[TBD: NAME OF TOOL]

TBD: A tool to study microclimates in an orchard

By

Adam Sidnell

Supervised by Professor Ruzanna Chitchyan

Department of Computer Science
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

September 2025

Word count: TBD

Executive summary

Dedication and acknowledgments

Author's declaration

Table of contents

List of Figures

1 Introduction

In this report, I give details of an online tool called [TBD: NAME OF TOOL] which was deployed alongside an IoT weather sensor network that I built myself. The tool aims to provide farmers with accessible, specific climate data to support real-time decision-making.

The IoT hardware deployed consisted of four separate custom built components: two field nodes capable of reading weather data and transmitting this using LoRa; a repeater that boosts received LoRa signals; and an internet-connected gateway that uploads weather data to an online database.

The nodes were placed at two different locations on the farm that the owner had identified as having distinct climate characteristics (see section on microclimates), meaning wider-area weather forecasts were unrepresentative.

Part I

Background

2 Microclimates

2.1 Microclimates in agriculture

A microclimate is generally understood as a set of distinct climatic conditions within a small, localised area [1]. The maximum size of a microclimate is debated, but the World Meteorological Organisation (WMO) regards it as occupying an area of anywhere from less than one metre across to several hundred metres [2]. In practice, microclimates can occur in spaces such as gardens, valleys, caves, or fields. Even human-made structures can generate their own microclimates; for example, tall buildings can create *street valleys* that reduce wind flow and lead to the formation of localised pockets of warmer air, which can also trap higher concentrations of pollution from vehicle emissions [3]. Vegetation plays a critical role in influencing microclimates. The addition of trees to an urban environment can reduce air temperature by as much as 2.8 °C [4].

This localised climatic variation, characteristic of microclimates, is therefore significant in agriculture. The climate that crops are exposed has an enormous impact on overall agricultural yields. Indeed, farmers have modified the microclimate of crop fields for millennia, a clear example of this being the use of fencing to reduce soil erosion and damage to edible plants [5]. Therefore, the relationship between microclimates and agriculture has been the subject of extensive research - particularly as climate change introduces new threats to food security.

2.2 Microclimates in apple orchards

3 *Internet of Things*

3.1 What is IoT?

The Internet of Things (IoT) refers to the concept of integrating networking capability in a range of devices, allowing for cooperation to reach common goals [6]. To this end, IoT is a very broad idea and there has been rapid ongoing growth of IoT in both domestic and business settings.

IoT therefore often involves the integration of wireless networking into objects which traditionally would not have had any such capability. As an example the use of smart lightbulbs, with the ability to control their status, brightness or even colour via a smartphone application, is a fairly standard demonstration of IoT. However

3.2 Examples of IoT in agriculture

3.3 IoT enabling technologies

4 LoRa

4.1 What is LoRa?

https://www.youtube.com/watch?v=jHWepP1ZWTk https://www.youtube.com/watch?v=T3dGLqZr
- see further reading on second one

LoRa stands for **Lo**ng **Ra**nge and it is a relatively modern radio modulation technique that allows for the transmission of data very long distances. Another comparable modulation technique is WiFi where LoRa has 4000 times the range (REFERENCE), with a theoretical maximum of 800km (although far less in realistic conditions).

This long distance can be achieved with remarkably little power, WiFi typically transmits data with 100mW-200mW of power for a range of 100-200m. LoRa meanwhile uses only 25mW of power.

4.2 How LoRa works

The preamble of a LoRa packet sends the same symbol multiple times in a row to allow the receiver to synchronise.

The most technically impressive aspect of LoRa is that it a receiver can demodulate signals below the noise floor. The noise floor is the sum of all unwanted signals in an environment. To use an example a stadium full of people talking will create a certain level of background noise, in order to speak to someone else in the stadium you would have to speak at a volume which can overcome this background noise. The fact that LoRa receivers can successfully demodulate below the noise floor is akin to being able to whisper in such a stadium and still be understood.

4.3 Benefits and limitations

Fresnell zone

4.4 Current applications of LoRa

5 Related work

5.1 Studies of IoT in agriculture

The use of IoT in agriculture has become widespread in recent years as it offers the opportunity for farmers to improve yields and cut costs by bringing digital solutions that would not have previously been viable without access to power and internet connectivity. The use of IoT in agriculture is often wrapped up in the moniker of "Smart Farming", which encompasses a range of digital, robotic, and internet-enabled approaches to improving efficiency.

A 2019 review by Farooq et al. [7] reveals the scope of IoT applications in agriculture. These include precision farming (soil quality, moisture, and weather monitoring), automated irrigation, pest and disease monitoring and prediction with machine learning, and even the deployment of agricultural drones for spraying, mapping and imaging.

5.2 Microclimate prediction using machine learning

There have been a number of recent studies where machine learning has been used to help predict micro climate conditions. A 2021 study by Kumar et al developed a machine learning framework that is able to predict a variety of climatic variables such as soil moisture, wind speed and temperature. They were able to get up to 90% accuracy with a 12-120 hour forecast range.

5.3 IoT weather stations on the market

Part II

Hardware development

6 Overview of hardware

7 Design

7.1 Nodes

7.1.1 Components

The first step in the design process was selecting hardware components that could operate autonomously without mains power. This required devices that were highly power-efficient, while also capable of transmitting small data packets via LoRa. An equally important consideration was ensuring the final devices could be fully weatherproofed to protect the sensitive electronics from water ingress and environmental damage.

Challenger RP2040 LoRa

The iLabs Challenger RP2040 LoRa is an embedded computer that uses the Raspberry Pi RP2040 chip that was released in 2021. The RP2040 itself is a low-cost and power-efficient processor with ample power to perform the data encoding and transmission in my use case. Additionally, the chip is extremely popular with over 10 million units being produced in the first two years of release [8]. This popularity means there is ample documentation for developing with this processor and it is compatible with circuit python which was my preferred language for development.

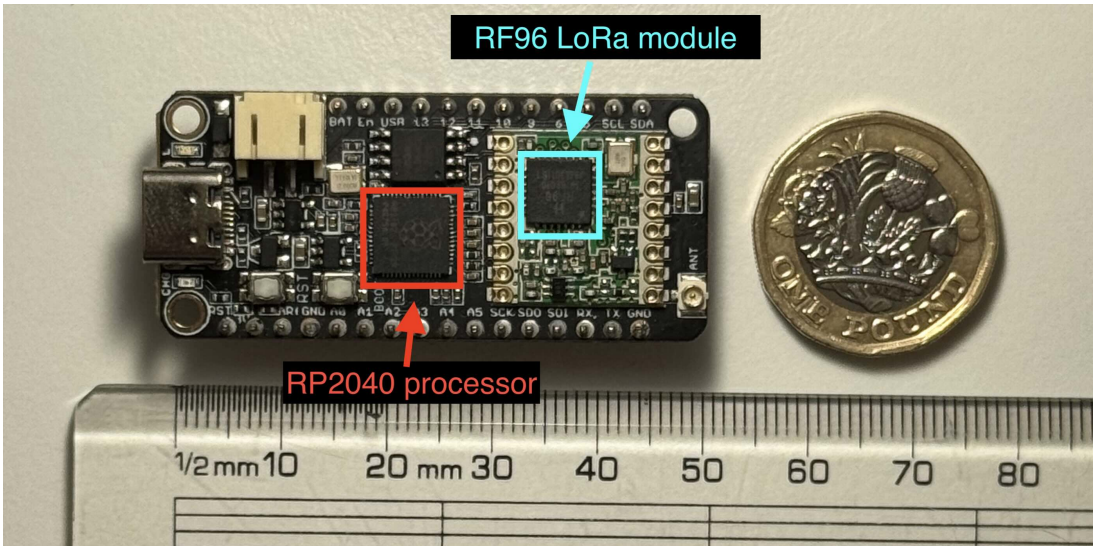

Figure 1: iLabs Challenger RP2040

The challenger board itself is well suited for this project for several reasons. It uses the compact Adafruit feather form factor, giving a board dimension of just 5cm by 2cm, making it easy to mount in a small enclosure. The onboard Hope RF96 LoRa modem

is built directly into the board and the U.FL antenna connector allows for the swapping of antenna's to different varieties. This board's LoRa module is also set to transmit at a frequency of 868mhz which is a standard UK frequency for LoRa and gives a good balance between range and bandwidth.

Another useful aspect of the board is the abundance of GPIO pins (20 in total) allowing for a large number of sensors to be fitted to the board.

Antennae

The selection of antennae is one of the largest determinants of range and reliability in the context of wireless communication systems [9]. Initially I used a simple PCB antenna as shown in Figure 2, however as explained in the next chapter, the range of this was insufficient for my use.

Figure 2: Low range PCB antenna

To improve overall range I switched to a more capable omnidirectional whip antenna that was made specifically for the Challenger RP2040. The antenna is tuned to perform best at the 868mhz frequency range - which is the range I was using.

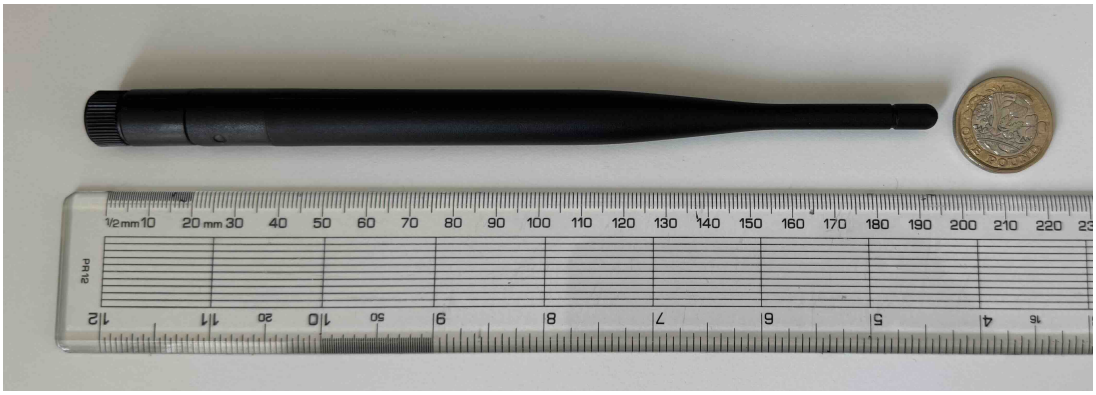

Figure 3: iLabs whip style LoRa antenna (868mhz)

Sensor selection

Temperature and humidity sensor

I used a DHT11 temperature/humidity sensor for each node to provide basic readings. It was chosen for it's low cost, availability and compatibility with both the RP2040 Chal-

lenger and CircuitPython (via libraries). The sensor can be connected to the microcontroller using a single GPIO pin as well as the usual power and ground pin.

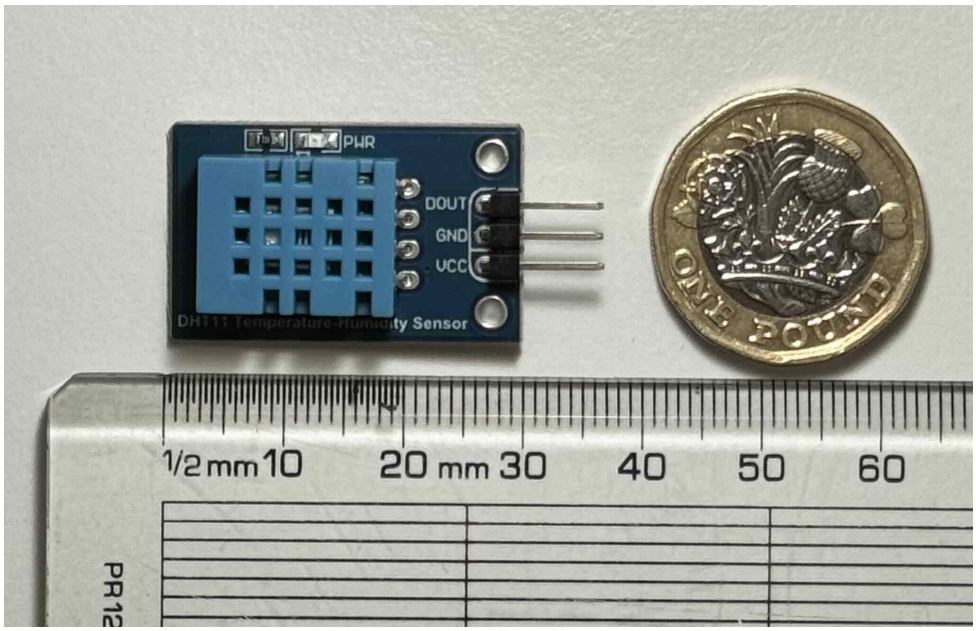

Figure 4: Waveshare DHT11 temperature/humidity sensor

Soil moisture sensor

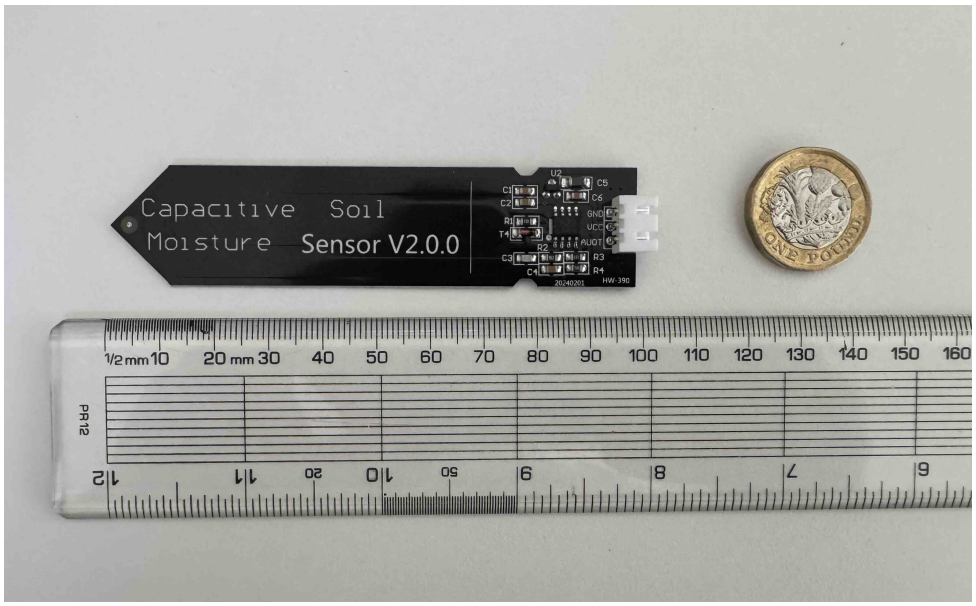

Figure 5: The Pi Hut capacitive soil moisture sensor

Wind speed sensor

Figure 6: DFROBOT wind speed sensor

Powering the node

To allow for continuous operation away from power sources, I set up a 5.5v Monocrystalline Silicon Solar Panel to each of the nodes. As the output from the solar panels was too high voltage to directly power the nodes I also installed a solar power management module onto the Challengers. This module powers a battery that is installed on it using a solar panel. It then is able to power the RP2040 using this battery, allowing for overnight or overcast usage of the challenger.

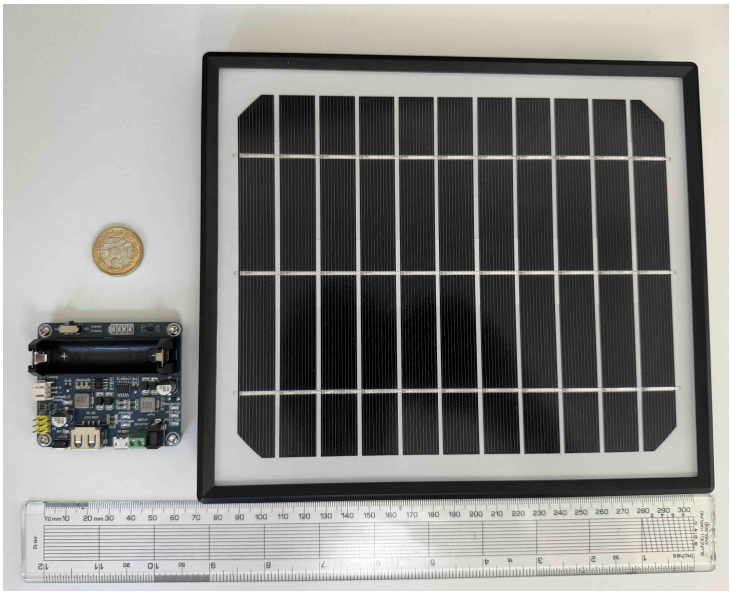

Figure 7: Waveshare solar power management module (left), solar panel (right)

Weather proofing

I 3D printed a water proof enclosure and a separate Stevenson weather shield for the temperature/humidity sensor to allow for accurate outdoor readings.

7.1.2 Programming the nodes

On each challenger I flashed the drives with CircuitPython, an open source interpretted language similar to Python but simplified for use in microcontrollers.

7.1.3 Repeater node

To improve range I made one of the four RP2040 challengers I used act as a repeater. This meant it did not require sensors like I used for the other two nodes and instead it just had a solar panel and battery connected up to it. The repeater would receive signals from the two nodes before relaying them to the final gateway. This had the effect of significantly boosting the range.

7.2 Gateway

The gateway consisted of a fourth Challenger RP2040 connected to a Raspberry Pi (TBD version). The challenger only received messages from the repeater and then would transfer these to the Raspberry Pi. After receiving data the Raspberry pi uploaded this to the internet.

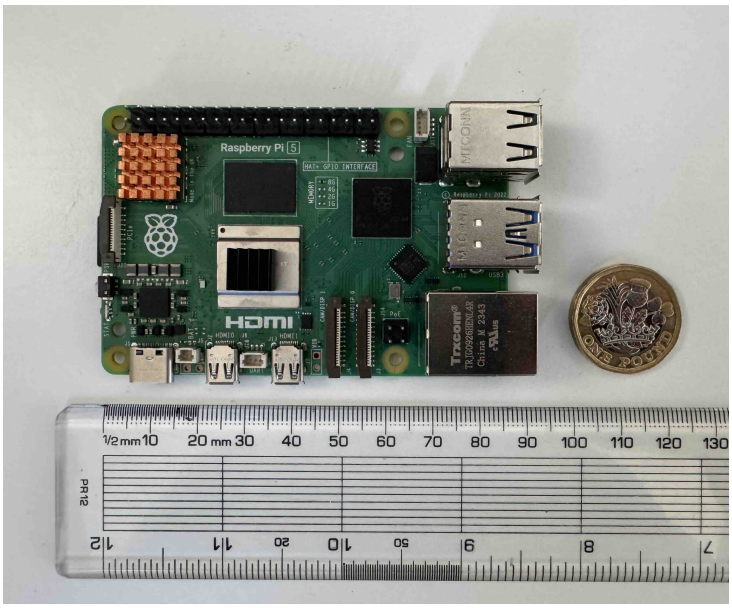

Figure 8: Raspberry Pi 5 (8GB)

7.3 LoRa settings

LoRa has many parameters that must be aligned for two devices to communicate successfully. These include:

1. **Spreading factor:**

2. **Frequency:** Must match across devices; this project uses 868 MHz, the UK LoRa ISM band.
3. **Bandwidth:** Determines how wide the signal is, I am using 125 kHz.
4. **Coding rate:**
5. **Transmit power:** Affects the power and therefore range. This must be set within legal limits (e.g., 14 dBm in the UK).

7.3.1 Compliance with regulatory limits on radio power

The UK has strict regulations on the usage of radio transmitters under the Ofcom ISM band rules. For the 868mhz band, the maximum effective radiated power that can be used is 25mW. This corresponds to a transmit power of roughly 14dB.

Additionally, the UK has rules on duty cycle rates. This is essentially how long radio signals are permitted to be on air. For example at a spreading factor of 12 a message may take approximately 1 second to send. The duty cycle limit in the UK is 1%, meaning you may only transmit for 1% of the time on a given day. 1% of a day is 864 seconds. This means to stay in line with UK regulations only 864 messages could be sent on a given day - or roughly 1 message every 2 minutes.

8 Development and testing

8.1 Range tests

One of the most important tests to carry out prior to deployment in the field was testing the range of the devices.

In this experiment I took four challenger RP2040s to The Downs, a large public park in Bristol. Here I tested four different antenna configurations to compare how well the signal travelled across an increasing distance. Signal strength can be measured using the received signal strength index (RSSI), a measure of the difference in signal from the transmitter to the receiver and measured in decibels (CHECK THIS).

For the test, two challengers were programmed as transmitters, sending an example data packet similar to the data that would eventually be used at the farm. One of the transmitters used a simple PCB antenna (Figure 2) while the other used a higher range whip style antenna (Figure 3). Then I programmed two challengers as receivers, again one had a low range antenna and the other a long range one.

This meant that four different antenna configurations could be tested concurrently, as the receivers could pick up the signal from each transmitter. An estimate of their estimated relative performance before testing is shown below:

1. Whip antenna to whip antenna (Likely best result)
2. PCB antenna to whip antenna
3. Whip antenna to PCB antenna
4. PCB antenna to PCB antenna (Likely poorest result)

The reason PCB transmitter to whip receiver will likely outperform the whip to PCB configuration is that generally it is more important that the receiver has better "hearing" capabilities than the transmitter can "shout".

Nine different distances from transmitter to receiver were tested, in 200m increments starting from 0m as a baseline to a distance of 1600m (Figure 9). An important aspect in getting a successful LoRa connection is whether there is line of sight between the transmitter and receiver. Figure 11 shows the elevation profile of the test area, with the initial large dip being an inaccuracy from google earth's topology data as the 0m point is near a cliff edge. The lowest elevation point is at the starting point at around 83m above sea level, while the highest point was at the 1km mark at around 94m making an elevation range of 11m. My hypothesis was that signal would likely drop off or stop entirely beyond this point as points beyond 1000m would be below the hill line. This would effectively mean that the receivers would be in a signal shadow point where the

transmission waves would not be able to reach them. The only possibility for signal to reach this area would be from reflections either from nearby buildings or topology. However the effects of reflections are virtually impossible to account for in the real world and therefore no accurate predictions could be made.

Figure 9: Google earth image of data collection points

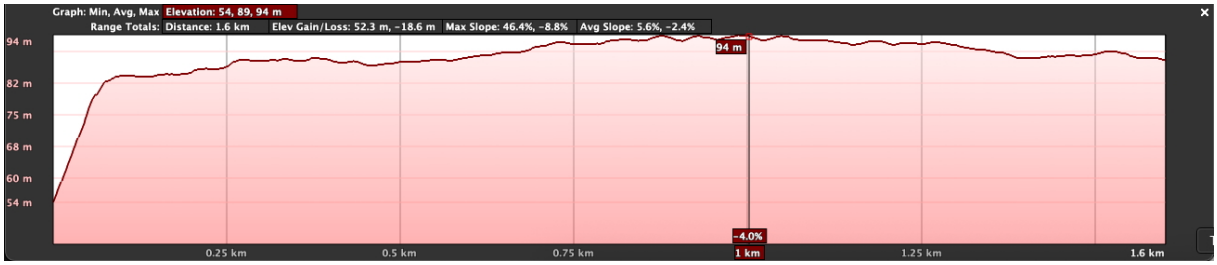

Figure 10: Elevation profile of test area (ignore large dip at start)

At the time of the actual test however a festival was being run between the 1200m and 1400m mark. The festival had a number of large tents and temporary buildings constructed which almost certainly blocked the signal. Therefore tests past this point should be heavily caveated. It is likely this contributed to the lack of signal past this point. Final data below:

Figure 11: Graph to show signal loss from different receiver and transmitter configurations (higher is better)

8.2 Battery tests

There is no mains electricity available in the apple field so the nodes must be able to operate without external power source. Even with the use of a solar panel the node must be able to work through the night and during periods of dense cloud coverage where the solar panel will not have sufficient power to keep the node running. This is where the use of a battery is particularly useful as the solar panel can charge the battery with excess energy during periods of excess solar radiation, such as during midday hours, and then store this energy for periods of low or no solar radiation.

However, while daylight will be of little concern during the summer months when this dissertation is being written, there will be far fewer days of usable sunlight over the winter period. As the UK has very high latitude, there is large seasonal variation in the length of a day. For the town of crediton (nearest town to Small Brook Farm), in the summer there are 16.5 hours of daylight while in winter it receives only 7.6 hours; meaning a night that is 16.4 hours long. Therefore the node must have a battery sufficiently large to power it for a minimum of 16.4 hours with some additional charge to account for high cloud cover during the early evening and morning period.

To see if the battery solution was sufficient for this environment a test was run on a fully charged 18650 battery that was then connected to the solar power manager to allow the voltage to be regulated up to the challenger's 5v input voltage.

A digital multimeter was installed between the solar power manager and the challenger's usb c input to view the voltage and current in real time as well as running a timer for the test and a calculation of the number of watt-hours consumed by the node. The node was then run with a full suite of sensors attached. During the test the node used 80ma at 5V for a wattage of 0.4W.

The node ran until the battery would no longer discharge, with the final battery life of the node being 19 hours and 17 minutes. The meter showed that the node consumed 7.96Wh of power. Considering the battery capacity is 9 Wh it may seem that the battery ran out too quickly. However, the solar power manager documentation tells us that the battery boost efficiency - being the conversion of battery voltage from native 3.7V to 5V output - is only 86%. With that in mind effective battery capacity is roughly 7.74Wh, which is very similar to the actual result.

The achieved result of 19 hours should be sufficient for running the node continuously for much of the year as this compares to the longest night period of 16.4 hours. However, during periods of heavy cloud cover in winter there is a chance the node would not be able to charge the battery sufficiently in the day to allow the node to run over night.

8.3 Solar panel testing

The solar panel is rated for 6W maximum power. This however would only realistically be achieved under optimal conditions with a clear sky and full sun around midday. This power rating was first verified using a multimeter on such a day where a voltage of (INSERT VOLTAGE) and amperage of (INSERT AMPERAGE) was measured for a wattage of (INSERT WATTAGE).

As explained in the battery section above, the node consumed about 7.96wH over a 19 hour and 17 minute period. We can therefore determine that the node requires roughly 10wH of energy per day to able to run continuously. It would be tempting to then say that the node would need less 2 hours of ideal sunlight to operate for an entire day (being 6W * 2h = 12Wh), however we must also account for the energy loss when converting from raw solar output to regulated 5V input that challenger accepts.

The solar panel manager rates it's conversion efficiency for solar at 78% meaning for each watt of solar energy received 22% of this will be wasted as heat when converted to the correct voltage. If we adjust for this loss we would need effectively 12.8Wh of energy just to power the node for a day, being 2 hours and 8 minutes of ideal conditions.

An additional 9Wh is needed to charge the battery which if we adjust again for solar efficiency factor we get 11.5Wh. This is an additional 1h:55m of ideal sun, leaving a total sunlight requirement each day of 4h:03m.

8.4 Hardware assembly and weatherproofing

Since most of the hardware in this project contains exposed electronics, the final assembled devices needed to be made wind and water resistant to prevent failure. However the sensors and solar panel would also need to be exposed to perform their function effectively - e.g. a solar panel must have a clear view of the sun throughout the day. The final design therefore needed to balance multiple conflicting purposes:

- The core electronics (challenger, solar manager, battery etc) must be kept dry, cool and away from wind which may damage electronics.
- The wind sensor must be exposed to the elements and securely fastened to withstand intense wind, it must also be kept high off the ground for more accurate readings.
- The solar panel must be exposed to the elements and be south facing at at an angle of 30-40 degrees to optimise solar efficiency.
- The soil moisture sensor must be inserted into the ground and as it has exposed electronics must be made waterproof.
- The antenna of the challenger must be placed as high as possible (at least 1.5m from the ground).
- The temperature/humidity sensor must be exposed to the elements to ensure accurate readings but cannot be exposed to rain due to the exposed electronics on it.

Due to the conflicting nature of some of these needs. The final device would need to allow for positioning of different components at different heights.

The final design therefore was to place most of the sensitive electronics inside a waterproof box with cut-outs for the to allow for external component wires. This could then be mounted onto a 2m pole to allow for a good antenna signal and more accurate wind speed readings.

8.4.1 Sensitive electronics

An IP65 waterproof junction box was selected to house the non-sensor portion of the electronics. An IP rating measures the Ingress Protection of a devices housing against water and dust defined by the International Electrotechnical Commission. The first number of an IP code is a measure of dust resistance while the second number is a measure of water resistance. An IP code of 65 therefore means the box used here has perfect dust resistance and can withstand water jets being sprayed at it from multiple directions [10]. This level of protection should be more than adequate to withstand the heavy rainfall it will experience.

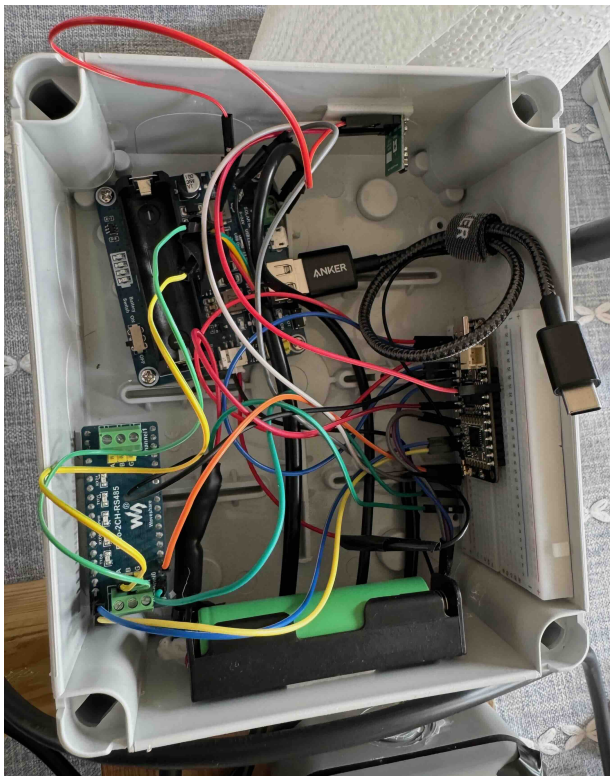

Figure 12: Open junction box showing internal wiring

The challenger microcontroller was inserted onto a half size breadboard and then stuck to one side of the box. The antenna cable was routed to the top right corner where a small hole was drilled to allow for the antenna to be mounted externally. All the other components were then wired into the breadboard and stuck down with double sided sticky pads to prevent damage during transport or heavy winds. The bottom of the box then had holes drilled for the anemometer, soil moisture sensor, solar panel and temperature/humidity sensor respectively. To prevent water ingress into these holes, sealant was deposited over the holes.

The entire box was then attached to a plank of wood measuring roughly 25cm by 60cm. This board holds all the sensors with the exception of the soil moisture sensor which must be able to hang freely.

8.4.2 Soil moisture sensor

The sensor that required the most thought into waterproofing and mounting was the capacitative moisture sensor. The first challenge was the need for the soil moisture sensor to be able to reach ground level while at the same time the microcontroller collecting it's readings was secured 2m above it. The solution to this was to swap the small included wires with a 2.5m weatherproof cable containing three cores (one for power, one for ground and one for data). Each cable core was then soldered onto the sensor as in figure 13, allowing for the sensor to be inserted into the ground.

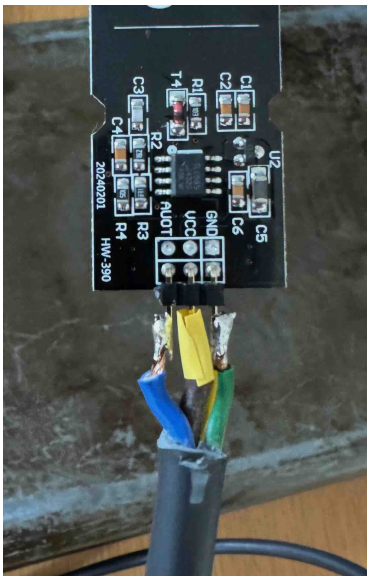

Figure 13: Soldered soil moisture sensor

Waterproofing the sensor was necessary as while the bottom three quarters of the board are designed to be inserted into soil, the part above this is made up of exposed electronics that cannot contact water (Refer to figure 5). This means this part of the board would be directly exposed to the elements unless waterproofing is applied

To waterproof this sensor I followed an online tutorial on a hobbyist website [11]. The method involves painting the electronics using nail varnish, while this sounds unconventional, nail varnish is a non-conductive compound that can be easily painted over electronics with the included brush. It is therefore quite a popular low-cost method for waterproofing electronics. After this is applied the portion of the board with the electronics is covered in heatshrink to form a tight seal over the board, with a layer of nail polish on the edges to reduces the chance of water ingress under the heat shrink's plastic.

ADD FIGURE HERE

8.4.3 Other external components

The DHT11 sensor was mounted inside a smaller IP55-rated junction box, which provides the same resistance against rain as the larger box. To allow for more accurate readings, small holes were drilled on the bottom panel of this so the air temperature inside the box would better match the external temperature/humidity. To further improve the sensor's accuracy, the box was painted white to reflect sunlight, and a solar shield made from a cut and reshaped aluminium can was mounted to the south facing side. This shield helps to prevent direct sunlight exposure to the sensor, reducing temperature distortion while still allowing airflow from the sides.

The solar panel was mounted below the main enclosure using the included adjustable gimbal mount. It was fixed at an angle of approximately 40° from horizontal, which balances solar efficiency throughout the year. This angle is steeper than the optimal summer setting but allows for improved performance during winter months when the sun is lower in the sky. The gimble allows for rotation in all planes which is useful in case the box itself cannot be mounted in a perfect south direction.

The anemometer was secured to a separate length of wood, positioned approximately 0.5 metres away from the main unit. This separation reduces turbulence caused by the box and pole, leading to more accurate wind speed readings. The sensor was screwed directly into the wood and positioned at a height of 2m.

ADD final assembled box on a pole with annotations

8.4.4 Gateway

The gateway node required less careful engineering as there was no requirement for it to be waterproofed or mounted externally. As this would eventually be placed inside a persons private home, the aim was therefore to make a minimally imposing and so a design similar to an internet router was chosen.

The two devices inside the box would be the raspberry pi used to upload data and the challenger which received incoming LoRa packets from the repeater. A plastic box was used with cut outs for the pi's outputs and power cable, with an additional cut out on the lid for the challenger's antenna. The pi and challenger were then secured to the box's base with velcro tape, as both devices would potentially need to be removed for any trouble shooting.

9 Deployment

9.1 Test deployment at home

Before setting up the nodes at the farm it was important to run the equipment in an area where repairs and updates could be more easily performed. The nodes were therefore placed at different locations in my garden and ran continuosly for a period of (ADD PERIOD).

9.1.1 Challenges and solutions from test deployment

One of the first challenges from the test deployment was the fact the batteries on the nodes was not quite sufficient to allow 24/7 readings. On day 4 of the run, after an overcast evening the night before, one of the nodes stopped transmitting at around 6am - just before sunrise. To remedy this another battery was fitted to each of the end nodes (but not the repeater as this had much lower battery consumption from the lack of any sensors). This doubling of battery capacity would make loss of power during the night much less likely. The fact the solar panels could charge the battery fully by mid morning pointed to the fact that solar capacity was sufficient but battery capacity was not.

A related problem was that when the node died it did not start repeating again after this despite solar power returning and the battery getting recharged. I discovered that when the microcontroller detected brownout - being a sudden dip in voltage - the device would enter a safeboot mode. In this safeboot mode the default code.py file would not automatically be run and instead the device would create and use a safemode.py file that without modification would essentially run infinitely unless the reset button on the device was pressed or the device was powered off and on again.

Clearly this behaviour was not appropriate for field deployment where brownouts would potentially occur whenever the battery was discharged completely. Even with the additional battery capacity provided by a second battery there would still be a high chance of this occurring on particularly dark days during winter, where continuous uptime could not be guaranteed.

To remedy this, I modified the safemode.py file using a template I found in the circuitpython documentation. This version of the safemode.py file was made specifically for the case I was using i.e. remote solar powered projects where manual reset of the board could not be achieved easily. Instead of entering an infinite loop this safemode would (DESCRIPTION OF SAFEMODE).

A seperate issue was the behaviour of the temperature-humidity sensor when exposed to direct sunlight. While the junction box used to encase the sensor was painted white to reduce any heat transfer from solar radiation, the readings taken on very sunny days showed

a large delta of around 5 celsius between the sensor readings and local air temperature readings that could not be explained with the existence of a microclimate.

To reduce this effect an additional solar shield was made to surround the sensor. Consisting a wooden frame and aluminium shield to block light falling on the junction box. While this method did help to reduce the delta at midday to around 3 celsius there was still clearly quite a pronounced effect. So to further reduce any warming the solar panel was mounted to the rear of the node. This would allow the temperature sensor to face north instead of south which in the northern hemisphere would result in much reduced solar warming to the sensors box as the sun tracks over teh southern area of the sky in the UK.

9.2 Deployment in the field

9.3 Placement of nodes

Part III

Software development

10 Programming the hardware

Section on programming: 1) The nodes, 2) the repeater, 3) The gateway

The challenger boards can be programmed in a few different languages. The most common of these are C (including C++) and python. I decided to program the boards in python due to the large availability of sensor libraries written in this language. Programming in C, while potentially more efficient, would likely have meant I would need to write my own libraries and functions to get many of the sensors to function properly and therefore was not selected.

The RP2040 processor that the challenger uses relatively low power, so the versions of python for the chip tend to be stripped down to accommodate this. The two main python versions available are circuit python and micropython. They both have extensive libraries and work with all the sensors I have selected. With no major difference in their functionality, I settled on circuitpython as this was the version recommended by iLabs (the makers of the challenger board) and included a library that supports challenger as well as example code.

10.1 LoRa settings

The most fundamental part of the program for each of the nodes in the LoRa network was that they could communicate with each other. This required the careful matching of LoRa configuration settings between them.

ADD LoRa settings

10.2 Sensor nodes

10.3 Repeater

10.4 Gateway

10.5 Remote diagnostics and error handling

Since all the sensors would not be accessible easily errors needed to be reported remotely to diagnose faults and problems. Unfortunately, while the sensor nodes and repeater are technically network connected with LoRa there is no easy way to send firmware updates over the air without developing a program to accept and create new files over LoRa (FACT CHECK THAT), which was outside the scope of a three month dissertation. Therefore instead the nodes would need to have robust error handling written into their programs.

Diagnostics on the gateway node was luckily much simpler as this was powered on 24/7 and network connected. (EXPLANATION OF TAILWIND AND RVC VIEWER)

11 Overview of software design

12 Backend: Database and SQL API

For the database, I chose Render as a good low cost option for hosting the web service as it offers a free trial period, constant uptime and backups. The free version of web services by Render (render.com), provides a limited set of resources at zero cost: 256 MB of RAM; 0.1 share of a CPU and 1 GB of storage. There are also limits on outbound bandwidth, build pipeline minutes and instance hours. A single instance of a POSTGRES database is available on this service - which is free for the first 30 days and then costs $5 a month. These resources should be sufficient for this project as only small amounts of data are involved.

The web service is running an API I have written, which will receive data from the raspberry pi used as the gateway. The data is sent using a POST request with a JSON body which is parsed, processed and added to the database by the API. Data can be retrieved for display by the frontend using a GET request.

The database design is very simple with a single table holding all the data so the choice of database management system is not crucial. Having said that, POSTGRES is a well established and reliable relational database management system that complies with ANSI SQL standards and most importantly for this project is supported by Render.

The 1 GB storage limitation allows for (x) rows of data with the current database design.

This solution should scale well. Additional tables can be added to the POSTGRES data base for additional sites. And in the unlikely event that the volumes exceed the Render limits it is easy to pay more for additional resources.

13 Frontend: Agriscanner webapp

Overview of front-end

14 Forecasting with machine learning

Part IV

Evaluation

References

[1] Met Office, "Factsheet 14: Microclimates," Met Office, Tech. Rep., 2023, Accessed: 16 June 2025. [Online]. Available: https://www.metoffice.gov.uk/binaries/content/assets/metofficegovuk/pdf/research/library-and-archive/library/publications/factsheets/factsheet_14-microclimates_2023.pdf.

[2] World Meteorological Organization (WMO), *Guide to Instruments and Methods of Observation: Volume III – Observing Systems*, 2024 edition. Geneva: World Meteorological Organization (WMO), 2025, ISBN: 978-92-63-10008-5. DOI: 10.59327/WMO/CIMO/3. [Online]. Available: https://library.wmo.int/idurl/4/68661.

[3] S. Yang, L. L. Wang, T. Stathopoulos, and A. M. Marey, "Urban microclimate and its impact on built environment–a review," *Building and Environment*, vol. 238, p. 110 334, 2023.

[4] D. Lai, W. Liu, T. Gan, K. Liu, and Q. Chen, "A review of mitigating strategies to improve the thermal environment and thermal comfort in urban outdoor spaces," *Science of The Total Environment*, vol. 661, pp. 337–353, 2019, ISSN: 0048-9697. DOI: https://doi.org/10.1016/j.scitotenv.2019.01.062. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0048969719300683.

[5] H. A. Cleugh, "Effects of windbreaks on airflow, microclimates and crop yields," *Agroforestry Systems*, vol. 41, no. 1, pp. 55–84, 1998, ISSN: 1572-9680. DOI: 10.1023/A:1006019805109. [Online]. Available: https://doi.org/10.1023/A:1006019805109.

[6] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2010.05.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128610001568.

[7] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of iot in agriculture for the implementation of smart farming," *IEEE Access*, vol. 7, pp. 156 237–156 271, 2019. DOI: 10.1109/ACCESS.2019.2949703.

[8] L. Pounder. "Raspberry pi produced 10 million rp2040s in 2021, more pi stores likely." Accessed: 02/07/25. (Jan. 11, 2023), [Online]. Available: https://www.tomshardware.com/news/raspberry-pi-10-million-rp2040s.

[9] A. Q. Khan, M. Riaz, and A. Bilal, "Various types of antenna with respect to their applications: A review," *International Journal of Multidisciplinary Sciences and Engineering*, vol. 7, no. 3, pp. 1–8, Mar. 2016, ISSN: 2045-7057. [Online]. Available: https://www.ijmse.org/Volume7/Issue3/paper1.pdf.

[10] Wikipedia contributors, *IP code*, [Online: accessed 11-August-2025]. [Online]. Available: https://en.wikipedia.org/wiki/IP_code.

[11] micromet, *Waterproofing a Capacitance Soil Moisture Sensor*, [Online; accessed 11-Aug-2025]. [Online]. Available: https://www.instructables.com/Waterproofing-a-Capacitance-Soil-Moisture-Sensor/.

Part V

Appendices

A Example