

1809802

by Mason Morgan

Submission date: 24-Apr-2022 01:53PM (UTC+0100)

Submission ID: 177012092

File name: 13695_Mason_Morgan_1809802_179584_1925625054.pdf (1.15M)

Word count: 3434

Character count: 22132

Course: BSc (Hons) Electrical and Electronic Engineering

Module: SGEE6005 Electronic Design

Assignment: Micro-controllers and the Internet of Things

Name: Mason Richard Morgan

University number: 1809802

Date submitted: 23rd April 2022

Word count: 3000

Currently:

3274

You sure?

40-45

- ① CITATION LIST IS VERY SHORT
- ② RESEARCH IS BRIEF AND HAS NO CONTENT.
- ③ REPORT IS VERY DESCRIPTIVE AND LACKS CRITICAL ANALYSIS.
- ④ WEATHER STATION WAS SUCCESSFULLY IMPLEMENTED.
- ⑤ PRESENTATION COULD HAVE BEEN BETTER WITH A LOT OF THE WEATHER STATION DIAGRAM IN THE APPENDIX.

Contents

1. Executive Summary	3
2. Research.....	4
IOT Definitions.....	4
Enabling Technologies	4
Connectivity Models.....	4
Transformational Potential	4
The origins and drivers of the Internet of Things	4
3.1. Build the sensor module to monitor atmospheric temperature, pressure, and humidity.....	8
3.2. Configure the IoT Cloud to monitor the data recorded by the sensors.....	9
3.3. Write the necessary Arduino code	10
3.4. Create an Arduino Dashboard	14
3.5. Configure and test the system	15
3.6. Modify the design	21
4. Conclusions.....	25
5. Bibliography	26
Figure 1: Setting up things for the Arduino	8
Figure 2: Setup page for selecting variables.....	9
Figure 3: Main code for Arduino weather station.....	10
Figure 4: More code	11
Figure 5: Connection of the Arduino to software device.....	12
Figure 6: Editor feature for transferring data	12
Figure 7: Monitoring the data transfer.....	13
Figure 8: Transferring data to Arduino board.....	13
Figure 9: Creating a dashboard.....	14
Figure 10: Dashboard display	14
Figure 11: Pressure display value	15
Figure 12: Temperature display value	15
Figure 13: Live Humidity reading	16
Figure 14: Live value for light.....	16
Figure 15: Arduino mobile app.....	17
Figure 16: Mobile dashboard.....	18
Figure 17: Dashboard configured on mobile	19
Figure 18: Using graphs on mobile app.....	20
Figure 19: Temp display modified	21
Figure 20: Humidity display modified.....	21
Figure 21: Light display modified	22
Figure 22: Pressure display modified	22
Figure 23: Finished device inside enclosure.....	23
Figure 24: Birds eye view of product	24

1. Executive Summary

This report will give extensive research into:

- The IOT Definitions
- Enabling Technologies
- Connectivity Models
- Transformational Potential
- The origins & drives of the internet of things
- Device-to-device
- Device-to-cloud
- Device-to-gateway
- Back-end-data sharing
- The security & privacy challenges faced by IOT

This will give a better understanding on how to carry out the next part of the task.

An important part of this report was the building a Personal Weather Station with Arduino IoT Cloud and the MKR IoT Carrier. The aim was to be able to record the local weather conditions, which was monitored in real time.

To make this application a success a weather station was setup that collected some basic data and give notifications about our local weather conditions. This was conducted through the cloud, so we could monitor it from wherever we wanted.

The goals of the project involved the following:

- 1) To familiarise ourselves with the sensors on the MKR IoT Carrier.
- 2) To learn how to build a dashboard to monitor data
- 3) Learn to use the display on the MKR IoT Carrier
- 4) Learn how to use the buttons on the MKR IoT Carrier

The components involved included the following:

- 1) Arduino MKR Wifi 1010
- 2) MKR IoT Carrier
- 3) Micro USB cable

The device could either be powered via a USB cable which could be plugged directly into a laptop / computer, or it can run off a 3.7V battery.

The Arduino is capable of tracking temperature, humidity, pressure, and light. While the temperature might provide the most immediately useful information, the other data can also be beneficial, with pressure values telling us whether there is warm or cold weather incoming, and light values letting us know if it is sunny outside.

Tracking local weather conditions is not only relevant to the present time;

2. Research

IOT Definitions

To define the term Internet of Things It can be said that network connectivity & computing capability extends to objects. Devices like everyday items / sensors etc. aren't normally considered computers. This then means devices can generate, exchange & consume data with minimal human intervention. (Karen Rose, 2022)

Enabling Technologies

An **enabling technology** is defined as “Equipment and/or methodology that, alone or in combination with associated technologies, provides the means to generate giant leaps in performance and capabilities of the user.” (University Birmingham, 2022)

It has been apparent that combining computers, sensors, and networks to monitor and control devices has been around for a long time.

Connectivity Models

Different communication models can be used for IOT implementations. There are three different communications methods which include the following: Device-to-Device, Device-to-Gateway, and Back-End Data-Sharing. (Channel Futures, 2022)

Transformational Potential

A shift in thinking about the implications and issues will be caused by the projections & trends towards IOT. Quiet often interactions from the internet don't come from active engagement with content but instead come from the engagement of connected objects.

The origins and drivers of the Internet of Things

The term “Internet of Things” (IoT) was first used in 1999 by British technology pioneer Kevin Ashton to describe a system in which objects in the physical world could be connected to the Internet by sensors. Ashton coined the term to illustrate the power of connecting Radio-

Frequency Identification (RFID) tags used in corporate supply chains to the Internet to count and track goods without the need for human intervention. Today, the Internet of Things has become a popular term for describing scenarios in which Internet connectivity and computing capability extend to a variety of objects, devices, sensors, and everyday items. While the term “Internet of Things” is new, the concept of combining computers and networks to monitor and control devices has been around for decades. By the late 1970s, for example, systems for remotely monitoring meters on the electrical grid via telephone lines were already in commercial use. In the 1990s, advances in wireless technology allowed “machine-to-machine” (M2M) enterprise and industrial solutions for equipment monitoring and operation to become widespread. Many of these early M2M solutions, however, were based on closed purpose-built networks and proprietary or industry-specific standards, rather than on Internet Protocol (IP)-based networks and Internet standards.

Device to Device

The direct connection and communication are represented by two or more devices rather than through an intermediary application server.

Some of the networks that promote communication involve IP networks or the Internet. A more modern way of communicating can be Bluetooth.

Device-to-device networks allow communication models for applications like home automation systems. This form of communication is smaller in size meaning there are smaller data packets and low data rate requirements.

Devices like light bulbs for example send small amounts of information to each other.

Device to Cloud

The IoT device connects directly to an Internet cloud service like an application service provider to exchange data and control message traffic. This approach frequently takes advantage of existing communications mechanisms like traditional wired Ethernet or Wi-Fi connections to establish a connection between the device and the IP network, which connects to the cloud service.

Device to Gateway

There is application software operating on a local gateway device, which acts as an intermediary between the device and the cloud service and provides security and other functionality such as data or protocol translation.

In many cases, the local gateway device is a smartphone running an app to communicate with a device and relay data to a cloud service. These devices do not have the native ability to connect directly to a cloud service, so they frequently rely on smartphone app software to serve as an intermediary gateway to connect the fitness device to the cloud.

Back-end Data Sharing

The back-end data-sharing model refers to a communication architecture that enables users to export and analyse smart object data from a cloud service in combination with data from other sources. This architecture supports “the [user’s] desire for granting access to the uploaded sensor data to third parties.” This approach is an extension of the single device- to-cloud communication model, which can lead to data silos where “IoT devices upload data only to a single application service provider.’ A back-end sharing architecture allows the data collected from single IoT device data streams to be aggregated and analysed. For example, a corporate user in charge of an office complex would be interested in consolidating and analysing the energy consumption and utilities data produced by all the IoT sensors and Internet-enabled utility systems on the premises.

The security and privacy challenges faced by IoT

It's very important that there is trust when using the internet but especially these days more than ever with all the digital threats that we're constantly exposed to daily.

The Internet of things is also a very similar process in the fact that it is fundamental to give the users a degree of trust.

It's almost certain that if the user cannot trust that their connected devices and their private information isn't safe then this may cause reluctance to use the internet.

This has global consequences to electronic commerce, technical innovation, free speech, and every other aspect of online activities. Indeed, ensuring security in IoT products and services should be considered a top priority for the sector.

Competitive cost and technical constraints on IoT devices challenge manufacturers to adequately design security features into these devices, potentially creating security and long-term maintainability vulnerabilities greater than their traditional computer counterparts.

Along with potential security design deficiencies, the sheer increase in the number and nature of IoT devices could increase the opportunities of attack. When coupled with the highly interconnected nature of IoT devices, every poorly secured device that is connected online potentially affects the security and resilience of the Internet globally, not just locally. For example, an unprotected refrigerator or television in the US that is infected with malware might send thousands of harmful spam emails to recipients worldwide using the owner's home Wi-Fi Internet connection. (HEWLETT PACKARD ENTERPRISE, 2022)

3. Methodology, testing, and results for Task 2

3.1. Build the sensor module to monitor atmospheric temperature, pressure, and humidity

To progress with this project, I firstly needed to start by setting up my device digitally rather than physically. I needed to use the IOT CLOUD software on the Arduino website to do this. (Project Hub, 2022)

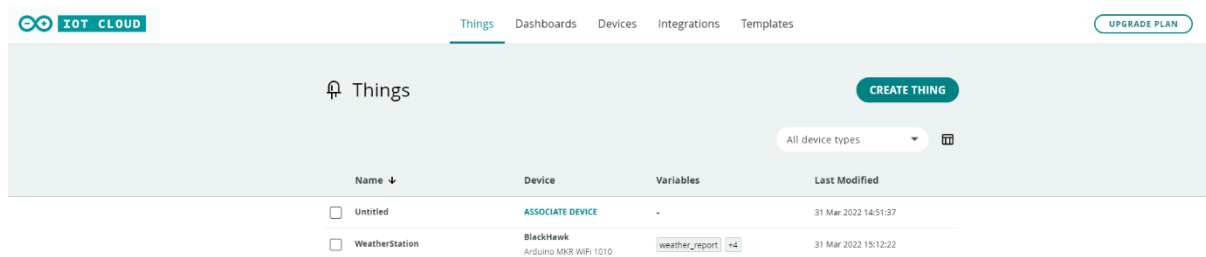


Figure 1: Setting up things for the Arduino

At this early stage, a file needed to be created to identify the “thing” meaning the Arduino device. I named the device “Blackhawk” because I thought the name was different to just using my standard name.

After setting up the Thing I was then directed to the setup page in which I found the Setup option. The setup allowed me to enter what data that I wanted to record for my feedback. The Arduino can read many parameters however, the variables I chose were:

1. Humidity
2. Light
3. Pressure
4. Temperature

3.2. Configure the IoT Cloud to monitor the data recorded by the sensors

All variables which affect us in our everyday life which is why they were chosen. A variable for “weather report” was setup to allow all data collected to be compiled in one area to be read properly.

The screenshot shows the 'WeatherStation' IoT Cloud setup interface. At the top, the 'Setup' tab is selected, and the 'Sketch' tab is also visible. The 'Thing ID' is f2a77647-e468-4bfe-85be-092beea1d1c6.

Variables

	Name ↓	Last Value	Last Update
☐	Humidity float humidity;	-	⋮
☐	Light int light;	-	⋮
☐	Pressure float pressure;	-	⋮
☐	Temperature float temperature;	-	⋮
☐	weather_report String weather_report;	-	⋮

Device

BlackHawk

ID: 25eb57c4-f22f-46e8-b6dd-7...
Type: Arduino MKR WiFi 1010
Status: Offline

Change **Detach**

Network

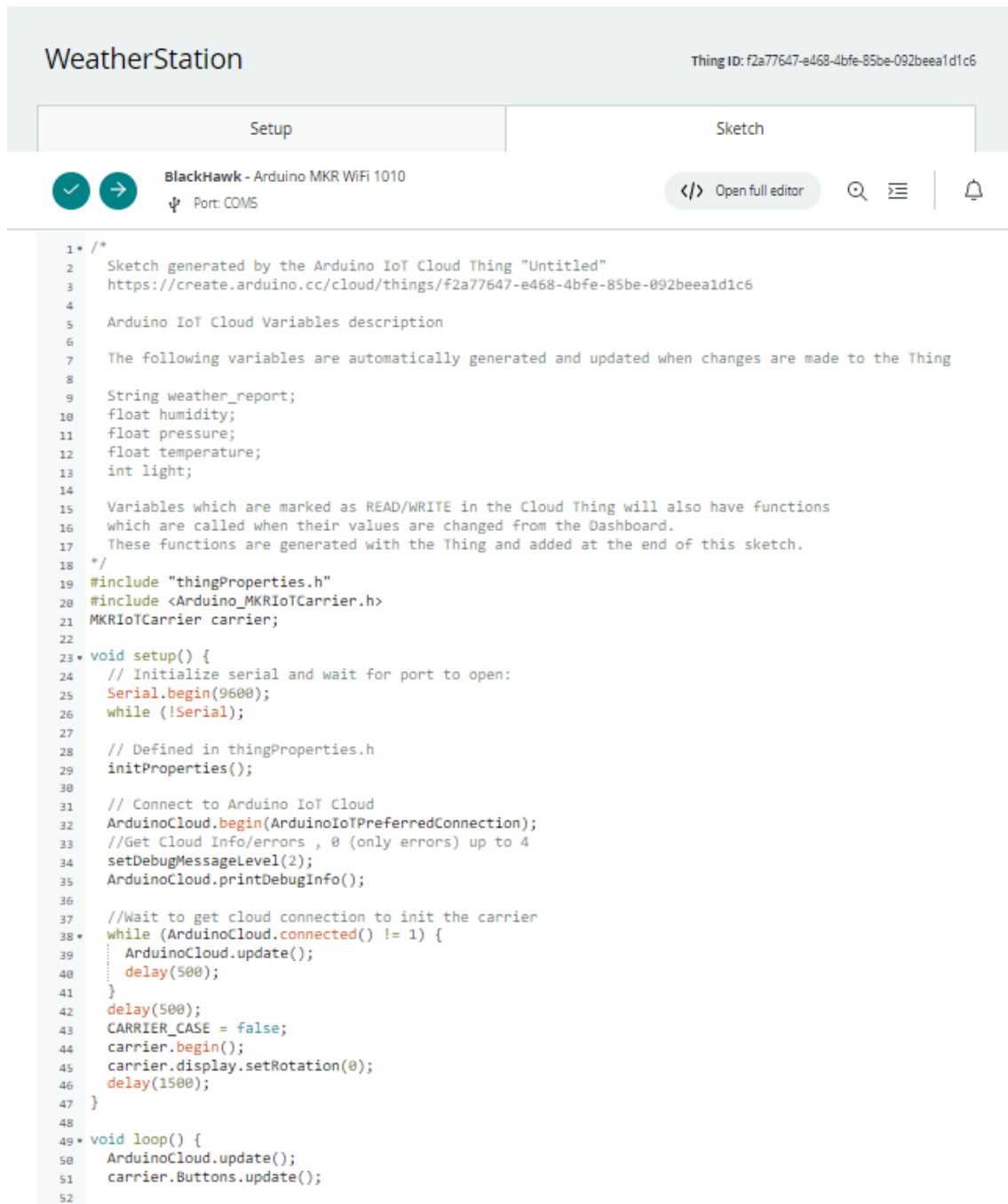
Wi-Fi Name: EE-JKC2FR
Password:
Change

Figure 2: Setup page for selecting variables

To allow access to the Arduino from anywhere the device was also capable of transmitting all data to a mobile phone through the Arduino app from anywhere in the world. The local Network was setup to my home Wi-Fi, and this now allowed me to go online.

3.3. Write the necessary Arduino code

Once the device had been Setup, I later went on to constructing the correct program which can be displayed by figure 3 below.



```
1 /*
2  Sketch generated by the Arduino IoT Cloud Thing "Untitled"
3  https://create.arduino.cc/cloud/things/f2a77647-e468-4bfe-85be-092beea1d1c6
4
5  Arduino IoT Cloud Variables description
6
7  The following variables are automatically generated and updated when changes are made to the Thing
8
9  String weather_report;
10 float humidity;
11 float pressure;
12 float temperature;
13 int light;
14
15 Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
16 which are called when their values are changed from the Dashboard.
17 These functions are generated with the Thing and added at the end of this sketch.
18 */
19 #include "thingProperties.h"
20 #include <Arduino_MKR101Carrier.h>
21 MKR101Carrier carrier;
22
23 void setup() {
24   // Initialize serial and wait for port to open:
25   Serial.begin(9600);
26   while (!Serial);
27
28   // Defined in thingProperties.h
29   initProperties();
30
31   // Connect to Arduino IoT Cloud
32   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
33   //Get Cloud Info/errors , 0 (only errors) up to 4
34   setDebugMessageLevel(2);
35   ArduinoCloud.printDebugInfo();
36
37   //Wait to get cloud connection to init the carrier
38   while (ArduinoCloud.connected() != 1) {
39     ArduinoCloud.update();
40     delay(500);
41   }
42   delay(500);
43   CARRIER_CASE = false;
44   carrier.begin();
45   carrier.display.setRotation(0);
46   delay(1500);
47 }
48
49 void loop() {
50   ArduinoCloud.update();
51   carrier.Buttons.update();
52 }
```

Figure 3: Main code for Arduino weather station

The code below is displaying the carrier display parameters but is just being continued from figure 3.

```
52
53 while(!carrier.Light.colorAvailable()) {
54     delay(5);
55 }
56 int none;
57 carrier.Light.readColor(none, none, none, light);
58
59 temperature = carrier.Env.readTemperature();
60 humidity = carrier.Env.readHumidity();
61 pressure = carrier.Pressure.readPressure();
62
63
64 * if (carrier.Button0.onTouchDown()) {
65     carrier.display.fillScreen(ST77XX_WHITE);
66     carrier.display.setTextColor(ST77XX_RED);
67     carrier.display.setTextSize(2);
68
69     carrier.display.setCursor(30, 110);
70     carrier.display.print("Temp: ");
71     carrier.display.print(temperature);
72     carrier.display.print(" C");
73 }
74
75 * if (carrier.Button1.onTouchDown()) {
76     carrier.display.fillScreen(ST77XX_WHITE);
77     carrier.display.setTextColor(ST77XX_RED);
78     carrier.display.setTextSize(2);
79
80     carrier.display.setCursor(30, 110);
81     carrier.display.print("Humi: ");
82     carrier.display.print(humidity);
83     carrier.display.print(" %");
84 }
85
86 * if (carrier.Button2.onTouchDown()) {
87     carrier.display.fillScreen(ST77XX_WHITE);
88     carrier.display.setTextColor(ST77XX_RED);
89     carrier.display.setTextSize(2);
90
91     carrier.display.setCursor(30, 110);
92     carrier.display.print("Light: ");
93     carrier.display.print(light);
94 }
95
96 * if (carrier.Button3.onTouchDown()) {
97     carrier.display.fillScreen(ST77XX_WHITE);
98     carrier.display.setTextColor(ST77XX_RED);
99     carrier.display.setTextSize(2);
100
101     carrier.display.setCursor(30, 110);
102     carrier.display.print("Pressure: ");
103     carrier.display.print(pressure);
104 }
105
106
107 * if (humidity >= 60 && temperature >= 15) {
108     weather_report = "It is very humid outside";
109
110 * }else if (temperature >= 15 && light >= 700) {
111     weather_report = "Warm and sunny outside";
112
113 * }else if (temperature <= 16 && light >= 700) {
114     weather_report = "A little cold, but sunny outside";
115 }
116
117 }
118
```

Figure 4: More code

The code in image four shows the different variants that the Arduino can control. Different ranges for the different parameters have been selected which have been programmed to give different responses depending on live feedback.

After the setup had been completed, I could then move on to the connection of the Arduino ready to transfer the code onto the device ready to evaluate feedback responses. The device came with an adaptable lead which allowed transfer of signals from my laptop which contained the code to the Arduino board.



Figure 5: Connection of the Arduino to software device

Once a successful connection had been determined the IOT cloud was used to start transferring the code that had been written. The “Editor” feature means that the code can be transferred to the Arduino device so, the code constructed previously just needs to be copied and pasted over.

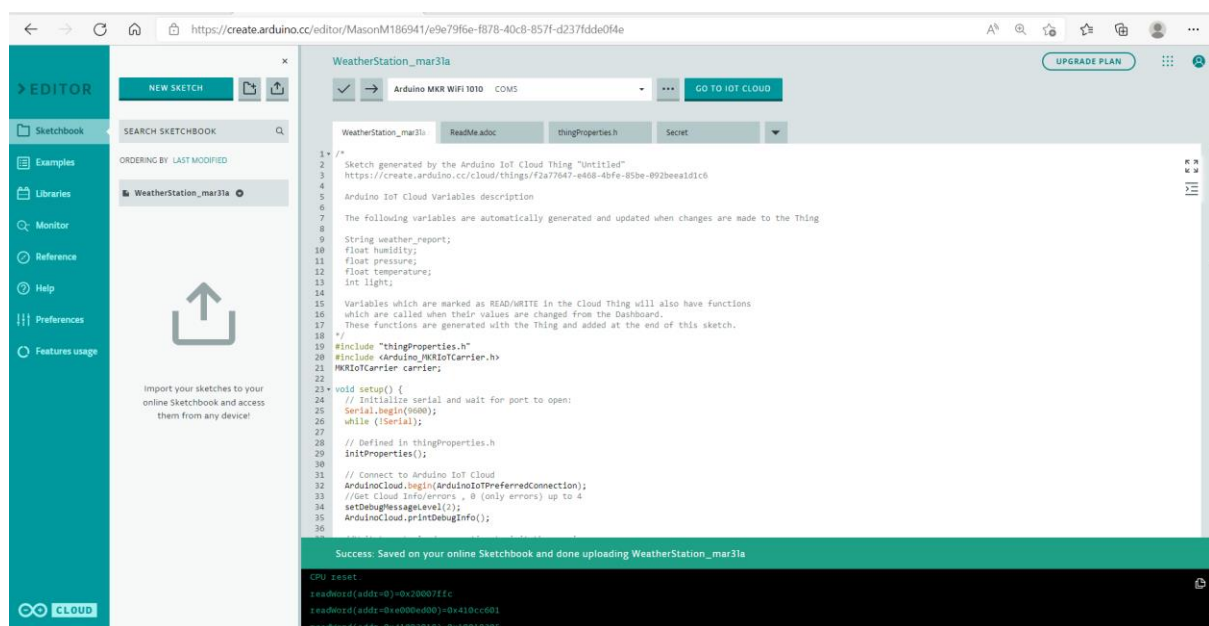


Figure 6: Editor feature for transferring data

Like figure 6 however, this time an image has been shown of the data being successfully transferred over to the Arduino board. The “Monitor” feature has been used to allow the transfer to occur.

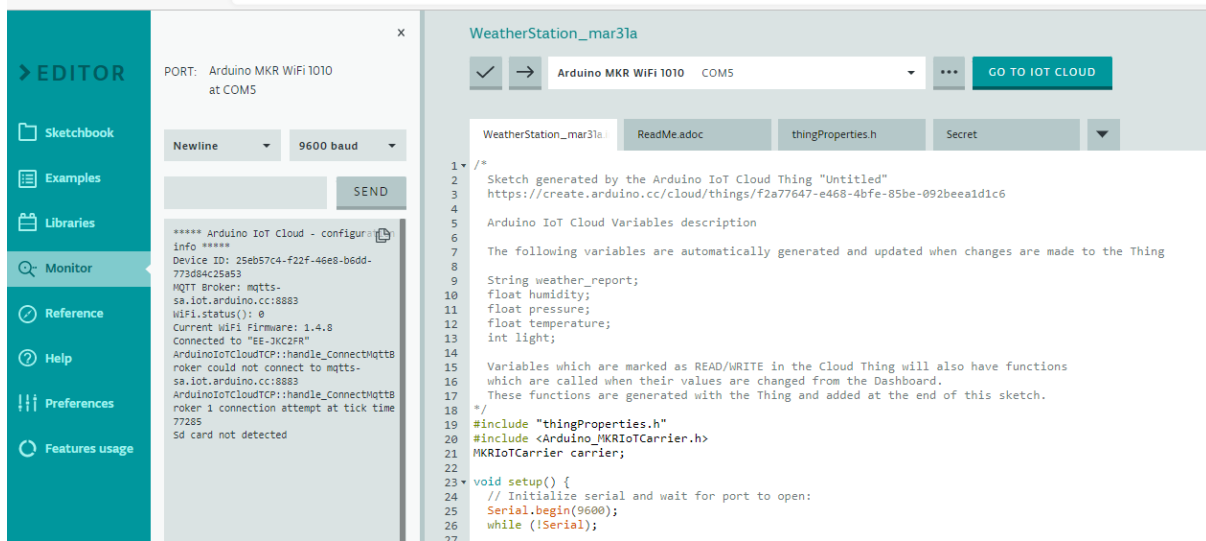


Figure 7: Monitoring the data transfer

A physical image of the Arduino device has been shown below by figure 8. The image clearly shows the data transfer cable being plugged in with an illuminated green LED to signify the device is ready. Upon data transfer you can hear the small relays switching.

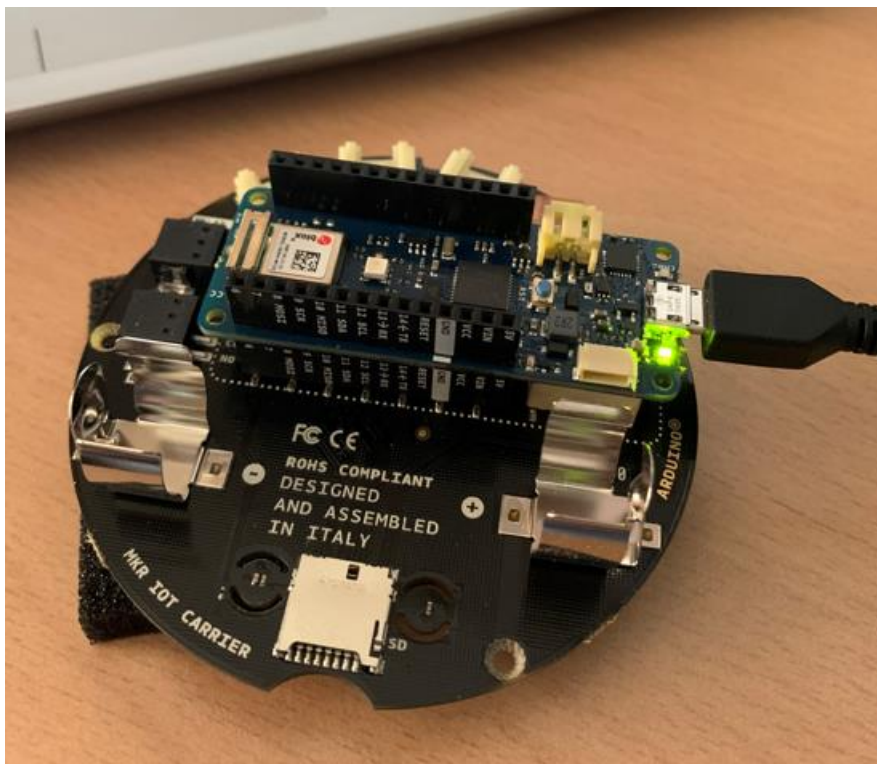


Figure 8: Transferring data to Arduino board

3.4. Create an Arduino Dashboard

Once happy with the transfer a Dashboard can be created to display all data that the Arduino is outputting in a form that we can read.

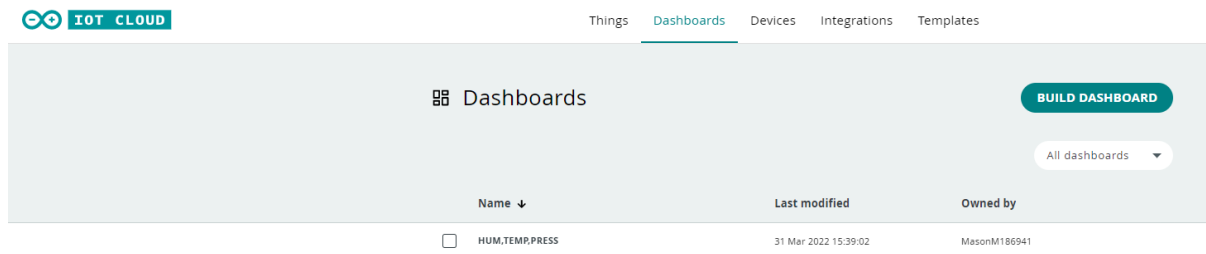


Figure 9: Creating a dashboard

The dashboard can be seen below in figure 10. There are plenty of different dials that can be used to read the data, but I thought it was best to use circles, semi circles and digital displays to show this data. The dashboard also allows for a weather report display which as shown in the code some images ago what sort of margin the weather is in.

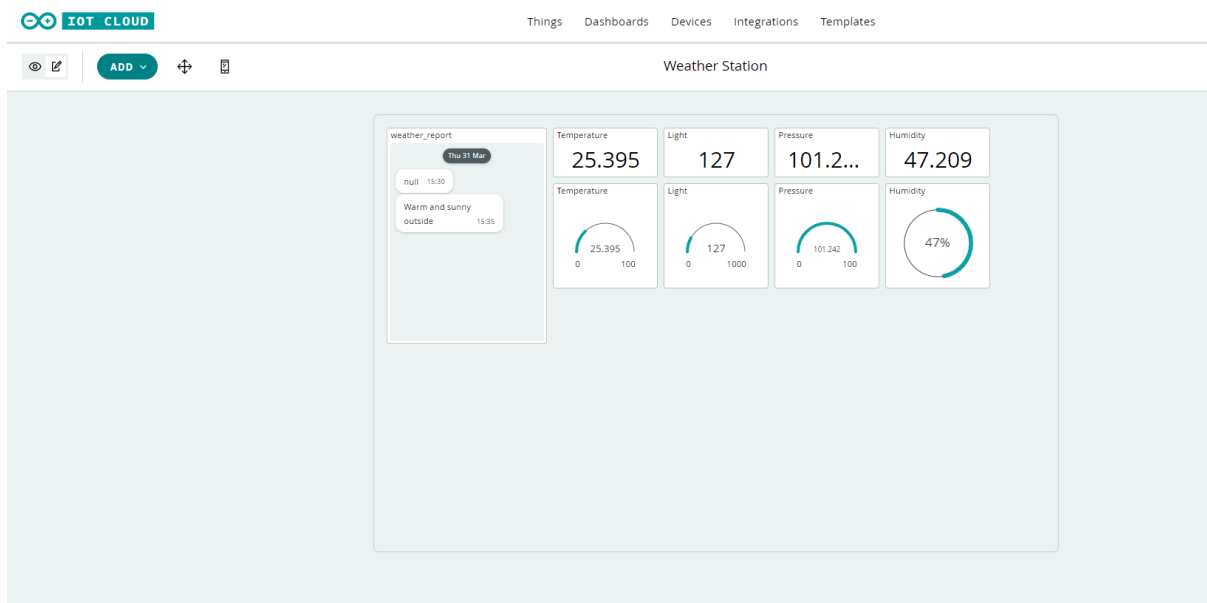


Figure 10: Dashboard display

3.5. Configure and test the system

Once the Arduino was powered a display was shown straight away. Here are five interactive buttons around the outside of the device. As shown, my finger is selecting one of the buttons and the pressure has been displayed with a live figure.



Figure 11: Pressure display value

The next button selected was the temperature reading shown by figure 12.



Figure 12: Temperature display value

Figure 13 shows the Humidity reading.



Figure 13: Live Humidity reading

Figure 14 displays live data for the Light reading.



Figure 14: Live value for light

The next stage for me was to download the Arduino mobile app. The mobile app allowed me to travel away from my Arduino device yet still access the live data.

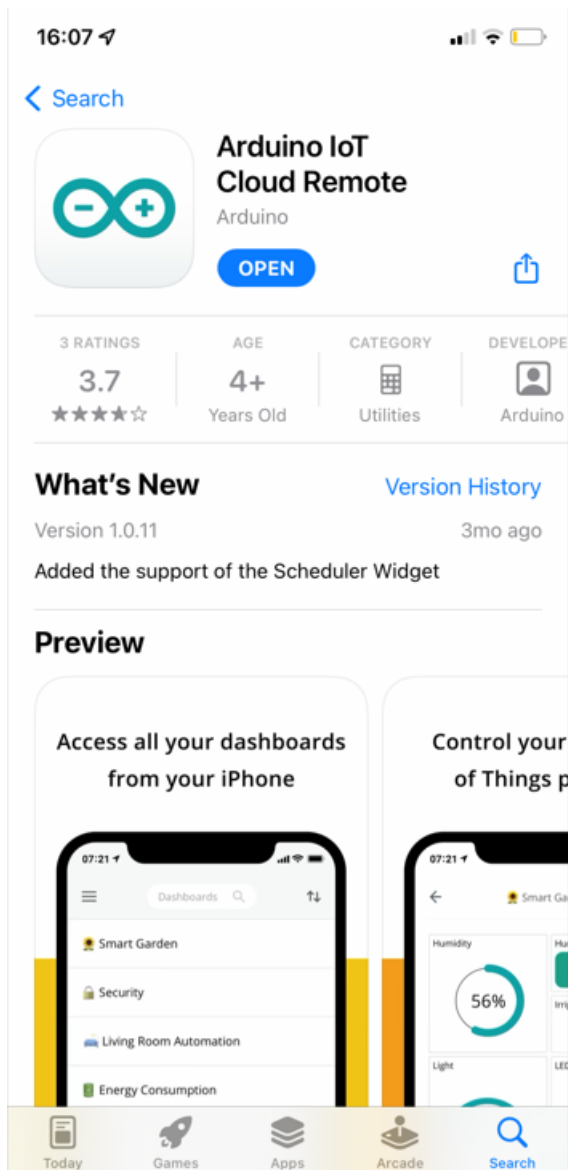


Figure 15: Arduino mobile app

Once I downloaded the app, I started to setup the dashboard on my phone. Like the dashboard that I created previously on my laptop software I again used the weather report along with digital displays and circle and semi-circles.

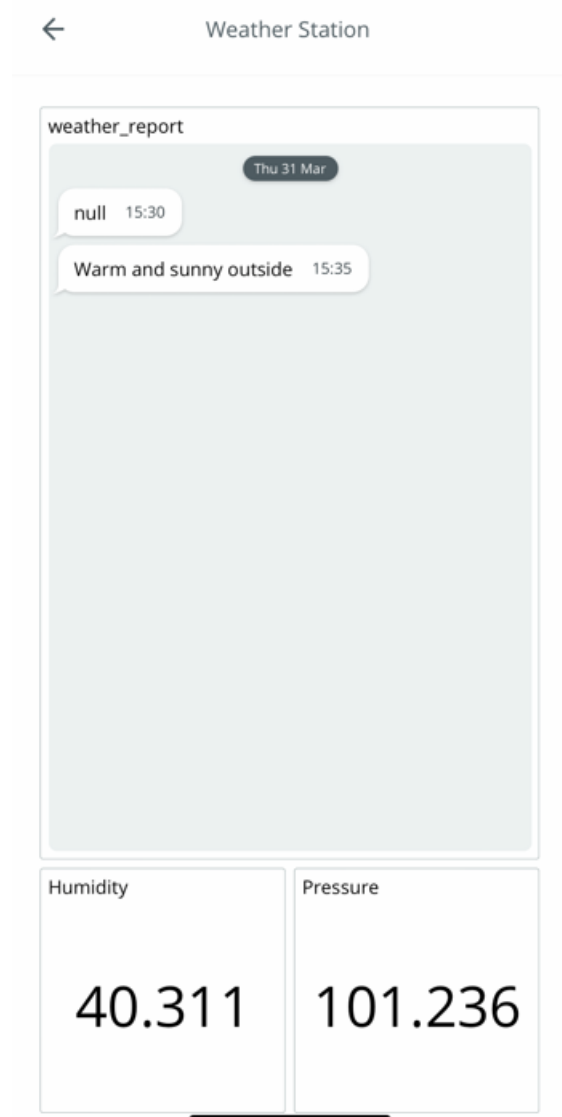


Figure 16: Mobile dashboard

Below in figure 17 the digital displays for Humidity, Pressure, Temperature and Light values can be seen live.

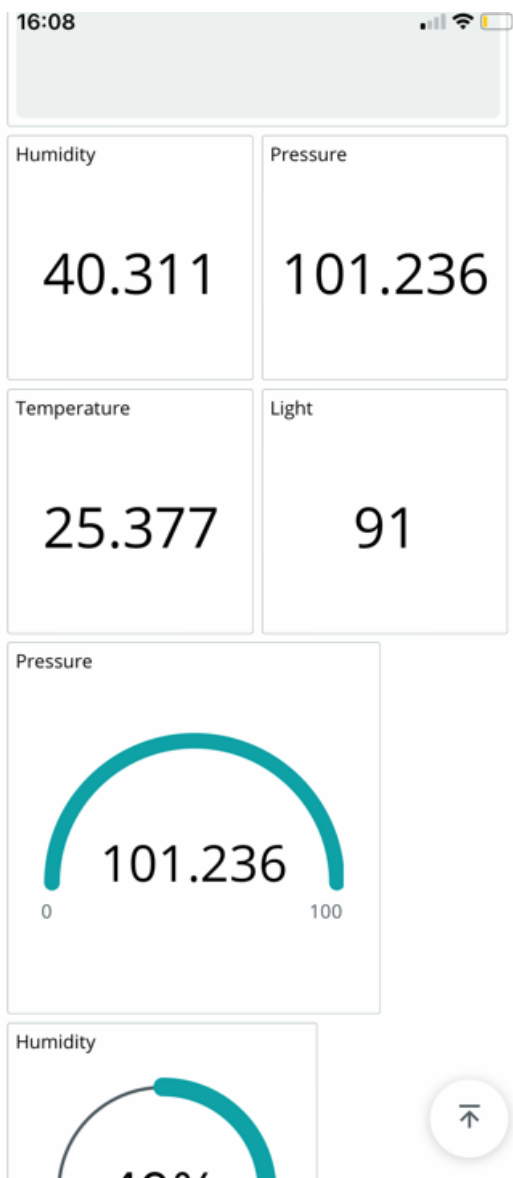


Figure 17: Dashboard configured on mobile

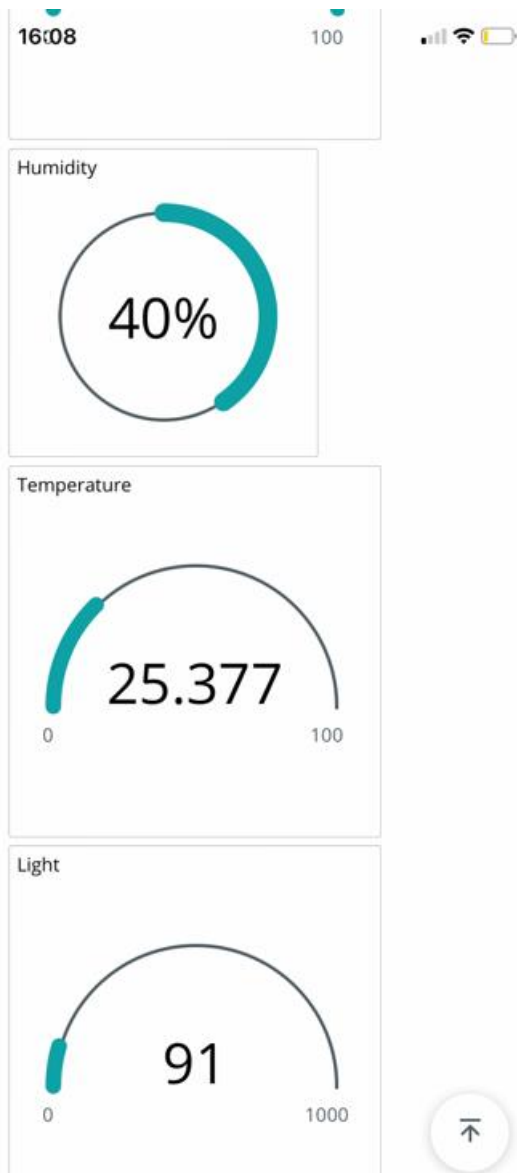


Figure 18: Using graphs on mobile app

3.6. Modify the design

There were some small concerns when setting up the device, some of these were the visibility of the screen, when selecting through each parameter it was hard to tell the difference all the time without properly straining at the small display therefore, I modified the code to alter the colour of the background and the font to allow each parameter to be read better. As shown by the following images this has made a drastic improvement.



Figure 19: Temp display modified



Figure 20: Humidity display modified



Figure 21: Light display modified



Figure 22: Pressure display modified

Figure 21 displays the finished product inside its enclosure next to my laptop with the running programme.

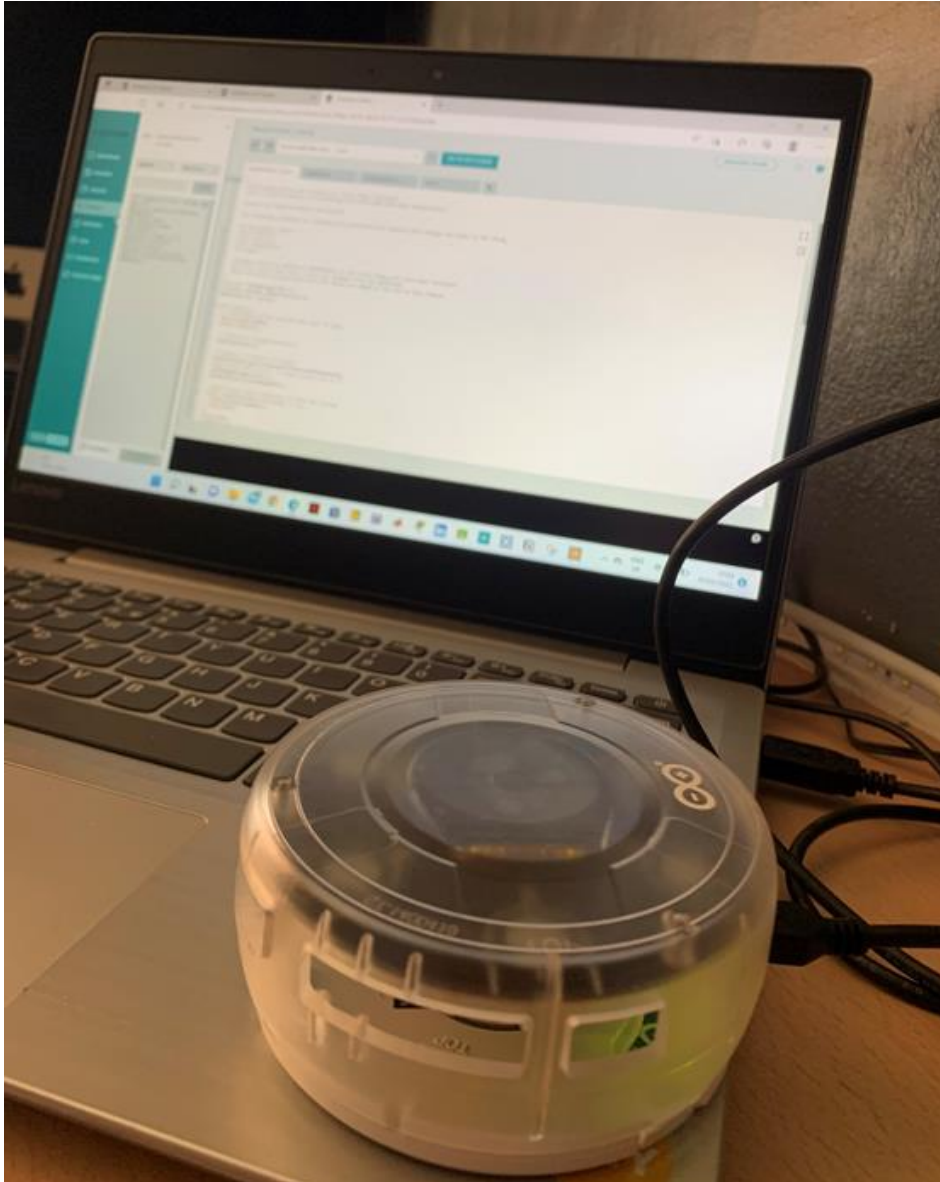


Figure 23: Finished device inside enclosure



Figure 24: Birds eye view of product

4. Conclusions

To conclude, the research was an essential part to this assignment. I can now understand the definitions of IOT. After conducting research into this I now know that the internet of things refers to scenarios where network connectivity and computing capability extends to objects, sensors, and everyday items.

Then going into Enabling technologies this made me understand that recent confluence of several technology market trends, however, is bringing the internet of things closer to widespread reality.

Next onto Connectivity models I was able to understand that there are three common communications used, these being: Device-to-Device, Device-to-Gateway, and Back-End Data-Sharing.

The transformational Potential allowed me to establish that the potential realisation of this outcome a hyperconnected world is testament to the general-purpose nature of the internet architecture itself.

To clarify a lot was learnt during the practical task. To start setting up the device was the first task to overcome. The task involved setting up the device to a laptop and then connecting that device to the internet. Thankfully, there was a lot of help with regards to setting up the device through a step-by-step video. The video was able to make the process a lot easier and at this stage I had my device set up with Wi-Fi capability.

The next stage was to set-up variables for the device which included parameters for the humidity, temperature, light, and pressure. Once these parameters were set-up, I could move onto creating the code for the device.

This involved me taking a copy of the code and pasting it into my editor. From the editor I was able to transfer this information to my device by selecting the monitor option. This taught me that I could now send these functions over to the device through an adaptable lead. I could hear the device switching which allowed me to understand that the data was being transferred successfully. Once the download was complete, I could then see from my laptop that the code had been transferred correctly.

The screen on the device lit up and started to display the data. To make this more realistic I wanted to be able to view the live data of the device from anywhere away from the device.

I then downloaded the mobile app in which I could select different ways of reading the live data from the Arduino. Once the mobile app was setup I felt that the device was in need of some modifications. I started by utilising the code to display different background colours and font colours, this was a much better Idea as before it was hard to read all the parameters.

If I had more time I would put something in the programme which allowed me to read the data when I selected it which was constantly updated, currently once you select the parameter It only gives you that current figure however after a matter of seconds the data is invalid and you have to reselect the value to update it, If I could I'd like to be able to make sure even after selecting the parameter that it's constantly updating it's value.

5. Bibliography

- Channel Futures, 2022. *The Four Internet of Things Connectivity Models Explained*. [Online]
Available at: <https://www.channelfutures.com/best-practices/the-four-internet-of-things-connectivity-models-explained#:~:text=%20The%20Four%20Internet%20of%20Things%20Connectivity%20Models,access%20a...%204%20Backend%20Data%20Sharing%20More%20>
[Accessed 20 04 2022].
- HEWLETT PACKARD ENTERPRISE, 2022. *INTERNET OF THINGS*. [Online]
Available at: https://www.hpe.com/uk/en/solutions/internet-of-things.html?jumpid=ps_5dmn24bsp3_aid-520061462&ef_id=162cace4ba641f54a99fce80b13782c2:G:s&s_kwcid=AL!13472!10!81570104738057!81570231358725
[Accessed 20 04 2022].
- Karen Rose, S. E. L. C., 2022. *IOT INTERNET OF THINGS SOLUTIONS FOR INSIGHTFUL DATA*. [Online]
Available at: https://www.hpe.com/uk/en/solutions/internet-of-things.html?jumpid=ps_5dmn24bsp3_aid-520061462&ef_id=837297629cfb1ba526c0f53bd2d50148:G:s&s_kwcid=AL!13472!10!81570104738057!81570205364921
[Accessed 10 04 2022].
- Project Hub, 2022. *ARDUINO IOT CLOUD*. [Online]
Available at: <https://create.arduino.cc/projecthub/products/arduino-iot-cloud>
[Accessed 20 04 2022].
- University Birmingham, 2022. *What is Enabling Technology*. [Online]
Available at: <https://www.birmingham.ac.uk/research/facilities/enabling-technologies/et-definition.aspx>
[Accessed 20 04 2022].

6. Appendices

FINAL GRADE

GENERAL COMMENTS

Instructor

45/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

GRADING FORM: CSG-RUBIC

MASON MORGAN

0

RESEARCH-INFORMED LITERATURE



There was some evidence of further reading and research and your interpretation and understanding of the additional research was OK. The sources referenced were insufficient at this level given the amount of potential research areas for this assignment and relied on a small selection of mainly inappropriate and non-authoritative sources.

7/20

KNOWLEDGE AND UNDERSTANDING



The knowledge and understanding of the subject demonstrated by this assignment was generally adequate. There was little evidence of any critical understanding. Your report lacked depth and detail at times and didn't really extend your knowledge beyond that contained in the course notes. Further work – Research results need to be in much more detail and written in a more logical and clearer way.

15/30

ANALYSIS



Your analysis of the task was mainly descriptive and superficial and lacked a lot of the required detail. Your conclusions were also mainly descriptive and there were several instances of flawed understanding. There was little evidence of critical analysis with most results accepted uncritically as fact and without explaining the underlying principles. You did however demonstrate some basic analytical skills.

10/30

PRACTICAL APPLICATION AND DEP.



Your application of practical skills for this task was generally very good. You used the resources appropriately when designing the dashboard and you used appropriate options for testing and evaluating the performance. You managed to design an effective dashboard that functioned within the specification and achieved this by only asking for assistance when appropriate thus

7/10

demonstrating some advanced problem solving skills.

SKILLS FOR PROFESSIONAL PRAC.



Your written report was reasonably well presented in terms of content and layout and was mostly structured in a coherent way. You need to be much more selective with where you put diagrams and graphs, only put those essential for the explanation in the report, the rest can be referred to in the appendix.

6/10