10	SHT 31 sensor	1	R336,36
11	Antenna	1	R68,00
12	5000 mAh LiPo battery	1	R126,00
Sensor node setup		**Sum 2:**	**R1 044,86**
		Total:	**R1 523,14**

Table 4-15 represents three sensor nodes. The first, Node 1, was the cost price of the sensor developed by De Oliveira (2023). Node 2 was the sensor developed for this study, represented in Table 4-14, while the third is a popular sensor produced by Seeed Studios used as a market price comparison.

Table 4-15 Sensor node cost comparison

Node 1	Node 2	Seeed studio
R4 277,93	R1 523,14	R1 805,00

Table 4-15 shows that Node 2 proved to be considerably more affordable than Node 1 and moderately more affordable than the Seeed studio sensor node.

4.2.5 Summary of results

The analysis of temperature data collected by 14 sensor nodes, from September to December 2023, divided equally between shaded and unshaded spaces, showed that temperature was consistently cooler in shaded spaces. Differences, between shaded and unshaded spaces, in peak temperature on the warmest recorded days of each month were as follows: 28 September, shaded spaces peaked at 28 °C, while unshaded areas reached 31 °C. 19 October, shaded spaces recorded an average peak temperature of 34 °C, compared to 37 °C in unshaded areas, reflecting a 3 °C difference. November, shaded regions peaked at 34 °C versus 37 °C in unshaded spaces, and in December, shaded areas recorded an average peak of 29 °C, while unshaded areas reached 32 °C.

The results also show that temperature increased and decreased at different rates in shaded and unshaded spaces. Shaded spaces showed a more gradual change in temperature when compared to unshaded areas. Using the same dates of warmest recorded days of each month, Table 4-6 summarises the rate of heating and cooling of sensors that outputted the warmest and coolest temperatures. An overview of monthly results, highlighting the pattern of rates in shaded and unshaded spaces, is as follows:

- **September**, shaded areas warmed at 0.5 °C/h and cooled at 0.4 °C/h, while unshaded spaces exhibited heating at 0.6 °C/h and cooling at 0.7 °C/h.
- **October**, shaded areas increased their heating rate to 0.7 °C/h, compared to 0.9 °C/h for unshaded areas; cooling rates were also higher for unshaded areas at 1.6 °C/h versus 1.3 °C/h for shaded areas.
- **November**, both shaded and unshaded areas maintained similar heating rates of 0.7°C/h and 0.8 °C/h, respectively, with unshaded areas cooling slightly faster at 0.9 °C/h compared to 0.7 °C/h for shaded areas.
- **December**, the heating rates were stable for both shaded (0.7 °C/h) and unshaded (0.9 °C/h) areas, while cooling rates were lower for shaded areas at 0.5 °C/h compared to 0.7 °C/h for unshaded areas.

Analysis of the humidity data showed that it shared an inverse relationship with temperature. Multiple linear regression analysis showed that humidity in unshaded spaces, compared to the shaded spaces, were on average lower by 5.33 % in October, 9.31% in November and 3.1% in December. These values all had p values less than 0.05, indicating strong significance. The model suggested that the humidity was not significantly different between shaded and unshaded spaces for the month of September. Another finding was that the humidity in shaded spaces decreased at a statistically significant rate, although slight, to unshaded spaces.

The Heat Index calculation illustrates the sensation or the ‘feeling’ of temperature. It was seen that when temperature readings surpassed 27 °C, the low humidity readings in shaded spaces reduced the ambient temperature experience or sensation (it seemed cooler than what was being recorded). Heat Index calculations showed that the perceived temperature in shaded spaces were consistently 0.5 °C to 0.92 °C lower than the actual recorded temperature value.

The IoT sensor network’s ability to reliably transmit data, power supply, robustness and security were assessed. Data transmission statistics show that 146 437 messages were transmitted, 84 148 of those were during the data collection period from 7 September to 31 December. Of those

84 148 transmissions, a mere 0.10 % TER was measured. Power supply for shaded sensors utilised two 5000 mAh LiPo batteries in parallel, amounting to an operational time of 562 hours, approximately 23 days. In contrast, unshaded sensors operated for 116 days without requiring a battery change. Robustness and security measures put in place to protect sensor nodes were deemed successful. The sensors remained functional through all 27 days of rain during the data collection period, and no evidence of device tampering was detected. However, occasional wire/pin disconnections were noted.

The cost to build each of these sensor nodes were R1 523,14, which was less than commercially available options. Thus, indicating that IoT sensor networks are economically feasible in South Africa.

5. Conclusion

The implementation of IoT technology in South Africa offers a promising solution to the challenges associated with hydrological data collection and the monitoring of UHIE. The affordability and availability of IoT hardware components in South Africa make this technology not only feasible but also practical for widespread use. Building on the practical implementation of this project, a temperature and humidity IoT sensor was deployed on UCT campus to assess the feasibility of using IoT technology for studying UHIs and the cooling effects of shaded green spaces. The case study demonstrated the effectiveness of IoT in providing real-time, accessible data, which is crucial for understanding and managing urban micro-climates. As urbanisation continues to expand, resulting in the creation of more UHIs, the need for reliable and extensive data becomes increasingly critical. The deployment of IoT technology can play a pivotal role in addressing this need, enabling the development of effective strategies to mitigate the adverse impacts of UHIE, thus contributing to sustainable urban development and improved climate resilience in the region.

Hydrological phenoms require data to be collected over spatial and temporal scales. A priority of this study was to establish how to build an IoT network capable of collecting hydrological data. To meet this requirement, multiple sensors must be able to operate simultaneously whilst covering large areas over extended periods of time. TTN, utilising LoRaWAN technology, was used to successfully facilitate the communication of data between a gateway and 14 individual sensor nodes simultaneously, enabling the sending of ‘real-time’ data. The sensor nodes in this research were built around the Maduino ATmega328 MCU with an onboard LoRa radio transmitter. The MCU was integrated with a SHT 31 sensor which collected temperature and humidity data. The sensor nodes output temperature and humidity, accompanied with a timestamp.

An issue related to hydrological data in the past was accessibility to the data. Data collected in this study was webhooked from TTN to the MongoDB database, where it was stored as objects. These objects stored in the database were used to update a live feed map of the data, representing the temperature and humidity values at each location in real-time. MongoDB allowed for the seamless download of data stored within these objects into Comma Separated Value (CSV) format. This function was used to transfer data into Excel format for the analysis of results. The architecture used in this project managed to mitigate issues regarding access to the data.

Sensor nodes were deployed on site for a period of four months and were programmed to collect and transmit data at intervals of 15 minutes. Two approaches were taken to power sensor nodes through the duration of the data collection period, both sets of sensors were programmed with the same sleep mode. The first approach was for shaded sensors to increase the available battery capacity by using two LiPo 5000 mAh batteries in parallel. The second, for unshaded sensor nodes was to harvest the available ambient energy in the form of solar energy. Unshaded sensor nodes were fitted with a 6V 4.5 W solar panel and charging module and made use of only a single LiPo 5000 mAh battery. The approach used for the shaded sensors underperformed as battery life only achieved 49 % of the expected duration. This resulted in a required battery change every 23 days, making this an unviable solution for energy supply. Unshaded sensors completed the data

collection period of 116 days without requiring a single battery change/replacement. When compared to the shaded sensors, the unshaded sensor nodes operated for a minimum of 85 days longer, making this a viable alternative for supplying power to IoT sensor nodes collecting hydrological data. It is important to note that ‘a minimum’ was used to describe the difference in days between the two sets of sensors as the figure of 85 days could have been far higher if the data collection period had been extended.

The cost to build the sensor node used in this project was R1 523,14 which is a significant improvement from the cost of the sensor, R4 277,93, built by De Oliveira (2023) and moderately more affordable than the Seeed Studio sensor node, R1 805,00. All components used in the build of this node and in the cost breakdown were easily procured from within the South African market and bought off South African ‘shelves’. The build of the sensor node in this project showed that it is certainly feasible and affordable to use IoT technology as an application to collect hydrological data in South Africa.

The data collected by sensors in this research project show that urban shaded spaces are cooler with more moderate climates than those of unshaded spaces, showing that shaded green spaces play a role in mitigating the UHIE. Patterns from the temperature show that the largest difference in temperature is seen at the time of peak temperature readings from the unshaded spaces. The temperature in shaded spaces also increased and decreased at a lower rate compared unshaded which showed evidence of more moderate climate conditions, as well as often reaching peak temperature values at a later stage/time of day than unshaded spaces. The collected data showed that humidity was inversely proportional to temperature. Statistically significant variations between shaded and unshaded humidity patterns were identified, where humidity values in shaded spaces were higher on average but also decreased at a sharper rate as temperature increased compared to unshaded spaces. Using the Heat Index formula with the given humidity reading, the temperature ‘felt’ in shaded spaces was consistently less than the true temperature reading at the time the peak temperature reading was recorded. Thus, further corroborating the role that shaded spaces have on cooling ambient air temperature.

In conclusion, the IoT sensor network built for the purpose of this study was capable of collecting and transmitting live/real-time hydrological data consistently, as well as being easily accessible by an end user. The sensor node was built affordably with components available in South Africa; and a feasible method of power supply for the built sensor node, utilising energy harvesting, was established. The WSN also proved to be capable of collecting temperature and humidity data to monitor the UHIE and impact shaded spaces had on cooling ambient air temperature on UCT campus. In accordance with the reviewed literature discussed in section 2, the built WSN met the requirements to perform as an IoT WSN suited for hydrological data collection, as well as being financially feasible for use in South Africa.

6. Recommendations

This research project focused on building and testing an IoT sensor network as a hydrological application to monitor the UHIE. The developed network has managed to meet the initial set out requirements, but should be seen as a conceptual build and a platform from which to improve. Many potential avenues and areas of improvement have been identified that may offer a more robust sensor node and WSN, as well as yielding further key insights into the UHIE.

The MCU that the sensor node was built around was a Maduino, powered by an ATmega328 chip. The primary attraction to using this board was its low cost, and it being readily available. Despite meeting the requirements of an IoT sensor node, many challenges were met particularly relating to the energy consumption and 'user-friendliness' capability of the device. In future research, if possible, alternative MCU options should be investigated, particularly ones that utilise a chip other than the Atmega328. In a previous project, De Oliveira (2023) used the Pycom Lopy 4 MCU which used an ESP32 chipset, which appeared to allow for far more comprehensive sleep mode libraries. The ESP32 chip allows for the programming language Python to be used to program the MCU, which could be seen as a more intuitive and 'user-friendly' language compared to C++ used to code the Atmega328P used by the Maduino. A step further would be to develop a board based on the specific requirements of a sensor node intended to collect hydrological data. Developing this board would be more time consuming but would solve many of the challenges related to energy consumption. An additional approach to achieve more optimal energy consumption would be to strip the MCU of components not required for its function as a hydrological sensor node e.g. built in LED lights and replacing linear voltage regulators with switch voltage regulators. This would also reduce the cost of the build as smaller batteries would be required and the developed board would be cheaper to reproduce than to purchase MCUs for projects that require scalability.

Early in the methodology it was decided to utilise energy harvesting techniques, namely, solar energy, as it proved to be a more sufficient solution to address problems associated with energy consumption in unshaded spaces. Future research should investigate the feasibility of applying energy harvesting techniques to sensors in shaded spaces.

To counter maintenance issues encountered throughout the data collection period, changes to the layout of electrical components, wires and connections should be made. If possible, designing a housing unit with divisions or separate compartments to secure the different electrical components would make servicing/trouble-shooting deployed sensor nodes a simpler task. In tandem with divisions within the housing unit, clear paths for wires connecting the various electrical components along with sturdier more secure pin connections, would also increase the robustness of the sensor node.

In this study, temperature and humidity data from shaded and unshaded spaces were compared to each other to determine whether UHIs were present and if shaded spaces had a cooling effect on ambient air temperatures. To better understand the patterns in this data, it is recommended that future research should investigate the following factors:

(i) The impact of variation of vegetation and paving within the shaded and unshaded regions have on the temperature and humidity (that is, what vegetation is present, what percentage land cover, and what paving material is used); and

(ii) Sensor nodes that could also produce data on more niche processes such as evaporation and 'breeze' patterns in shaded spaces.

These additions would bring about key insights and may provide a more rounded explanation and improve the R^2 statistical model fit value.

An additional recommendation would be to leverage the data collected by the WSN to evaluate the impact UHIs have on health issue trends in the City of Cape Town. Further, by building on the existing IoT WSN infrastructure used in this study, a dynamic model which incorporates data from the sensors as well as the health department can be developed.

A map displaying the live feed of temperature and humidity data of sensors at the respective locations was developed during this study. The map itself had no role in the data analysis of this study but did show the potential for providing accessible live streamed data for hydrological applications. In future research, steps toward building a model that uses data collected by deployed IoT sensor nodes, combined with Land Surface Temperature (LST) data collected by thermal imagery, should be explored. The combination of IoT and LST data could be used to develop comprehensive and adaptive exothermic heat maps of the region being monitored.

Overall, while the initial deployment of the IoT sensor network has provided valuable insights and confirmed the feasibility of using such technology for monitoring the UHIE, the study has also highlighted several areas for improvement. Addressing issues related to energy consumption, maintenance, and data collection accuracy, will be crucial for future research. It is recommended that future studies explore alternative MCU options, such as, (i) enhancing energy harvesting techniques; and (ii) refining the design and layout of sensor nodes; this will assist future projects to be more robust and user-friendly and decrease reliance on retail components. Furthermore, integrating advanced data analysis techniques and expanding the scope of environmental parameters, will enhance our understanding of the UHIE and contribute to more effective urban planning and climate adaptation strategies.

7. References

Adepoju, F. (2021, June 20). *Webhooks Explained: What Are Webhooks and How Do They Work? | by Fikayo Adepoju | Hookdeck | Medium*. https://medium.com/hookdeck/webhooks-explained-what-are-webhooks-and-how-do-they-work-90c613c055a4

Ajao, L. A., Agajo. J, Kolo, J. G., Maliki, D., & Adegboye, M. A. (2017). *WIRELESS SENSOR NETWORKS BASED-INTERNET OF THING FOR AGRO-CLIMATIC PARAMETERS MONITORING AND REAL-TIME DATA ACQUISITION*. https://doi.org/10.18488/journal.2.2017.76.240.252

Akorede, M. F. (2022). Design and performance analysis of off-grid hybrid renewable energy systems. *Hybrid Technologies for Power Generation*, 35–68. https://doi.org/10.1016/B978-0-12-823793-9.00001-2

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, *17*(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

Al-Qurabat, A. K. M., & Abdulzahra, S. A. (2020). An Overview of Periodic Wireless Sensor Networks to the Internet of Things. *IOP Conference Series: Materials Science and Engineering*, *928*(3). https://doi.org/10.1088/1757-899X/928/3/032055

Amghar, S., Cherdal, S., & Mouline, S. (2018). Which NoSQL database for IoT Applications? *2018 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, 131–137. https://doi.org/10.1109/MoWNet.2018.8428922

Arduino, Bagur, J., & Chung, T. (2023, December 29). *Arduino Memory Guide | Arduino Documentation*. https://docs.arduino.cc/learn/programming/memory-guide/

Banu, M. (2017). *IoT Architecture a Comparative Study*. https://doi.org/10.12732/ijpam.v117i8.10

Barthwal, A., & Sharma, K. (2021). Analysis and prediction of urban ambient and surface temperatures using internet of things. *International Journal of System Assurance Engineering and Management*, *13*. https://doi.org/10.1007/s13198-021-01502-3

Bellouch, A., Boujnoui, A., Zaaloul, A., & Haqiq, A. (2021). *Modeling and Performance Evaluation of LoRa Network Based on Capture Effect* (pp. 249–263). https://doi.org/10.1007/978-981-33-6129-4_18

Bhavsar, A. (2018). *A Guide for selecting the right microcontroller for your IoT project – IIoT World*. https://iiot-world.com/connected-industry/a-guide-for-selecting-the-right-microcontroller-for-your-iot-project/

Bolic, M. (2023). Transducers. *Pervasive Cardiovascular and Respiratory Monitoring Devices*, 37–71. https://doi.org/10.1016/B978-0-12-820947-9.00004-0

Brooke Anderson, G., Bell, M. L., & Peng, R. D. (2013). Methods to calculate the heat index as an exposure metric in environmental health research. *Environmental Health Perspectives*, *121*(10), 1111–1119. https://doi.org/10.1289/EHP.1206273

Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., & Lemmens, R. (2011). New Generation Sensor Web Enablement. *Sensors*, *11*(3), 2652–2699. https://doi.org/10.3390/s110302652

Cerchecci, M., Luti, F., Mecocci, A., Parrino, S., Peruzzi, G., & Pozzebon, A. (2018). A Low Power IoT Sensor Node Architecture for Waste Management Within Smart Cities Context. *Sensors 2018, Vol. 18, Page 1282*, *18*(4), 1282. https://doi.org/10.3390/S18041282

Chapman, L., Muller, C., Young, D., Warren, E., Grimmond, S., Cai, X., & Ferranti, E. (2014). The Birmingham Urban Climate Laboratory: An Open Meteorological Test Bed and Challenges of the Smart City. *Bulletin of the American Meteorological Society*, *96*. https://doi.org/10.1175/BAMS-D-13-00193.1

Chen, Y., Shan, B., & Yu, X. (2022). Study on the spatial heterogeneity of urban heat islands and influencing factors. *Building and Environment*, *208*, 108604. https://doi.org/https://doi.org/10.1016/j.buildenv.2021.108604

Das, A., Sarma, M. P., Sarma, K. K., & Mastorakis, N. (2018). Design of an IoT based Real Time Environment Monitoring System using Legacy Sensors. *MATEC Web of Conferences*, *210*, 03008. https://doi.org/10.1051/MATECCONF/201821003008

De Oliveira, R. (2023). *Transition to novel Internet of Things technology for measurement in hydrology – case study in Cape Town, South Africa*.

Difference between Active Transducer and Passive Transducer - GeeksforGeeks. (2022, July 7). https://www.geeksforgeeks.org/difference-between-active-transducer-and-passive-transducer/

Difference between Sensors and Actuators - javatpoint. (2021). https://www.javatpoint.com/difference-between-sensors-and-actuators

End Device Activation | The Things Network. (n.d.). Retrieved May 1, 2024, from https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/

Ghapar, A. A., Yussof, S., & Abu Bakar, A. (2018). Internet of Things (IoT) Architecture for Flood Data Management. *International Journal of Future Generation Communication and Networking*, *11*(1), 55–62. https://doi.org/10.14257/ijfgcn.2018.11.1.06

GitHub - microrobotics/YN-868MHZ-5DBI_datsheet: Yetnorson YN-868MHZ-5DBI Antenna datasheet. (n.d.). Retrieved May 1, 2024, from https://github.com/microrobotics/YN-868MHZ-5DBI_datsheet

Hamami, L., & Nassereddine, B. (2020). Application of wireless sensor networks in the field of irrigation: A review. *Computers and Electronics in Agriculture*, *179*. https://doi.org/10.1016/j.compag.2020.105782

Hao, D., Qi, L., Tairab, A. M., Ahmed, A., Azam, A., Luo, D., Pan, Y., Zhang, Z., & Yan, J. (2022). Solar energy harvesting technologies for PV self-powered applications: A comprehensive review. *Renewable Energy*, *188*, 678–697. https://doi.org/https://doi.org/10.1016/j.renene.2022.02.066

Hasan, K., Tom, N., & Yuce, M. R. (2023). Navigating Battery Choices in IoT: An Extensive Survey of Technologies and Their Applications. *Batteries 2023, Vol. 9, Page 580*, *9*(12), 580. https://doi.org/10.3390/BATTERIES9120580

Hassan, Q. F., Atta ur, R. K., & Madani, S. A. (2017). *Internet of Things*. https://doi.org/10.1201/9781315155005

Hidayat, D. J., & Soekirno, S. (2021). Development of temperature monitoring and prediction system for urban heat island (UHI) based on the internet of things. *Journal of Physics: Conference Series*, *1816*(1), 012054. https://doi.org/10.1088/1742-6596/1816/1/012054

Hoperf Electronic. (n.d.). *RFM95/96/97/98 - Low Power Long Range Transceiver Module V1.0*.

Houghton, J. T., Ding, Y., Griggs, D. J., Noguer, M., van der Linden, P. J., Dai, X., Maskell, K., & Johnson, C. A. (2001). *Climate change 2001: the scientific basis: contribution of Working Group I to the third assessment report of the Intergovernmental Panel on Climate Change*. Cambridge university press.

Inter-Integrated Circuit (I2C) Protocol | Arduino Documentation. (n.d.). Retrieved May 1, 2024, from https://docs.arduino.cc/learn/communication/wire/

J P Bentley. (1984). Temperature sensor characteristics and measurement system design. *Journal of Physics E: Scientific Instruments*, *17*(6), 430. https://doi.org/10.1088/0022-3735/17/6/002

Jabraeil Jamali, M. A., Bahrami, B., Heidari, A., Allahverdizadeh, P., & Norouzi, F. (2020). *IoT Architecture BT - Towards the Internet of Things: Architectures, Security, and Applications*. https://doi.org/10.1007/978-3-030-18468-1_2

Jiang, X., Polastre, J., & Culler, D. (2005). *Perpetual Environmentally Powered Sensor Networks*.

Kansal, A., Hsu, J., Zahedi, S., & Srivastava, M. B. (2007). Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems*, *6*(4). https://doi.org/10.1145/1274858.1274870

Kaynak, B., Mermer, O., Sermet, Y., & Demir, I. (2025). HydroSignal: open-source internet of things information communication platform for hydrological education and outreach. *Environmental Monitoring and Assessment*, *197*(4), 359. https://doi.org/10.1007/s10661-025-13812-1

Keim, R. (2019). *What Is a Microcontroller? The Defining Characteristics and Architecture of a Common Component - Technical Articles*. https://www.allaboutcircuits.com/technical-articles/what-is-a-microcontroller-introduction-component-characteristics-component/

Khatri, P. (2019, September 12). *Arduino Sleep Modes and How to use them to Save the Power*. https://circuitdigest.com/microcontroller-projects/arduino-sleep-modes-and-how-to-use-them-to-reduce-power-consumption

Khriji, S., Houssaini, D. El, Kammoun, I., & Kanoun, O. (2018). Energy-efficient techniques in wireless sensor networks. *Energy Harvesting for Wireless Sensor Networks: Technology, Components and System Design*, 287–303. https://doi.org/10.1515/9783110445053-017

Kishorebabu, V., & Sravanthi, R. (2020). Real Time Monitoring of Environmental Parameters Using IOT. *Wireless Personal Communications*, *112*(2), 785–808. https://doi.org/10.1007/S11277-020-07074-Y/TABLES/9

Kumar, N. M., & Mallick, P. K. (2018). The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. *Procedia Computer Science*, *132*, 109–117. https://doi.org/10.1016/j.procs.2018.05.170

Lazarescu, M. T. (2013). Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *3*(1), 45–54. https://doi.org/10.1109/JETCAS.2013.2243032

Leonardi, L., Lo Bello, L., Battaglia, F., & Patti, G. (2020). Comparative Assessment of the LoRaWAN Medium Access Control Protocols for IoT: Does Listen before Talk Perform Better than ALOHA? *Electronics*, *9*. https://doi.org/10.3390/electronics9040553

Leonardi, L., Lo Bello, L., Patti, G., Pirri, A., & Pirri, M. (2023). Combined Use of LoRaWAN Medium Access Control Protocols for IoT Applications. *Applied Sciences*, *13*(4). https://doi.org/10.3390/app13042341

Levermore, G., Parkinson, J., Lee, K., Laycock, P., & Lindley, S. (2018). The increasing trend of the urban heat island intensity. *Urban Climate*, *24*, 360–368. https://doi.org/https://doi.org/10.1016/j.uclim.2017.02.004

Li, M., Lu, J., Chen, Z., & Amine, K. (2018). 30 Years of Lithium-Ion Batteries. *Advanced Materials*, *30*(33), 1800561. https://doi.org/10.1002/ADMA.201800561

Li, Y., Wang, L., Zhang, L., Liu, M., & Zhao, G. (2019). Monitoring Intra-annual Spatiotemporal Changes in Urban Heat Islands in 1449 Cities in China Based on Remote Sensing. *Chinese Geographical Science*, *29*(6), 905–916. https://doi.org/10.1007/s11769-019-1080-9

LoPy4 - Pycom - Quadruple Bearer MicroPython enabled Dev Board. (n.d.). Retrieved June 20, 2022, from https://pycom.io/product/lopy4/

LoRa Alliance Technical Committee. (2018). *LoRaWAN 1.0.3 Specification*.

Lutkevich, B. (2019). *What is a Microcontroller and How Does it Work?* https://www.techtarget.com/iotagenda/definition/microcontroller

Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, *03*(05), 164–173. https://doi.org/10.4236/jcc.2015.35021

Maduino ATmega328 + RFM96 868MHz - Micro Robotics. (n.d.). Retrieved May 2, 2024, from https://www.robotics.org.za/OAC915MLR?search=maduino

Maduino LoRa Radio - Makerfabs Wiki. (n.d.). Retrieved May 28, 2024, from https://wiki.makerfabs.com/Maduino_LoRa_Radio.html

Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2018). Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT. *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 197–202. https://doi.org/10.1109/PERCOMW.2018.8480255

Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, *5*(1), 1–7. https://doi.org/10.1016/j.icte.2017.12.005

Meque, A., Gamedze, S., Moitlhobogi, T., Booneeady, P., Samuel, S., & Mpalang, L. (2021). Numerical weather prediction and climate modelling: Challenges and opportunities for improving climate services delivery in Southern Africa. *Climate Services*, *23*, 100243. https://doi.org/https://doi.org/10.1016/j.cliser.2021.100243

Michel, F. (2017). *Integrating Heterogeneous Data Sources in the Web of Data*.

Misra, S., Mukherjee, A., & Roy, A. (Eds.). (2021). IoT Sensing and Actuation. In *Introduction to IoT* (pp. 97–114). Cambridge University Press. https://doi.org/DOI: 10.1017/9781108913560.007

Mohammad Harmay, N. S., Kim, D., & Choi, M. (2021). Urban Heat Island associated with Land Use/Land Cover and climate variations in Melbourne, Australia. *Sustainable Cities and Society*, *69*, 102861. https://doi.org/https://doi.org/10.1016/j.scs.2021.102861

Nakas, C., Kandris, D., & Visvardis, G. (2020). Energy Efficient Routing in Wireless Sensor Networks: A Comprehensive Survey. *Algorithms*, *13*, 72. https://doi.org/10.3390/a13030072

Noura, M., Atiquzzaman, M., & Gaedke, M. (n.d.). *Interoperability in Internet of Things: Taxonomies and Open Challenges*. https://doi.org/10.1007/s11036-018-1089-9

Onojakpor, O. (2023, November 20). *SQL vs NoSQL Databases: A Guide*. https://www.dbvis.com/thetable/sql-vs-nosql-databases-which-is-better/#

Oshana, R. (2006). 4 - Overview of Digital Signal Processing Algorithms. In R. Oshana (Ed.), *DSP Software Development Techniques for Embedded and Real-Time Systems* (pp. 59–121). Newnes. https://doi.org/https://doi.org/10.1016/B978-075067759-2/50006-5

Pop, A.-I., Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks? *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 1–6. https://doi.org/10.1109/GLOCOM.2017.8254509

Prasad, A., & Chawda, P. (2018). Power management factors and techniques for IoT design devices. *2018 19th International Symposium on Quality Electronic Design (ISQED)*, 364–369. https://doi.org/10.1109/ISQED.2018.8357314

Pretorius, M., & Mthethwa, S. N. (2019). *A survey of distributed trust mechanisms suitable for IoT devices*.

Raj, A., & Steingart, D. (2018). Review—Power Sources for the Internet of Things. *Journal of The Electrochemical Society*, *165*(8), B3130. https://doi.org/10.1149/2.0181808jes

Rautmare, S., & Bhalerao, D. M. (2016). MySQL and NoSQL database comparison for IoT application. *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, 235–238. https://doi.org/10.1109/ICACA.2016.7887957

Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys and Tutorials*, *19*(2), 855–873. https://doi.org/10.1109/COMST.2017.2652320

Renesas. (n.d.). *MCU Basic Structure/Operation*. Retrieved May 6, 2024, from https://www.renesas.com/us/en/support/engineer-school/mcu-01-basic-structure-operation

Rhee, S., Seetharam, D., & Liu, S. (2004). *Techniques for Minimizing Power Consumption in Low Data-Rate Wireless Sensor Networks*.

Rouse, M. (2015). *What is a System Bus? - Definition from Techopedia*. https://www.techopedia.com/definition/2307/system-bus

Sendra, S., Lloret, J., García, M., & Toledo, J. F. (2011). *Power saving and energy optimization techniques for Wireless Sensor Networks*. https://doi.org/10.4304/jcm.6.6.439-459

SenseCAP Gateway - LoRaWAN EU868, US915, AU915, AS923 - Seeed Studio IIoT Solutions. (n.d.). Retrieved April 25, 2024, from https://solution.seeedstudio.com/product/sensecap-gateway-lorawan/

Sethi, P., & Sarangi, S. R. (2017). *Internet of Things: Architectures, Protocols, and Applications*. https://doi.org/10.1155/2017/9324035

SHT31 Weather-proof Temperature & Humidity Sensor - DFRobot. (n.d.). Retrieved June 20, 2022, from https://www.dfrobot.com/product-2160.html

Somov, A., & Giaffreda, R. (2015, November 9). *Powering IoT Devices: Technologies and Opportunities - IEEE Internet of Things*. https://iot.ieee.org/articles-publications/newsletter/november-2015/powering-iot-devices-technologies-and-opportunities.html?TB_iframe=true&width=921.6&height=921.6

Stanbury, P. F., Whitaker, A., & Hall, S. J. (2017). Instrumentation and control. *Principles of Fermentation Technology*, 487–536. https://doi.org/10.1016/B978-0-08-099953-1.00008-9

Sudevalayam, S., & Kulkarni, P. (2011). Energy Harvesting Sensor Nodes: Survey and Implications. *IEEE Communications Surveys & Tutorials*, *13*(3), 443–461. https://doi.org/10.1109/SURV.2011.060710.00094

Teel, J. (2022). *Microcontroller or Microprocessor: Which is Right for Your New Product?* https://predictabledesigns.com/microcontroller-or-microprocessor-which-is-right-for-your-new-product/#

Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*, *79*, 849–861. https://doi.org/10.1016/J.FUTURE.2017.09.020

What Are Webhooks And How Do They Work. (n.d.). Retrieved April 26, 2024, from https://hookdeck.com/webhooks/guides/what-are-webhooks-how-they-work

What is LoRaWAN® Specification - LoRa Alliance®. (n.d.). Retrieved October 25, 2021, from https://lora-alliance.org/about-lorawan/

Yang, X., & Yao, L. (2022). Reexamining the relationship between surface urban heat island intensity and annual precipitation: Effects of reference rural land cover. *Urban Climate*, *41*, 101074. https://doi.org/https://doi.org/10.1016/j.uclim.2021.101074

Yang, Z., Yue, Y., Yang, Y., Peng, Y., Wang, X., & Liu, W. (2011). Study and application on the architecture and key technologies for IOT. *2011 International Conference on Multimedia Technology, ICMT 2011*, 747–751. https://doi.org/10.1109/ICMT.2011.6002149

Yelamarthi, K., Aman, M. S., & Abdelgawad, A. (2017). An Application-Driven Modular IoT Architecture. *Wireless Communications and Mobile Computing*, *2017*, 1350929. https://doi.org/10.1155/2017/1350929

Young, J. K. (2008, July 1). *A Practical Guide to Battery Technologies for Wireless Sensor Networking | Fierce Electronics*. https://www.fierceelectronics.com/components/a-practical-guide-to-battery-technologies-for-wireless-sensor-networking

Zaki, I. M., Amin, M., Ercan, A., Ishida, K., Levent Kavvas, M., Chen, Z. Q., & Jang, S.-H. (2019). *Impacts of Climate Change on the Hydro-Climate of Peninsular Malaysia*. https://doi.org/10.3390/w11091798

Zhao, L., Lee, X., Smith, R. B., & Oleson, K. (2014). Strong contributions of local background climate to urban heat islands. *Nature*, *511*(7508), 216–219. https://doi.org/10.1038/nature13462

8. Appendices

Appendix A: Extract from document made as a request for permission to mount sensor nodes on lamp posts at UCT

Appendix B: Maduino MCU Schematic

Appendix C: LoRa Transceiver

Appendix D: LoRaWAN gateway

Appendix E: Temperature graphs

Appendix F: Humidity vs Temperature graphs

Appendix G: Monthly breakdown of data transmission

Appendix A: Extract from document made as a request for permission to mount sensor nodes on lamp posts at UCT

Introduction

The Urban Heat Island Effect (UHIE) is a phenomenon where there is a detectable heat increase in paved areas. The purpose of my research is to investigate UHIE on UCT campus using IoT technology.In order to do so, ambient temperature and humidity data from green (unpaved and shaded areas) and paved (open paved areas) will be collected and compared.

It is proposed that sensors will be mounted on trees (green areas) and lamp posts (paved areas) to collect this data. Approval for the mounting of sensors on to trees has already been received. This document serves as an application to gain approval for the mounting of sensors on to lamp posts from 22/03/2023to 31/01/2024.

Lamp posts to be used

Below are images of the three variations of lamp post being identified as possible sensor mounts, for this project.

Figure 1: Lamp post 1

Figure 2: Lamp

Figure 3: Lamp post 3

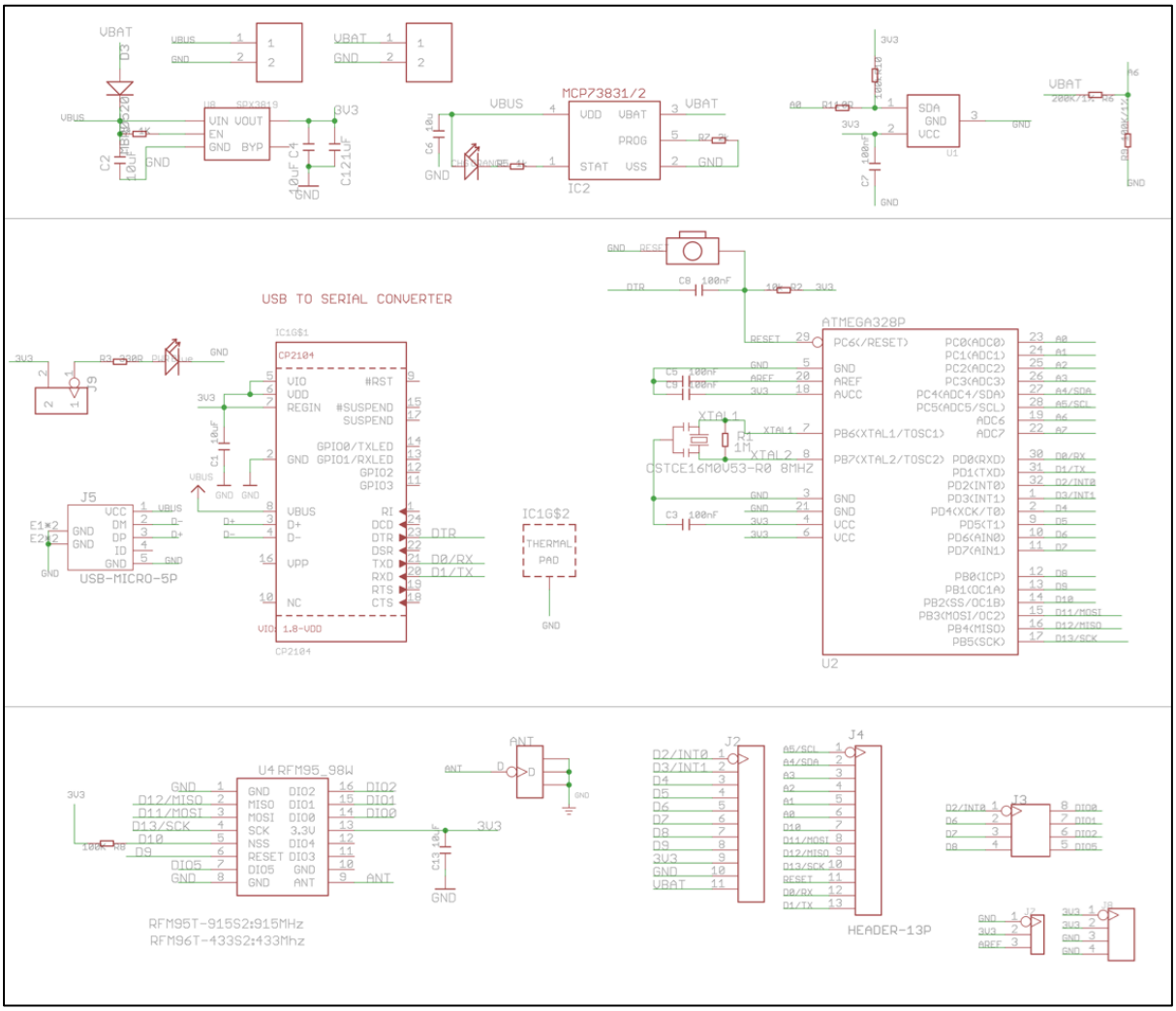
Proposed Design

Figure 4 shows the proposed sensor design. The sensor is protected by a radiation shield (labelled 1), with the components powering the sensor housed in an electrical box (labelled 2). Both the box and theshield are fixed onto an aluminium bracket (labelled 3) and mounted to the lamp post with the use of stainless-steel ties (labelled 4). The ties are easily tightened and as these sensors are intended to be temporary, they are also easily removed. No damage to the lamp post mount will be incurred.

The lowest point of each sensor node will be placed 2.3 m above the ground to be unobstructive, for safety and to avoid tampering.

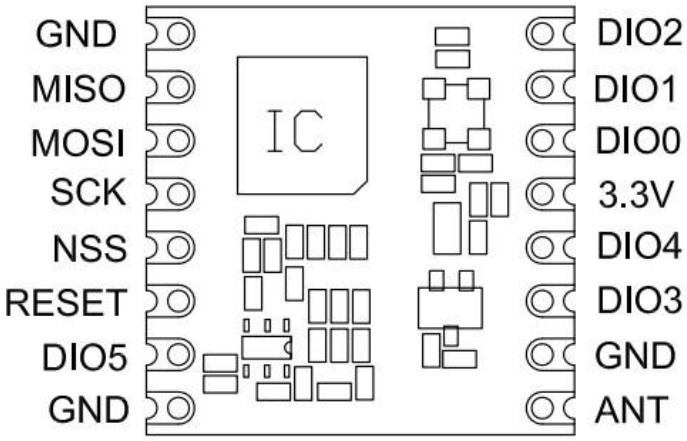
Figure 4: Proposed Design

Appendix B: Maduino ATmega328 MCU schematic



(Makerfabs, n.d.)

Appendix C: LoRa transceiver schematic/pin diagram and pin description



.

Number	Name	Type	Description Description Stand Alone Mode
1	GND	-	Ground
2	MISO	I	SPI Data output
3	MOSI	O	SPI Data input
4	SCK	I	SPI Clock input
5	NSS	I	SPI Chip select input
6	RESET	I/O	Reset trigger input
7	DIO5	I/O	Digital I/O, software configured
8	GND	-	Ground
9	ANT	-	RF signal output/input.
10	GND	-	Ground
11	DIO3	I/O	Digital I/O, software configured
12	DIO4	I/O	Digital I/O, software configured
13	3.3V	-	Supply voltage
14	DIO0	I/O	Digital I/O, software configured
15	DIO1	I/O	Digital I/O, software configured
16	DIO2	I/O	Digital I/O, software configured

(Hoperf Electronic, n.d.)

Appendix D: Seeed studio LoRaWAN gateway Specifications

LoRaWAN Gateway

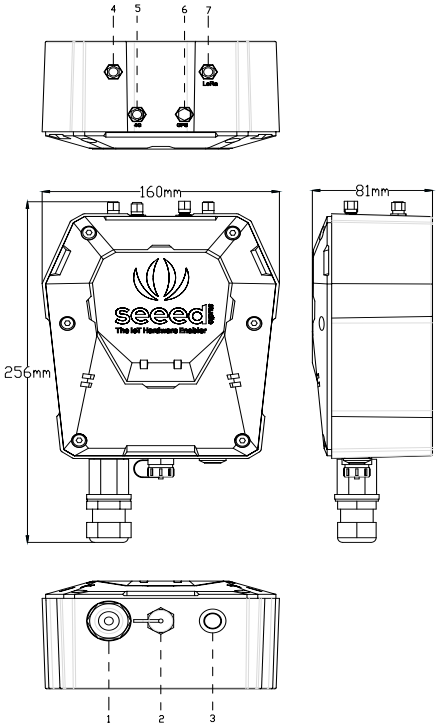
Specifications

Product Model	
Model	Region
LoRa-G-470-E/4G	Asia(China)
LoRa-G-868-E/4G	European, Africa, Asia(India etc.)
LoRa-G-915-E/4G	North America, South America, Oceania,Asia(Japan, Korea, Thailand, etc.)

LoRa Parameters			
Channel Plan	470~510MHz	863~870MHz	902~928MHz
Power Output	25dBm	27dBm	26dBm
Sensitivity	140.5dBm (SF12BW125)	139.5dBm (SF12BW125)	141.5dBm (SF12BW125)

General Parameters	
CPU	TI AM3358 Cortex-A8 1GHz
System	Linux Debian
RAM	DDR3 512MB
Memory	4GB eMMC
Ethernet	100Mbps FE (RJ-45)
4G Frequency	B3/B7/B20/GSM900/GSM1800 (Europe/APAC) B2/B4/B12(AT&T, T-Mobile) B4/B13(Verizon) LTE Speed Category: Cat.1 LTE Speed: down link 10.3Mb/s, up link 5.2Mb/s
Antenna	LoRa Antenna *1, 4G Antenna *1
LED Indicator	Indicating network condition (online/ offline)
Grounding	Reserved 1 screw hole for GND
Power Consumption	3.6W
Power Supply	DC 12V/1.5A
IP Rating	IP66
UV Resistance	anti-aging (from rain/sun exposure): UL746C F1
Enclosure Material	PC+PBT
Operating Temperature	-40 ℃ to +70 ℃
Operating Humidity	0 to 100 %RH (not solid condition)
Installation Method	Wall or pole mounting
Device Weight	840g

Device Dimensions

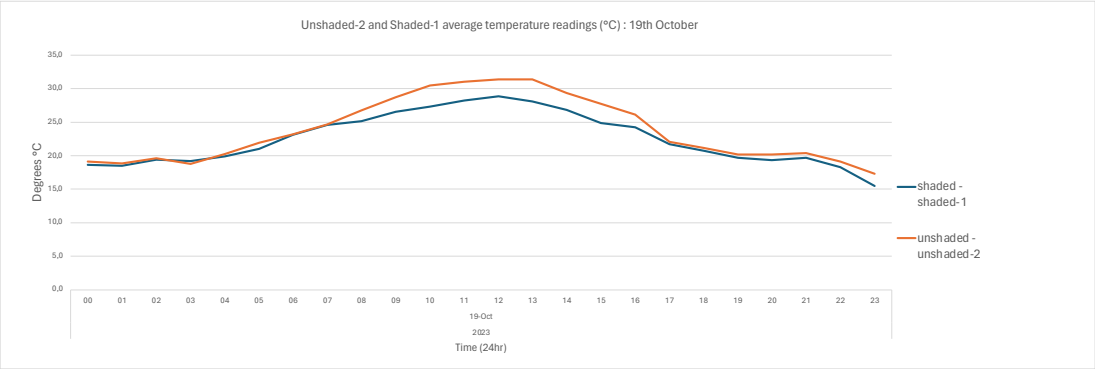


1. Ethernet Port
2. Power Connector
3. LED
4. Reserved
5. 4G Antenna Connector
6. Reserved
7. LoRa Antenna Connector

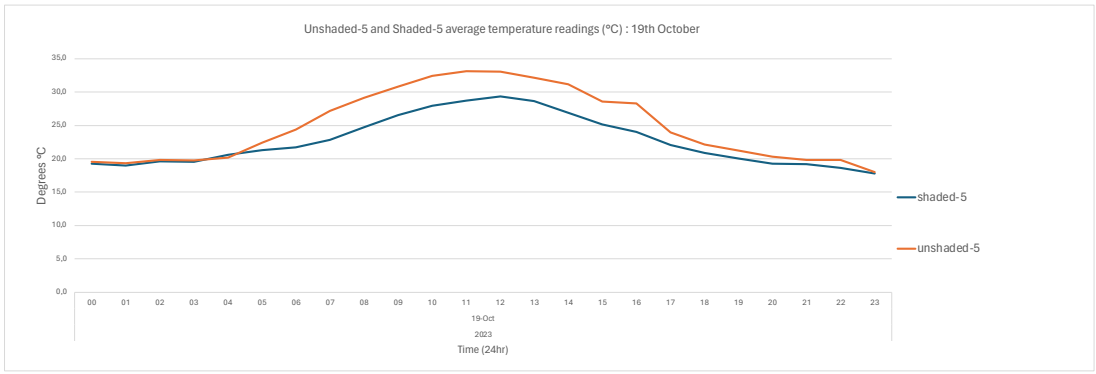
(Seeed Studio IIoT Solutions, n.d.)

Appendix E: Temperature Graphs

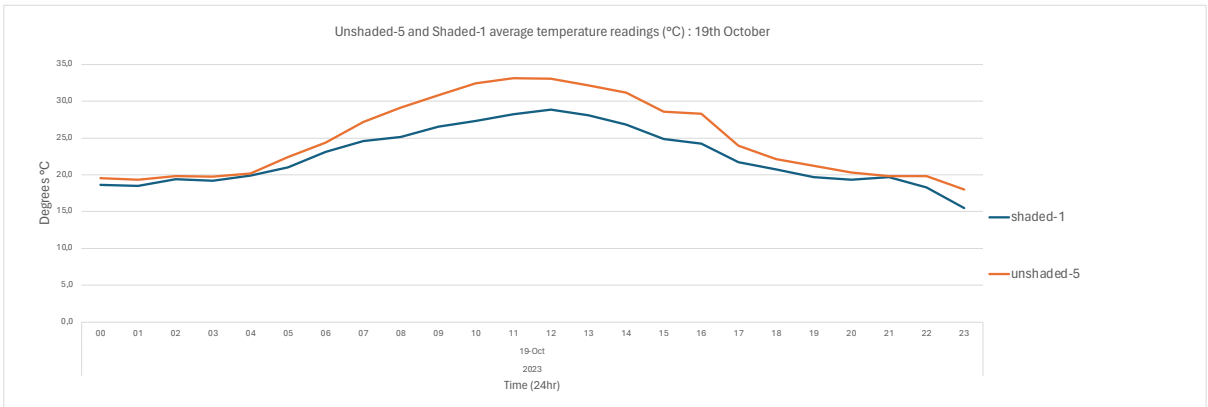
Appendix E-1: Unshaded-2 vs Shaded-1 average temperature readings (°C) – 19thOctober 2023



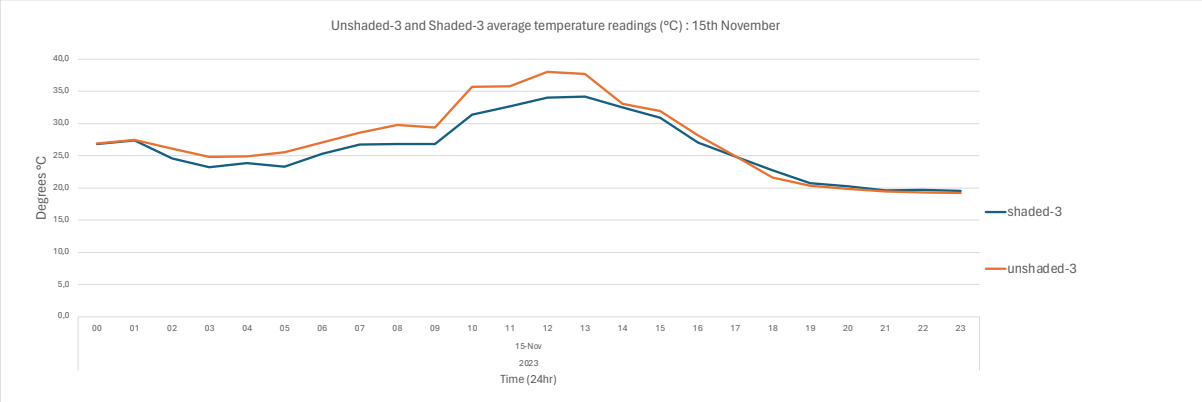
Appendix E-2: Unshaded-5 vs Shaded-5 average temperature readings (°C) – 19th October 2023



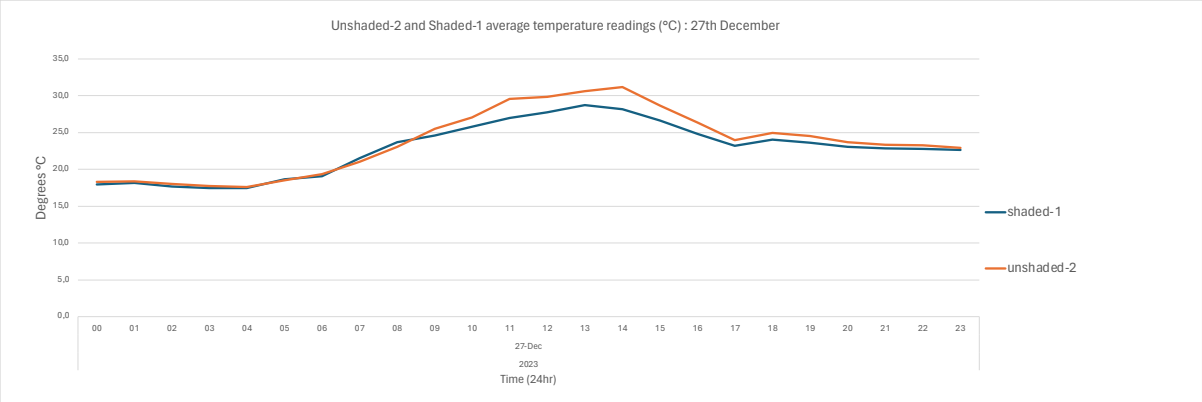
Appendix E-3: Unshaded-5 vs Shaded-1 average temperature readings (°C) – 19th October 2023



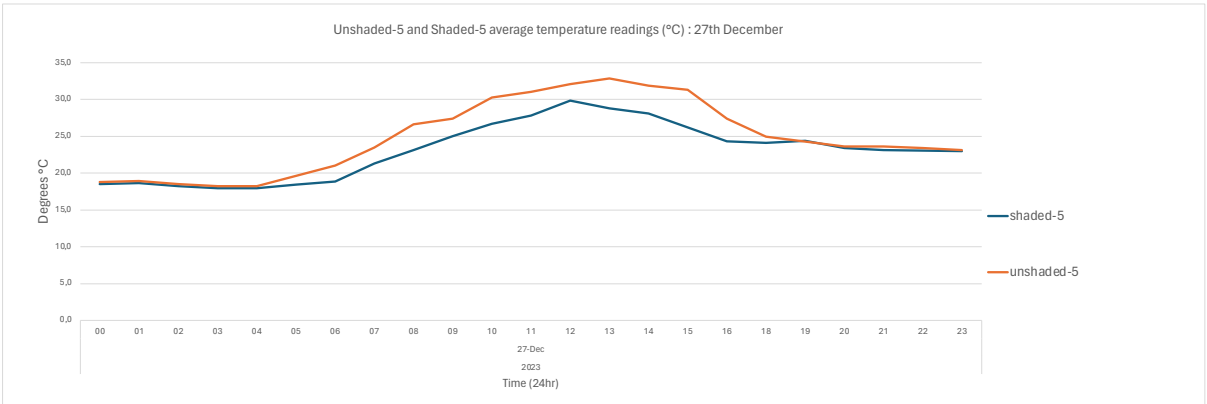
Appendix E-4: Unshaded-3 vs Shaded-3 average temperature readings (°C) – 15th November 2023



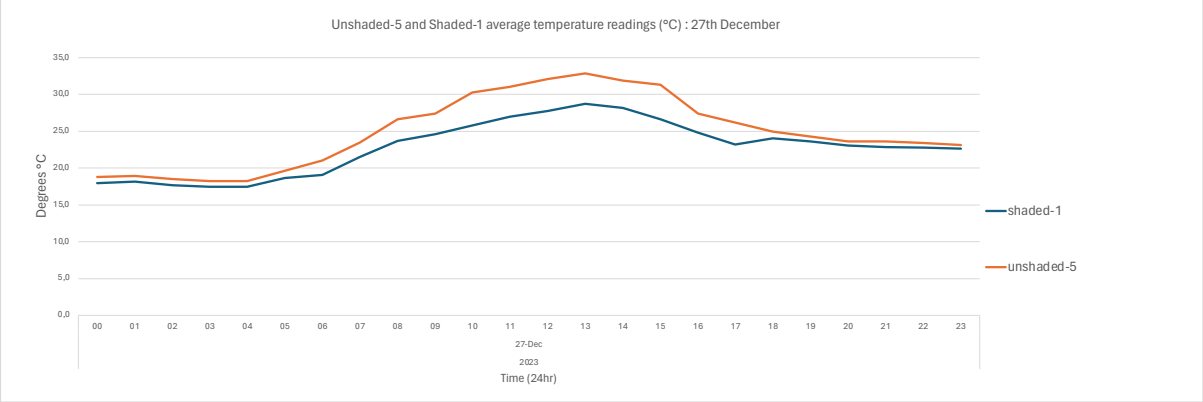
Appendix E-5: Unshaded-2 vs Shaded-1 average temperature readings (°C) – 27th December 2023



Appendix E-6: Unshaded-5 vs Shaded-5 average temperature readings (°C) – 27th December 2023

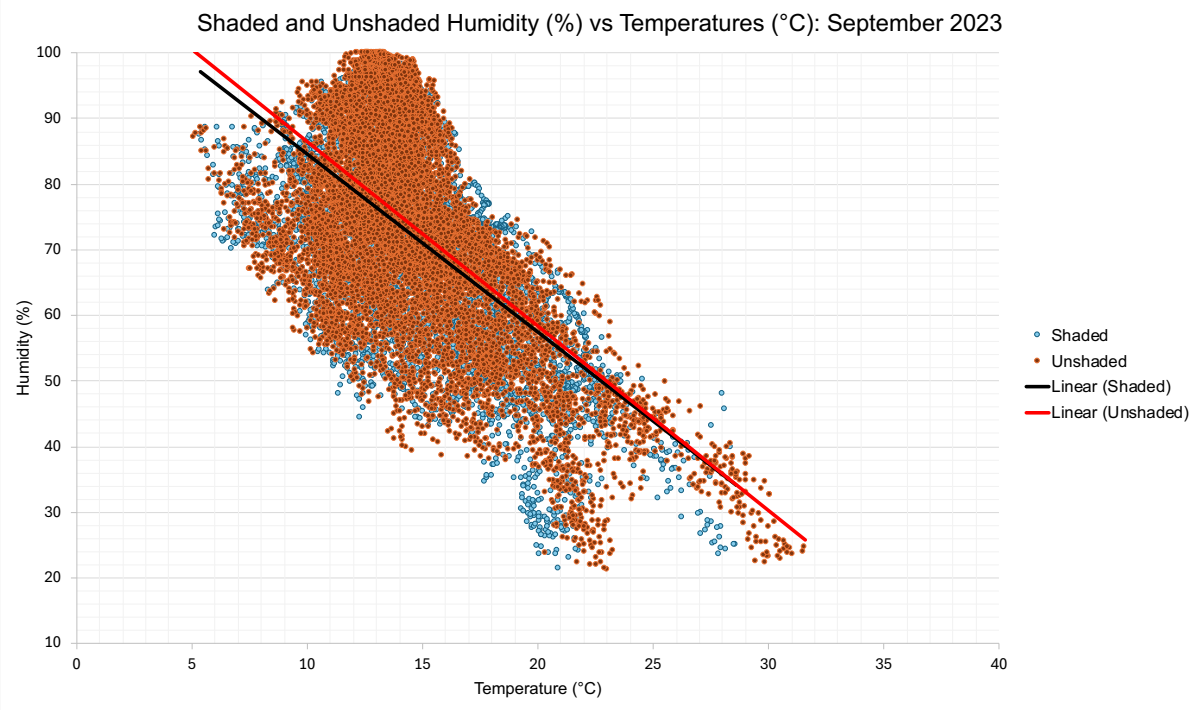


Appendix E-7: Unshaded-5 vs Shaded-5 average temperature readings (°C) – 27th December 2023

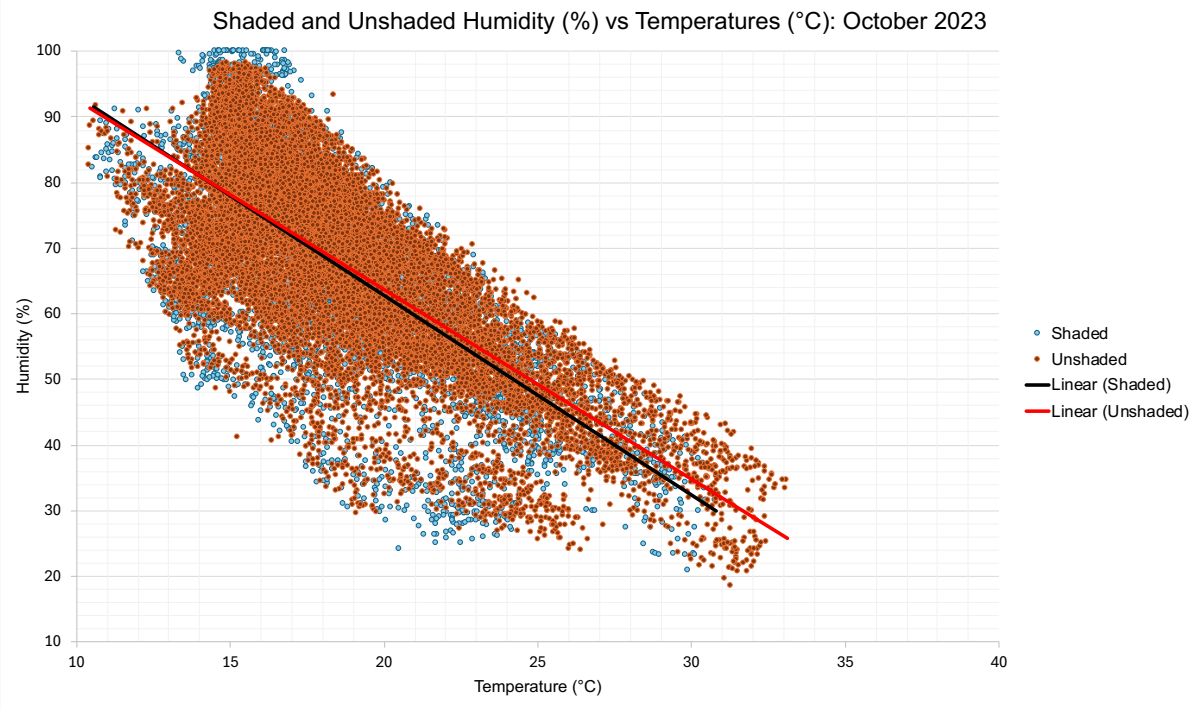


Appendix F: Humidity vs Temperature graphs for October, November and December

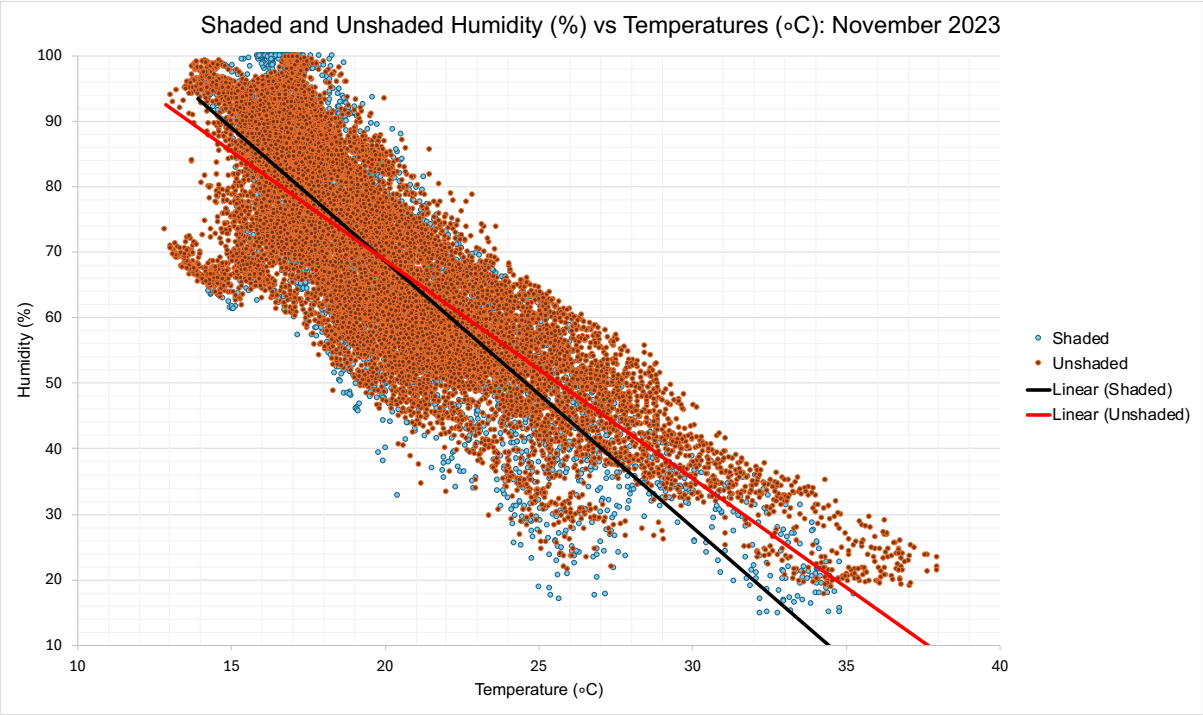
Appendix F-1: Shaded and unshaded humidity (%) vs temperature (ºC) – September



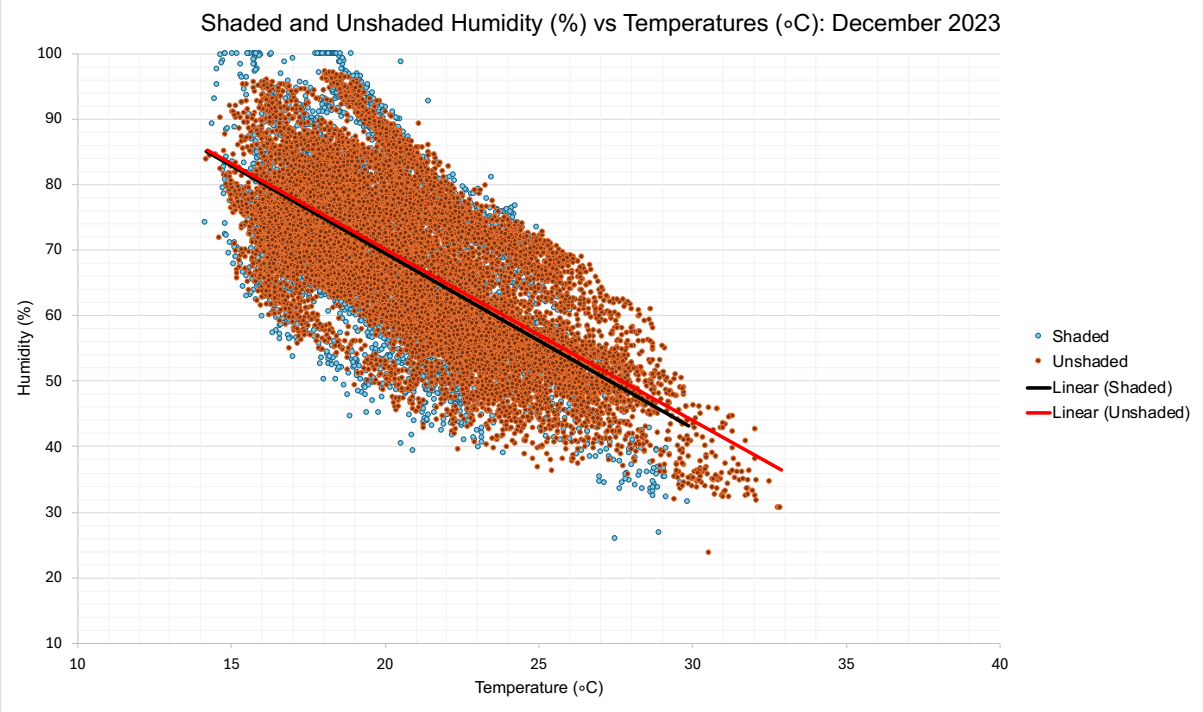
Appendix F-2: Shaded and unshaded humidity (%) vs temperature (ºC) – October



Appendix F-3: Shaded and unshaded humidity (%) vs temperature (ºC) – November



Appendix F-4: Shaded and unshaded humidity (%) vs temperature (ºC) – December



Appendix G: Monthly breakdown of data transmissions

Month	Day	Total transmissions
Sep	07-Sep	91
	08-Sep	399
	09-Sep	664
	10-Sep	668
	11-Sep	744
	12-Sep	922
	13-Sep	993
	14-Sep	946
	15-Sep	908
	16-Sep	916
	17-Sep	913
	18-Sep	853
	19-Sep	838
	20-Sep	827
	21-Sep	831
	22-Sep	835
	23-Sep	837
	24-Sep	822
	25-Sep	853
	26-Sep	822
	27-Sep	832
	28-Sep	837
	29-Sep	820
	30-Sep	782
Sub total		**18953**
Oct	01-Oct	647
	02-Oct	570
	03-Oct	577
	04-Oct	570
	05-Oct	494
	06-Oct	503
	07-Oct	488
	08-Oct	477
	09-Oct	533
	10-Oct	644
	11-Oct	717
	12-Oct	820
	13-Oct	819
	14-Oct	826
	15-Oct	814
	16-Oct	812
	17-Oct	830
	18-Oct	822

	19-Oct	813
	20-Oct	841
	21-Oct	807
	22-Oct	811
	23-Oct	833
	24-Oct	813
	25-Oct	835
	26-Oct	825
	27-Oct	822
	28-Oct	836
	29-Oct	821
	30-Oct	815
	31-Oct	754
Sub total		**22589**
Nov	01-Nov	657
	02-Nov	652
	03-Nov	598
	04-Nov	573
	05-Nov	573
	06-Nov	578
	07-Nov	570
	08-Nov	579
	09-Nov	571
	10-Nov	571
	11-Nov	582
	12-Nov	523
	13-Nov	558
	14-Nov	758
	15-Nov	740
	16-Nov	737
	17-Nov	755
	18-Nov	742
	19-Nov	749
	20-Nov	745
	21-Nov	742
	22-Nov	751
	23-Nov	743
	24-Nov	742
	25-Nov	744
	26-Nov	743
	27-Nov	740
	28-Nov	753
	29-Nov	745
	30-Nov	747
Sub total		**20261**
Dec	01-Dec	745

	02-Dec	741
	03-Dec	752
	04-Dec	745
	05-Dec	742
	06-Dec	753
	07-Dec	739
	08-Dec	719
	09-Dec	728
	10-Dec	708
	11-Dec	705
	12-Dec	728
	13-Dec	705
	14-Dec	717
	15-Dec	716
	16-Dec	706
	17-Dec	726
	18-Dec	702
	19-Dec	693
	20-Dec	722
	21-Dec	707
	22-Dec	698
	23-Dec	733
	24-Dec	706
	25-Dec	714
	26-Dec	723
	27-Dec	718
	28-Dec	716
	29-Dec	705
	30-Dec	714
	31-Dec	719
Sub total		**22345**
Grand Total		**84148**