**Contents**

Intro; Page 3

Analysis of Brief; Page 4-6

First Ideas; Page 7

Jobs to be Completed; Page 8

Processes Each Week; Page 9-21

Our Code; Page 22-24

Analysis of Collected Data; Page 25-29

Research; Page 30-34

What went well/What we can improve; Page 34

Reflections; Page 35

Research Links; Page 36-38

**Intro**

We have decided to take on the Arduino project; to create a weather station. This project diary will detail our thought processes and progress throughout our six weeks on this project.

Our team consists of;

* Jordan Ahern who will write in red for personal comments in this diary
* Cian Turner who will write in blue for personal comments in this diary
* Sergio Pop who will write in yellow for personal comments in this diary
* Comments in black have been discussed by the group and recorded by Jordan

**Analysis of Brief**

**Arduino/Raspberry Pi – Python/C**

In October 2017, Ex Hurricane Ophelia tore across Ireland, leading to the death of 3 people and the worst storm in 50 years. Students are tasked to create a model weather station that can accurately record environmental conditions. Each model should be capable of recording 3 distinct measurements. The weather station should be constructed in such a way that it can be left outdoors.

Students will work in groups (max 3 per group) to achieve this project. Clear work from each group member is required. You will be required to keep a project diary outlining the process of creating the project, from conception to realisation. This diary needs to include all research conducted and the different stages that you went through creating the project. An evaluation by your peers will occur after the project and you will have a chance to reflect on your learning and work during the project. This should also be included in the diary.

**Breakdown**

* **Arduino/Raspberry Pi:** Miniature computers that can run programs and control external electronics.
* **Python/C:** Two different programming languages which can be used to program Arduinos and Raspberry Pis.
* **Model:** Must function correctly but is not full-sized.
* **Weather Station:** Main function is to record current, and likely future, environmental conditions.
* **Accurately:** Its records are correct (i.e. it will not say the weather is mild, in the middle of a storm).
* **3 Distinct Measurements:** Records three different variables that are not directly related.
* **Can be Left Outdoors:** It is water proof and will not blow away.

So, in short, this brief is asking us to use an Arduino or a Raspberry Pi to create a functioning miniature weather station which can correctly record at least three different variables and will not be damaged if left outside.

**Why we Chose this Brief and our Thoughts on the Other Options**

* **I chose this brief because I enjoyed working with Arduinos in class and felt that this project could challenge both my abilities in Coding and my previous knowledge of electronics from Technology.**

**In terms of the other briefs;**

* + **I believe the Scratch project would have become tedious eventually.**
  + **I don’t have any interest in creating a business for the HTML/CSS/JavaScript CBA.**
  + **I didn’t enjoy working with Unity as much as I have in other areas.**
  + **I was somewhat interested in the Blended Language option as I enjoyed working with both Python and websites in the past, but I felt that I would enjoy the Arduino project more overall.**
* **I chose the Arduino project for my Junior Certificate Project because when we worked with it as a class previously I really enjoyed it and had a lot of fun using them. I love the idea of building electronics and using code to make them operate. Other than that, here is my thoughts on the other projects:**
* **For my summer project last year, I done a scratch project that met the expectations for the scratch project this year, but I don’t think it would challenge my ability to do it again.**
* **The HTML/CSS/JavaScript didn’t catch my eye because I find making websites and coding plain text boring.**
* **When we done Unity as a class group I didn’t like the idea of making a game with an engine programme.**
* **In result of me joining late in 2nd year I did not get to work on Blended Language with the class. What I’ve heard about it from classmates It seems to be a good programme.**
* **I chose the Arduino project for my Junior Cert because I really enjoyed working with it and I had a lot of fun using it. I really like the idea of creating a weather station.**

**In terms of the other projects:**

* **I don’t think the scratch project would challenge my abilities.**
* **I didn’t enjoy doing the HTML/CSS/JAVASCRIPT as much as the Arduino.**
* **I didn’t like Unity because I don’t like making games and it is very difficult.**
* **I really enjoyed using Python because I found it easy, but I enjoyed the Arduino more.**

**First Ideas**

After choosing to complete the Arduino weather station we brainstormed some ideas;

Variables that we could record;

* Temperature
* Atmospheric Pressure
* Wind
* Rain
* Light level
* Anemometer

Forms of indication;

* A buzzer would alert anyone monitoring the system of a change but could be annoying to nearby members of the public
* A LED would not be annoying but would be less likely to grab anyone’s attention unless it was closely monitored
* A LCD screen could be programmed to give on the spot readings of recent changes but would need a buzzer or LED to tell someone to check it

Environmentally Sound;

* A plastic ice-cream carton or lunchbox was suggested by our teacher, Micheal Griffin, as it would be water proof and small sections could be easily cut to allow instruments to take external readings.

Data Collection;

* To collect the data, we could take the data from our weather station and send them to an excel spreadsheet which can be used to easily visualise collected data.

**Jobs to be Completed**

**Research (when we began much more was completed over time);**

* How to connect an Arduino to a computer wirelessly
* How to send data from an Arduino to Excel
* How to display information on an LCD

**Raspberry Pi (previously Arduino);**

* Connected to three working sensors
* Collected data is accurate
* Can connect to a computer wirelessly
* Uses a form of indication when certain factors are met
* Can be left outside

**Code;**

* Fully explained with comments

**Data;**

* Collected and clearly presented on graphs
* Analysed and compared to data from a national weather service (such as Met Éireann)

**Project Diary;**

* Fully completed
* Very clear and detailed (include research and thought processes)
* Many contributions from the entire team
* Include clear reflections
* Importantly; Team work

**Process Each Week**

**Week One;**

* Week one was focused on discussion and research, we chose the Arduino brief, we then analysed the brief and talked about our first ideas and jobs to be completed. I then researched ways to analyse collected data in excel and to display updates on a LCD.
* Week one was when we chose what project we were going to take on as a group and read what the expectations for exceptional grade were. After reading over them we discussed what sensors to use. I then personally researched how to connect the Arduino to the PC wirelessly.
* Week one we chose what project we were going to work on and we were discussing ideas. We assigned each person’s job and then I researched the code needed to connect an Arduino to pc wirelessly.

**Week Two;**

* In week two, we began testing different ideas and continued our research. I tested the LDR program which we had previously used in class as this would-be part of our final project and it allowed me to test the PLX-DAQ addon for Excel, which would allow us to easily plot data from our weather station, unfortunately this did not work for reasons I explain in the research section of this diary. Luckily, I was able to find another method of plotting our data which I will test next week. I then talked to Phillip, who teaches Metalwork and Technology, who is coordinating the Com-Sat competition in Kishoge, about good methods of wirelessly connecting an Arduino to a Computer, which I expand on in the research section. Finally, I went to Maplin in Blanchardstown, to look for extra sensors and components, unfortunately, they do not sell any components I was hoping to find in store.
* Unfortunately, I missed the first class in week two, but in the next class we continued to research, and we wrote why we chose the Arduino project and not the other projects in our diary. We started setting up the Arduino and try to connect to a pc wirelessly. We tried to find a way to plot our data from our weather station on excel but it didn’t work out. Finally, we looked for sensors online that we could use, and our teammate Jordan went Blanchardstown to look for extra sensors.
* During Week Two, we went on with our research and further into how we setup certain sensors. We tested out how to setup and the code for a LDR. We went on and done that without hassle due to it being easy enough. We also searched online for different sensors to buy but we couldn’t find any close to our area. Last thing we done was look up on how to connect the Arduino to the PC wirelessly, but all the websites said we need a WI-FI adapted shield to do so.

**Week Three;**

* This week we decided that we should focus on connecting the Arduino to the internet and connecting the LCD screen which we decided we would use to display information about the weather station. I primarily focused on connecting the Arduino to the WIFI using some methods which I researched over the weekend. I go into more detail about setting up the WiFi connection in the Research section. I also assisted Sergio and Cian with wiring the LCD screen until they had a hand of it themselves. Unfortunately, we ran into continues problems with the WIFI connection and in the end our teacher, Micheal, decided to take the Arduino home with him to try figure out what the problem was.
* In Week Three, I went on with the research of the LCD screen and in the end, I finally found a working circuit and code to set it up. We set it up as a team and managed to get it working completely but we didn’t get to display the results on the LCD screen. Our teacher, Micheal, could not understand what was not working so he decided to work on the screen in his own time. We also attempted to connect the Arduino to the WIFI but to do so we needed an external router.
* In week three, I had to research the code to set up the LCD screen. We set it up together and got it to work but the results from the sensors weren’t displaying on the screen. Our teacher Micheal decided to take it home and try to fix the issue. We also attempted to connect the Arduino to the WIFI, but I didn’t work because we needed an external router.

**Week Four;**

* I began this week working with Micheal to setup the WIFI connection. We spent an hour working on it but could not find a way to make it work for reasons I will go into in the research section. Micheal was also able to get the LCD screen to function correctly. Despite this Micheal told us that we might be more successful if we change to using a Raspberry Pi. The next day I discussed this with Cian and Sergio.

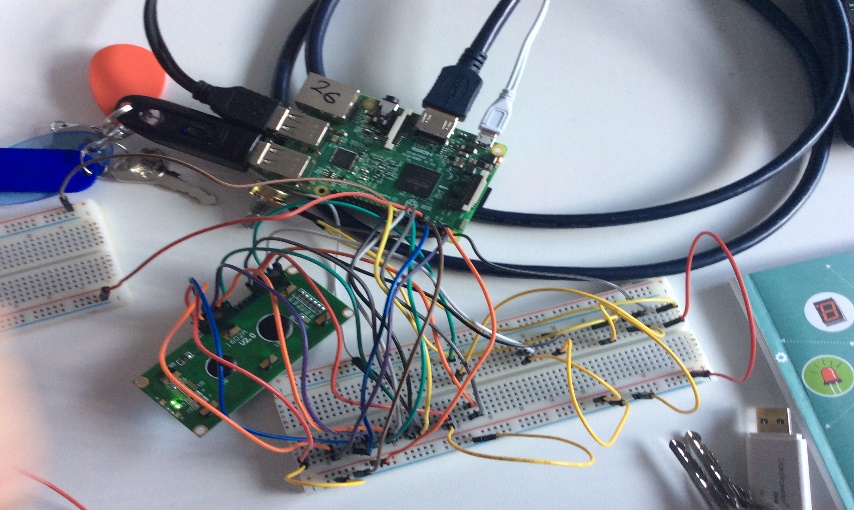
**Our thoughts on changing to a Raspberry Pi;**

We all agreed that the best decision we could make for our project would be to change to using a Raspberry Pi (specifically Raspberry Pi 3 Model B). The main advantage to this is that the Pi3 came with WIFI built in which will remove the need for an external router and will make connecting wirelessly much simpler. On the other side, this will render all work we have previously completed relatively useless. Also, we have never used a Raspberry Pi before and Cian has no experience with Python. Despite this, we have all agreed to put even more work in, to complete this project within our deadline, which we are fully competent we can achieve.

* Once we decided to use a Raspberry Pi we spent the rest of the week familiarising ourselves with the Raspberry Pi and researching new solutions to our tasks. I also found and tested a program for plotting our data named Gnu Plot. This seems to be a good way to graphically display our collected data.
* We started week four of with trying to get the WIFI to work with the Arduino but unfortunately it did not work. In the end we came to an agreement that we should switch to Raspberry Pi for easier remote access over the WIFI. None of us previously worked with Raspberry Pi and I’ve never worked with Python before which is the code used with this engine. Due to this we thought we would find it very difficult to work with but we decided to still go ahead with it and challenge ourselves.
* In week four we continued to try and get the Arduino to work with WIFI, but it didn’t work. We talked to our teacher Micheal what we should do and as a group we decided to use a raspberry pi which would be easier wo connect with the WIFI. None of us had worked with it so it was very challenging.

**Week Five;**

* I was out for much of the time this week, so I only had time to get the LCD screen working. The wiring looks very complicated but did work very well. We did have a slight bit of trouble with the code we were using were it was not displaying text but found this was just due to the brightness being too high which was easily fixed. I also gave Cian some quick help finding and using some components he needed to set up the LDR.



* This week Jordan was absent so me and Sergio had to work together without him. We researched shields for the Raspberry Pi and came across one that suited our project brilliantly. It was called the Sense Hat and it had a built-in sensors which went with our project. Thermometer, Barometer, Humidity Sensor, Gyroscope and Accelerometer were the sensors that were built into the Sense Hat. We did not require the Gyroscope and Accelerometer in our Weather Station, so they were not used but the other 3 came in very handy.

* This week me and Cian worked on setting up the LDR and wiring up the raspberry pi. Me and Cian came across a shield for the raspberry pi that was called the Sense Hat. It had five built in sensors, but we only needed three. A Barometer, thermometer and humidity sensor.

**Week Six;**

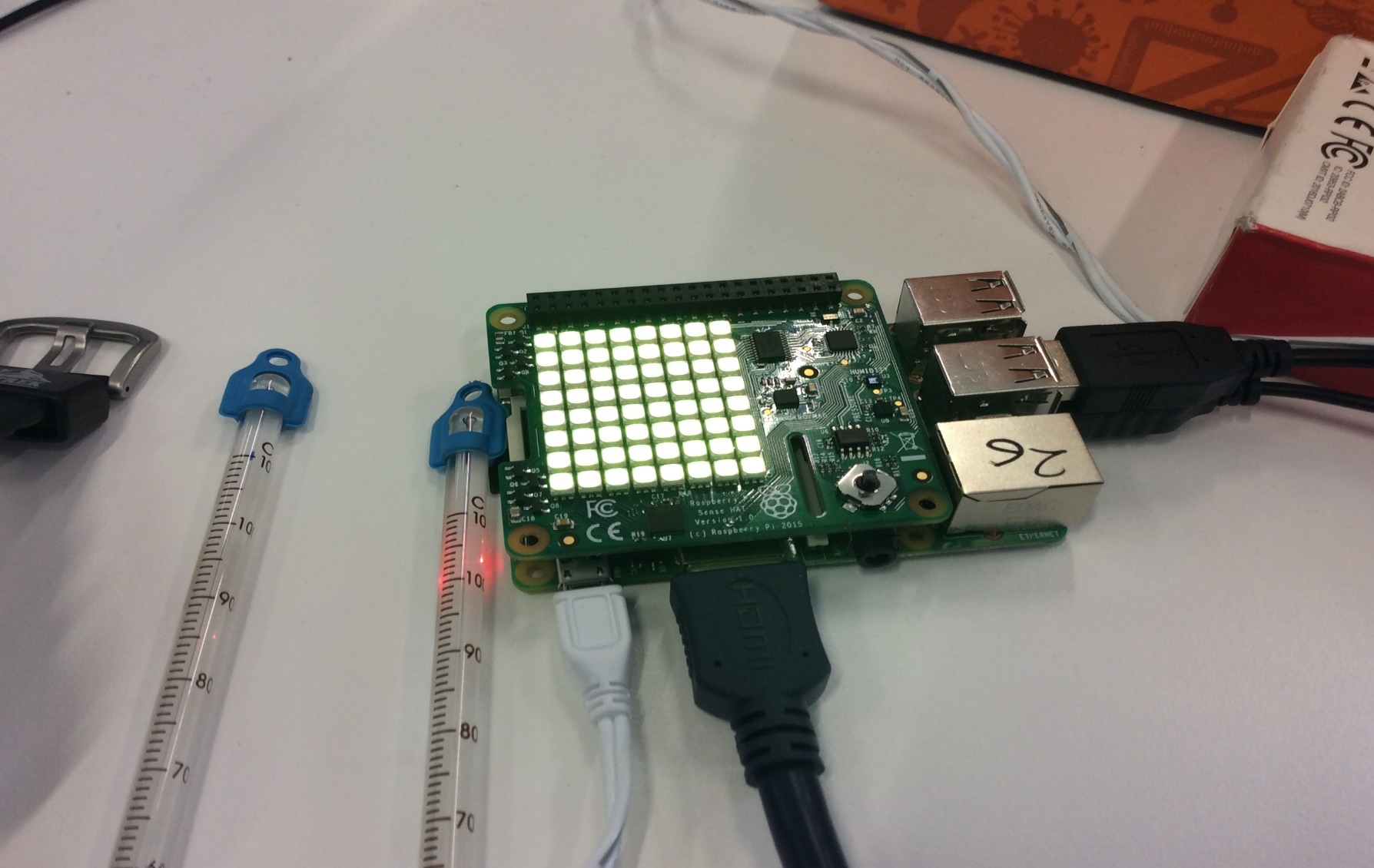
* After returning from my brief absence I received a pleasant surprise that Micheal had purchased us a shield for our Raspberry Pi named the Sense Hat. In my absence, I discovered, Cian and Sergio had found this shield online after running into many problems with our sensors that not even Micheal could figure out. The Sense Hat| will undoubtedly make our task easier and allow us to focus more on the code, rather then, attempting to wire all our sensors. I then spent the week coding the sensors (Thermometer, Barometer, and Humidity Sensor) in school and at home. I also added a time and date data point. I also adapted the code I used to create some data points to test Gnu Plot to place the data into a file. I then tested graphing the data with Gnu Plot but found it did not work for reasons that I cannot guess. Not to waste time I began testing Libre Office Calc. I found that the collected data was formatted in such a way as to allow me to simply copy and paste data into Calc without needing to separate it into boxes myself, which would have been a near imposable job. I also played with some different graphs, finding that a xy scatter graph works best. Interestingly, creating graphs is incredibly resource intensive for the Pi. To save time it may be easier to transfer our collected data to a PC and graph the data with Excel. I go into further detail about the Sense Hat and adding time in the research section.
* At the start of this week Jordan returned from his absence and we went back into class with all three of us. Micheal treated us to a little surprise when we came in. He bought us a shield for our project. It was called the Sense Hat. While Jordan was not in myself and Sergio done some research on this because the wiring was starting to become too complicated. The shield came with built in sensors (Thermometer, Barometer & Humidity Sensor) and allowed us to wire more to it with minimum amount of wires. Due to sensors being built in this allowed us to focus more so on the code than the wiring. The shield was working miracles for us. I also discover an app on our iPads called TeamViewer with allowed us to remotely control our PC and Raspberry Pi from the iPad itself.
* This week Jordan returned from his absence. Micheal had a surprise for us which would help us a lot with our project. He bought a shield for our raspberry pi called the Sense Hat. While Jordan was not in me and Cian came across this shield because the wiring was getting too complicated. The shield came with three built in sensors. A Barometer, thermometer and humidity sensor. This helped to focus more on our code rather than the wiring. We discovered an app on our iPads called TeamViewer which allowed us to remotely control raspberry pi and PC from our iPads.

**Week Six;**

* This was the week of our Christmas exams, so we ended up missing a double class and had less free time in general. Despite this I managed to comment all my code. I also began researching a way to externally connect the Sense Hat to the Raspberry Pi, which would free up some pins to use to add a LCD screen or a moisture sensor. I also began looking into ways to directly write our data into a LibreOffice Calc file and graph it automatically.
* In week six we didn’t get a chance to work on our project due to us having our Christmas exams. The only thing we got to do was research and try to connect the Sense Hat with wires but unfortunately it was applying too many short circuits to our Raspberry Pi.
* In week six we had our Christmas exams, so we didn’t get a chance to work on our project. The only thing we’ve done was researching and connect the Sense Hat with wires.

**Week Seven & Eight (Christmas Break);**

* I volunteered to take the Raspberry Pi home with me over Christmas, as I had some things I wanted to add and test if I had time. I first attempted to connect the Hat to the Pi with wires rather than slotting it on top in the general sense. Despite the time spent looking online I could not find any definite answers as to what GPIO pins are used so I experimented with what I found online. After one too many short circuits, I decided I would be wiser to program the Sense Hat to display information rather than chancing frying the Hat. I then attempted to use the python addon, pyoo, to send data to LibreOffice Calc and graph it. Sadly, when attempting to run this code I continuously received an error telling me that Python Uno, which pyoo required could not be found. I then attempted to find a solution online but, as far as I can see no one could solve this error. I then programmed the Heat Index calculation into the code. Finally, I programmed the Sense Hat LED screen to change colour based on the temperature.



* In week seven and eight we had our Christmas holidays, so we were not going to get a chance to work on our project, but Jordan volunteered to take it home and work on it there.
* In week seven and eight we had our Christmas exams, so we didn’t get a chance to work on our project, but Jordan took it home to work on it.

**Week Nine;**

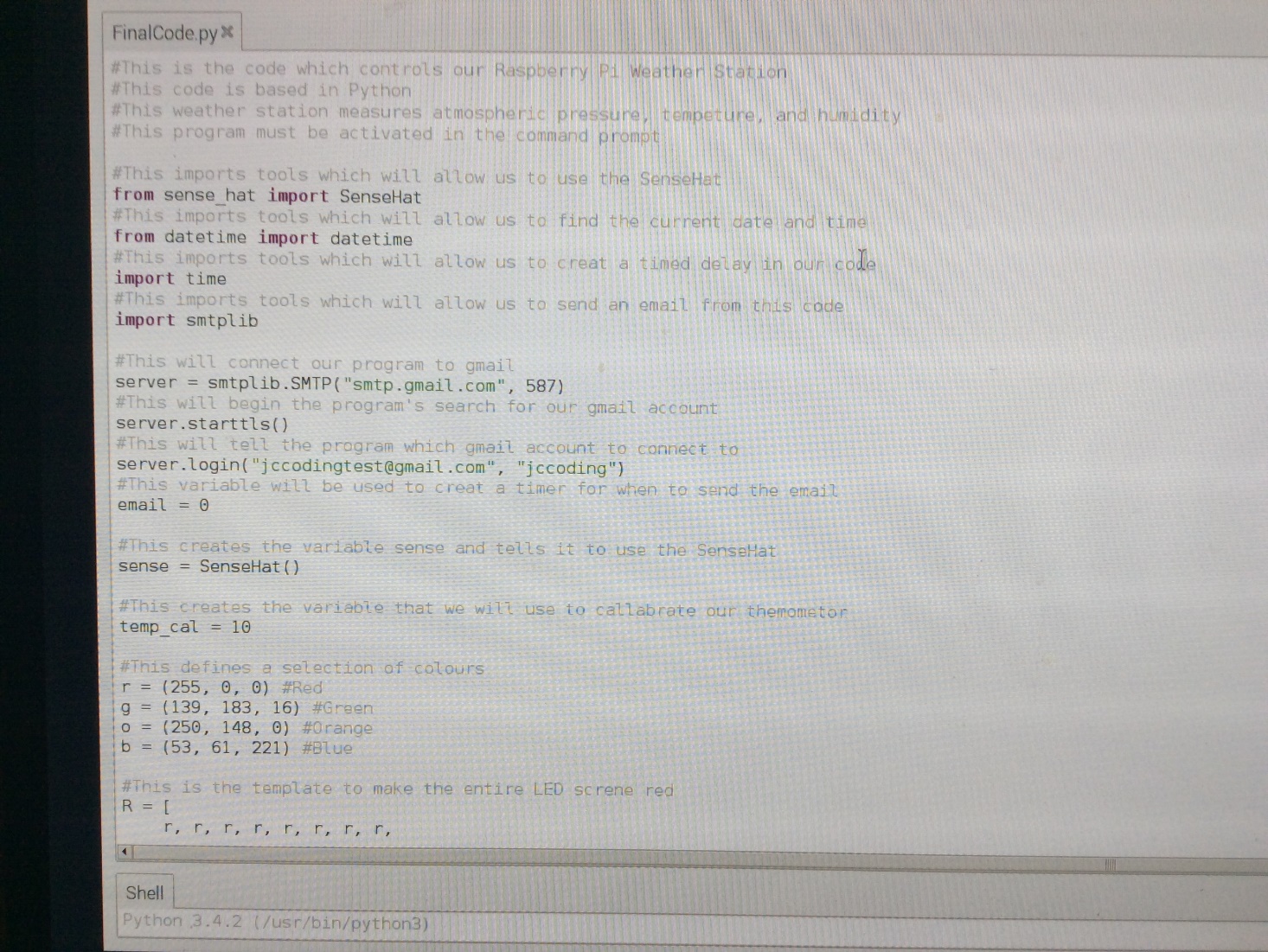
* I began this week by collecting data to recalibrate the temperature sensor with Cian. We decided to do this because we have previously observed that the Pi gets warmer over time, which was no accounted for in the flat -7C we previously used, due to the movement of electricity. We observed that when turned on the Pi there was a difference of roughly 10C, we also observed a constant increase of 0.2C per minute over a period of about 25 minutes, after which time the temperature appeared to stop rapidly increasing which I will account for with a while loop. To make this experiment as authentic as possible we placed the Pi in our box with the lid closed as much as possible (the small opening can account for the small ventilation hole we will have). Towards the end of the experiment, we observed a rapid increase on the thermometer we were using to calibrate the Pi, which the Pi did not register. We believe this was an experimental inaccuracy as the temperature increase was very dramatic, but we did not feel anything ourselves. This may have been caused by heat from the Pi spreading to the thermometer or by another student touching the thermometer when we were occupied. I also created a string that would display our current readings on the Hat’s LED screen.

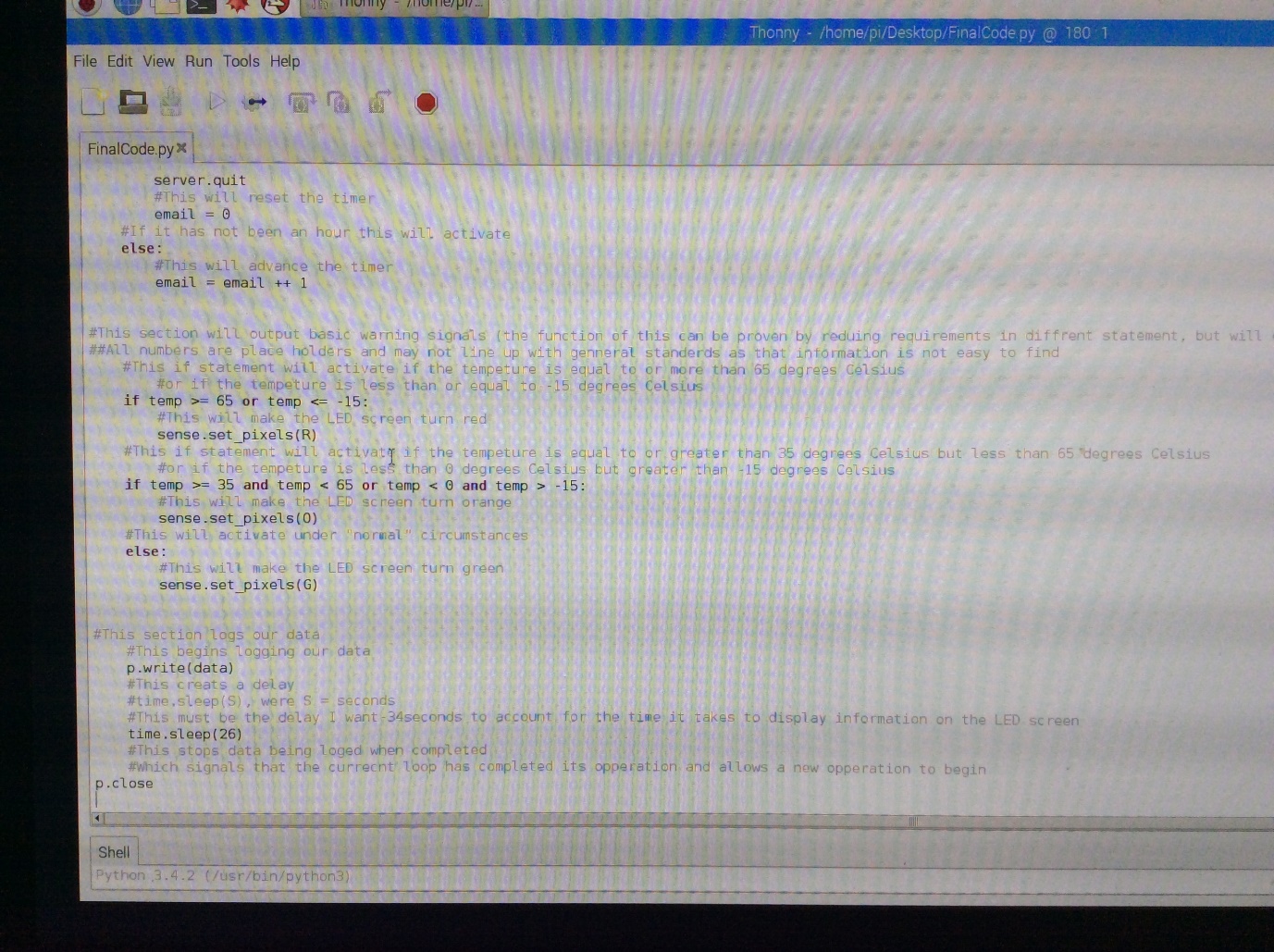
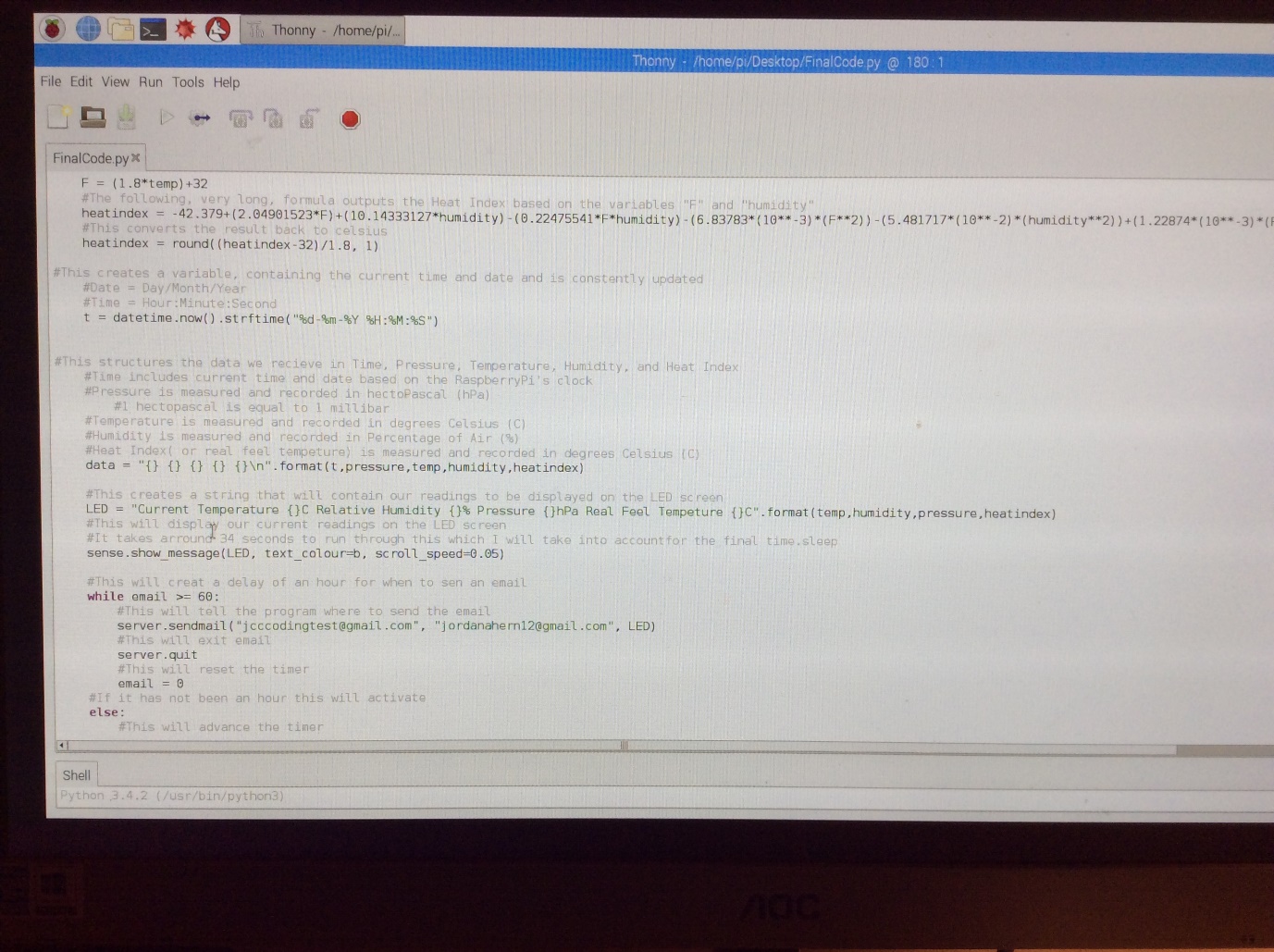
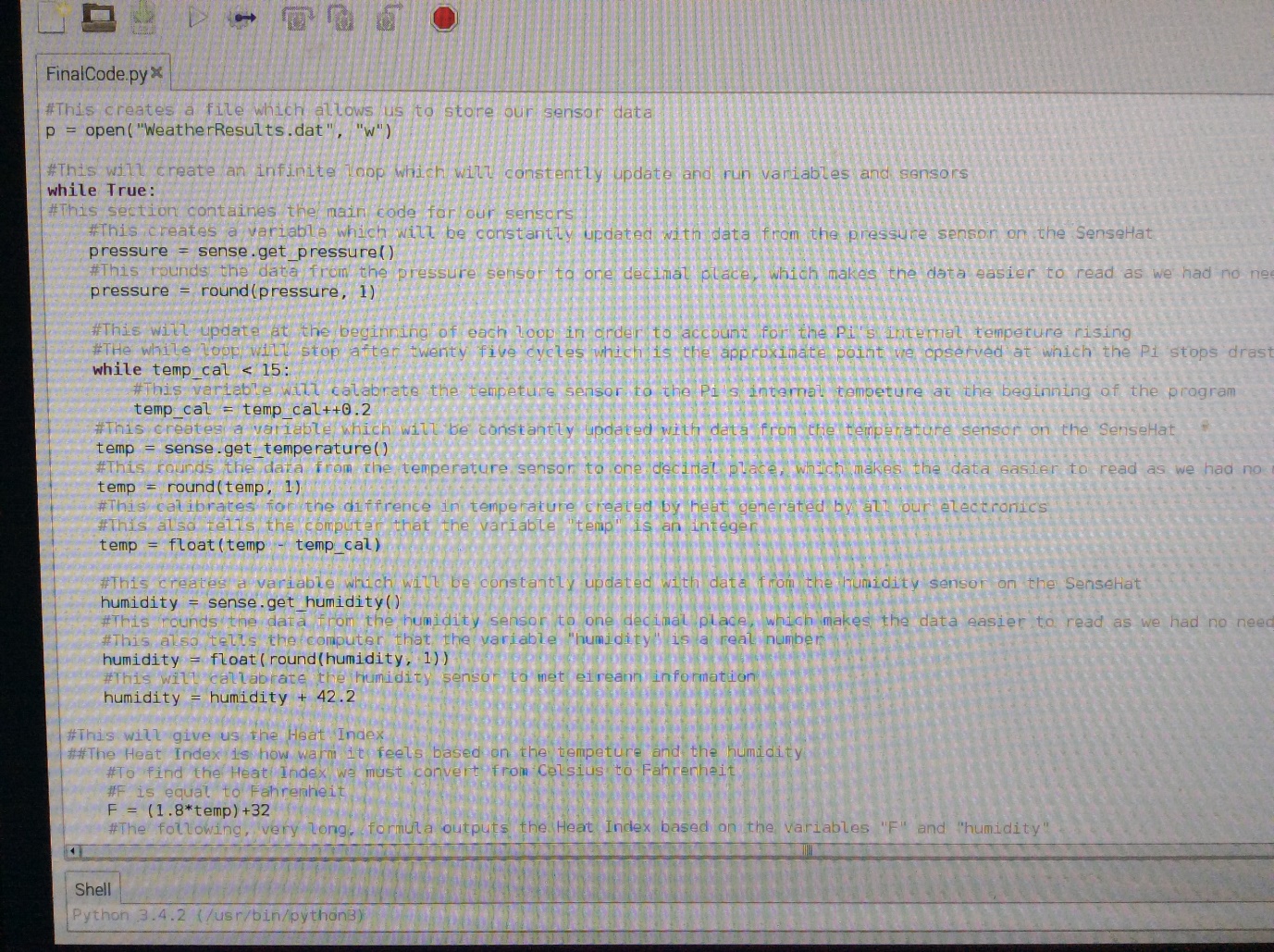
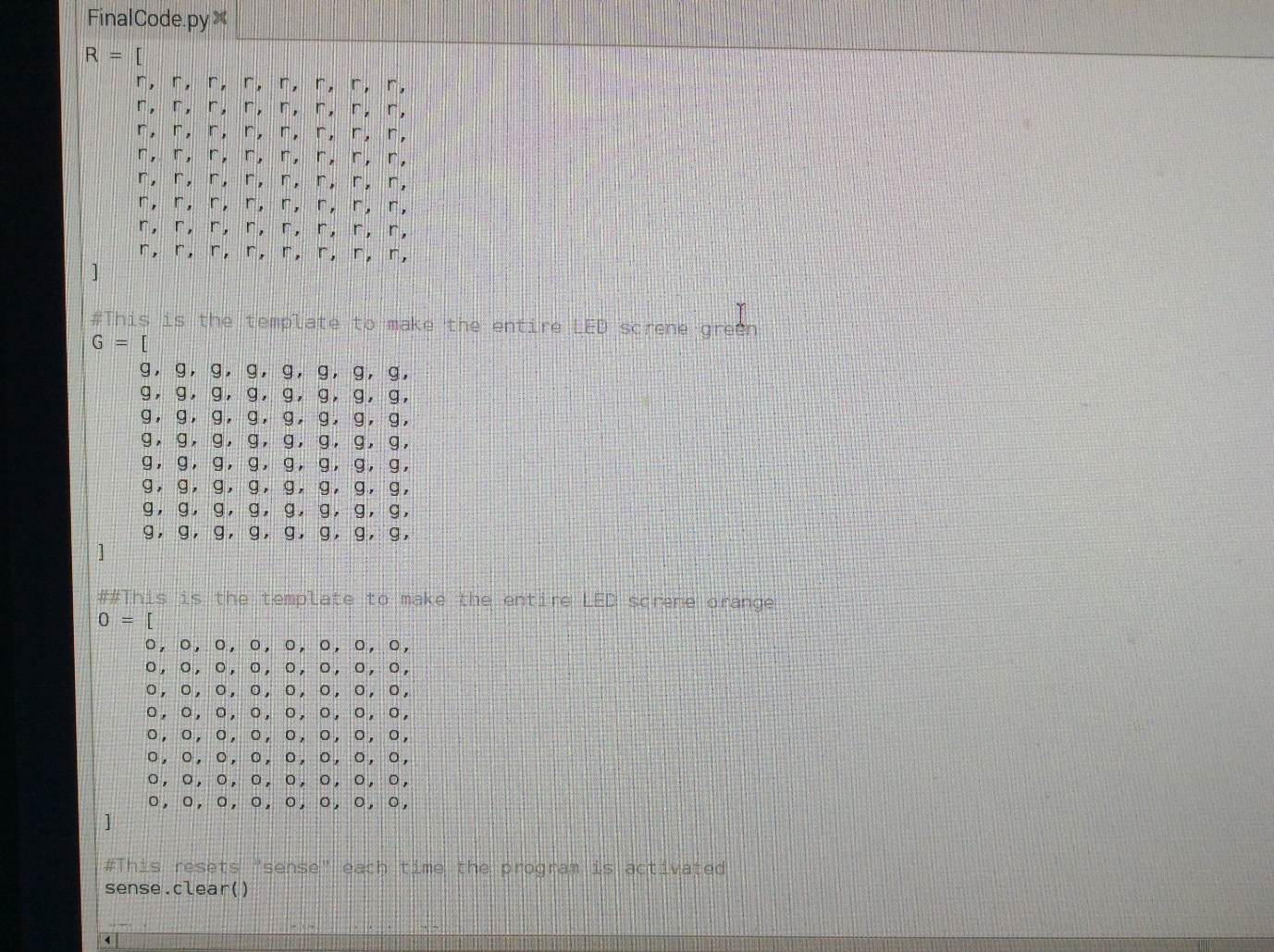
Finally, I researched proof of concept for a way to power the Pi with an external source, which would be required if this was to be left outside over a long-time period. This is more difficult with a Raspberry Pi then with an Arduino as it cannot take a 9V supply and must be powered through a USB port. I found a simple solution using a Linear Voltage Regulator, a 9V battery, some wires and a female USB Jack. The link to this can be found in research links.

* In week nine I was surprised that Jordan had added a light affect to the Sense Hat. It turned green when the temperature is safe for living in, orange when the temperature is slightly dangerous to the health of people in the area and finally red when the temperature is extremely dangerous to live in. When we settled down after having a look at Jordan’s work I collected all the data to help him calibrate the temperature sensor and I also completed and filled out my research section of the project diary.
* I was absent for all of week nine, so I was unable to continue with the project.

**Week Ten;**

* As this is our last week most of our time was spent making finishing touches to our project. I incorporated the new temperature calibrations. I then tested the Pi’s barometer against another barometer and found no major difference. I also expanded the parameters under which the LED screen will turn from Green to Orange or Red, to include if the temperature becomes too cold. I then had the last-minute idea to make further use of the Raspberry Pi’s WIFI connection, by setting it up to email me the current readings at set intervals. I go into further detail about this in the Research section. After Cian and Sergio cut a hole in the new lunch box I set up a quick test to check my calibrations. I created conditions exactly as I will run the final test, with the Pi sealed in the box beside an open window. I then ran my program over a short five-minute period. Over this time, I observed a very small margin of error between my temperature and pressure readings and readings from the Casement Aerodrome provided by Met Éireann, which can be put down as changeover distance. However, my humidity sensor was well under the Met’s data, on average I observed a difference of 44.2. This was easily fixed by adding this to my readings at the bottom of my code.
* This week when we came into class we asked Micheal where on our plastic container would be best to put a hole for the power supply to enter and then me and Sergio went down to the metalwork teacher, Philip, and got the hole cut but unfortunately, he cracked right through it. I had to buy a new container and bring it in on the second class during the week. We then brought the new container to our woodwork teacher, Michael Forrestal, and he pierced a hole in it with the pillar drill. We then completed the project diary and the project that day because it was due that Friday.
* This week me and Cian went to Philip our metalwork teacher to put a hole through the plastic container but unfortunately it broke. Cian volunteered to buy a new one and bring it in the next day to our woodwork teacher, Michael Forrestal to put a hole through it. We also worked on finishing our project diary.

**Our Code**

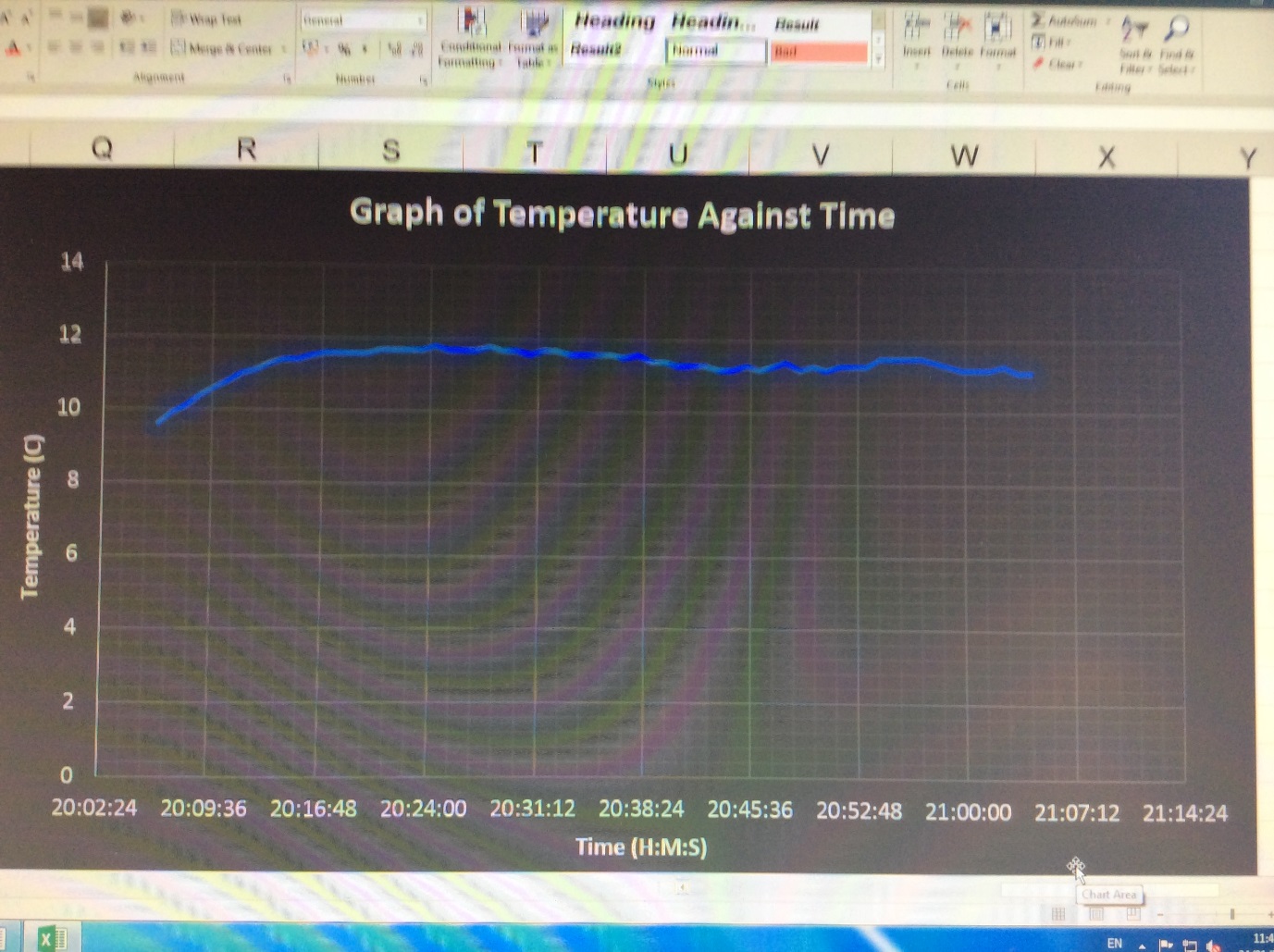
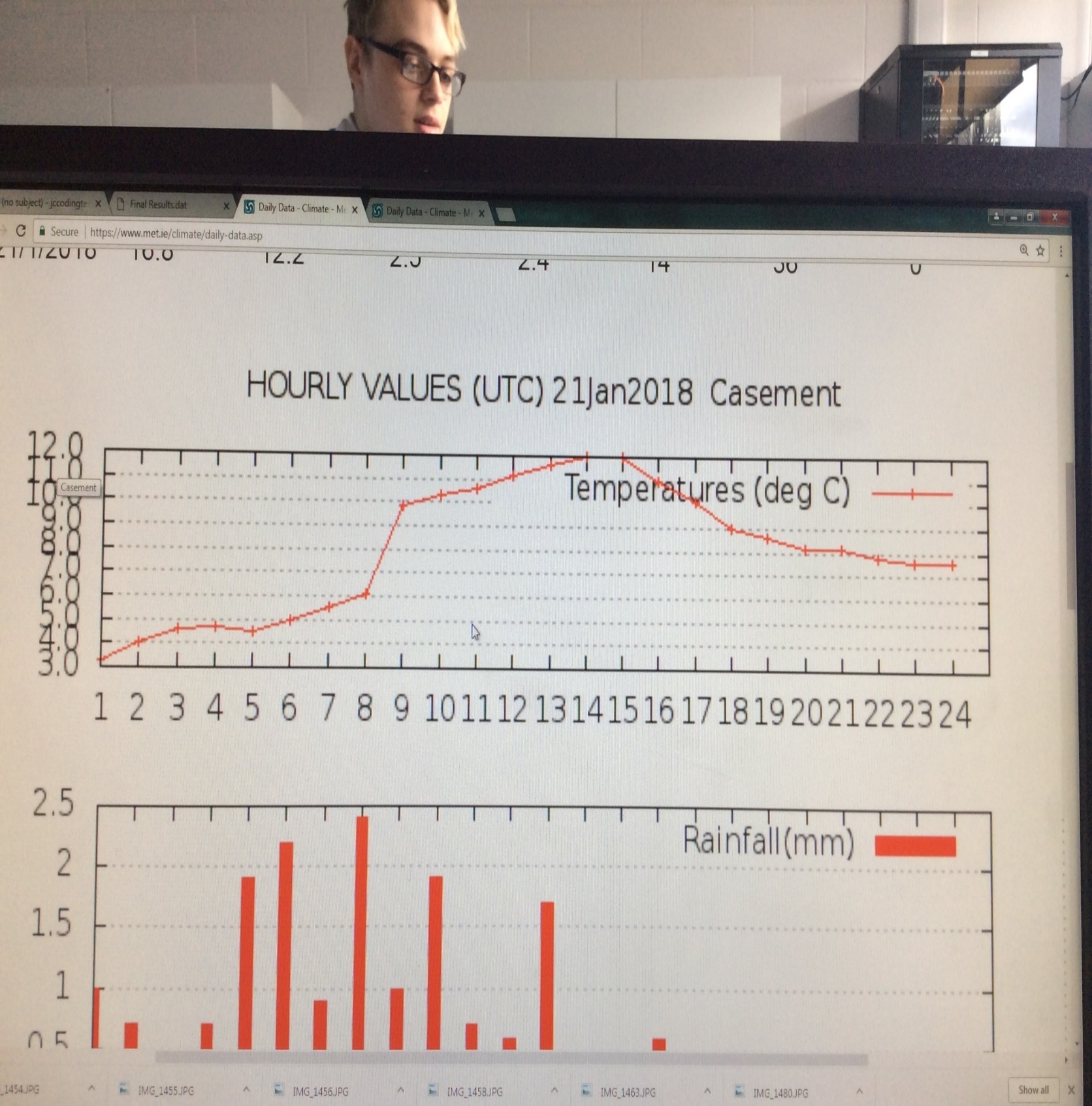


**Analysis of Collected Data**

We collected our data over the course of about an hour from six past eight pm to four past nine pm, on Sunday the 21st January 2018. The test was conducted by Jordan at his house. Our data was graphed using XY scatter graphs with Microsoft Excel and is displayed below alongside data collected by Met Éireann, at the Casement Aerodrome. (The Met’s data is only applicable from 8 to 9pm)

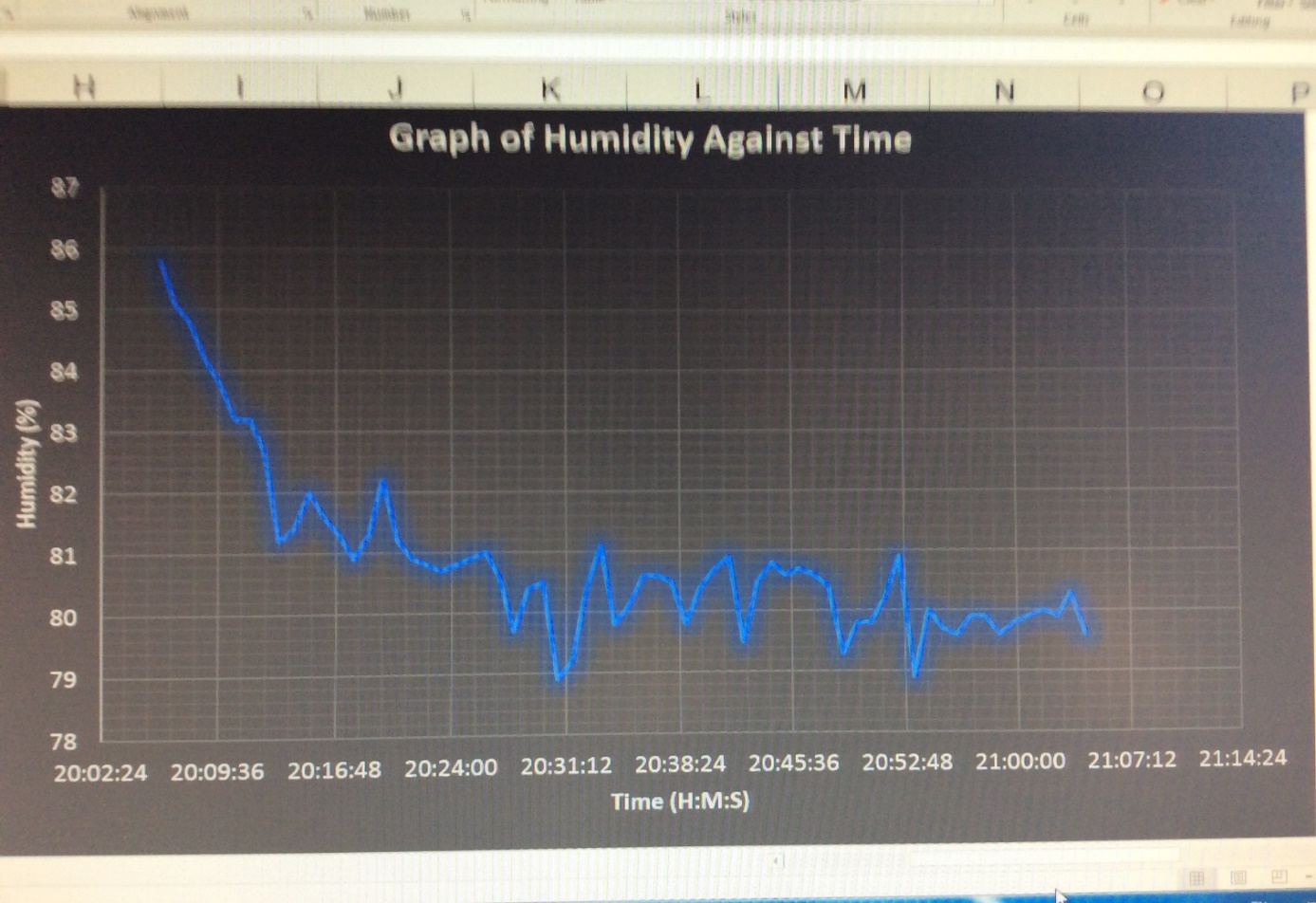
**Temperature;**

Our collected data appears to consistently remain three degrees warmer than what was recorded by the Met. I, Jordan, believe that this is due to a slight flaw in my test rather than in the code. This being that I kept the Pi on a windowsill beside a wide-open window. This means that the sensor may have been affected by my house’s central heating. This was necessary as it was a relatively windy night and I could not risk our project being blown off a window ledge and smashing.



**Humidity;**

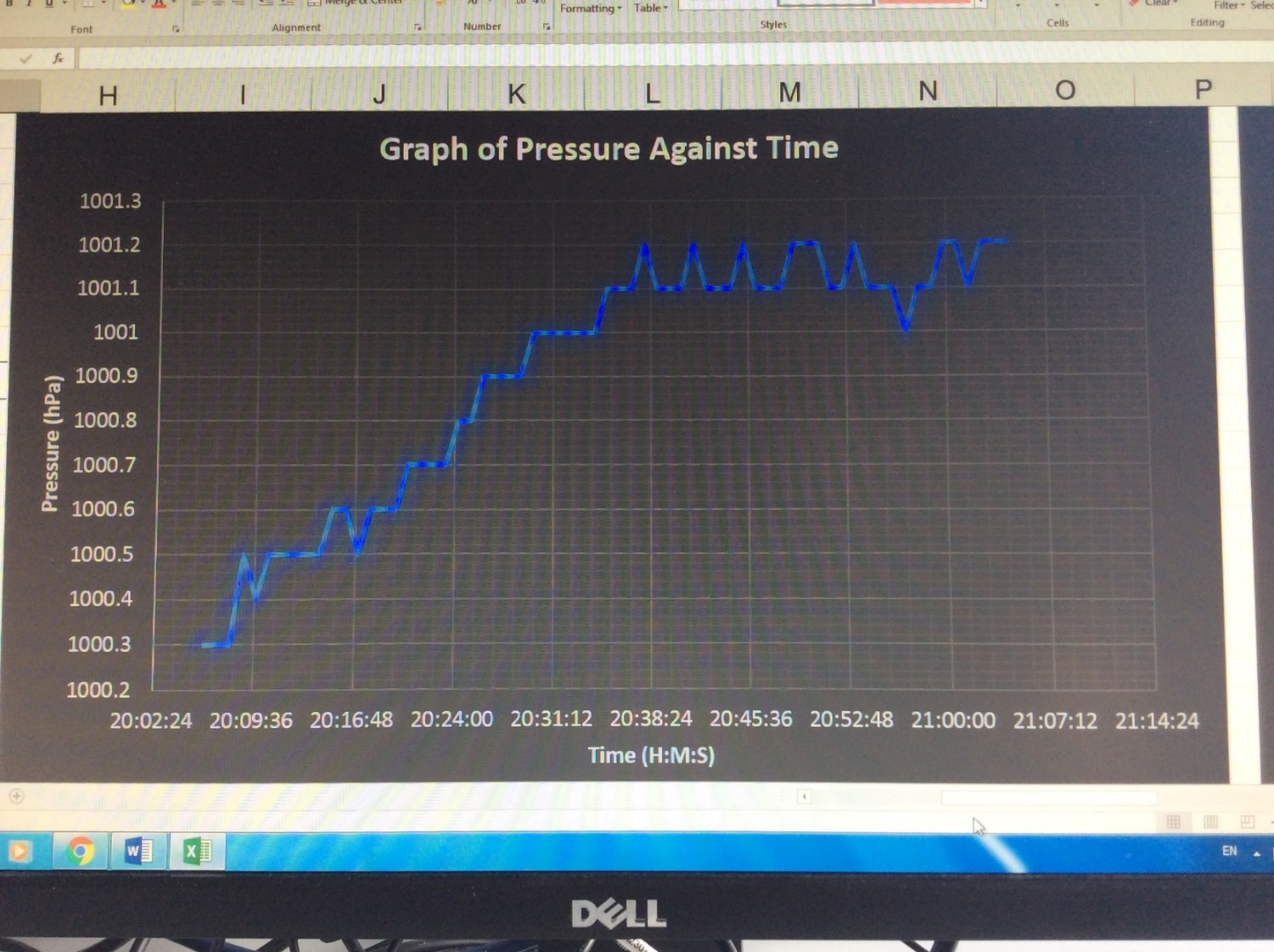
We were forced to use data collected by the station in the Phoenix Park rather than our local station at Casement as the Met does not record humidity data collected a Casement in their records. For this reason, we believe that our Humidity sensor worked very well considering the relatively small difference over such a distance.

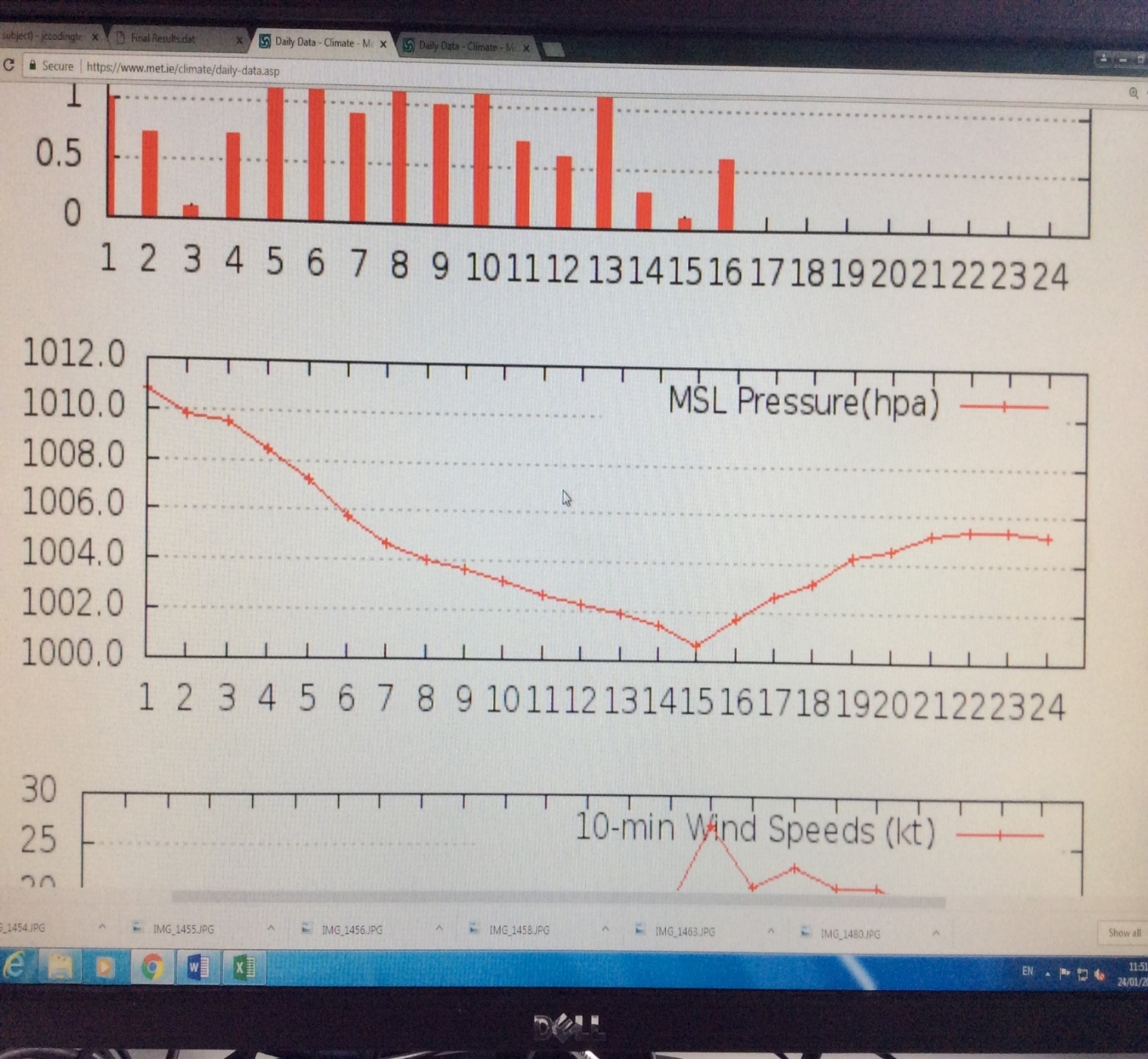




**Atmospheric Pressure;**

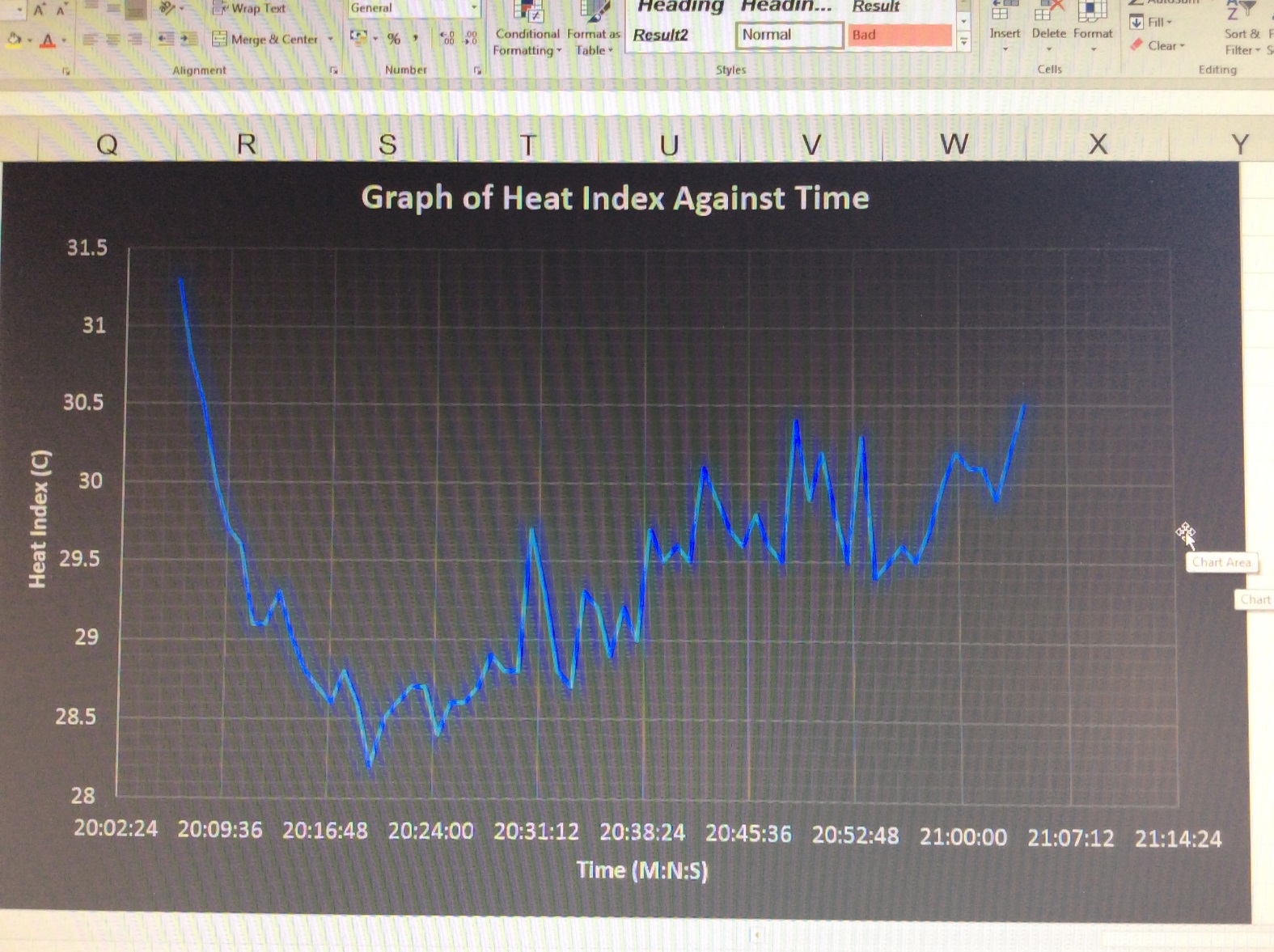
Our pressure sensor remained withen a margen of 4hPa(or millibars). Sutch a diffrence can easily be seen a variations over distance.





**Heat Index;**

Unfortinaltly, we could not find any national source to compare our collected heat index data to. Despite this we are certain that our data is incorrect. We are not sure what may have caused this error as having dubble checked my formula we can find no diffrences to our source. We believe that there may be problems with the calculation at lower tempetures but we am yet to confirm that.



**Research**

In this section we will go into further detail about our research and the results of that research. All links and references can be found under *Research Links*

**Research by Jordan;**

* **Transferring Data from the IDE to Software to be Graphed**

When I first began researching ways to do this I thought of using Microsoft PowerPoint to visualise our data, so I looked for a way to send data from the Arduino to PowerPoint.

I easily found the PLX-DAQ addon for PowerPoint from what I could see was exactly what I was looking for. So, in the next class I set up as simple LDR sensor (which we will also be using in the weather station), which could be used to test PLX-DAQ. While PLX was successful in collecting data, I found the data could not be graphed.

Believing that I was making a mistake, I began looking for video guides on how to use PLX, these were no help until I found a video which contained an annotation explaining that PLX is no longer supported and does not correctly function past Office 2010 (we are using Office 2013). Thankfully this video linked me to an updated version using new opensource software which does not use PowerPoint, named the Telemetry Viewer.

Unfortunately, it was clear that something was wrong from the start. Telemetry Viewer loaded fine and appeared to have access to my data but was unable to graph it. By reading through the website I found that Telemetry Viewer required a Java Runtime Environment which I was informed is not installed on the school computers. I received permission to install JRE on my computer which did not take long. I then attempted to run Telemetry Viewer, which I worryingly observed had lost its icon. I soon found that Telemetry Viewer refused to start. I reinstalled Telemetry Viewer to resolve this, but I received the same result. I believe that this is caused by conflicts between JRE and other versions of Java on the computer, or a conflict with school system. This is well outside my current understanding and timeframe, so I must look elsewhere.

After converting from an Arduino to a Raspberry Pi I found Gnu Plot which is a plotting software that can be downloaded and run in the Raspberry Pi console. I used a piece of code provided by the person who created the tutorial, which I have linked to, to create a file with a few data points. I also adapted some of this code to log data in my final code. Gnu Plot could then plot the data from this file and could alter the graph in multiple ways. Such as, axes, colour, and orientation.

Unfortunately, for some reason that I can’t figure out Gnu Plot seems to be unable to plot data that has been collected by the Sense Hat. I have attempted using both .dat and .csv files which I know should work to no avail.

At this point I decided that there was no need to continue searching for fancy ways to graph our data as we are only required to display it graphicly. As a result of this I decided to simply copy and paste the data into LibreOffice Calc and Graph it from there.

* **Remotely accessing/controlling the Arduino/Raspberry Pi**

From the start we saw this as a potentially very difficult challenge. The main reason for this are that Arduinos do not come with WIFI or Bluetooth connectivity built in, though the Intel Galileo Gen 2 that we use has an Ethernet port. At first, I had the idea to use two APC220 Radio Communication Modules, which I have come across in the CanSat Competition. This would allow us to create a two-way link between the Arduino and a PC. This hit a stonewall when I found that these modules cost quite a lot and often did not ship to Ireland.

I then began looking at other modules that would connect to WIFI or Bluetooth. I decided this may not work well as these would often be expensive too and had a lack of documentation as to how they worked.

As I go into in the Weekly Processes section, we ran into trouble attempting to connect the Arduino with an Ethernet connection. In the end it became apparent that there was a conflict between the Arduino and the school system.

After we transferred to the Pi, Cian and Sergio took on the role of remotely accessing the Pi. They came back with the ingenious idea to use TeamViewer. This worked so well that I spend much of my time using TeamViewer rather than directly connecting to the Pi when I am programming it.

* **My Attempt at using Multiprocessing**

When I decided to add a scrolling string to the LED screen I knew that the current loop would only continue after the string had run through. This could mess up timings if the string varies in length depending on the current data.

I found a solution which could resolve this. The multiprocessing module for Python would allow me to run two loops at the same time. This would allow me to run the main process of collecting the data and logging it in one loop and to control the LED screen from another.

My initial tests went well I could run two separate loops, as advertised, easily. However, I quickly discovered that I could not update a variable in one loop and use it in another. This was a problem as I relied on collected data to change the string and the warning system. Based on my understanding, I believe this occurred because each loop was a subset of the global system running parallel to each other. Which means that data outside the loops can be used by both but data inside each loop is confined to that loop alone.

I attempted to find a solution to this online but all info I could find was far above my current skill level. With time I would work my way up to a better understanding but for now I need to move on.

* **Sending Email from Python**

Surprisingly, I found this was very simple in comparison to what I expected. I added some code from the website which we have linked to and created a dummy test Gmail account. This allowed me to send email to any other Gmail account effortlessly with Python. I then decided that it would be a major annoyance to receive contestant emails every time the program loops. I created a while loop which would count at every loop, until an hour has passed at which time an email is sent and the count is reset.

* **Heat Index**

The Heat Index is how warm you feel despite what temperature really is. This is affected by air humidity. The most difficult part of this was typing up the long formula, but I did run into a problem returning a result. When the code was run in the console it would return a result of 0.0 consistently. I fist attempted to resolve this by declaring the formula a number by using a float. This made no difference, so I removed my conversion back to Celsius, from Fahrenheit, and received a result. But upon inspection I found that it was the incorrect result, while this was not ideal it did tell me that the problem likely stemmed from my conversions. The three components of the conversions are 9/5, temperature (in degrees C or F depending on which conversion it is), and 32. Of these components 9/5 seamed most likely to cause some confusion. To resolve this, I replaced 9/5 with 1.8, this resolved the problem and returned the correct result every time. While this works perfectly I still want to know why the conversions worked fine in the text editors built-in IED but not in the console.

Research by Cian;

* Wi-Fi Connection

When I started to research how to connect the Arduino to the PC using WIFI I came across a couple of websites which showed me how to do so. It told me that I needed an external router to do it. We could do this but unfortunately with the sensors that we had wired to the Arduino we could not spare the time and enough ports to completely connect it. We found a simple way around this and it was by just switching from Arduino to Raspberry Pi. Raspberry Pi has its own built in WIFI adapter so once we switched over we got it connected within seconds.

* Water Sensor

The water sensor on the Arduino was an interesting sensor that we all agreed that it would be nice to add to our project. Unfortunately, our Arduino was not compatible with this sensor, we needed the Arduino Uno. So that idea then went down the drain and when we switched to Raspberry Pi we didn’t even need to use this sensor.

* LDR to lux converting

I researched this conversion to assist Jordan with his readings of the LDR sensor on the PC that he was using. To read this conversion we needed to connect LDR with a 10 Kila Ohms resistor. But we decided not to include an LDR in our final project.

* LCD Screen

I researched the LCD Screen and successfully found multiple websites which showed us ways to set up the screen with wires and the code attached which we ran through but unfortunately, we were not successful with doing this due to the wiring being incorrect and the code had a good few issues also.

* TeamViewer

Me and Sergio found an app which allowed us to control a PC remotely through our iPads which solved a lot of our problems in our struggle with the Remote Controlling of the Raspberry Pi.

**Research by Sergio**

* Arduino WIFI connection

At the start of the project I researched how to connect an Arduino to a pc wirelessly and I found a couple of websites that could help us but all of them said that we need an external router. We tried to connect it but unfortunately, we could not do it, so we decided to change to a raspberry pi which had a built in WIFI adapter and it would be easier to connect to.

* Thermometer

I started researching on how to wire the thermometer to the Arduino and the code needed. I found a few websites that were helpful, and we used this to wire up the sensor. We got it to work but the only issue that we had is that the results from the sensor wouldn’t display on the LCD screen. Our teacher Michael.

took it home to work on it in his own time but he couldn’t figure out the issue. This is one of the reasons why we decided to change to a raspberry pi.

* TeamViewer

Me and Cian found an app on our iPads called TeamViewer. This app allowed us to control the pc remotely through the iPad. This app helped us a lot because we couldn’t find a different way to do it.

* Barometer

When I was researching about the barometer I found some websites, but the wiring was too difficult, and we didn’t have the necessary pins on the Arduino. We decided to try it and use a second breadboard for the pins that didn’t have space on the Arduino we couldn’t get it to work because the wiring was incorrect, and the code had some issues.

* LCD screen

While I was researching the LCD screen I found a couple of websites that showed us how to wire the LCD to the Arduino and it also had the code attached which we used but unfortunately, it didn’t work because the code was incorrect, and nothing would display on the screen.

**What Went Well**

* Our project we completed all required tasks using the Raspberry Pi
* We added effective secondary missions such as Emailing, and information on the LED display.
* Our collected data was accurate
* The LED screen worked very well and could be seen very easily through our box
* Our code worked very well

**What we could Improve**

* We could find a way to send data to be Graphed in real-time
* We could find a way to add extra sensors to the Pi
* We could test the temperature sensor again to ensure it was affected by an experimental error
* We could create an external power supply to allow us the create a fully portable Weather Station

**Reflections**

Overall, we are very happy with how our project turned out. The code worked very well and was concise and explained. We believe we worked very well over these weeks. Based on our feedback our project was a success. We created an easily usable weather station that has many commercial uses, such as teaching young children about coding and the weather in a fun and visual way. We can definitively improve this project in many ways. These have been listed above. We are confident that most of these goals could have been completed; had we decided to work with the Raspberry Pi from the beginning. We also believe that it would be a good idea to add a fan of sorts that would measure wind speed. This would require us to find a way to connect the Hat with wires rather than putting it directly on the Pi.

**Research Links**

**Cian;**

* + - **Arduino Wi-Fi connection**
      * <https://openhomeautomation.net/arduino-wifi-cc3000/>
      * <https://openhomeautomation.net/wireless-relay-arduino-wifi/>
      * <http://forum.arduino.cc/index.php?topic=14498.0>
    - **Arduino Water Sensor**
      * <http://www.instructables.com/id/How-to-use-a-Water-Level-Sensor-Arduino-Tutorial/>
* **LDR to lux converting**
* <http://forum.arduino.cc/index.php?topic=141815.0>
* <https://raspberrypi.stackexchange.com/questions/12955/how-do-i-send-data-from-the-raspberry-pi-to-the-pc>
* **LCD Screen**
* <https://github.com/Seeed-Studio/Grove_LCD_RGB_Backlight>
* <http://www.instructables.com/id/Grove-Lucky-Dumpling/>
* <https://create.arduino.cc/projecthub/microBob/lcd-liquid-crystal-display-e72c74?ref=search&ref_id=LCDScreen&offset=0>
* **Remote Control of Raspberry Pi**

<https://lifehacker.com/how-to-control-your-raspberry-pi-from-any-computer-usin-1788592777>

* **TeamViewer**
* <https://www.teamviewer.com/en/download/windows/?pid=google.tv_ex_sl1.s.gb&gclid=EAIaIQobChMIlPDGm_vP2AIVbL7tCh0qfAzqEAAYASADEgKKh_D_BwE>
* **LCD Screen with Pi3**
* <http://www.circuitbasics.com/raspberry-pi-lcd-set-up-and-programming-in-python/>

**Sergio;**

* **Arduino WIFI**
* <http://www.instructables.com/id/Accessing-Arduino-over-internet/>
* <http://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>
* **Thermometer**
* <https://create.arduino.cc/projecthub/TheGadgetBoy/making-lcd-thermometer-with-arduino-and-lm35-36-c058f0>
* **TeamViewer**
* <https://www.google.ie/search?q=how+to+download+teamveiwer+on+raspberry+pi&rlz=1C1GGRV_enIE774IE774&oq=how+to+download+teamveiwer+on+raspberry+pi&aqs=chrome..69i57.53382j0j7&sourceid=chrome&ie=UTF-8#kpvalbx=1>
* **Barometer**
* <https://www.sunfounder.com/learn/sensor-kit-v2-0-for-raspberry-pi-b-plus/lesson-31-barometer-sensor-kit-v2-0-for-b-plus.html>
* **LCD Screen**
* <https://github.com/Seeed-Studio/Grove_LCD_RGB_Backlight>
* <http://www.instructables.com/id/Grove-Lucky-Dumpling/>

**Jordan;**

* **Transferring Data from the IDE to Software to be Graphed**
* <http://www.instructables.com/id/Sending-data-from-Arduino-to-Excel-and-plotting-it/>
* <http://www.instructables.com/id/Live-data-to-excel-from-an-Arduino-Light-Sensor/>
* <https://www.youtube.com/watch?v=lFZ26gD7OIE>
* <http://www.farrellf.com/TelemetryViewer/>
* <https://www.raspberrypi-spy.co.uk/2014/04/how-to-use-gnuplot-to-graph-data-on-the-raspberry-pi/>
* <https://www.youtube.com/watch?v=9k-l_ol9jok>
* <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
* <https://pypi.python.org/pypi/pyoo>
* <https://www.codeproject.com/Tips/1065823/PyOOCalc-Python-Libre-Open-Office-Calc-Interface-A>
* <https://www.raspberrypi-spy.co.uk/2014/04/how-to-use-gnuplot-to-graph-data-on-the-raspberry-pi/>
* <https://www.youtube.com/watch?v=vcOE2XAQHzY>
* <https://www.youtube.com/watch?v=jgKPmjQtJG4>
* **Displaying Information on a LCD screen**
  + <http://www.instructables.com/id/How-to-use-an-LCD-displays-Arduino-Tutorial/>
* **Wirelessly accessing an Arduino/Raspberry Pi**
  + <http://www.instructables.com/id/How-to-Receive-Arduino-Sensor-Data-on-Your-Android/>
  + <http://www.instructables.com/id/Connect-Arduino-Uno-to-Android-via-Bluetooth/>
  + <http://www.instructables.com/id/How-control-arduino-board-using-an-android-phone-a/>
  + <http://www.instructables.com/id/Connect-Arduino-Router-without-any-sheilds/>
  + <http://www.instructables.com/id/How-to-Connect-your-Arduino-to-the-Internet-as-a-W/>
* **Sense Hat**
* <https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat>
* <https://pimylifeup.com/raspberry-pi-weather-station/>
* **Sending Email with Python**
* <http://naelshiab.com/tutorial-send-email-python/>
* **Pins used by the Sense Hat**
* <https://www.raspberrypi.org/forums/viewtopic.php?t=118964#p1048288>
* <https://www.raspberrypi.org/app/uploads/2015/08/Sense-HAT-V1_0.pdf>
* <https://pinout.xyz/pinout/sense_hat>
* **Power a Pi with a 9v battery (proof of concept)**
* <https://www.youtube.com/watch?v=h6Bl0Bckl2k>
* **National Weather Service**
* <http://www.met.ie/default.asp>
* **Heat Index**
* <https://www.weather.gov/media/epz/wxcalc/heatIndex.pdf>
* <https://www.rapidtables.com/convert/temperature/celsius-to-fahrenheit.html>
* <http://www.wpc.ncep.noaa.gov/html/heatindex.shtml>
* **Multiprocessing**
* <https://docs.python.org/2/library/multiprocessing.html>
* <https://stackoverflow.com/questions/3474382/how-do-i-run-two-python-loops-concurrently>
* <https://www.youtube.com/watch?v=Lu5LrKh1Zno>
* **Miscellaneous**
* <https://www.w3schools.com/colors/colors_picker.asp>

**Thanks to our teacher Micheal for teaching us how to code and for his assistance during this project**