Data Logger for solar power using Arduino

Cian Hannon

N00173800

Supervisor: Timm Jeschawitz

Year 4 2020-21

DL835 BSc (Hons) in Creative Media Technologies

Acknowledgments

Throughout writing this thesis I have received a huge amount of help and support.

I would first like to thank my supervisor Timm Jeschawitz, who's knowledge and experience in this area has been unbelievably valuable to formulating my project and my report. Your in-depth feedback and suggestions pushed me to expand my work and report to the best of my ability and brought my work to a completely higher level.

I would like to acknowledge all the lecturers and staff at IADT for their wonderful participation and work in guiding me to where I am today. The knowledge and skills that I have learned throughout my 4 years would not have been possible without them. I want to kindly thank you all for your extreme patience and support you have provided for all the opportunities I have been given throughout my time here.

I would also like to thank my lecturers, Paul Comiskey, Conor Brennan, Shane Byrne, and Louise Glynn for their valued guidance from the past and current years I spent studying here. You all provided me with the tools and guidance to point me in the right direction to complete my thesis successfully.

Lastly, I would like to thank IADT for shaping me as a person and giving me a chance to progress in life and figure out what I love.

Abstract

The world is suffering from harsh methods of powering and heating your home as a lot of these methods cause air pollution. They also cost a lot more then solar power. It has been proven that incorporating solar power into powering and heating your home is guaranteed to have major benefits to the eco system. It is also a way more inexpensive way of powering and heating your home. The study aims to figure out how much power just a small 12V solar panel can produce in a few hours in Ireland. Solar power is the next big thing in producing energy for homes and is 100% ecofriendly as all it requires is rays from the sun. People who invest in solar power will find they are saving money in the long run.

To test the hypothesis readings are taken and the data results are compared from start to finish to see how much the energy produced can fluctuate over the time. Multiple soldering methods are used to solder all the wires and components together. Arduino IDE coding methods are used to write and upload code onto the Arduino board. The data is organized in an excel spreadsheet and is visually represented in Tableau. The results showed that there was an average of around 8V being produced from the solar panel with a maximum of 12V. This shows the effectiveness of solar panels even with a small solar panel that was used for this project it produced a high consistent voltage. The solar panel worked correctly as it should.

These results show that switching to solar power has more benefits that disadvantages. The panels are high costing initially but long term you will save money from bills and you are contributing to helping repair the eco-system. The panels are highly effective and easy to install. All readings and testing were done to replicate the panel on a roof at 45 degrees to ensure accurate results.

Table of Contents

Acknowledgments 2

Abstract 3

Table of figures 5

1. Introduction 7

2. Research 8

2.1 Technologies 8

2.1.1 Arduino 8

2.1.2 PV cell/ solar panel 9

2.1.3 Data logger shield 9

2.1.4 LED 10

2.1.5 Excel 10

2.1.6 Tableau 10

2.2 Applications 10

2.3 Requirements 11

2.3.1 Similar systems 11

2.3.2 List of requirements 11

3. Design 12

3.1 Diagrams and schematics 13

3.1.1 Block diagram 13

3.2 Schematic 14

3.3 list of components 15

3.4 Software design 16

3.4.1 Arduino IDE 16

3.4.2 Excel 18

3.4.3 Tableau 18

3.5 Conclusion 19

4. Implementation 20

4.1 Safety 20

4.1.1 Fire/burns 20

4.1.2 Electrocution 20

4.1.3 Fumes 20

4.1.4 Cuts/ scrapes 20

4.2 Prototype 1 21

4.2.1 Process 21

4.2.2 Features 22

4.2.3 Issues 22

4.2.4 Resolution of issues 22
4.3 Prototype 2 23
4.3.1 Process 23
4.3.2 Features 38
4.3.3 issues 38
4.3.4 Resolution of issues 38
4.4 Conclusion 38
5. Testing and results 38
5.1 Test set up 39
5.2 Hardware testing 40
5.3 Software testing 41
5.4 Conclusion 42
6. Conclusion 43
7. References 45
Appendix 47
'Low Voltage Directive' Compliance Statement. 47
'Medium Risk' 48

Table of figures

Figure 2.0 Cave pearl project 11
Figure 2.1 Log interval value 12
Figure 3.1 Block diagram, start to finish process. 13
Figure 3.2 High level schematic 14
Figure 3.3 Arduino code flow chart 16
Figure 3.4 The defines 16
Figure 3.5 CSV file is created. 17
Figure 3.6 Excel readings 18
Figure 3.7 Overall comparison example. (DataFlair,2019). 18
Figure 3.8 Line graph comparison. (Gateway, 2021). 19
Figure 4.0 Fritzing prototype circuit layout 21
Figure 4.1 Prototype 2 setup 23
Figure 4.2 Data logger mounted. 24
Figure 4.3 Male headers pins to attach the boards. 25
Figure 4.4 LED pins soldered. 26
Figure 4.5 Voltage divider 27
Figure 4.6 Panel wires stripped. 28
Figure 4.7 Solar panel pins 29
Figure 4.8 Wires soldered to the solar panel pins. 30
Figure 4.9 Solar panel connected to voltage divider. 31
Figure 4.10 Libraries and defines 31
Figure 4.11 Defining the RTC and error message. 32
Figure 4.12 Starting the logging 33
Figure 4.13 Error messages and spreadsheet column names 34

Figure 4.14 Data values and date and time organised pt.1 ... 34
Figure 4.15 Data values and date and time organised pt.2 ... 35
Figure 4.16 CSV file data readings ... 36
Figure 4.17 Tableau data dashboard ... 37
Figure 5.0 Circuit on breadboard ... 39
Table 1 Hardware testing ... 40
Table 2 Software testing ... 41

1. Introduction

The Data logger for solar is a device that automatically records and monitors the parameters over a period. The information is gathered from the solar panel and is stored in an excel sheet on the SD card. The data is then sorted into columns in the excel spreadsheet with the main columns being the date and time, voltage, current and power. The results from these columns are then imported into the Tableau software app to be visually represented into line graphs and bar charts. Time can be adjusted in between logs by just changing the Millis in the Arduino IDE code. The data taken can let the user know how much energy their solar panel generates on any certain area of their home. Using the data taken the user can find out where the best place for the solar panels to be placed to ensure getting the maximum potential out of their panel.

The introduction Is followed by the research chapter which describes the area of research and explains in detail how the technology works and how it plays it part in the project.

The research chapter is followed by the design chapter which shows you the overall design. High level and low-level schematics are constructed based on the project. The code used for this design is shown on this chapter.

The implementation chapter details the instruction of the hardware and the development of the software. The chapter runs through the safety procedures of the project and shown how the project has progressed throughout each prototype.

The testing and results chapter shows the tests ran on each component and a test on the device. Results from the data logging taken.

2. Research

The research chapter describes the area of research and explains in detail how the technology works and how it plays it part in the project. This project is based around data logging. Data logging is data that is collected over a period. It is usually done automatically as it is more capable and more time efficient this way. The data can be monitored on a software app such as excel, and the values can be organised. (Tag, 2019).

The aim is to conduct sufficient research so that reliable and inexpensive components can be purchased. The correct Excel formulas must be created in order for the data to be able to be logged Multiple readings taken from the solar panel to be able to organise and show enough results. Using specific research, the prototype of the Data logger will be fully functional by the end of the build.

This cost affordable data logger monitors real time readings on the PV panel taking the voltage, current and power. Using an Arduino which is low cost will be the main piece of this project. Everything that will be bought will be cheap and homemade if possible. PLX-DAQ data will be taken on excel to assist with the formula layout.

All logged data in excel will be imported into Tableau to put the information into graphs to see the results easier (Tableau Software,2020). Tableau is a free to use software app that allows you to broadly visualize data in a more in-depth way with dashboards multiple tools etc.

2.1 Technologies

There are multiple technologies that can be used to create a data logger. There are multiple micro controllers that can be used for logging data. Microcontrollers are small circuit designed devices that allow the user to give it a specific command or function to operate to their needs. A microcontroller consists of a processor, memory, input, and output peripherals. (Lutkevich, 2019). In this case the microcontroller is given a function to create a CSV file and plot the date and time, voltage, current and power onto the spreadsheet.

Adafruit, Arduino, Raspberry pi, Altera, and Atmel are all microcontrollers that can be used for this data logging device. (List of common microcontrollers. 2021). Arduino UNO R3 is the microcontroller that is going to be used for this system as it comes with its own coding software Arduino IDE which makes it easier to upload code onto the microcontroller.

2.1.1 Arduino

Arduino is an electronic platform that provides free use to easy hardware and software services. For this project, the Arduino UNO R3 and Arduino IDE are used. The Arduino UNO R3 was chosen as deep research was done to choose one of the cost affordable and high-performance boards that they offer. The UNO R3 is very inexpensive and is an open-source device. The Arduino reads the inputs from light on a sensor and turns it into an output for this project. This creates data and transfers it to a CSV file inputted by the user in Arduino IDE. The Arduino can also read buttons pressed and messages and turn them into outputs too and gives them a function e.g., Turns on an LED.

You give the board a function by using code in Arduino IDE to send information to the microcontroller on the board. (Arduino, 2021).

Arduinos are little microcontrollers that are used to make interactional projects as (Team, T, 2020) points out "The Arduino can sense different parts of its environment by receiving inputs from sensors, and then based on those inputs, can change its surroundings by controlling lights, motors, and other devices. The software can be written for the Arduino to instruct it what to do in different situations" (pg.1)

Another one of their brilliant ideas was to make the software free to all as students cannot afford to pay for software. They decided if you want to pay you can pay what you want or nothing at all. It was made for anyone who wanted to work with interactive objects. The Arduino can interact with many things from buttons and motors to cameras and phones. The list just goes on.

The Arduino boards are a fairly simple design, just like the software is easy to pick up and learn hence why there is such a big community of people who have added and modified the code making it even easier to make your desired project.

2.1.2 PV cell/ solar panel

Also known as a solar cell, this component generates electricity from light or photons. This is a more environmentally friendly way to power something. Solar panels are becoming more common in newly built homes.it is also cheaper in the long run.

How it works?

Firstly, the panel absorbs the photon cells(light), and the electrons separate. The separated electrons then flow through the panel which makes current and then flows through the wires and into its desired destination (e.g., homes).

2.1.3 Data logger shield

The data logger saves the data files retrieved from the PV panel onto the SD card, which will then be transferred easily onto excel or any type of spreadsheet. It will record the Time, voltage, power and current. This allows the user to know what happened and a certain time.

The data logger is highly effective because even when the Arduino is turned off the logger can keep track of the time. The Realtime clock has a battery holder to power it when the Arduino is not on using a coin cell battery.

There is an SD holder on shield that can fit an SD card up to 32GB which is more than enough space for what is needed. There is an SD LED on the device that activates The LED when the data transfers by the SPI. This means that you can insert or remove the SD card. Beside the holder is a level shift which acts as a resistor in a way. Any signal that is in between 3.3 and 5V will be lowered down to 3.3V so the SD card will not be damaged.

2.1.4 LED

An LED (Light Emitting Diode) is a semiconductor source that produces light when current is flowing through it. LED's come in almost any colour you can think of. The light is emitted as a form of energy as atoms called photons. In an LED there are released in a result of electrons. This happens when an electron drops from high energy levels to low energy to low energy levels. Meaning it loses energy emitting photons which means it lights up.

2.1.5 Excel

Excel is a free to download software program. You can make your own spreadsheets to organise your data. The data taken from the SD card will be put onto a spreadsheet. From here the user then puts in their own formulas and sets up their tables.

2.1.6 Tableau

Tableau is a free to download software for data visualisation. It allows you to transform your collected data and put them into a visual format. From bar charts to scatter plots there are multiple ways to layout your data whatever way you want (Tableau Software, 2020). This application will help visualize the data taken from this solar panel as it allows broad in-depth changes to the data letting the user make all kinds of graphs to plot the information clearly and properly coloured and labelled.

2.2 Applications

Solar panels can be a substitution for almost any energy generation source. They are used all around the world and mostly in china with a capacity of 175,019 MW. A lot of homeowners are starting to use solar panels instead as it is reducing their monthly bills on electricity. So, for this project we will be able to see the amount of electrical energy the panel generates, and the data taken will show how much you could be saving by using this panel alone. Depending by how many panels the user wants to use you just multiply your results by the number of panels being used.

Businesses which also require a lot of power for example large supermarkets. These supermarkets use solar panels to play their part in being as eco efficient as possible. The main user would be homeowners that would require any sort of electrical energy. Not only are they saving money but there also saving time as the data logger system already works out the numbers for them so they can see how much and how little power they would have at any given time by just opening excel.

2.3 Requirements

2.3.1 Similar systems

Black box for trucks:

Just like the data logger for Arduino, this black box logs the data of the trucks GPS, acceleration, speed, wind speed, temperature and many more. This is where the idea came to log the data that the PV panel generates.

There are black boxes on planes helicopters and some cars too. It also logs phone calls and text messages in case of an accident the black box can be retrieved, and you can see by the data what happened. (Van De Berg, 2017).

The cave pearl project:

In short term this device is a water flow sensor. This is where the idea came to use an Arduino to code the device as it is cheap and easy to use. The device is used to monitor flooded caves and costal water around the world using sensors and to use the data to see what can be done to reduce or improve the waters.

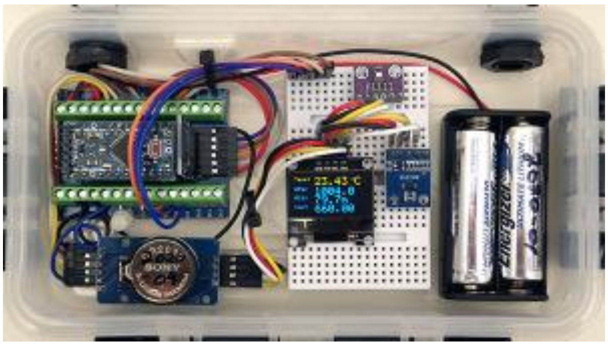


Figure 2.0 Cave pearl project

Figure 2.0 shows a prototype of the cave pearl project and you can see it does not need to be big or bulky and shows its simplicity. (Mallon, 2017).

2.3.2 List of requirements

Safe use of machinery is required to prevent injury. Material is going to have to be drilled and cut the make the stand for the PV panel and a home for the data logger.

Functional requirements:

- The ability to operate Microsoft excel and creates formulas for your data set. The date and time columns must update to whatever second in between each log is set to on the Millis code. Also, to be able to organise your data and transfer it.
- Tableau is an essential part to make your data easily visualised by the user. Use of tableaus systems is needed to complete the final part of the project.

- You need to be able to operate an Arduino properly including wiring up the Arduino to the circuit and to create the appropriate code for your project. The Arduino must power the data logger and send the code to it. Making the data logger log everything into the SD card. This is mandatory.

Non-functional requirements:

- Data imported into Tableau from Excel is visualised with line graphs and bar charts.
- To change the time between logs, the Log interval number can potentially be changed according to the time in between logs that is desired in Millis as shown in figure 2.1.

```
// how many milliseconds between grabbing data and logging it. 1000 ms is once a second
#define LOG_INTERVAL  1000 // mills between entries (reduce to take more/faster data)
```

Figure 2.1 Log interval value

2.4 Conclusion

The research chapter described the area of research and explains in detail how the technology works and how it plays it part in the project. The suitable microcontroller was chosen to suit the project with its ease of use and effectiveness. By this stage, the research done in the technologies section will provide enough information on what suitable parts are needed to complete this project. With all the essential research done the first prototype will be ready to be made and the design process should start. With the research that was done the user should be comfortable with building this project and feel that they have been provided the appropriate plan to approach this project.

3. Design

The design chapter shows in detail the start to finish process on how it works using block diagrams and showing components function. Schematics and components will be shown, and their functions are described. The use of tableau software, Excel, and Arduino uno coding applications will be describes with their role in the project.

The design will meet the requirements from the research chapter in terms with the build. The aim is to have a complete prototype and to improve from previous prototypes ensuring the data logger is completed.

3.1 Diagrams and schematics

3.1.1 Block diagram

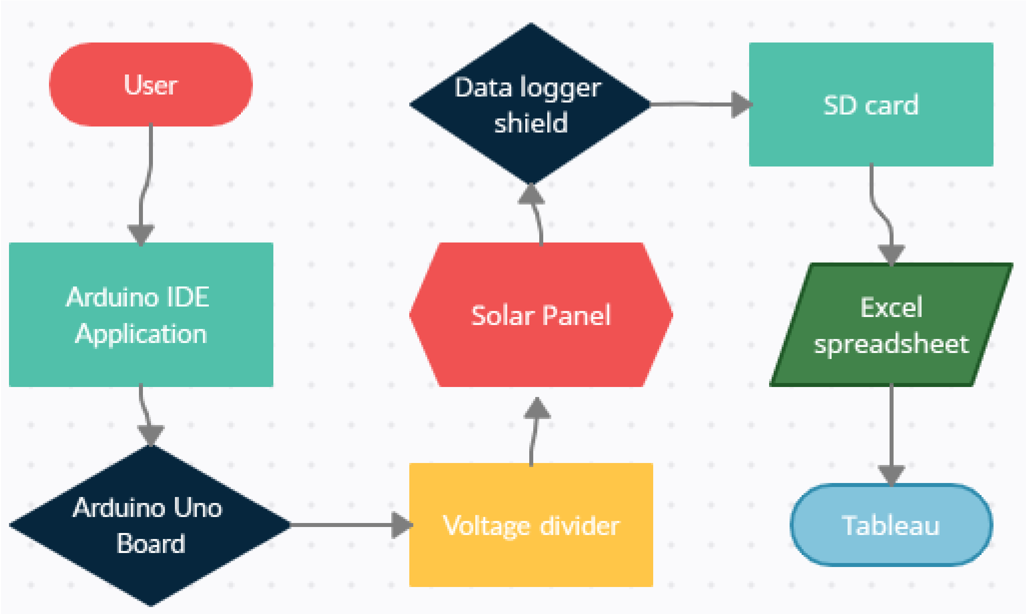


Figure 3.1 Block diagram, start to finish process.

Figure 3.1 is a block diagram of the start to finish process of the data logger shield. It starts off with the user uploading and running the code from the Arduino IDE app onto the Arduino uno board. The user names the csv file that the code is going to create and sets all the define values to suit their needs. From there the Arduino is connected to the voltage divider. The voltage divider Insurers that 5V is provided to one of the Arduino's analogue pins when the solar panel reaches its (MPP) maximum power point using two resistors.

The solar panel is connected to each of the resistors on the voltage divider. When the solar panel generates energy all the information is sent back to the data logger shield by the log interval value inputted by the user which is connected on top of the Arduino IDE board using male and female headers. All this information is then stored on the SD card. An excel spreadsheet is created in the SD card using the "create file" code and the information is split into columns with the date and time, voltage, current and power printed. When sufficient information is taken it will be transferred onto Tableau to be visually represented with the data taken from the Excel spreadsheet.

3.2 Schematic

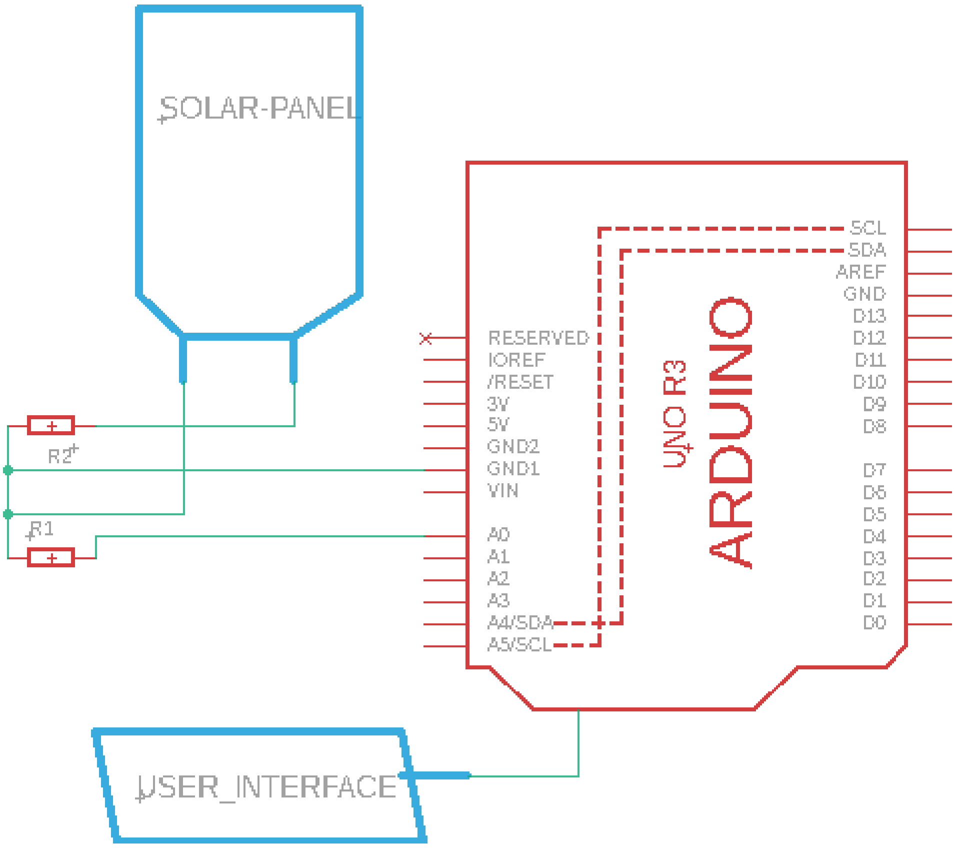


Figure 3.2 High level schematic

Figure 3.2 shows the high-level schematic for the data logger. It starts with the Arduino being powered with a power supply or laptop etc. The voltage divider is connected in between the Arduino and solar panel. The voltage divider is made on a breadboard for this build.

R1 is connected to the analogue pin 0 and connected to ground 1. R2 is connected to ground 1 and the negative pin on the solar panel. The positive pin on the solar panel is connected to R1.

The Two LED pins (L1, L2) are connected to the 2 and 3 digital write pins. This is to show us if the SD card is inserted or not present.

3.3 list of components

Materials:

- PV cell (solar panel). Off grid tec solar panels are the most reliable and cheap. (Technology,2013).
- Arduino uno R3
- Adafruit Data logger shield. (Earl, 2013).
- Breadboard
- Male to male wires
- Male to female header pins
- Male to male header pins
- 2 resistors
- 0.75mm cables
- Crimps
- 3 button USB cable
- SD card
- Coin cell battery

Software:

- Excel
- Tableau
- Arduino IDE
- SD card formatter

Tools:

- Soldering iron
- Wire strippers
- Crimping tool
- Pliers
- Screwdriver

3.4 Software design

3.4.1 Arduino IDE

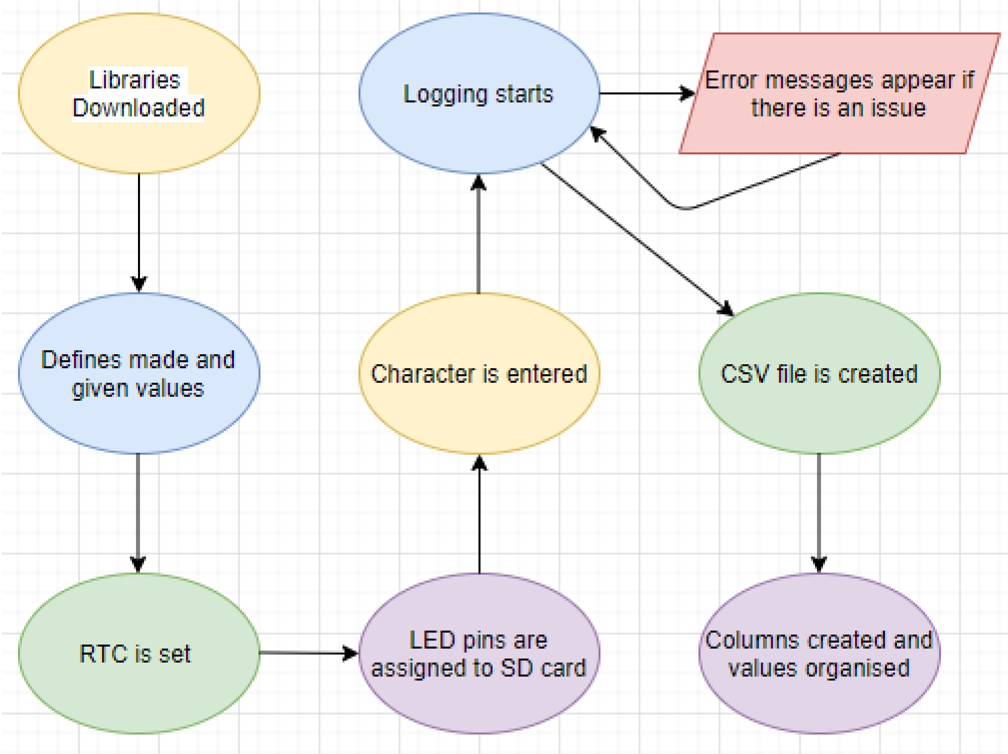


Figure 3.3 Arduino code flow chart

Arduino IDE is used to upload the code onto the Arduino board. Examples of code that will be used is shown below and in figure 3.3.

```
// how many milliseconds between grabbing data and logging it. 1000 ms is
#define LOG_INTERVAL  1000 // mills between entries (reduce to take more/f

// how many milliseconds before writing the logged data permanently to dis
// set it to the LOG_INTERVAL to write each time (safest)
// set it to 10*LOG_INTERVAL to write all data every 10 datareads, you cou
// the last 10 reads if power is lost but it uses less power and is much f
#define SYNC_INTERVAL 10000 // mills between calls to flush() - to write d
uint32_t syncTime = 0; // time of last sync()

#define ECHO_TO_SERIAL   1 // Echo data to serial monitor (Setting it to 0
#define WAIT_TO_START    1 // Wait for serial input in setup() (Setting it

// the digital pins that connect to the LEDs (Solder pads L1 (green) and L
#define redLEDpin 2
#define greenLEDpin 3

// The analog pin connected to the PV cell voltage divider
#define sensorPin A0
```

Figure 3.4 The defines

In figure 3.4 are all the defines. These are all the lines of code that can be adjusted. Log interval is the number of milliseconds between the readings on the solar panel. Echo to serial sends the echo data to the serial monitor. 1 means on and 0 means off. Red LED pin means the red led on the data logger shield is assigned to pin 2 and the same for the green LED pin. The sensor pin is assigned to analogue 0.

```
#if WAIT_TO_START
  Serial.println("Type any character to start");
  while (!Serial.available());
#endif //WAIT_TO_START

  // initialize the SD card
  Serial.print("Initializing SD card...");
  // make sure that the default chip select pin is set to
  // output, even if you don't use it:
  pinMode(10, OUTPUT);

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    error("Card failed, or not present");
  }
  Serial.println("card initialized.");

  // create a new file
  char filename[] = "DATA_readings.CSV";
  for (uint8_t i = 0; i < 100; i++) {
    filename[6] = i/10 + '0';
    filename[7] = i%10 + '0';
    if (! SD.exists(filename)) {
      // only open a new file if it doesn't exist
      logfile = SD.open(filename, FILE_WRITE);
      break;  // leave the loop!
    }
```

Figure 3.5 CSV file is created.

A character is typed into the serial monitor to start the file creating process. The SD card initializes, and the CSV file is created in the SD card. This gives the data taken from the solar panel a destination in the SD card as shown in figure 3.5.

RTC (real time clock) and SD card libraries must be downloaded and included for the project to work These were retrieved from the Adafruit tutorial page (Earl, 2021).

The real time clock library download:

https://learn.adafruit.com/adafruit-data-logger-shield/using-the-real-time-clock

The SD card library download:

https://learn.adafruit.com/adafruit-data-logger-shield/using-the-sd-card

The project must be fully built and working for the code to work as it needs to verify each component that is being used. A coin cell battery is vital as the shield wont function properly without it. The battery does not have to be working it just needs to have one in its slot.

3.4.2 Excel

The data taken from the panel is imported into an automatic CSV file and organised into a spreadsheet with the main data readings being date and time, voltage, current, and power.

millis	stamp	datetime	voltage U2 [V]	voltage U [V]	current I [A]	power P (W)	energy [Ws]	R1 [Ohm]	R2 [Ohm]
999	1461434035	23.4.2016 17:53:55	0.13	0.48	0.01	0.00	3989.47	48.5	18
1999	1461434036	23.4.2016 17:53:56	0.13	0.48	0.01	0.00	0.00		
3000	1461434037	23.4.2016 17:53:57	0.12	0.44	0.01	0.00	0.00		
3999	1461434038	23.4.2016 17:53:58	0.12	0.44	0.01	0.00	0.00		
5000	1461434039	23.4.2016 17:53:59	0.10	0.37	0.01	0.00	0.00		
5999	1461434040	23.4.2016 17:54:0	0.32	1.18	0.02	0.02	0.01		
7000	1461434041	23.4.2016 17:54:1	0.49	1.81	0.03	0.05	0.04		
7999	1461434042	23.4.2016 17:54:2	0.52	1.92	0.03	0.06	0.05		
8998	1461434043	23.4.2016 17:54:3	0.52	1.92	0.03	0.06	0.06		
9999	1461434044	23.4.2016 17:54:4	0.53	1.96	0.03	0.06	0.06		
10999	1461434045	23.4.2016 17:54:5	0.54	2.00	0.03	0.06	0.06		

Figure 3.6 Excel readings

All the values taken from the solar panel are transferred to the excel spreadsheet on the SD card. In figure 3.6 you can see the values of each unit. Chunks from different dates and times will be taken and compared.

3.4.3 Tableau

Tableau is used to visually represent data sheets. Columns of information from the excel spreadsheet will be inputted into this application and compared using graphs and bar charts etc.

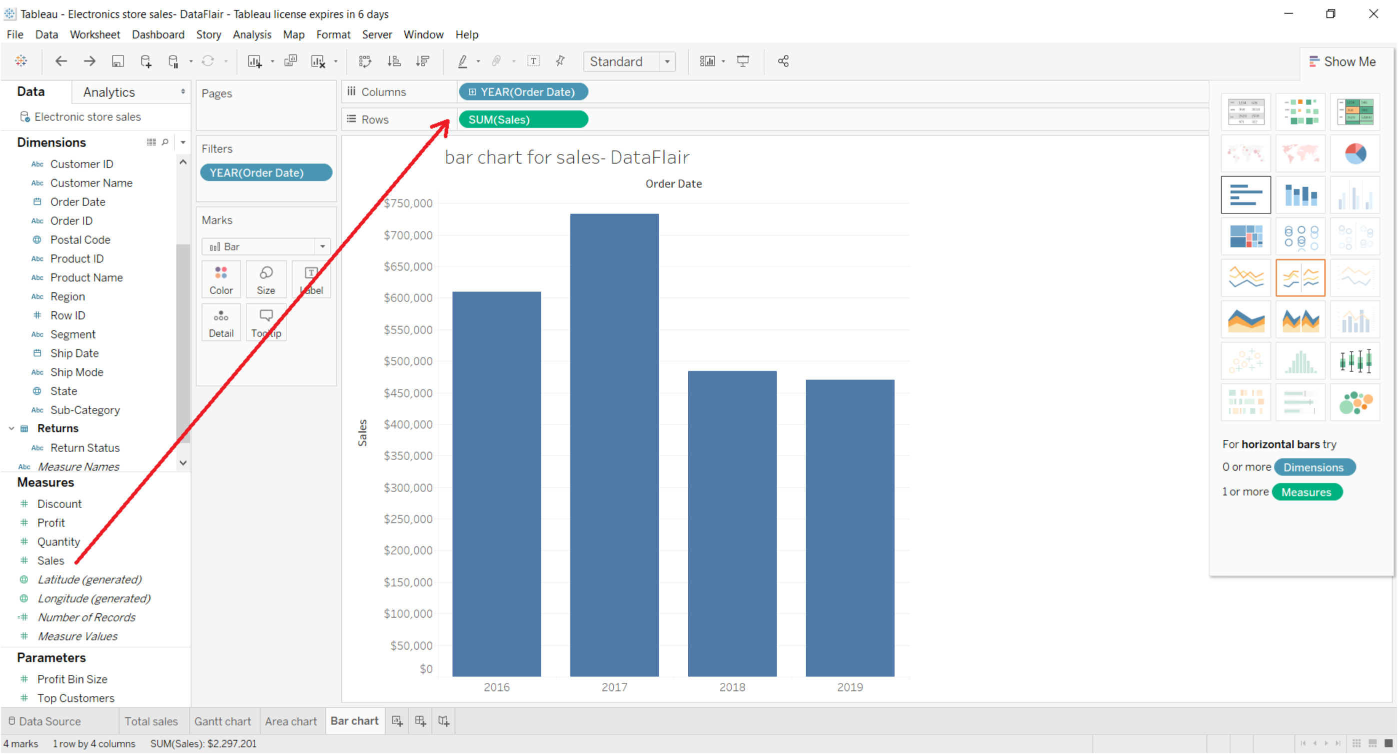


Figure 3.7 Overall comparison example. (DataFlair,2019).

Figure 3.7 is the overall comparison from data taken each year for sales. The data is compared side by side to give an easier view of the differences between the 4 data readings. This is how the data visualisation will be set up for all the data taken. This is how the visualisation should look when the data is imported into tableau with the further prototypes. The colour will be changed in the prototype to make the charts easier to tell apart and signing a colour with a value e.g., Blue is current.

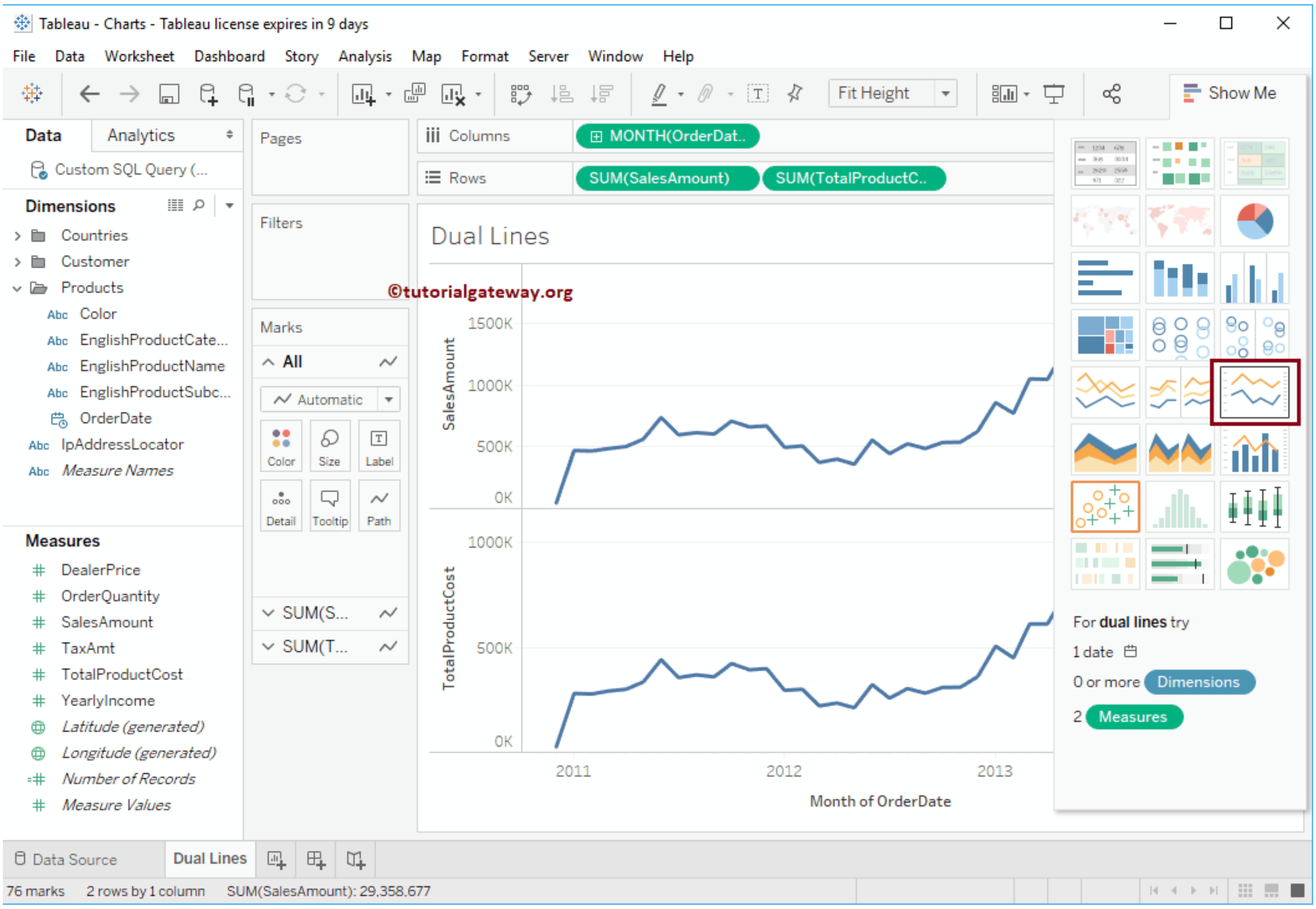


Figure 3.8 Line graph comparison. (Gateway, 2021).

Figure 3.8 shows the line graph comparison of the 2 data sets with a value every month for 3 years. In this example each data type is the same colour. In the final project the line graphs so that the chart is easier to read and identify. The sizes are adjusted so each line is a similar size to see the comparison easier. The lines are relatively the same, but the slight differences are clear due to using the size tool. This is how the data visualisation will be set up for all the data taken. This is how the visualisation should look when the data is imported into tableau with the further prototypes.

3.5 Conclusion

The block diagram and the schematic show you how the data logger operates from start to finish. Each part of the process is explained in detail including the software side of things. Every line of code is commented to give a full understanding of how it works. The list of components is outlined in case anyone else wants to give it a go. The Tableau chart examples show what the visualised data will look like.

4. Implementation

4.1 Safety

Safety during project construction is hugely important. Multiple accidents can happen in the workplace when working on a hands-on project. This list is used to remind the user the dangers and precautions to be taken to avoid these. Several the hazards listed below are from the low-risk standard risk assessment document related to the data logger.

4.1.1 Fire/burns

In this project a soldering iron is the main fire/burn hazard. Do not touch the tip when the iron is plugged in as the iron can reach up to 400 degrees. Soldering wires together can be a task as it is extremely easy to burn your hand with the iron accidentally. Also, the wire could be extremely hot from the iron and will be uncomfortable to hold as it will burn your hand. A helping hand is a good tool to use when soldering wires or components together to reduce the chance of endangering yourself.

4.1.2 Electrocution

A chance of being electrocuted from constructing this project is possible. Ensure all wires are covered and there is no exposed wiring. When soldering and connecting wires make sure that the device is turned off and not powered. Wear gloves while working with live wires to prevent and shocks. Always make sure the power supply Is turned off or plugged out when working with the wires, components etc.

4.1.3 Fumes

Fumes from lead solder or any other solder is not safe to breathe in. these harmful fumes can lead to serious permanent damage and could be detrimental to people with raspatory issues. The fumes can also be harmful to your eyes and could possibly reduce your vision. To prevent these hazards the use of safety goggles and a face mask would majorly reduce the chances of any harm caused.

4.1.4 Cuts/ scrapes

When cutting wires and components there is a risk of cutting or pricking yourself. Be cautious of any wires or components sticking out of the board as they could prick you. Safety gloves are recommended to avoid any of these hazards.

4.2 Prototype 1

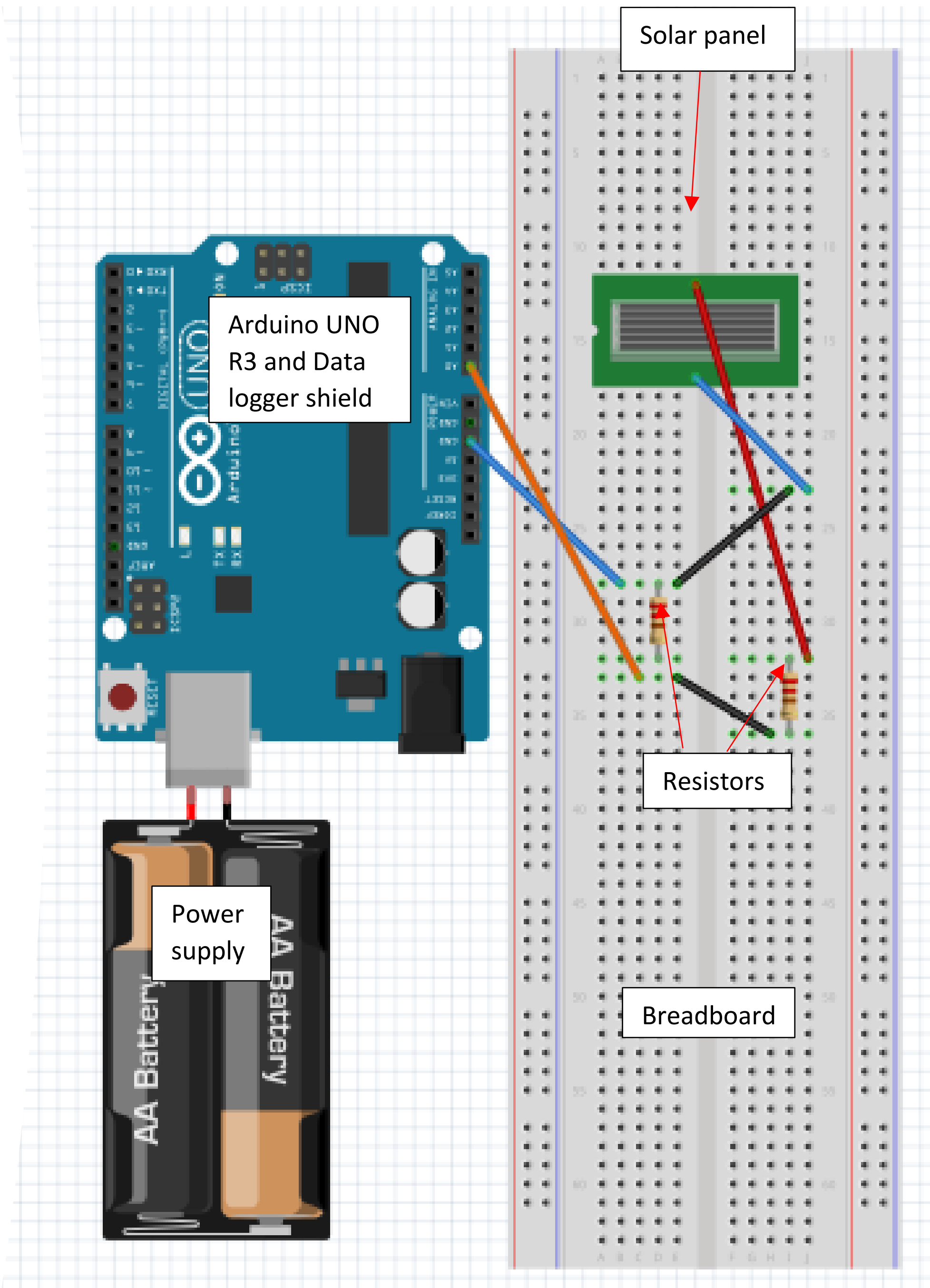


Figure 4.0 Fritzing prototype circuit layout

4.2.1 Process

The first prototype was made on fritzing. Fritzing is used to make a circuit build digitally. (Fritzing, 2021). This was used to see how the data logger would be built. It consists of an Arduino, PV panel, battery pack, breadboard, two resistors and a few wires. This layout from Figure 4.0 shows that the voltage divider can be built on the breadboard as it allows easy connections from the PV panel and the Arduino.

4.2.2 Features

- Takes the date and time, voltage, current and power onto a excel spreadsheet.
- Real time clock is always on with a coin cell battery to keep an accurate reading
- Time taken between readings can be adjusted in the Arduino IDE code.
- LED beside card holder will light up green when the card is inserted correctly and will light up red when the card is not inserted.

4.2.3 Issues

- Incorrect battery was used for the data logger shield and the code would not run because of it.

4.2.4 Resolution of issues

- Coin cell battery was used instead of a button cell battery.

4.3 Prototype 2

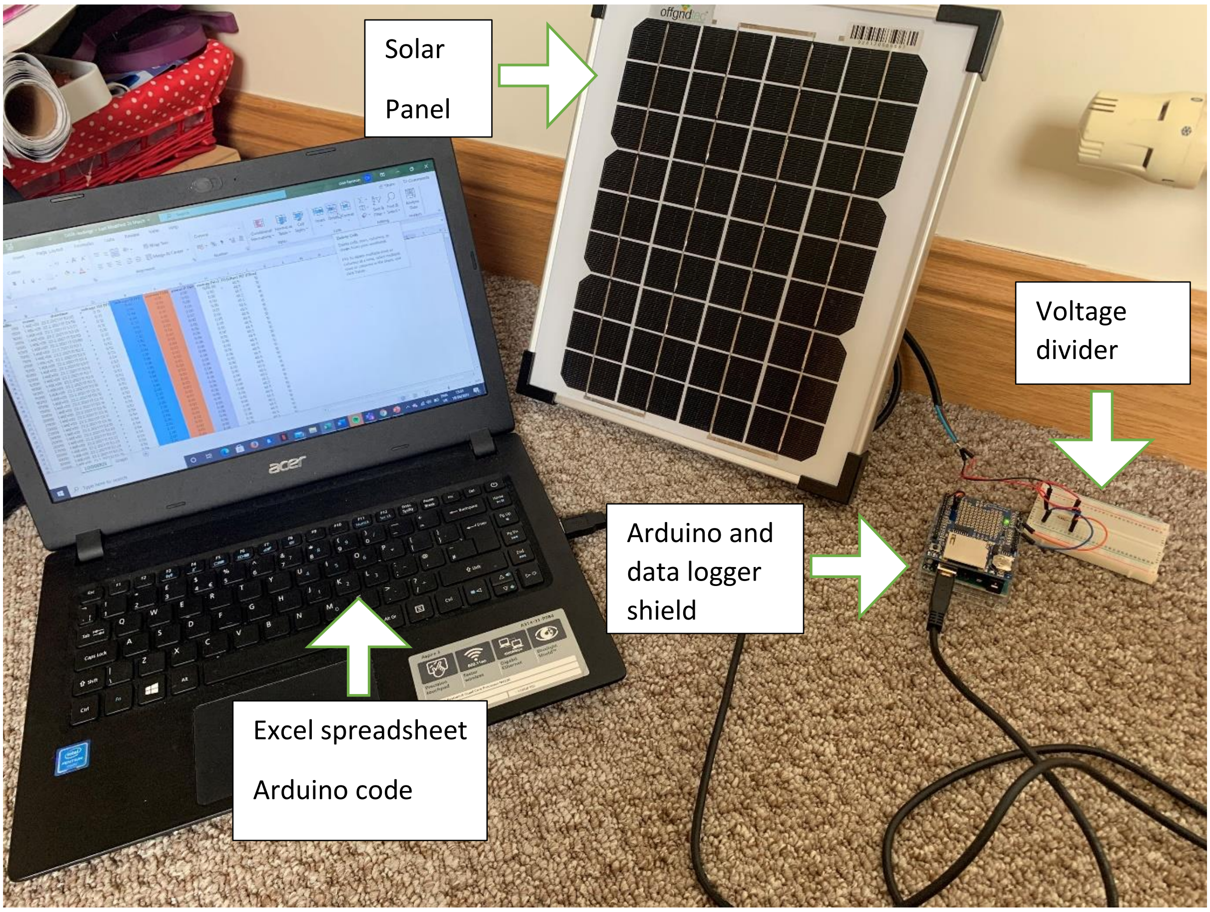


Figure 4.1 Prototype 2 setup

4.3.1 Process

The data logger shield is the main device of the project. It stores all the data using the code from the Arduino Uno R3. It also holds the real time clock which is a vital source of information for this data. The prototype 2 setup consists of the laptop that holds the Arduino IDE code, Excel spreadsheet, and Tableau data charts. The solar panel that generates the energy. The Arduino UNO R3 and the voltage divider.

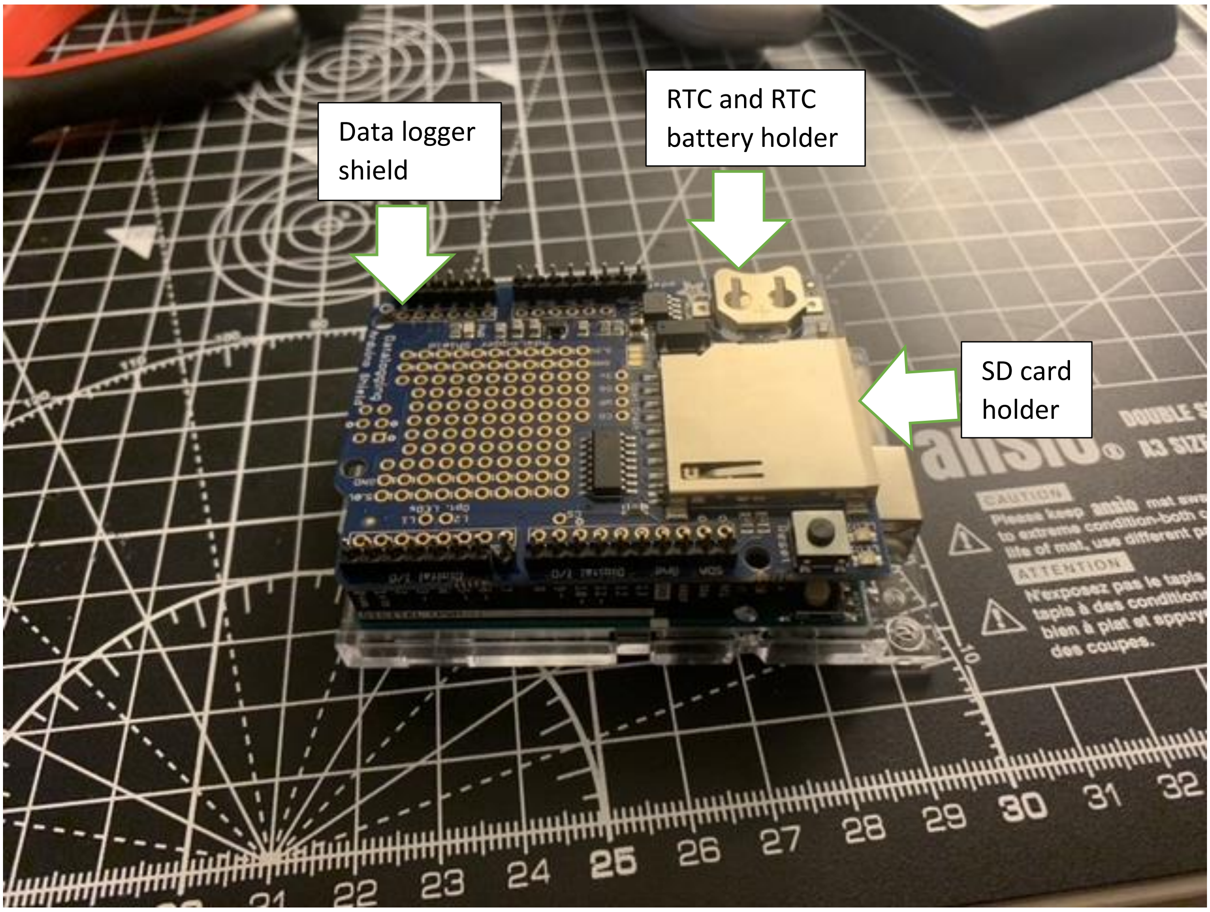


Figure 4.2 Data logger mounted.

The data logger shield is mounted on top of the Arduino as shown in figure 4.2. The Arduino already comes with female header pins built in. Male headers were used to insert the shield into the Arduino shield as shown in figure 4.3.

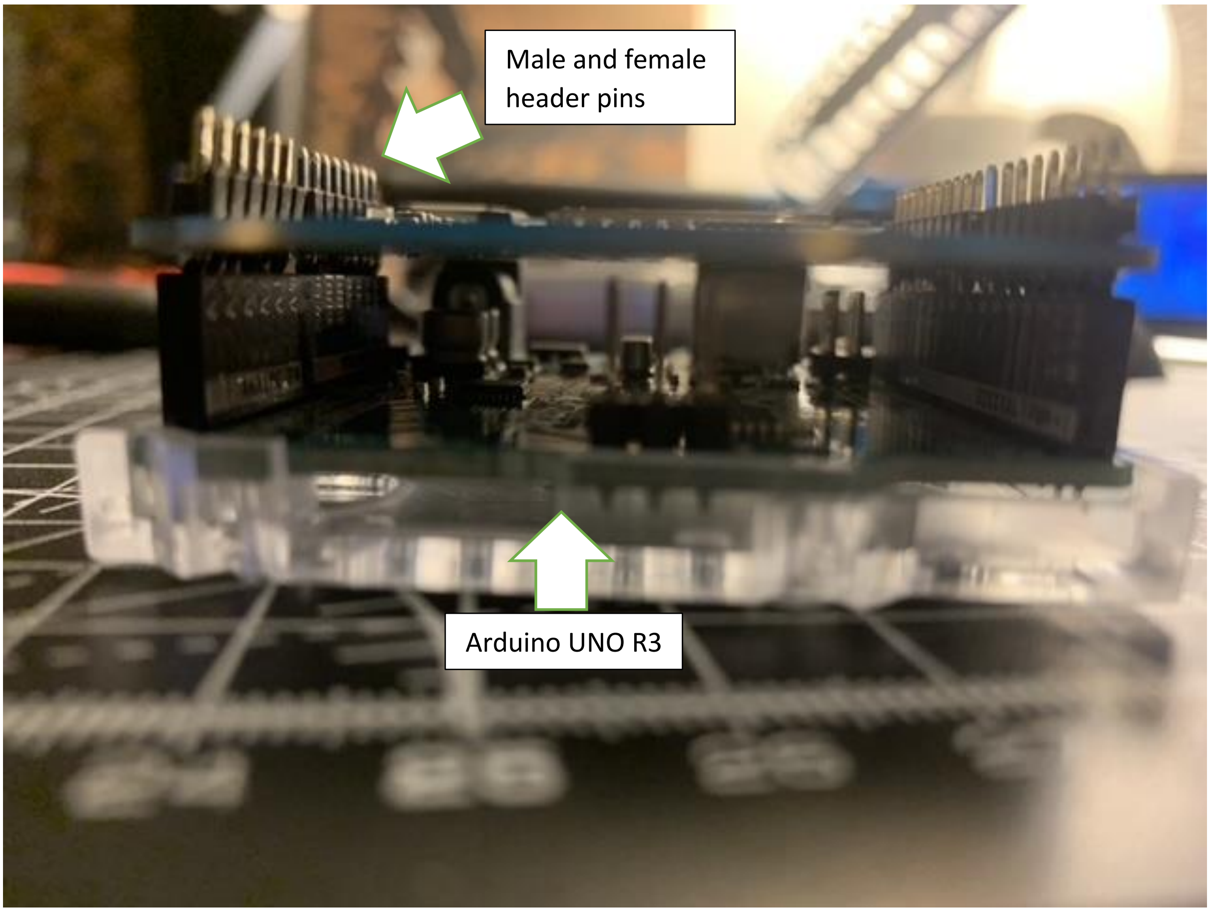


Figure 4.3 Male headers pins to attach the boards.

The male header pins are soldered onto the shield to ensure they do not fall off as shown in figure 4.3. If there are any cold solder joints or spikes, they could touch the pin beside the board and when turned on the board could break. The male header pins slot into the female header pins to connect each pin of the data logger shield to each pin of the Arduino UNO R3. Wires are soldered onto the male header pins from the voltage divider.

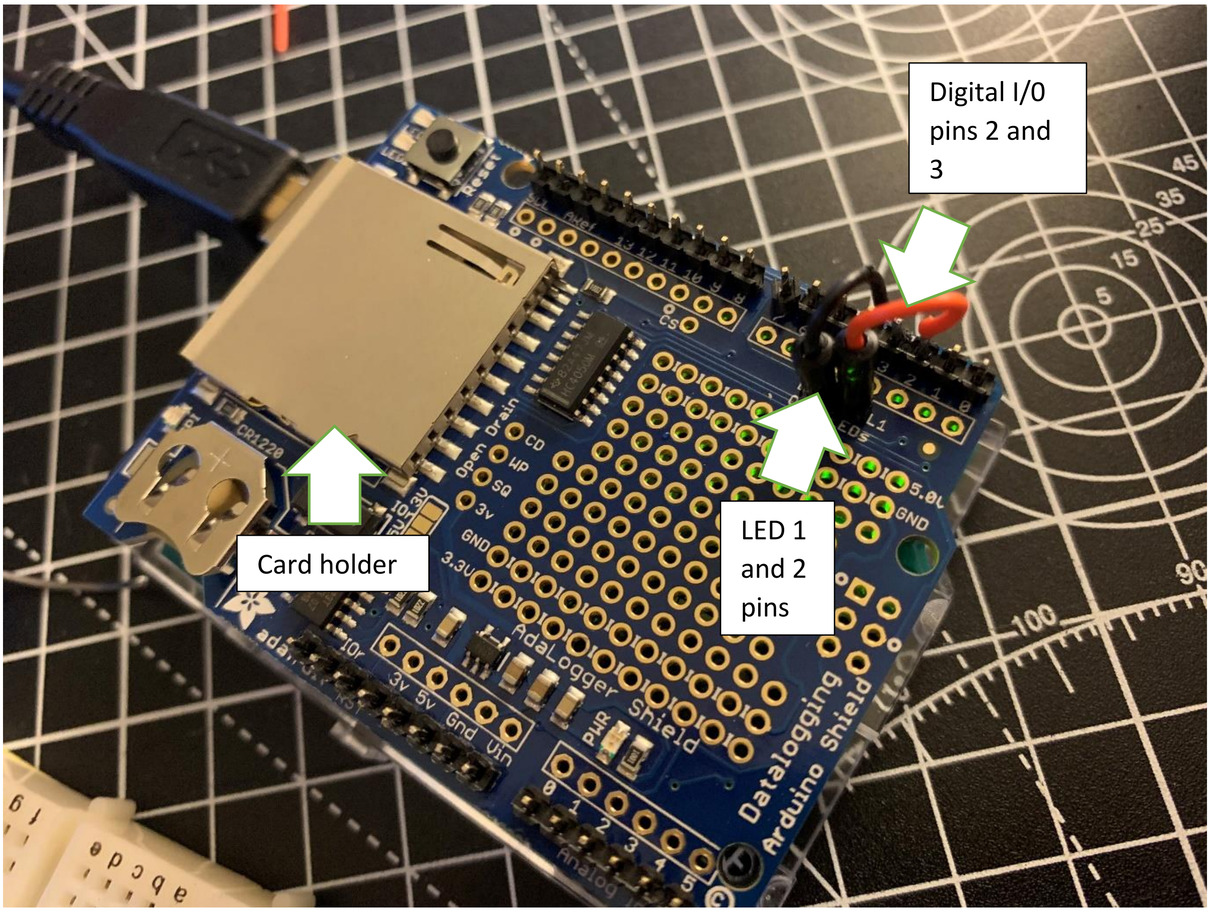


Figure 4.4 LED pins soldered.

The LED1 and 2 pins are soldered from the shield onto the digital I/0 pins 2 and 3 on the Arduino as shown in figure 4.4. Using the Arduino code from figure 4.10 and 4.13 the LED beside the card holder the LED will flash green when the card is inserted and will flash red when the data is syncing and uploading onto the spreadsheet.

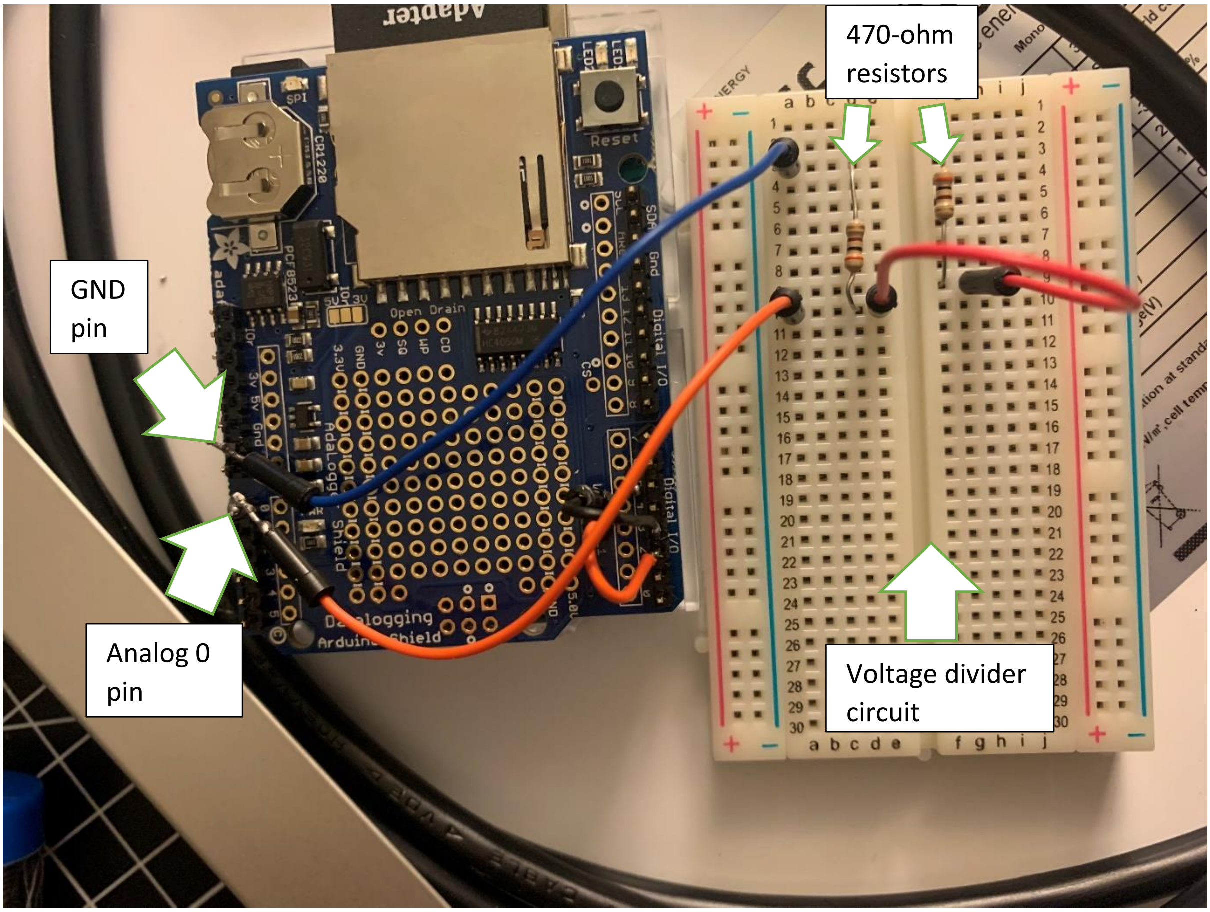


Figure 4.5 Voltage divider

The voltage divider is made on a breadboard as it is easy to change wires around if needed. The voltage divider outputs a fraction of the input voltage. It acts as a big resistor in a way. 2 470-ohm resistors and a few wires are all that is needed. The resistors are connected at one end with a wire and the other end of one resistor goes to the Arduino ground pin and the other end of the resistor goes to the Analog 0 pin as shown in figure 4.5. The other resistor is connected to the solar panel.

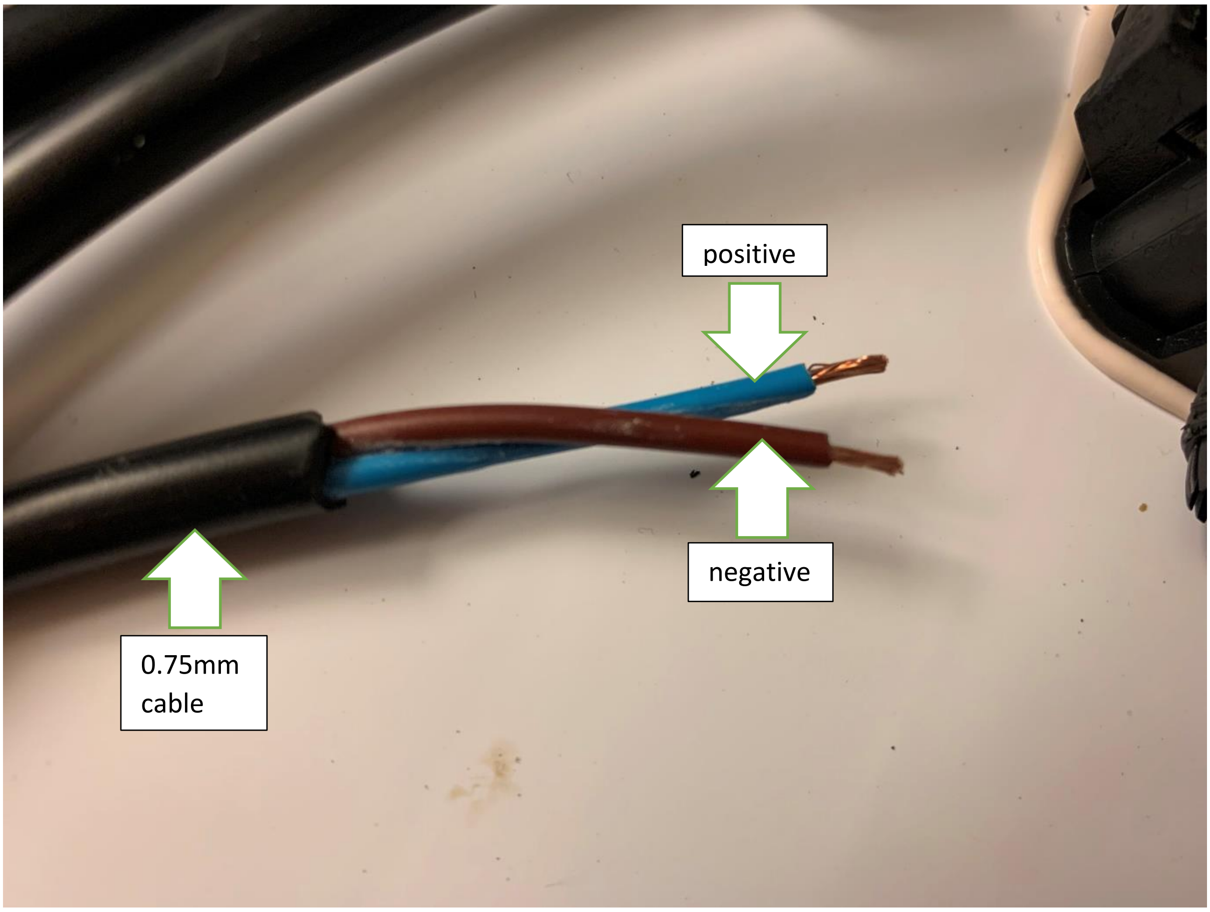


Figure 4.6 Panel wires stripped.

The 0.75mm cable is cut and stripped so the two wires are revealed as shown in figure 4.6. The brown wire is negative, and the blue wire is positive. The two exposed wires are thinned with solder to keep the copper wire together so they can be inserted into the breadboard. These wires are connected to the remaining resistor in figure 4.5.

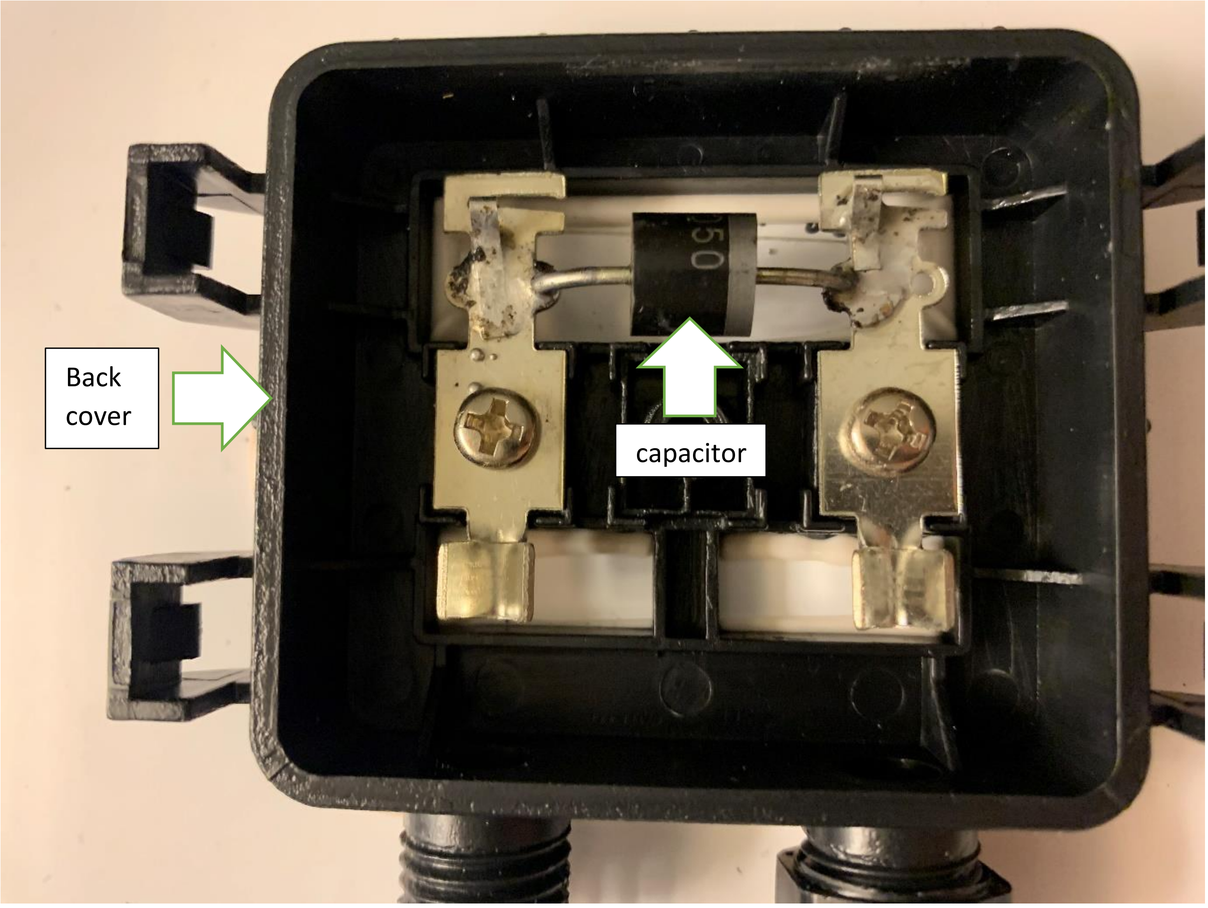


Figure 4.7 Solar panel pins

The back cover of the solar panel is opened to reveal its positive and negative pins as shown in figure 4.7. A capacitor is soldered between the two pins to store the electric charge and release it when needed.

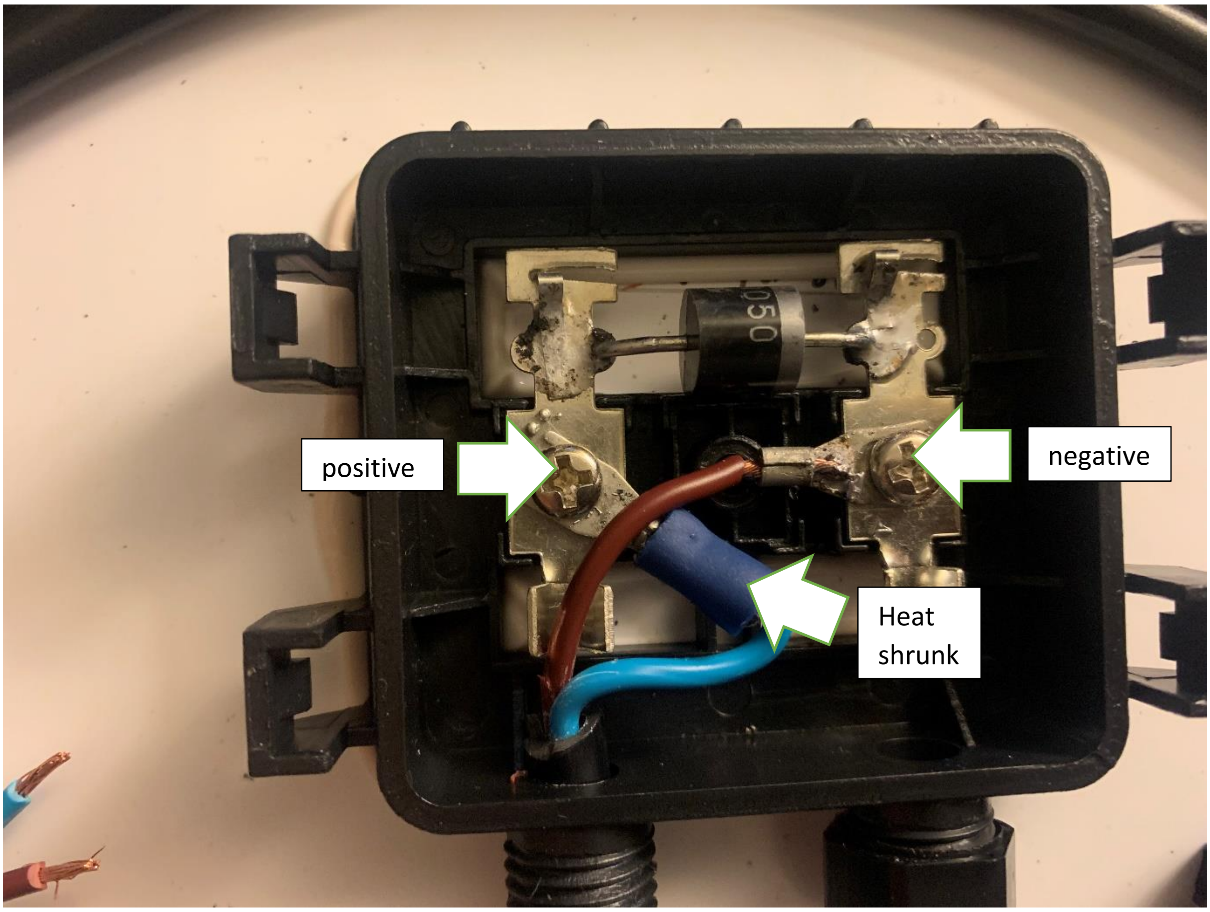


Figure 4.8 Wires soldered to the solar panel pins.

The positive wire is soldered to the left pin and the negative is soldered to the right pin as shown in figure 4.8. These wires are covered with plastic that is heat shrunk to avoid any electrocution hazards. The other end of the wires is inserted onto the voltage divider.

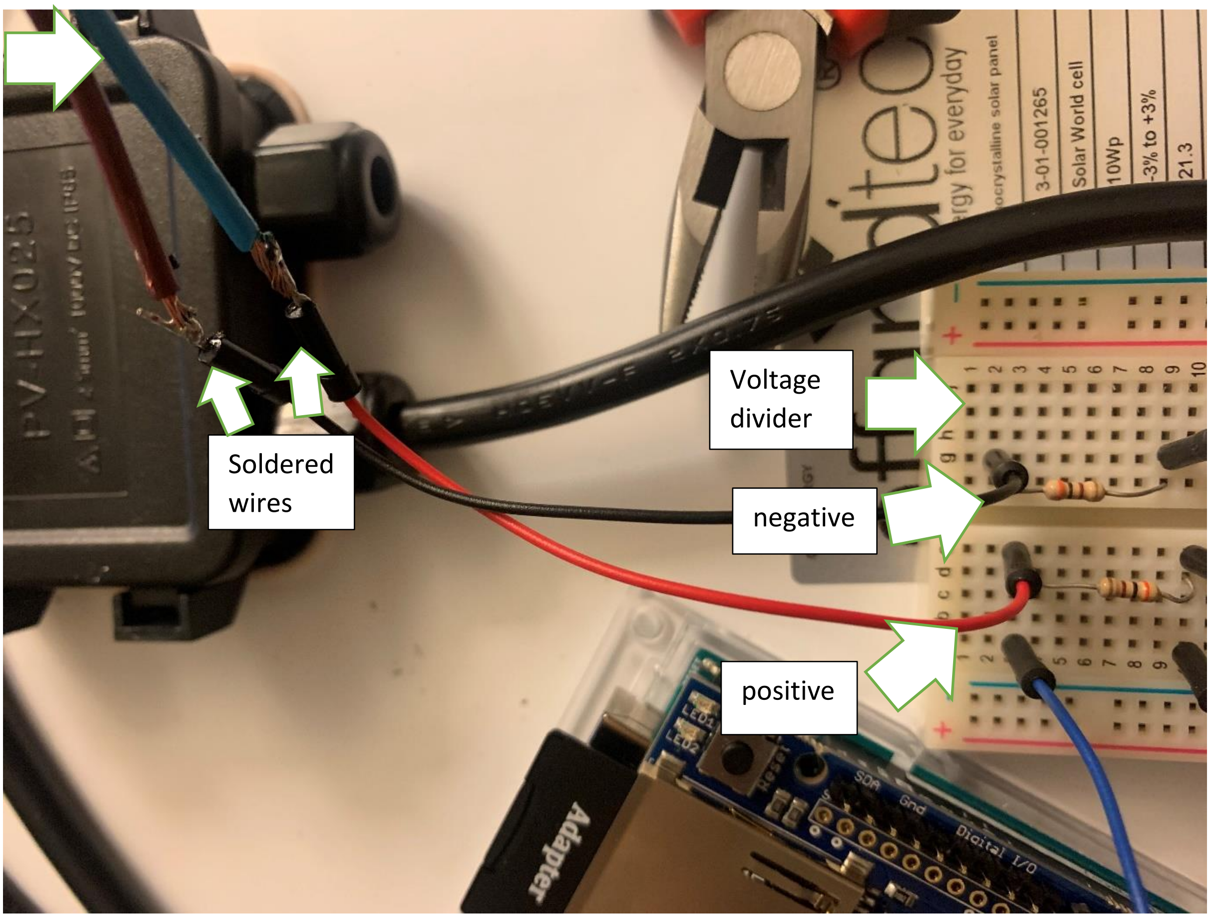


Figure 4.9 Solar panel connected to voltage divider.

The wires from the solar panel are then connected to the voltage divider. Each wire is connected to either of the resistors at the ends as shown in figure 4.9.

All of this would not work without the Arduino IDE code.

```
#include <SD.h>
#include <Wire.h>
#include "RTClib.h"
#include <SPI.h>

// A simple data logger for the Arduino analog pins

// how many milliseconds between grabbing data and logging it. 1000 ms is once a second
#define LOG_INTERVAL  1000 // mills between entries (reduce to take more/faster data)

// how many milliseconds before writing the logged data permanently to disk
// set it to the LOG_INTERVAL to write each time (safest)
// set it to 10*LOG_INTERVAL to write all data every 10 datareads, you could lose up to
// the last 10 reads if power is lost but it uses less power and is much faster!
#define SYNC_INTERVAL 10000 // mills between calls to flush() - to write data to the card
uint32_t syncTime = 0; // time of last sync()

#define ECHO_TO_SERIAL   1 // Echo data to serial monitor (Setting it to 0 will turn it off)
#define WAIT_TO_START    1 // Wait for serial input in setup() (Setting it to 1 you have to send a character to zhe Arduino´s Serial port to start the logging)

// the digital pins that connect to the LEDs (Solder pads L1 (green) and L2 (red) on the datalogger shield)
#define redLEDpin 2
#define greenLEDpin 3

// The analog pin connected to the PV cell voltage divider
#define sensorPin A0
```

Figure 4.10 Libraries and defines

In figure 4.10 the SD, wire, RTC and SPI libraries are included. the time between data is taken is set to 1000 Millis. Type “1” into echo to start to start the device and type 1 into wait to start to start logging data. The red led pin 2 and the green led pin 3 are defined and set to the L1(Green) and L2(Red) solar pads on the shield. The analogue “0” pin is assigned as the sensor or the solar panel.

```
RTC_DS1307 RTC; // define the Real Time Clock object

// for the data logging shield, we use digital pin 10 for the SD cs line
const int chipSelect = 10;

// the logging file
File logfile; // logfile is a shortcut for the CSV filename

void error(char *str) // starts the void error loop
{
 Serial.print("error: "); // if there is a problem an error message will appear
 Serial.println(str); // prints the string

// red LED indicates error
 digitalWrite(redLEDpin, HIGH); // If there is an error the red LED will stay lit
 while(1); // resets at 1
}
```

Figure 4.11 Defining the RTC and error message.

The real-time clock is defined as shown in figure 4.11. On the data logger shield the digital pin 10 is selected for the SD card line. The file starts to log. If there is a problem with starting the file in the SD card an error message will appear using the string code and the red LED will light up to notify the user on the Arduino also.

```
void setup(void)
{
  Serial.begin(9600);
  Serial.println();
  
  // use debugging LEDs
  pinMode(redLEDpin, OUTPUT);
  pinMode(greenLEDpin, OUTPUT);

#if WAIT_TO_START
  Serial.println("Type any character to start");
  while (!Serial.available());
#endif //WAIT_TO_START

  // initialize the SD card
  Serial.print("Initializing SD card...");
  // make sure that the default chip select pin is set to
  // output, even if you don't use it:
  pinMode(10, OUTPUT);

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    error("Card failed, or not present");
  }
  Serial.println("card initialized.");

  // create a new file
  char filename[] = "DATA_readings.CSV";
  for (uint8_t i = 0; i < 100; i++) {
    filename[6] = i/10 + '0';
    filename[7] = i%10 + '0';
    if (! SD.exists(filename)) {
      // only open a new file if it doesn't exist
      logfile = SD.open(filename, FILE_WRITE);
      break;  // leave the loop!
    }
  }
```

Figure 4.12 Starting the logging

Figure 4.12 is the section of code that stars the logging and initializes the SD card. The serial begin is set at 9600 which is the default value. The green and red LED pins are made outputs. The program waits for a character to be inputted in the serial monitor so it can start the logging process. The SD card is then initialized. This is to see if the SD card is working or inserted correctly. The digital pin 10 is set to an output. If the SD card is not working or is inserted incorrectly an error message will appear saying “Card failed, or not present”. If the SD card is working a “card initialized” message will appear.

The CSV file is then created, and the filename is set to a certain name made by the user. If the file is not created already the file will create itself in the SD card and if the file is already created the data will start to log into the spreadsheet.

```
  if (! logfile) {
    error("couldnt create file");
  }

  Serial.print("Logging to: ");
  Serial.println(filename);

  // connect to RTC
  Wire.begin();
  if (!RTC.begin()) {
    logfile.println("RTC failed");
#if ECHO_TO_SERIAL
    Serial.println("RTC failed");
#endif  //ECHO_TO_SERIAL
  }


  logfile.println("millis; stamp; datetime; voltage; current; power; energy;");
#if ECHO_TO_SERIAL
  Serial.print("millis");
  Serial.print("stamp");
  Serial.print("datetime");
  Serial.print("voltage");
  Serial.print("current");
  Serial.print("power");
  Serial.print("energy");
#endif //ECHO_TO_SERIAL
  }
```

Figure 4.13 Error messages and spreadsheet column names

If there is an error with the logging process an error message will appear with either a “couldn’t create file” message if there is an issue with the file and a “RTC failed” message if there is a problem with the real-time clock. When the CSV file is created the column, headings are printed in order as shown in figure 4.13.

```
void loop(void)
{
  // stores the value read on analog input A0
  int sensorVal = analogRead(sensorPin);
  // set the range of values to be considered for datalogging
  if(sensorVal>=3.25) // 3.25 corresponds roughly to 12V on the PV cell
 {
  DateTime now;

  // delay for the amount of time we want between readings
  delay((LOG_INTERVAL -1) - (millis() % LOG_INTERVAL));

  digitalWrite(greenLEDpin, HIGH);

  // log milliseconds since starting
  uint32_t m = millis();
  logfile.print(m);           // milliseconds since start
  logfile.print("; ");
#if ECHO_TO_SERIAL
  Serial.print(m);         // milliseconds since start
  Serial.print("\t");
#endif

  // fetch the time
  now = RTC.now();
  // log time
  logfile.print(now.unixtime()); // seconds since 4/4/2021
  logfile.print("; ");
  // logfile.print();
  logfile.print(now.day(), );
  logfile.print(".");
  logfile.print(now.month(), DEC);
  logfile.print(".");
  logfile.print(now.year(), DEC);
  logfile.print(" ");
  logfile.print(now.hour(), DEC);
  logfile.print(":");
  logfile.print(now.minute(), DEC);
  logfile.print(":");
  logfile.print(now.second(), DEC);
```

Figure 4.14 Data values and date and time organised pt.1

```
  Serial.print(now.month(), DEC);
  Serial.print(".");
  Serial.print(now.year(), DEC);
  Serial.print(" ");
  Serial.print(now.hour(), DEC);
  Serial.print(":");
  Serial.print(now.minute(), DEC);
  Serial.print(":");
  Serial.print(now.second(), DEC);
  Serial.print("\t");
#endif //ECHO_TO_SERIAL

  // converting the value of sensorVal to voltage
  float voltage = (sensorVal/1024.0)*5.0;

  logfile.print("; ");
  logfile.println(voltage);

#if ECHO_TO_SERIAL
  Serial.println(voltage);
#endif //ECHO_TO_SERIAL

// At the end of data logging the green LED is turned off
digitalWrite(greenLEDpin, LOW);

  // Now we write data to disk! Don't sync too often - requires 2048 bytes of I/O to SD card
  // which uses a bunch of power and takes time
  if ((millis() - syncTime) < SYNC_INTERVAL) return;
  syncTime = millis();

  // blink LED to show we are syncing data to the card & updating FAT!
 digitalWrite(redLEDpin, HIGH);
  logfile.flush();
 digitalWrite(redLEDpin, LOW);
 }
 else {
  #if ECHO_TO_SERIAL
  Serial.print("sensorVal is out of range: ");
  Serial.println(sensorVal);
  #endif // ECHO_TO_SERIAL
  }
}

//https://learn.adafruit.com/adafruit-data-logger-shield/using-the-real-time-clock-3
//https://www.arduino.cc/en/Tutorial/LibraryExamples/Files#code
```

Figure 4.15 Data values and date and time organised pt.2

Figures 4.14 and 4.15 show the data values and what pins they are assigned to. They also show how the date and time is organised. The sensor value is set to the "analogRead" pin. The Range of values it set at equal to or less than 3.25 to match the 12V on the solar panel to avoid any issues.

The delay is set to -1 on the log interval to slightly delay the time in between readings to assure the program does not crash. When the logging starts, the milliseconds will start to log also with the correct date and time as set below this line of code. The day, month, year, hour, minute and second are set for the real-time clock and printed onto the excel spreadsheet. These values are also printed on the serial.

The sensor value is then converted to voltage using "(sensorVal/1024.0) *5.0" and the voltage is printed into the spreadsheet and serial. When the data logging is complete the green LED is turned off. The data is then synced when complete to the SD card. You should not sync too often as it will make copies and can fill up the SD card.

When the data is syncing the red LED will start blinking if there is an error a "SensorVal is out of range" message will appear.

From the file created in figure 4.12 the data is logged onto an excel spreadsheet.

	A	B	C	D	E	F	G	H	I	J
1	millis	stamp	datetime	voltage U2 [V]	voltage U [V]	current I [A]	power P [W]	energy [Ws]	R1 [Ohm]	R2 [Ohm]
2	999	1.46E+09	23.2.2021 17:53:55	0.13	0.48	0.01	0.00	3606.88	48.5	18
3	1999	1.46E+09	23.2.2021 17:53:56	0.13	0.48	0.01	0.00	0.00	48.5	18
4	3000	1.46E+09	23.2.2021 17:53:57	0.12	0.44	0.01	0.00	0.00	48.5	18
5	3999	1.46E+09	23.2.2021 17:53:58	0.12	0.44	0.01	0.00	0.00	48.5	18
6	5000	1.46E+09	23.2.2021 17:53:59	0.10	0.37	0.01	0.00	0.00	48.5	18
7	5999	1.46E+09	23.2.2021 17:53:60	0.32	1.18	0.02	0.02	0.01	48.5	18
8	7000	1.46E+09	23.2.2021 17:53:1	0.49	1.81	0.03	0.05	0.04	48.5	18
9	7999	1.46E+09	23.2.2021 17:53:2	0.52	1.92	0.03	0.06	0.05	48.5	18
10	8998	1.46E+09	23.2.2021 17:53:3	0.52	1.92	0.03	0.06	0.06	48.5	18
11	9999	1.46E+09	23.2.2021 17:53:4	0.53	1.96	0.03	0.06	0.06	48.5	18
12	10999	1.46E+09	23.2.2021 17:53:5	0.54	2.00	0.03	0.06	0.06	48.5	18
13	12000	1.46E+09	23.2.2021 17:53:6	0.53	1.96	0.03	0.06	0.06	48.5	18
14	12998	1.46E+09	23.2.2021 17:53:7	0.53	1.96	0.03	0.06	0.06	48.5	18
15	14000	1.46E+09	23.2.2021 17:53:8	0.53	1.96	0.03	0.06	0.06	48.5	18
16	14999	1.46E+09	23.2.2021 17:53:9	0.53	1.96	0.03	0.06	0.06	48.5	18
17	15998	1.46E+09	23.2.2021 17:53:10	0.53	1.96	0.03	0.06	0.06	48.5	18
18	16999	1.46E+09	23.2.2021 17:53:11	0.53	1.96	0.03	0.06	0.06	48.5	18
19	17998	1.46E+09	23.2.2021 17:53:12	0.53	1.96	0.03	0.06	0.06	48.5	18
20	19000	1.46E+09	23.2.2021 17:53:13	0.53	1.96	0.03	0.06	0.06	48.5	18
21	19999	1.46E+09	23.2.2021 17:53:14	0.54	2.00	0.03	0.06	0.06	48.5	18
22	20999	1.46E+09	23.2.2021 17:53:15	0.54	2.00	0.03	0.06	0.06	48.5	18
23	21998	1.46E+09	23.2.2021 17:53:16	0.53	1.96	0.03	0.06	0.06	48.5	18
24	23000	1.46E+09	23.2.2021 17:53:17	0.54	2.00	0.03	0.06	0.06	48.5	18
25	23999	1.46E+09	23.2.2021 17:53:18	0.54	2.00	0.03	0.06	0.06	48.5	18
26	24999	1.46E+09	23.2.2021 17:53:19	0.54	2.00	0.03	0.06	0.06	48.5	18
27	25999	1.46E+09	23.2.2021 17:53:20	0.54	2.00	0.03	0.06	0.06	48.5	18
28	26999	1.46E+09	23.2.2021 17:53:21	0.54	2.00	0.03	0.06	0.06	48.5	18
29	27999	1.46E+09	23.2.2021 17:53:22	0.54	2.00	0.03	0.06	0.06	48.5	18
30	28999	1.46E+09	23.2.2021 17:53:23	0.54	2.00	0.03	0.06	0.06	48.5	18
31	30000	1.46E+09	23.2.2021 17:53:24	0.54	2.00	0.03	0.06	0.06	48.5	18
32	30999	1.46E+09	23.2.2021 17:53:25	0.56	2.07	0.03	0.06	0.06	48.5	18
33	32000	1.46E+09	23.2.2021 17:53:26	0.54	2.00	0.03	0.06	0.06	48.5	18

Figure 4.16 CSV file data readings

The data is spread into 9 columns with Date and time, Voltage, Current and power being the main ones as shown in figure 4.16. 20 minutes was recorded in this data logging session. This data is then imported into Tableau to be further visualised.

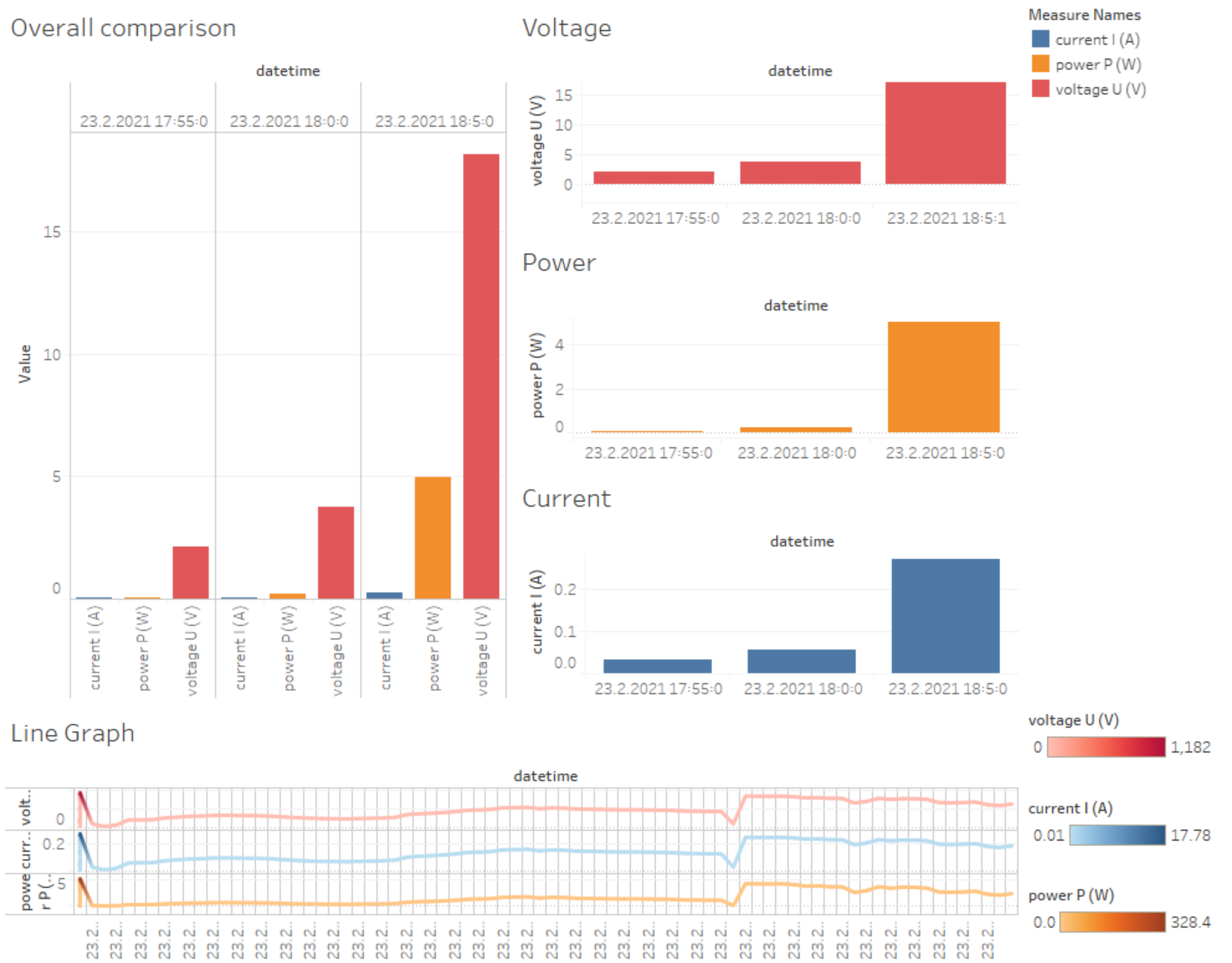


Figure 4.17 Tableau data dashboard

All the data is imported from the excel spreadsheet into Tableau. This data is all broken down into multiple bar charts and a line graph. Readings are taken 5 minutes apart and are put into bar charts to compare the voltage, current and power values. Each value is given a different colour to easily identify the differences between them. These readings are put into a bar chart to clearly show a visual representation of the energy produced. Data was taken 5 minutes apart in the evening, this chart shows there can be a huge difference in the current, power and voltage being generated. On the first chart there is 2.14V being generated and then just 10 minutes later there is 18.2V being generated. This is most likely a cloud covering the sun then moving away which can make such a huge difference, this could also be a glitch with the device which is unlikely.

Figure 4.17 shows the line graph comparison of the 3 data sets with a value every second for 20 minutes. In this example each data type is a different colour so that the chart is easier to read and identify. The sizes are adjusted so each line is a similar size to see the comparison easier. The lines are relatively the same, but you can the slight differences due to using the size tool. This is how the data visualisation will be set up for all the data taken. This is how the visualisation should look when the data is imported into tableau with the further prototypes.

A line graph is made with 20 minutes of data to show the differences a few minutes can make in the power produced. The line graph is a better reading as you can see a more accurate change in the power coming from the solar panel. All these graphs are inserted into a dashboard to showcase them all at once as shown in figure 4.17.

4.3.2 Features

- Takes the date and time, voltage, current and power onto a excel spreadsheet.
- Real time clock is always on with a coin cell battery to keep an accurate reading
- Time taken between readings can be adjusted in the Arduino IDE code.
- LED beside card holder will light up green when the card is inserted correctly and will light up red when the card is not inserted.

4.3.3 issues

- Code would not upload onto Arduino board due to incorrect battery in RTC.
- The Arduino and the voltage divider were being blown in the wind and getting rained on

4.3.4 Resolution of issues

- Coin cell battery was used instead of button battery and code uploaded fine.
- The Arduino and voltage divider are slotted into the back of the PV panel to avoid any weather obstructions

4.4 Conclusion

The safety hazard and precautions are resolved above with why and how these hazards can be dangerous and how to avoid them. Each prototype is explained from start to finish. There is a major improvement between prototype 1 and prototype 2 as prototype to is the real build. A step-to-step guide is described on each prototype build with a list of features shown. The issues from both prototypes are listed with their resolutions and how it has improved.

5. Testing and results

Testing is used to ensure any faults or hazards are found and resolved. There is the test set up, hardware testing, software testing, unit testing and systems testing. There are tests taken to see what materials, wires and code works best with results showing why.

5.1 Test set up

The first test done was what the circuit was going to be tested on. Breadboard seemed like the best option as wires and components can be moved with ease if the circuit did not work or something was in the wrong place. The circuit is kept on the breadboard as there is no reason to solder onto Perfboard etc as you cannot see the breadboard. the circuit on the breadboard is the voltage divider.

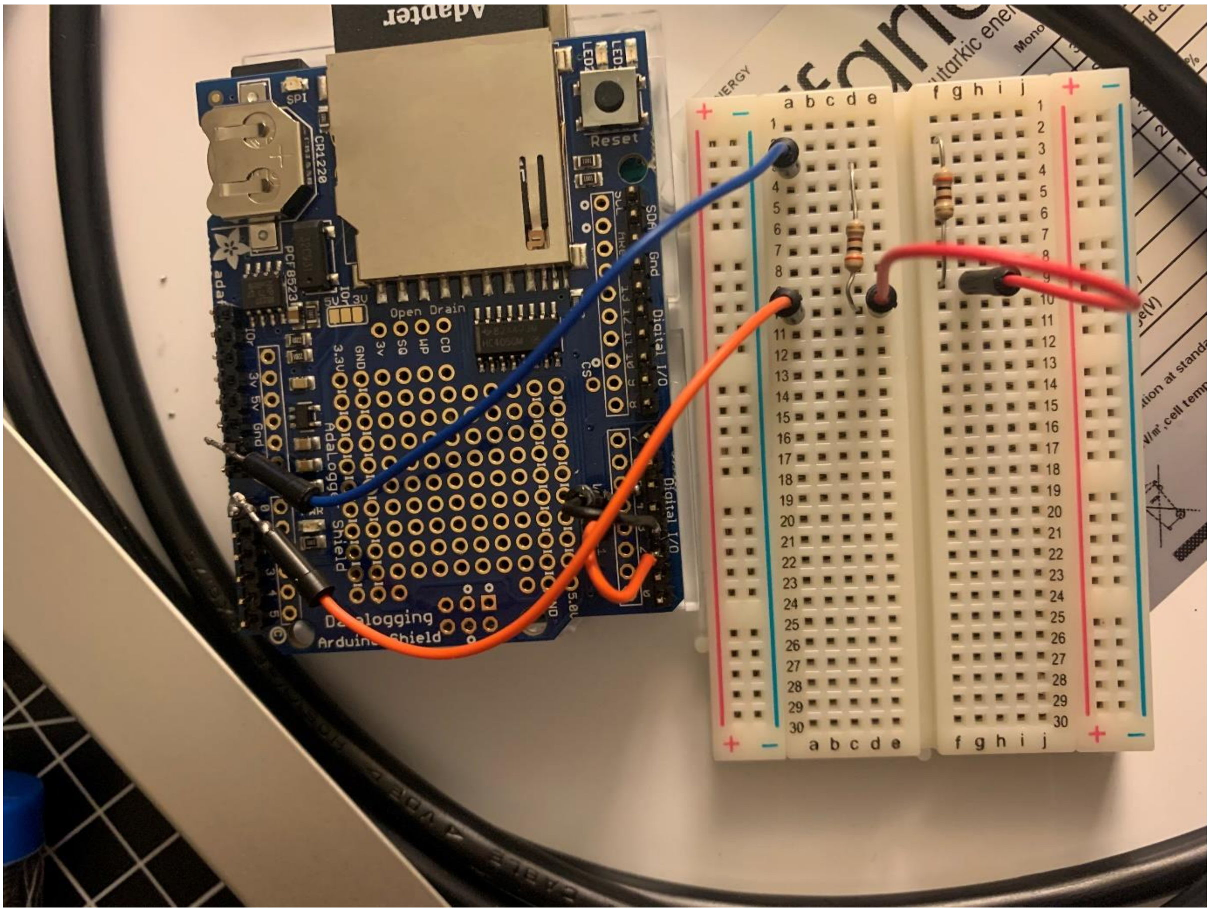


Figure 5.0 Circuit on breadboard

Figure 5.0 shows the voltage divider circuit on the breadboard between the Arduino and solar panel. There were no problems using the breadboard.

5.2 Hardware testing

Test no.	description	Input	Expected output	Actual output	Comments
1	Voltage flowing across the device from probe to probe	9V	5v	4.7v	Successful test as the voltage divider minimizes the volts to under 5V
2	Arduino and voltage divider weather test	Devices are Beside the solar panel on the floor.	Devices are secure and safe	Devices got oved by the wind and rained on	Unsuccessful test as the devices were not secure and got damaged
3	Arduino and voltage divider weather test	Devices are slotted in behind the panel	Devices are secure and safe	Devices were protected from the wind and rain	Successful test as the devices were secure and safe

Table 1 Hardware testing

A multimeter was used to test each component ensuring the correct voltage and current was running through it. The positive probe is placed at one end of the component or circuit and the negative probe is placed at the other end of the component or circuit. A certain reading will appear on the screen when the probes are placed at either end of a component or circuit as shown in table 1.

9V is more than enough to power this circuit as efficiently possible. If the Arduino receives more than 5V the board could short circuit or break completely. The voltage divider is used to bring the 9V down below 5V to ensure the Arduino is not fried. A 3v battery pack was used at first and was enough to power the Arduino but not the whole circuit. A 9v battery pack and laptop work as a power source to power the whole device safely. The voltage divider worked with the 9v battery pack and laptop which made it a successful test as shown in table 1.

Another test was made to see how the Arduino and breadboard would cope in the weather. The Arduino and breadboard were originally just beside the panel in the open but if it were windy the boards would be blown around and if it rained the boards could be damaged or the water could jump the circuit onto other pins making the Arduino short circuit. A second test was done for this. The solar panel has a slot behind the panel allowing the boards to sit in the slot snugly. The boards are also stuck to the back of the panel. As the panel is tilted at 45 degrees the rain cannot touch the boards as the panel acts as a roof. This test was successful as the device was able to be left outside unattended without any issues as shown in table 1.

The Problem-solving strategy is to learn from the unsuccessful tests and try and use the new learned knowledge to complete the test second time round.

5.3 Software testing

Test no.	description	Input	Expected output	Actual output	Comments
1	Excel and Tableau recovery test	Both files are closed unexpectedly	Files will be recovered on restart	Files were recovered with ease	Successful test as the files were automatically recovered
2	Arduino IDE recovery rest	File is closed unexpectedly	File will be recovered on restart	File was recovered on restart	Successful test as the file was automatically recovered
3	Code is verified	Verify button is pressed	Verification is successful with no errors	Multiple errors and misspells appeared	Unsuccessful test as there were multiple errors
4	Code is verified	Verify button is pressed	Verification is successful with no errors	Verification was successful with no errors	Successful test as there were no errors

Table 2 Software testing

The first software test done was to see if both the Excel and Tableau files were turned off unexpectedly would they be recovered when I reopened the software. The test was successful and both files appeared back on the software straight away as shown in table 2.

The second software test done was to see if the Arduino IDE file were turned off unexpectedly would it be recovered when the app was reopened. The test was successful as the file appeared back on the app when it was reopened as shown in table 2.

The third software test done was to verify the Arduino IDE code. When the code is finished the verify, button is pressed to validate the code and check for errors. The test was Unsuccessful as a few errors came up for unexpected character which was spelling mistakes and incorrect variables. A detailed run through the code was preformed to correct all the errors listed in the code as shown in table 2.

The fourth test was done to verify the Arduino uno IDE code. When the code is finished the verify, button is pressed to validate the code and check for errors. The test was successful as no errors appeared and the code was ready to be uploaded as shown in table 2.

5.4 Conclusion

The tests are a vital part of building a project as you learn and improve from the faults. There were multiple tests done to ensure the project was working to the best of its ability. Testing is done to avoid any future faults or hazards that could be a setback. The test set up was successful as the voltage divider was built on the breadboard and is the best option for ease of use.

The hardware testing was successful as each component and materials were functioning correctly. This testing is vital as it is the best way to find out If a component is not functioning as it should be and could be the reason the device is not working.

The software testing was successful as all recovery tests, coding verifications were successful. Both recovery tests worked perfectly. The first code verification failed as there were many errors and faults as outlined in the table above. The errors and faults were assessed and fixed and a second verification test was ran. The second code verification test was a success as the code had no errors and the code was ready to upload.

6. Conclusion

The report has described the development of the data logger for solar power using Arduino project which is capable of logging accurate data from any location that holds a source of light and is visually represented to ensure the data is extremely clear. The project build worked as it was described from the research chapter with the use of the hardware testing to ensure any faults were overcome. The excel spreadsheet was formulated with PLX-DAQ. All data was made out clear using columns and rows and colouring.

The data was exploded in Tableau with the use of bar charts and line graphs data was compared clearly using multiple tools such as the size tool, colouring tool, detail tool and label tool. Different presents were used to shape that graphs to make them as appealing and as clear as possible.

The system was developed using the following technologies.

- Arduino UNO R3
- Off grid Tec 12V solar panel
- Voltage divider
- Data logger shield
- LED's
- Excel
- Tableau

The steps involved in the project had been outlined in the report. The Research chapter describes what research was needed to identify the potential technologies that could be used and what were the best suited for the project. The applications are listed to help aid the visual representation process. Similar systems were outlined like the cave pearl project to give inspiration from where the idea arose from. The skills and requirements were researched to reveal what skills were needed to complete the project.

The design chapter outlined the design plan and what the project, code and visual aspects were going to be based about. Multiple block diagrams and schematics were used to help plan the prototypes and set the foundations. The components are listed and researched to give a clear understanding of each component and why they are needed for the project. The design of the code, CSV spreadsheet and tableau is shown to give an outline of what to expect in the implementation chapter.

The implementation chapter outlined the safety procedures needed to ensure the user is safe when building the project. The hazards are clearly outlines with the precautions taken to avoid them. The prototypes are described with a step-by-step process outlining how it was made. The issues with each prototype are explained and resolved later. The prototypes improve as they progress because they learn from the previous prototype.

The testing and results chapter outlined how each test is ran in the setup, hardware, and software areas. Tables are used to clearly show the tests done in order, what was expected, and the actual result of the tests. The tests removed any room for future errors and setbacks which was vital. The testing allowed for future improvements to ensure the project was working to the best of its ability.

Overall, the project was a big success. The aims were achieved and improved by the end. The data logger is a perfect device for any homeowner looking to reduce their monthly electric bill while helping the environment. The device was tested at 45 degrees to replicate it resting on a roof to ensure the maximum accuracy for the product. If the project were to be done again a humidity option would be added to predict oncoming weather conditions. The visual aspects were represented deeply and improved from the original design inspiration in the design chapter. Each graph was easy to read and labelled correctly so it was as clear as possible. The data comparisons were made clear with the use of colour and sizes as the aims were achieved. This can be improved in the future by exploding the data even more with a more detailed in-depth graphs with the use of Gannt bar graphs.

7. References

Arduino. (2021). Introduction. Retrieved April 29, 2021, from https://www.arduino.cc/en/guide/introduction

Tag, T. (2019). What is a data Logger?: Understanding loggers vs data acquisition. Retrieved April 23, 2021, from https://www.geminidataloggers.com/info/what_is_a_data_logger

Lutkevich, B. (2019, November 07). What is a microcontroller and how does it work? Retrieved April 27, 2021, from https://internetofthingsagenda.techtarget.com/definition/microcontroller#:~:text=A%20microcontroller%20is%20a%20compact,peripherals%20on%20a%20single%20chip.

List of common microcontrollers. (2021, April 17). Retrieved April 27, 2021, from https://en.wikipedia.org/wiki/List_of_common_microcontrollers

Team, T. (2020). What is Arduino? Retrieved November 10, 2020, fromhttps://www.arduino.cc/en/Guide/Introduction

Tableau Software. (n.d.). Retrieved November 10, 2020,

from https://en.wikipedia.org/wiki/Tableau_Software

Van De Berg, B. (2017). Black box data logger for heavy vehicles/trucks. Retrieved November 9, 2020, from

https://hackaday.io/project/27135-black-box-data-logger-for-heavy-vehiclestrucks

Mallon, E. (2017). Pin on Arduino Data Logger Research. Retrieved November 9, 2020, from

https://www.pinterest.ie/pin/481111172683188490/?nic_v2=1a5Yr9Hq4

Technology, O. (2013, March 11). Offgridtec® 5W Poly Solar panel 12V. Retrieved February 07, 2021, from https://www.offgridtec.com/offgridtecr-5w-poly-solarpanel-12v.html

Earl, B. (2013, April 12). Adafruit data Logger Shield. Retrieved February 07, 2021, from

https://learn.adafruit.com/adafruit-data-logger-shield/overview

Gateway, T. (2021). Dual lines chart in Tableau. Retrieved May 01, 2021, from https://www.tutorialgateway.org/dual-lines-chart-in-tableau/

DataFlair. (2019, November 19). Bar chart in Tableau - the art of PORTRAYING DATA. Retrieved May 01, 2021, from https://data-flair.training/blogs/tableau-bar-chart/

Earl, B. (2013, April 12). Adafruit data Logger Shield. Retrieved April 25, 2021, from https://learn.adafruit.com/adafruit-data-logger-shield/using-the-real-time-clock-3

Team, T. (2018, February 5). Using the SD library to create and remove files on a SD card. Retrieved April 25, 2021, from https://www.arduino.cc/en/Tutorial/LibraryExamples/Files#code

Fritzing. (2021). Fritzing. Retrieved May 01, 2021, from https://fritzing.org/

Appendix

CMT Projects

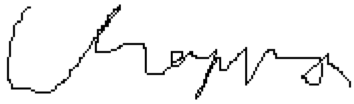
'Low Voltage Directive' Compliance Statement.

DIRECTIVE 2006/95/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 12 December 2006 on the harmonisation of the laws of Member States relating to electrical equipment designed for use within certain voltage limits (codified version) (Text with EEA relevance)

Otherwise known as the Low Voltage Directive (LVD).

If the electrical aspects of your project have a voltage between 50 and 1000 V for alternating current and between 75 and 1500 V for direct current, for voltages at the **electrical input or output** (not internally), then you are signing to **confirm that it complies with all the safety requirements of the LVD**. You must read the LVD and analyse your project before signing, taking advice where necessary.

Student name: Cian Hannon

Student signature: 

Date: 26/03/2021

Project supervisor or delegated representative who is an engineer by discipline:

Name: Timm Jeschawitz

Signature: 

Date: 26 March 2021

CMT (DL835) Student Projects

‘Medium Risk’

Standard Risk Assessment Template
For activities carried out in the School of Creative Technologies facilities.
These are projects where all Hazards are Ranked as a 3rd Rating.
See Risk Rating Matrices on Pages 10 and 11.

Project Risk Assessments and the methodology are needed to comply with the Safety, Health, and welfare at work act 2005 and all other relevant Legislation. This document is based on the ‘Joint Risk Assessments’ procedure – IADT – December 2010.

STUDENT NAME(S): Cian Hannon

PROGRAMME/YEAR: 4th Year

SIGNATURE:

DATE: 26/03/2021

SUPERVISOR: Timm Jeschawitz

SIGNATURE

DATE: 26 March 2021

By signing this assessment, it is agreed by all parties that:

- the full facts relating to the health and safety aspects of the project have been declared by the student.
- All parties are fully aware of the safety risks.

All parties will implement the control measures detailed, to reduce the contribution of the hazards to the level of the risks detailed.

Location of Work:	Home

Brief RELEVANT Details of project: 20 words	Data logger for solar power using Arduino. Multiple components are soldered and cut.

Step 1: Initial Hazards Identification

Risk Assessment No.	INITIAL HAZARD
1	Electrocution
2	Fire
3	Cutting injuries
4	Drilling injuries
5	Heavy equipment
6	Burn Injury
7	Fumes

Step 2: Risk Assessment Forms

(Start Overleaf)

Significant Hazard and consequences:	**1. Electrocution**

Who might be exposed to the hazards?	Students and staff

Proposed Control Measures – to be written in conjunction with project supervisor and revised at key project milestone dates	Institute Health and Safety Approval before development, where public safety is an issue. Circuit design must include features that will minimise likelihood of electrocution of anybody, when in an unsafe mode, e.g., use of fuses and circuit breakers. Short circuits should be identified and removed before the testing stage. Cable and insulation should be checked before testing stage. Power supply equipment should be PAT tested on a regular basis. Water sources should be kept away from the project, when in operation. Components, whether connected to power supplies or not, should be fully discharged before inspections – isolated from power supplies, are commenced. E.g., discharge capacitors greater than 50µF via a 100Ω resistor. Project checked by engineering member of staff, and certified health and safety before being switched on, where public safety is an issue. Switch on supervised by engineering member of staff and supervised by certified health and safety officer where public safety is an issue.

Significant Hazard and consequences:	**2. Fire**

Who might be exposed to the hazards?	Students and staff

Proposed Control Measures – to be written in conjunction with project supervisor and revised at key project milestone dates	Institute Health and Safety Approval before development, where public safety is an issue. Maintain tidy work practices on benches and the laboratory environment. Keep combustible materials, e.g., paper, plastic, away from heat sources, such as soldering irons. Stow heat sources away safely when not in use, e.g., use a sturdy soldering iron stand. Circuit design must include features that will minimise likelihood of a fire, when in an unsafe mode, e.g., use of fuses and circuit breakers. Short circuits should be identified and removed before the testing stage. Suitable cable and insulation should be used, with a safety margin on the rating and size. Water sources should be kept away from the project, when in operation. Project checked by engineering member of staff, and certified health and safety before being switched on, where public safety is an issue. Switch on supervised by engineering member of staff and supervised by certified health and safety officer where public safety is an issue. Be aware of the locations of first aid kit and fire extinguishers and use these items if suitably competent/trained. In the event of a fire, leave the laboratory and building in an orderly manner, and sound the fire alarm if it has not already automatically activated.

Significant Hazard and consequences:	**3. Cutting injuries**

Who might be exposed to the hazards?	Students

Proposed Control Measures – to be written in conjunction with project supervisor and revised at key project milestone dates	Maintain tidy work practices on benches and the laboratory environment. Develop projects with the minimum requirement for cutting any jagged edged in the final manufactured item. Use of quality, maintained tools and clamps if necessary. Use of a cutting board and goggles. Clear a space around the cutting area before commencing work. Tie back hair and loose clothing from the cutting area. Remove jewellery. Be aware of the location of the first aid kit and use the kit if suitably competent/trained.

Completed by:	Signatures: Date: 26/03/2021				
Supervisor:	Name: Timm Jeschawitz		Signature: Date:26/03/2021		
Dates of Reviews:	26/03/2021				

Significant Hazard and consequences:	**4. Drilling injuries**

Who might be exposed to the hazards?	Students and staff

Proposed Control Measures – to be written in conjunction with project supervisor and revised at key project milestone dates	Only use the drill equipment if training, given by staff, has been undergone. Use of quality, maintained drilling equipment and goggles. Secure drill bit and table. Use the drill guard. Check all fastenings are complete before switching on the drill machine. Drill machine should be tested and checked on a regular basis. Develop projects with the minimum requirement for cutting any jagged edged in the final manufactured item. Use of quality, maintained tools and clamps if necessary. Use of a cutting board and goggles. Clear a space around the cutting area before commencing work. Tie back hair and loose clothing from the cutting area. Remove jewellery. Be aware of the location of the first aid kit and use the kit if suitably competent/trained.

Significant Hazard and consequences:	**5. Heavy equipment**

Who might be exposed to the hazards?	Students and staff

Proposed Control Measures – to be written in conjunction with project supervisor and revised at key project milestone dates	Maintain tidy work practices on benches and the laboratory environment. Undertake heavy lifting only if suitable advised and/or trained. Correct posture and lifting procedures. Use mechanical lifting aids where possible and appropriate. One or more persons to be involved in lifting or supervising the lifting of heavy equipment. Clear a space around the lifting area before commencing work. Tie back hair and loose clothing from the cutting area. Remove jewellery. Use protective footwear, and headwear if necessary.

Significant Hazard and consequences:	**6. Burns**

Who might be exposed to the hazards?	Students and staff.

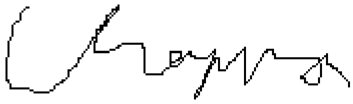
Proposed Control Measures – to be written in conjunction with project supervisor and revised at key project milestone dates	Maintain tidy work practices on benches and the laboratory environment. Only use soldering irons and other hot-works appliances if training, given by staff, has been undergone. Keep combustible materials, e.g., paper, plastic, away from heat sources, such as soldering irons. Stow heat sources away safely when not in use, e.g., use a sturdy soldering iron stand. Cable and insulation should be checked before using soldering irons, or electrically powered hot-works appliances. Use gloves, goggles, and other personal protection equipment where necessary. Use cooling equipment, such as wet sponges for soldering irons. Do not allow water from any source to penetrate electrical cables and wires. Tie back hair and loose clothing from the cutting area. Remove jewellery. Be aware of the locations of first aid kit and fire extinguishers and use these items if suitably competent/trained. In the event of a fire, leave the laboratory and building in an orderly manner, and sound the fire alarm if it has not already automatically activated. Electrically powered hot-works equipment, such as soldering irons, should be checked and tested on a regular basis.

Completed by:	Signatures: Date: 26/03/2021				
Supervisor:	Name: Timm Jeschawitz		Signature: Date: 26/03/2021		
Dates of Reviews:	26/03/2021				

Significant Hazard and consequences:	**7. Fumes**

Who might be exposed to the hazards?	Students and Staff.

Proposed Control Measures – to be written in conjunction with project supervisor and revised at key project milestone dates	Use fume extraction equipment, e.g., for solder fumes. Keep laboratories well ventilated. Take frequent breaks from activities generating fumes. Employ a higher level of control measures when an individual suffers from a respiratory condition, such as asthma, taking advice from a GP. Solder fume extraction equipment and other similar items should be maintained checked and tested on a regular basis.

Completed by:	 Signatures: Date:26/03/2021				
Supervisor:	Name: Timm Jeschawitz 		Signature: Date:26/03/2021		
Dates of Reviews:	26/03/2021				

Significant Hazard and consequences:	

Who might be exposed to the hazards?	Student and staff

Proposed Control Measures – to be written in conjunction with project supervisor and revised at key project milestone dates	

Completed by:	Signatures: Date: 26/03/2021				
Supervisor:	Name: Timm Jeschawitz Signature: Date:26/03/2021				
Dates of Reviews:	26/03/2021				

Risk Rating Matrices

By looking at the hazard and asking how many people will be exposed to it, decide on the probability of an incident/accident occurring.

Example: Take an extension lead trailing along the floor up against the wall. The extension lead is a hazard and if it is in an office with one person working in it the probability and likelihood is "improbable" (see table No.1) because the lead is along the wall. However, if the lead is in a corridor with 200 people walking by there is a chance that someone could kick it out from the wall accidentally and create a greater probability/likelihood of a loss occurring thus upping its rating to "remote".

When this is done you must decide on the seriousness of the loss, using the four columns on the left side of the Table No.2 below.

Example: Firstly, taking the one-person office example from above the possibility/likelihood is "improbable "but the result might be a "minor injury" e.g., scrape or a bruise. This gives us an "acceptable risk no action required" If we were to put the lead on a building site across an unguarded stairwell with 50 people using it the result is now possibly "fatal". This gives us a "1ST rank action".

Table No. 1:

PROBABILITY/LIKELIHOOD	DESCRIPTION
Likely/frequent	Occurs
Probable	Not Surprised. Will occur several times.
Possible	Could occur sometimes.
Remote	Unlikely, though conceivable.
Improbable	So unlikely that probability is close to zero.

Table No. 2:

	LIKELY	PROBABLE	POSSIBLE	REMOTE	IMPROBABLE
Fatal	1st	2nd	2nd	3rd	
Major Injury/ permanent disability	2nd	2nd	3rd		
Minor Injury	3rd	3rd			
No Injury					

By using the matrices above we now have an action needed ranking system. This means we can prioritize the hazards depending on their ranking.

Example: A 1st rank action requires immediate remedy

A 2nd rank action requires a less immediate remedy but must be dealt with quickly.

Table No. 3:

	1st rank actions
	2nd rank actions
	3rd rank actions
	Acceptable risk – no action

Taking all this information and pooling it in the Initial Hazard Identification log we can now prioritise the hazards in the left-hand column "Risk Assessment No." This number will appear on the top left of the risk assessment forms for easy referencing.

Medium Risk Project – Risk Rating Summary

Risk Assessment No.	INITIAL HAZARD	Probability	Ranking
1	Electrocution	Possible	2nd
2	Fire	Possible	2nd
3	Cutting injuries	Possible	Acceptable
4	Drilling injuries	Possible	3rd
5	Heavy equipment	Remote	3rd
6	Burns	Possible	3rd
7	Fumes	Probable	3rd