

Internet of Things Project

**Garage of the Future**

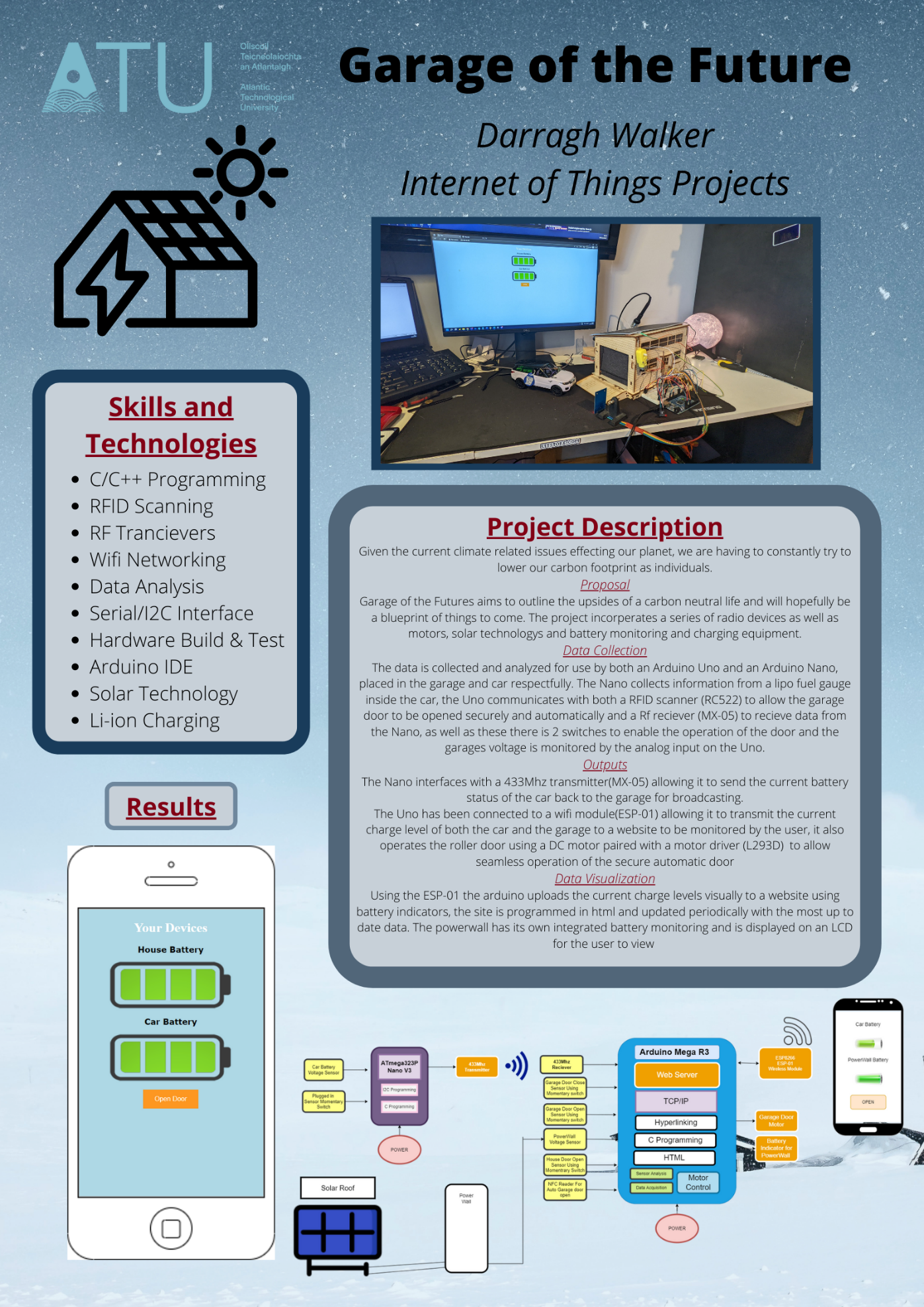
**(Solar Energy, Electric Cars and Power-Walls)**

Darragh Walker

Bachelor of Software & Electronic Engineering

Galway-Mayo Institute of Technology

2021/2022



**Declaration**

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Software & Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

Darragh Walker\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgements**

I would like to acknowledge my IOT lecturers Natasha rowan, Martin O Conghaile and Brian O Shea for all the help with the making of this project as well as supplying me with most of the components needed to complete it.

**Table of Contents**

[1 Summary 7](#_Toc102410997)

[2 Introduction 8](#_Toc102410998)

[3 The Internet of Things Overview 9](#_Toc102410999)

[4 Project Architecture 10](#_Toc102411000)

[5 Development Platform and Tools 11](#_Toc102411001)

[5.1 Hardware 11](#_Toc102411002)

[5.1.1 Arduino MEGA 2560 11](#_Toc102411003)

[5.1.2 Arduino Nano 11](#_Toc102411004)

[5.2 Software 12](#_Toc102411005)

[5.2.1 Arduino IDE 12](#_Toc102411006)

[5.2.2 Altium Designer 13](#_Toc102411007)

[5.2.3 W3Schools 13](#_Toc102411008)

[6 Sensors 14](#_Toc102411009)

[6.1 RFID Scanner 14](#_Toc102411010)

[7 RF devices 15](#_Toc102411011)

[7.1 RF 433Mhz Transmitter & Receiver 15](#_Toc102411012)

[7.2 RF 433Mhz Transmitter 16](#_Toc102411013)

[7.3 RF 433Mhz Transmitter Code 17](#_Toc102411014)

[7.4 RF 433Mhz Receiver 17](#_Toc102411015)

[7.5 RF 433Mhz Receiver Code 18](#_Toc102411016)

[8 Motors 18](#_Toc102411017)

[8.1 DC Motors 18](#_Toc102411018)

[8.2 Motor Control Code 19](#_Toc102411019)

[9 Web Server 20](#_Toc102411020)

[9.1 Web Server 20](#_Toc102411021)

[9.2 HTML & CSS Code 21](#_Toc102411022)

[9.2.1 HTML 21](#_Toc102411023)

[9.2.2 CSS 22](#_Toc102411024)

[10 Problem Solving 24](#_Toc102411025)

[10.1 Multiple TX and RX pins 24](#_Toc102411026)

[10.2 Switch sensitivity 24](#_Toc102411027)

[10.3 Using if else statements in HTML 24](#_Toc102411028)

[10.4 Implementing a button in HTML 25](#_Toc102411029)

[11 Ethical Considerations 26](#_Toc102411030)

[11.1 United Nations Sustainable Development Goals 26](#_Toc102411031)

[11.1.1 7\* Ensuring access to affordable, reliable sustainable and modern energy for all. 26](#_Toc102411032)

[11.1.2 13\* Take urgent action to combat Climate Change and its impact. 26](#_Toc102411033)

[11.2 Power 26](#_Toc102411034)

[11.3 Programming Style 27](#_Toc102411035)

[11.4 Sustainability - Component Reuse and Recycling 27](#_Toc102411036)

[12 Conclusion 28](#_Toc102411037)

[13 References 29](#_Toc102411038)

[Appendix 1: Code 31](#_Toc102411039)

[Appendix 2: Bill of Materials 32](#_Toc102411040)

[Appendix 3: Data Sheets 33](#_Toc102411041)

[Appendix 4: Schematic 34](#_Toc102411042)

# Summary

In the summer of 2021, the Irish government laid out its plans to switch the country to all electric vehicle EV sales by the year 2030 and to have nearly one million electric vehicles on the road the same year [1] My project aims to tackle some of the issues first time EV owners will likely run into daily and show the upsides of EV ownership and the use of clean sustainable energy.

For my project I’ve incorporated solar energy in the form of a solar roof like Tesla Solar Roofing. This will be a grid of solar collectors that double as rooftiles. While solar energy is a great option for sustainable home power, its main downside is that energy is normally created during the day when the usage of the household is at its lowest and in the evening and early night when energy consumption is at its highest the production of power is substantially lower. To combat this, I have incorporated a “Powerwall” which is often an array of lithium-ion cells used to store the energy created during the day for use during peak consumption hours. For the scale versions of these I’m using a mini solar panel for the roof and 4-6 18650 lithium-ion cells in a charger that has inbuilt cell balancing system as the cells must be balanced to prevent cell damage.

As you drive up to the garage there is a NFC scanner that communicates with an NFC chip in the side of the model car, allowing automated, secure operation of the garage door, for the garage door mechanism I’ve used a DC motor to open and close the door operated by an Arduino inside the garage. There is momentary switch at the bottom of the door to monitor its current state (open or closed) that also communicate with the Arduino.

When you enter the garage, you are greeted by your power wall as well as an EV charging lead and plug which will be powered by the wall. Inside the model car there is a second Arduino to monitor the battery voltage inside the car, these readings will both be accessible through a web-app as well as being communicated to the main Arduino inside the garage. The battery voltage level will also be compared with the max voltage of the EV and if it

# Introduction

For this project my hope is to demonstrate a model of what garages will look like in the future as well this it is an aspiration for what I hope my own home garage will look like in the future. Currently the technologies on display in this project are very much part of the luxury market as a large majority of lower middle-class families cannot afford the large upfront cost of an electric vehicle or solar panels. Given the current downward trend in the price of the technologies, I could see garages like this one becoming more and more common place.

NFC is becoming more and more vital in everyday life from contactless payment during covid to key cards for houses and apartments, NFC as a security feature for a garage just seemed like the correct choice as it is relatively low cost and requires very low computing power to operate. Batteries are becoming another area where there are massive amounts of money being put into R&D on new technologies such as the new Solid-State Battery that recently had a breakthrough at Harvard last year in production [2]. They are a very important component in our everyday life but also volatile if mishandled so working with them on this project was a good experience.

# The Internet of Things Overview

The Internet of Things, or IoT, refers to the billions of physical devices around the world that are now connected to the internet, all collecting and sharing data [3]. Thanks to the arrival of super-cheap computer chips and the ubiquity of wireless networks, it's possible to turn anything, from something as small as a pill to something as big as an aeroplane, into a part of the IoT. Connecting all these different objects and adding sensors to them adds a level of digital intelligence to devices that would be otherwise dumb, enabling them to communicate real-time data without involving a human being. The Internet of Things is making the fabric of the world around us smarter and more responsive, merging the digital and physical universes [4].

What is an example of an Internet of Things device?

Pretty much any physical object can be transformed into an IoT device if it can be connected to the internet to be controlled or communicate information.

A lightbulb that can be switched on using a smartphone app is an IoT device, as is a motion sensor or a smart thermostat in your office or a connected streetlight [5]. An IoT device could be as fluffy as a child's toy or as serious as a driverless truck. Some larger objects may themselves be filled with many smaller IoT components, such as a jet engine that's now filled with thousands of sensors collecting and transmitting data back to make sure it is operating efficiently [5]. At an even bigger scale, smart cities projects are filling entire regions with sensors to help us understand and control the environment [5].

# Project Architecture

The Arduino Mega R3 is the main micro controller I am using for my project. It handles the operation of the garage door as well as the hosting of the website and the battery monitoring. Inside the car there is an Arduino Nano V3 this allows me to monitor the current charge level and to communicate this information to the Mega. The website is facilitated using an Esp-01 Wi-Fi module connected to the Mega allowing it to transmit to the internet. The Nano and Mega communicate using a RF transmitter and receiver. An RC555 RFID module is used to allow entry into the garage with the correct key fob.

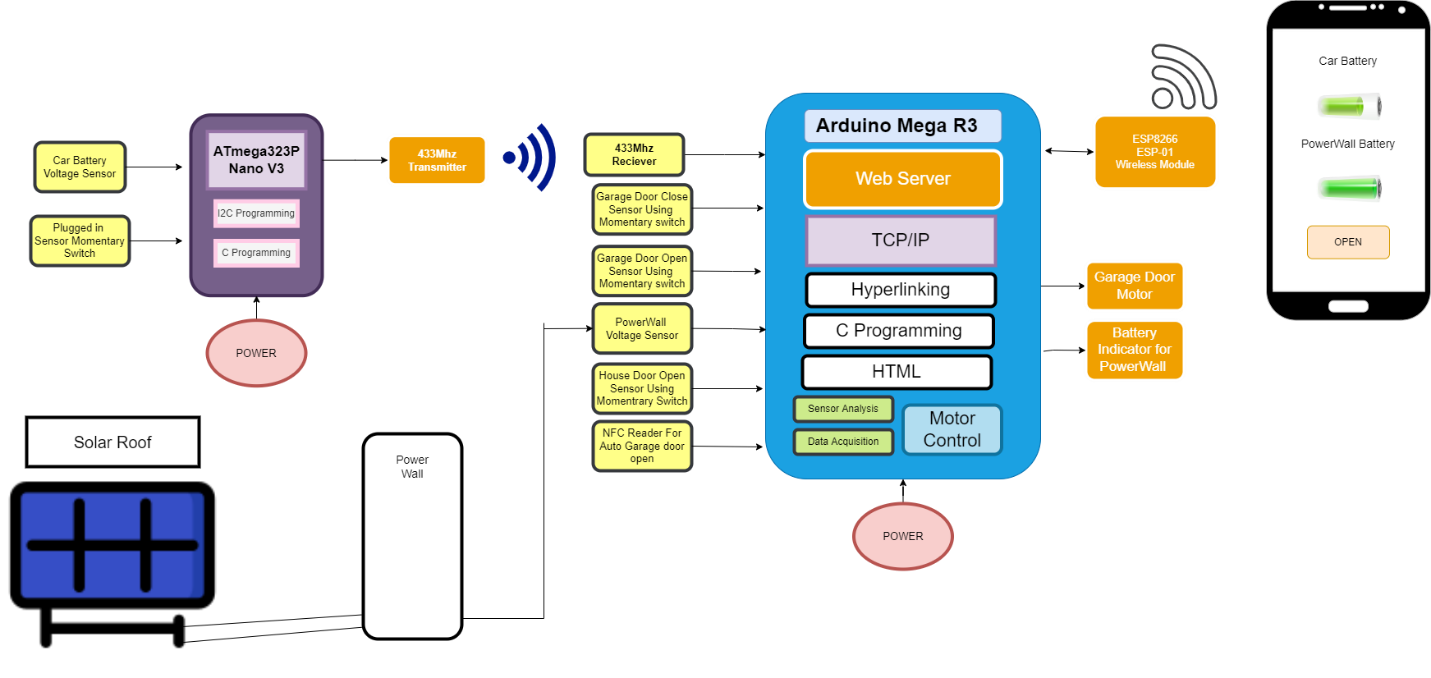


Figure 4‑1 Architecture Block Diagram

# Development Platform and Tools

## Hardware

### Arduino MEGA 2560

The Arduino Mega 2560 is the microcontroller that is used to aid the operation of most of the functionality of the garage. It collects data from the multiple sensors operates the motor and hosts the website. The main reason I chose this microcontroller is that it has 3 pairs TX and RX pins allowing me to use 3 different TX/RX devices.

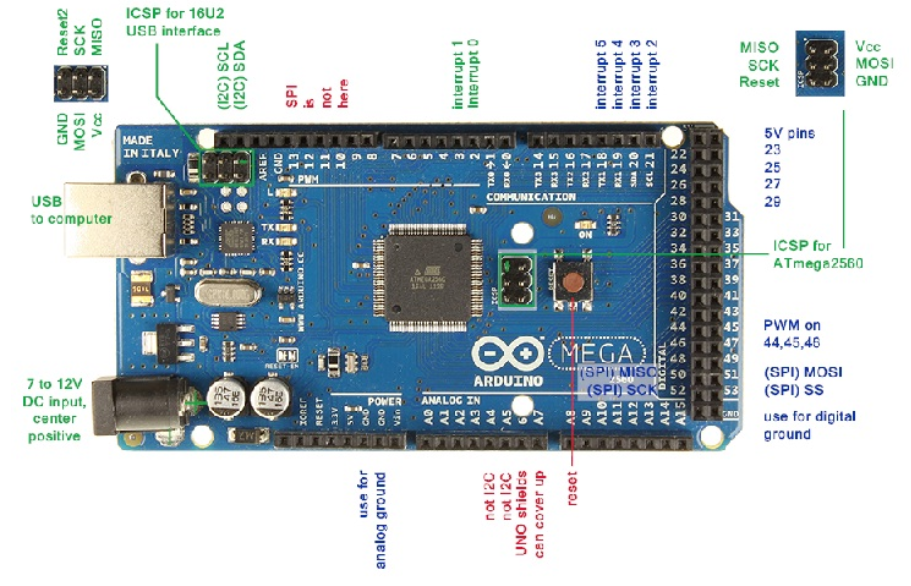


Figure 5-1 Arduino Mega 2560

### Arduino Nano

The Arduino Nano is the microcontroller used inside the car. It is used to monitor the cars voltage and send that information to the Arduino in the garage. I chose this controller as it is small enough to fit inside the car and can also be powered using the batteries inside the car.



Figure 5-2 Arduino Nano

## Software

### Arduino IDE

To program both Arduino’s I used Arduino IDE. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board. There is also many libraries already made to work with the Arduino ide that can be downloaded online.



Figure 5-3 Arduino IDE Logo

### Altium Designer

Altium Designer System Engineering (SE) is a fully-featured editor for schematics that includes powerful collaboration capabilities and a rich set of schematic capture tools to quickly create, edit, simulate, and document schematics. I mainly used the software to create schematics for my report.



Figure 5-4 Altium Designer Logo

### W3Schools

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. I used The HTML tutorial on W3Schools to design my HTML code and then copied it over to the Arduino.



Figure 5-5 W3Schools Log

# Sensors

## RFID Scanner

Radio Frequency Identification system is made up of two components the transponder and the transceiver (shown below). The transceiver creates a high frequency electromagnetic field, this field powers up the transponder in the paired tag and allows the information to be transmitted [6].

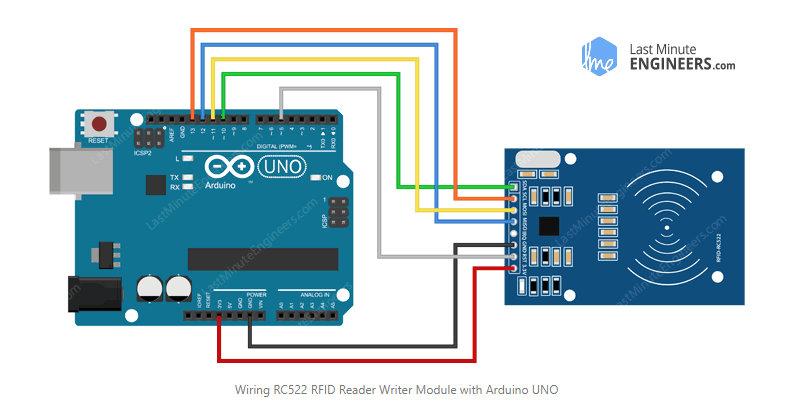
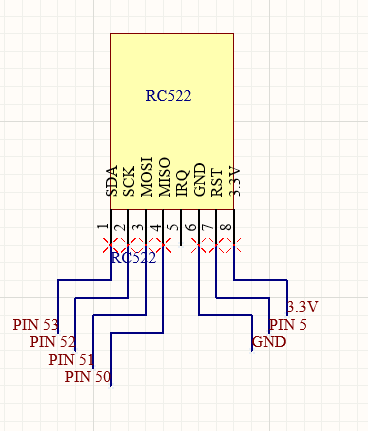


Figure 6‑1 RFID Scanner Circuit

The RC522 reader is designed to creates a 13.56MHz electromagnetic field to communicate with the tag. It communicates with the Arduino using a 4 pin Serial Peripheral Interface (SPI) The maximum data rate is 10Mbps [6]. To find out what the Unique Identifier (UID) is on the tag that I used for my project I opened the example code called dump info which shows the UID of any tag, phone, or other NFC capable device scanner by it. There is also another example code that can be used to reprogram the tag, which has a total of 1Kilobyte of memory that is made up of 16 sectors divided up into 4 block chunks that can store 16bytes of data [6]



Figure 6‑2 RFID Scanner Code

This code searches for a new card using the ariel on the RFID scanner once it senses its presents it will read the UID tag of the given card and prints it to the serial monitor if the UID tag matches the UID tag that is recognized by the code it will initiate the code to open the door.

# RF devices

## RF 433Mhz Transmitter & Receiver

The RF devices that I have used in my project send digital data over radio. The process used is known as Amplitude Shift Keying (ASK), this works using a technique like amplitude modulation [7]. When the data pin on the transmitter receives a digital 1 the carrier signal drives at full strength and if it receives a 0 it cuts the carrier off completely.

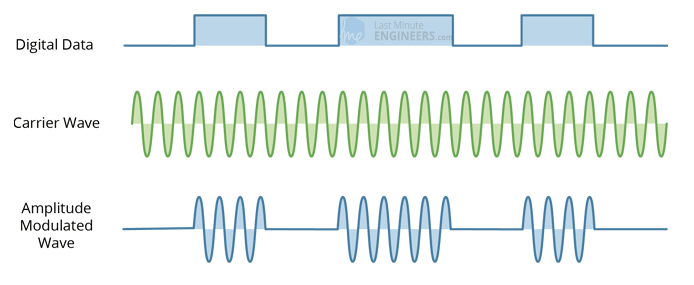


Figure 7‑1 RF Amplitude Modulation

## RF 433Mhz Transmitter

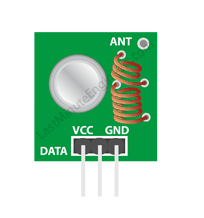
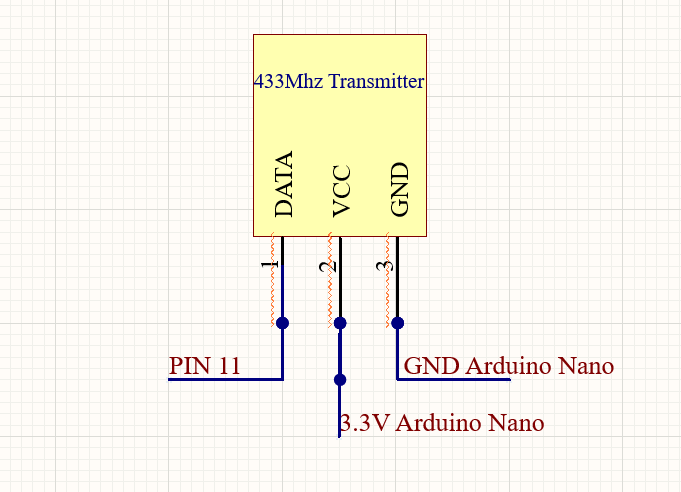
 

Figure 7‑2 RF 433Mhz Transmitter

Shown above is the transmitter module of the pair, the main element of the module is the SAW resonator that is set up for 433Mhz operation, as well as this there is a switching transistor an antenna and a few passive components [7]. Unfortunately, the antenna on the unit just was not powerful enough for the application I needed it for, so I had to make an extended antenna. For this, I used a solid copper core shielding wire wrapped 16 times around a 3-millimeter cylinder which greatly increased the range. I then connected it up to the Arduino Nano located inside the car so it could transmit the current voltage level of the car back to the Arduino Mega.

## RF 433Mhz Transmitter Code

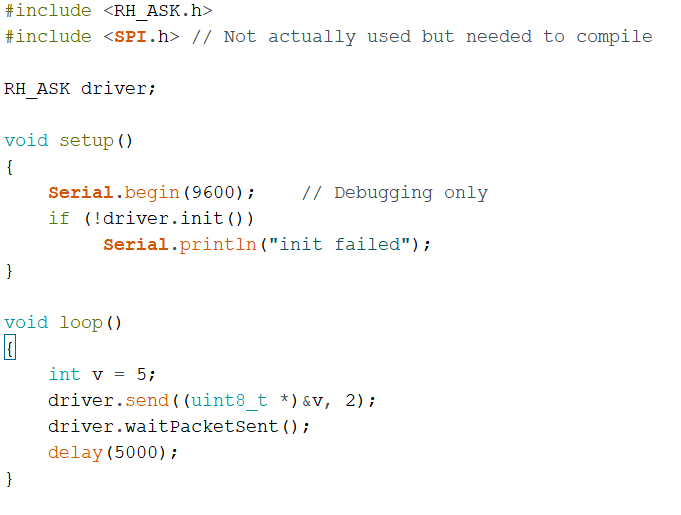


Figure 7‑3 Arduino Nano code for transmitter

The RH\_ASK.h Library is included so the functions located in it can be used in code, one of these is the RH\_ASK which creates an ASK object named driver. This is then initiated using rf\_driver.init(). The driver.send function is used to send the integer and then waits for confirmation that it has been sent.

## RF 433Mhz Receiver

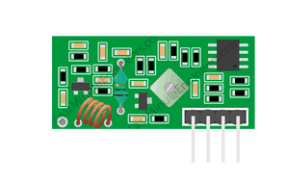
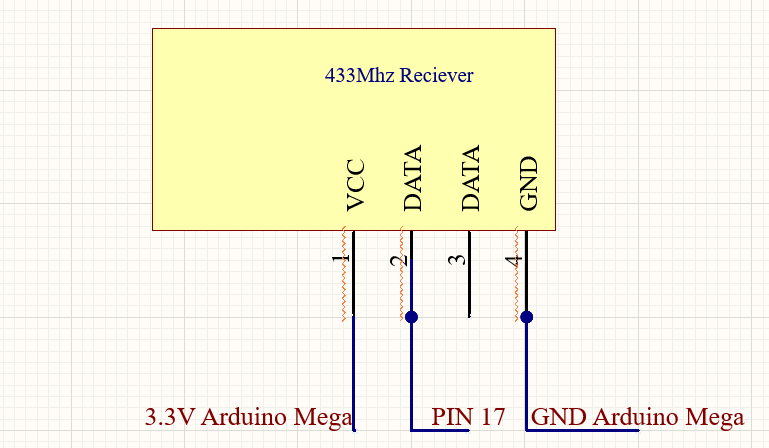
 

Figure 7‑4 RF 433Mhz Receiver

Shown above is the receiver module of the pair, it consists of an RF tuned circuit and 2 OP Amps to amplify the received signal, the signal is then sent to a Phase Lock Loop (PLL) that enables the decoder to catch the stream of bits which helps protect against noise and gives a better output. I made another external aerial for the receiver module. This module is connected to the RX pin 17 on the Arduino mega allowing it to receive the information from the Arduino Nano.

## RF 433Mhz Receiver Code

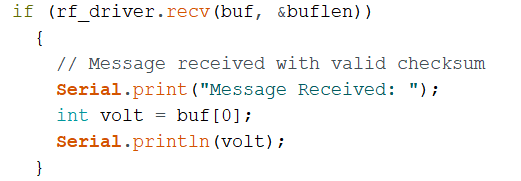


Figure 7‑5 RF 433Mhz Receiver

The above code is uploaded to the Arduino Mega, it checks to see if there is a packet received. The first element (the information) of the buffer is used to set the value of the car’s voltage. Not shown is the similar starting code to the transmitter code including the header file etc.

# Motors

## DC Motors

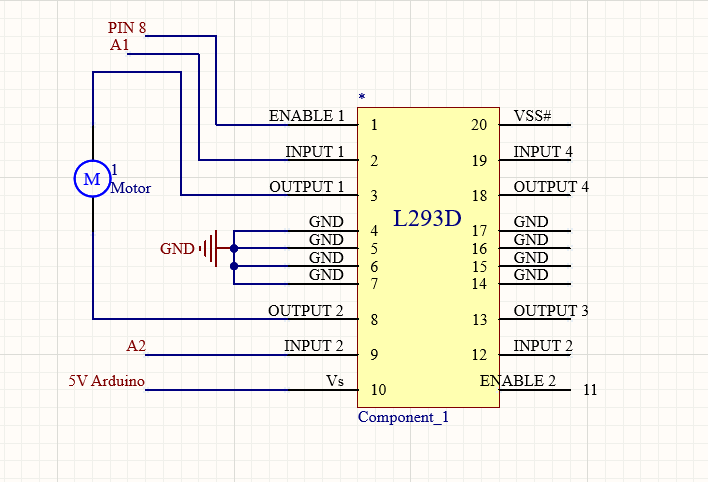
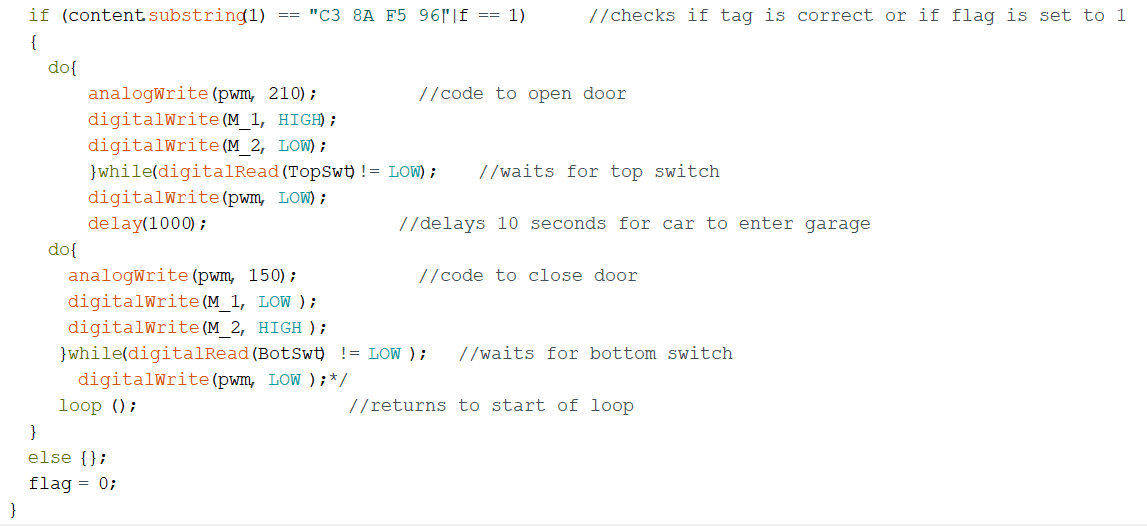
 

Figure 8-1 Motor Driver Circuit & Motor

I used a geared DC motor to control the operation of the rolling garage door as it didn’t require any fine variations in position. Given that the door needs to go up and down, I had to incorporate an L293D motor driver IC. This also allows the use of Pulse Width Modulation (PWM) on the enable pin of the chip so I could specify what speed the door goes up and down at.

## Motor Control Code

Explain your code in English, or with the aid of flow diagrams.

Figure 8-2 Motor Driver Code

This code is used by the Arduino mega, at the start of the code the program checks to see if the tags UID is equal to the saved UID. It also checks to see if the flag is set to one, if either of these parameters are set the code will execute. The motor driver code consists of two do, while loop. The first do while loop sets the PWM value to 210 (/255) as it is the opening cycle and takes more power to lift the weight of the door. It then starts the door on its upward cycle, this continues until the level switch at the top of the door is pushed in by the door itself. This brings the pin “TopSwt” which is set up as a pullup to low letting the program exit the loop. After waiting for a period, the program begins its downwards cycle. This continues until the metal bar that makes up the bottom of the door crosses 2 jumper cables placed at the entrance pulling the “BotSwt” which is also set up as a pullup to low, stopping the door at this point PWM is set to low stopping the motor completely, at the very end the flag is set back to zero.

# Web Server

To enable the use of Wi-Fi in my project I used an esp01 wireless module, this connects to the hotspot on a phone and allows the Arduino to host a webserver. This webserver then allows the user to monitor the charge levels of both the car and the garage, it also allows the user to open the door remotely.

## Web Server

The esp01 connected to the Arduino mega allows for a webserver to be created. This is facilitated using the WifiEsp library. A webserver is started at the beginning of the code on port 80  The ssid and password of the network being accessed is also specified at the beginning of the code.



Figure 9-1 Network information

The esp01 is then connected to the network and the ip address is printed to the serial monitor and the server is started.

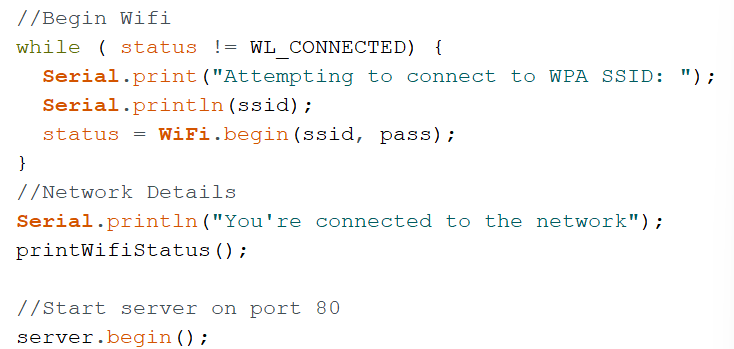


Figure 9-2 Network connection and server start code

## HTML & CSS Code

The website is used to display the battery level of both the car and garage as well as that it allows the user to open the door via a button. This is facilitated using both HTML for the website code and CSS for the styling.

### HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser [8]. In my code HTML is used for the display of the website.

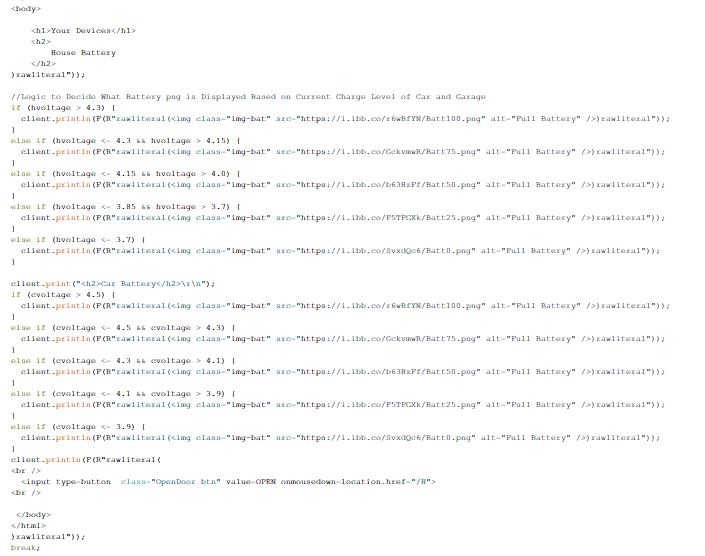


Figure 9-3 HTML Code

The html code is run by the Arduino and uploaded to the web using the esp01. The function of this code is to submit all the headings to the website. It also sends a specific link to an image hosted on a separate online hosting service based on the current charge level. This is sent to the esp using the rawliteral tags allowing HTML code to be directly inserted. There is also a button at the bottom of the webpage to open the door, this operates by appending a /H to

The end of the char c.

### CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML [9]. The css contained in my code sets the background colour to blue as well as specifying the styling of the button

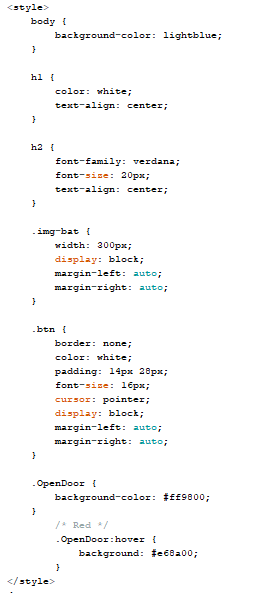


Figure 9-4 CSS code

The CSS code is placed inside the <style> tag as we cannot access separate files within the Arduino ide.

Graphical user interface, application

Description automatically generated

Figure 9-5 Finished Website

# Problem Solving

## Multiple TX and RX pins

Given that I have multiple devices that require serial communication with the Arduino I needed multiple TX and RX pins. Originally, I was using an Arduino uno that only has 1 set of the pins, the pins can however be simulated using software serial which is what was done for the esp01, however when I attempted to using the esp01, the rc522 and the 433Mhz receiver all together it would not work, the Arduino would not establish communication with the receiver. I attempted to set up another set of software serial ports to no avail. To rectify this, I was given an Arduino mega by my lecturer, as well as having 3 sets of TX/RX pins it also has far more digital I/O pins which came in handy later on.

## Switch sensitivity

Originally, I had intended on using two lever, operated microswitches to sense when the door was opened or closed, unfortunately the door just wasn’t heavy enough to push the lever switch when the door was closing. To try and rectify this I 3D printed a lever that got pushed by the door on one side and pushed the switch with the other, this worked ok but was prone to sometimes not fully press the switch. I decided that it made more sense to use the metal bar in the bottom of the door to cross 2 jumper wires across the entrance of the garage, I grinded the ends of the rod so there was exposed metal and once the door is fully closed the circuit is completed and the sensor pin is pulled down to ground.

## Using if else statements in HTML

For my website I needed to set up an array of if else statements to choose which battery png is displayed depending on the current charge level, the best way I could see to do the was using javascript but every time I tried to add a <script> to my HTML code it would crash the esp01 on startup. After many hours of research, I decided that I would just use c programming in the Arduino to handle the logics of the if else statements and used the client.print function to send the correct image link to the website.

## Implementing a button in HTML

As stated above anytime I tried to add a <script> tag to my HTML code it would crash the esp. This also limited the implementation of a button as I couldn’t add a button handler. After a bit of research, I decided to use href to append a /H to the end of client.read. I then checked for the /H after the HTML code and set a flag to 1 if it has been pressed this allows the door to open.

# Ethical Considerations

## United Nations Sustainable Development Goals

### 7\* Ensuring access to affordable, reliable sustainable and modern energy for all.

#### 7.1\* Universal access to modern energy.

By 2030, ensure universal access to affordable, reliable, and modern energy services [10]. The Home Garage of the Future is equipped with a solar panel on the roof that supplies charge to the Powerwall, solar covers all 3 of the goals parameters being affordable once installed, reliable and modern.

#### 7.2\* Increase global percentage of renewable energy.

By 2030, increase substantially the share of renewable energy in the global energy mix [11].

If implemented on a full-scale garage the surplus energy created by the solar panels could be sold back to the grid increasing the share of renewable energy.

### 13\* Take urgent action to combat Climate Change and its impact.

#### 13.2\* Integrate climate change measures into national policies and strategies and planning.

Integrate climate change measures into national policies, strategies, and planning [10].

There are many countries that already have programs in operation that provide subsidies for the purchase of electric vehicles, this makes them more accessible to working class people.

## Power

To power the car and hold the charge supplied by the solar panel, there are 18650 batteries used. These are a great option for hobbyist projects as they are readily available and relatively safe. They can however become hot or even explode if the are overcharged or balanced incorrectly. To combat this, I’ve used 2 18650 charge boards, one in the garage and on inside the car this allows me to ensure that the batteries are not overcharged making them safe for use in the project.

## Programming Style

The projects code is separated into tabs, so it is easier to follow, as well as tabbed it is also indented so it is clear to see what code falls into what logic scope.

## Sustainability - Component Reuse and Recycling

The 18650s used in the project are readily available and are normally found in devices like laptop batteries or power banks. The ones I used in this project were harvested from Makita drill batteries. The battery units were not holding a charge as some of the cells had died but once the plastic housing was removed the good cells could be harvested and the dead cells disposed of safely. The Powerwall charge board was originally a power bank charging board that was supposed to have 6 18650 batteries installed, I took this board and soldered the battery holder to it which allowed me to charge the Powerwall batteries.

# Conclusion

I’ve learned allot in the creation of this project and I think it’s a good blueprint for what homes and garages will look like in the future. After many months of research and testing I now realize that even the most complex seeming of technology can be learned and harnessed for use. I feel this project could be used as an example for prospective home builders and the technologies could be implemented in there builds. I also feel that it is a good representation of my ability in both software and hardware.

# References

[1]Department of Transport, “Government approves Electric Vehicle Policy Pathway Report,” *www.gov.ie*, 2021. <https://www.gov.ie/en/press-release/a6a2f-government-approves-electric-vehicle-policy-pathway-report/> (accessed May 02, 2022).

[2]L. Burrows, “Researchers design long-lasting, solid-state lithium battery,” *Harvard Gazette*, May 12, 2021. <https://news.harvard.edu/gazette/story/2021/05/researchers-design-long-lasting-solid-state-lithium-battery/> (accessed May 02, 2022).

[3]S. Ranger, “What is the IoT? Everything you need to know about the Internet of Things right now,” *ZDNet*, Feb. 03, 2020. <https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/> (accessed May 02, 2022).

[4]D. Rodelo, “Internet Of Things,” *www.linkedin.com*, 2021. <https://www.linkedin.com/pulse/internet-things-duvan-rodelo/> (accessed May 02, 2022).

[5]IIBA, “What is the Internet of Things (IoT)?,” *hongkong.iiba.org*, Mar. 27, 2018. <https://hongkong.iiba.org/news/what-internet-things-iot> (accessed May 02, 2022).

[6]Last Minute Engineers, “What is RFID? How It Works? Interface RC522 RFID Module with Arduino,” *Last Minute Engineers*, Jul. 30, 2018. <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/> (accessed May 02, 2022).

[7]“Insight Into How 433MHz RF Tx-Rx Modules Work & Interface with Arduino,” *Last Minute Engineers*, Jul. 05, 2018. <https://lastminuteengineers.com/433mhz-rf-wireless-arduino-tutorial/> (accessed May 02, 2022).

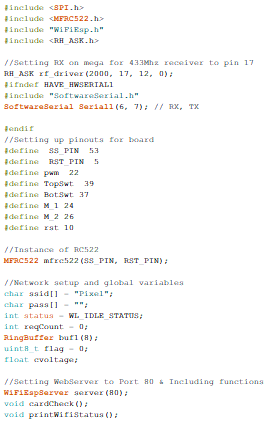
[8]“HTML,” *Wikipedia*, Apr. 13, 2022. https://en.wikipedia.org/wiki/HTML#:~:text=The%20HyperText%20Markup%20Language%20or (accessed May 02, 2022).

[9]“CSS,” *Wikipedia*, Sep. 17, 2020. <https://en.wikipedia.org/wiki/CSS>

[10]United Nations, “Climate Change - United Nations Sustainable Development,” *United Nations Sustainable Development*, 2021. https://www.un.org/sustainabledevelopment/climate-change/ (accessed May 02, 2022).

[11]“GOAL 7: Affordable and clean energy,” *UNEP - UN Environment Programme*, Oct. 02, 2017. https://www.unep.org/explore-topics/sustainable-development-goals/why-do-sustainable-development-goals-matter/goal-7#:~:text=Target%207.2%3A%20By%202030%2C%20increase (accessed May 02, 2022).

# Appendix 1: Code



Variable initialization and pin setup

# Appendix 2: Bill of Materials

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Item** | **Quantity** | **Manuf** | **Manuf No** | **GMIT Stores** | **Sourced from** | **Order No** | **Cost Euros** |
| Arduino Mega | 1 | Arduino | A000067 | y | Radionics | 715-4081 | 43.69 |
| 3K3 | 1 |  |  | y |  |  | 0 |
| 1K8 | 1 |  |  | y |  |  | 0 |
| ESP8266 | 1 | Sparkfun | WRL-17146 | y | Mouser | 474-WRL-17146 | 5.89 |
| RC522\* | 1 | JZK | B06X9PZSQN | n | Amazon |  | 6.50 |
| Arduino nano | 1 | Arduino | A000005 | y | Farnell |  | 22.2 |
| DC Motor | 1 | TT |  | y | Gleanntronics |  | 3.72 |
| Solar charger | 1 | DFRobot | DFR0264 | y | RS-Comp |  | 4.71 |
| Wooden Garage \* | 1 | Alex |  | n | Etsy |  | 60.47 |
| Solar panel | 1 | PowerFilm | P7.2-150F | y | Mouser |  | **15.50** |
| **Limit Switch\*** | **1** | Youmile |  | n | Amazon |  | **0.25** |
| 18650 batteryholder\* | **3** | Galdoep |  | n | Amazon |  | **5** |
| **18650 power bank\*** | **1** |  |  | n | ebay |  | **9.21** |
| **USB Charge Board\*** | **1** | Hayatec |  | n | Amazon |  | **1.25** |
| **RC Car\*** | **1** | CMJ |  | n | Amazon |  | **11.5** |
| **18650 Batteries\*** | **6** | STM |  | n | Salvaged |  | **Free** |
| **RF433Mhz Set\*** | **1** | kwmobile |  | n | Amazon |  | **2.5** |
| **L293D** | **1** | ST |  | y | Gleantronics |  | **1.5** |
| **Total** |  |  |  |  |  |  | **193.89** |

Components marked with an asterisk indicate that the student sourced this component themself.

# Appendix 3: Data Sheets

Arduino MEGA 2560:

<https://www.robotshop.com/media/files/PDF/ArduinoMega2560Datasheet.pdf>

ESP-01:

<https://www.microchip.ua/wireless/esp01.pdf>

MFRC522:

<https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>

433Mhz Transmitter & Receiver:

<http://www.energiazero.org/arduino_sensori/Complete%20Guide%20for%20RF%20433MHz%20Transmitter.pdf>

L293D:

<https://www.ti.com/lit/ds/symlink/l293.pdf>

Geared DC motor:

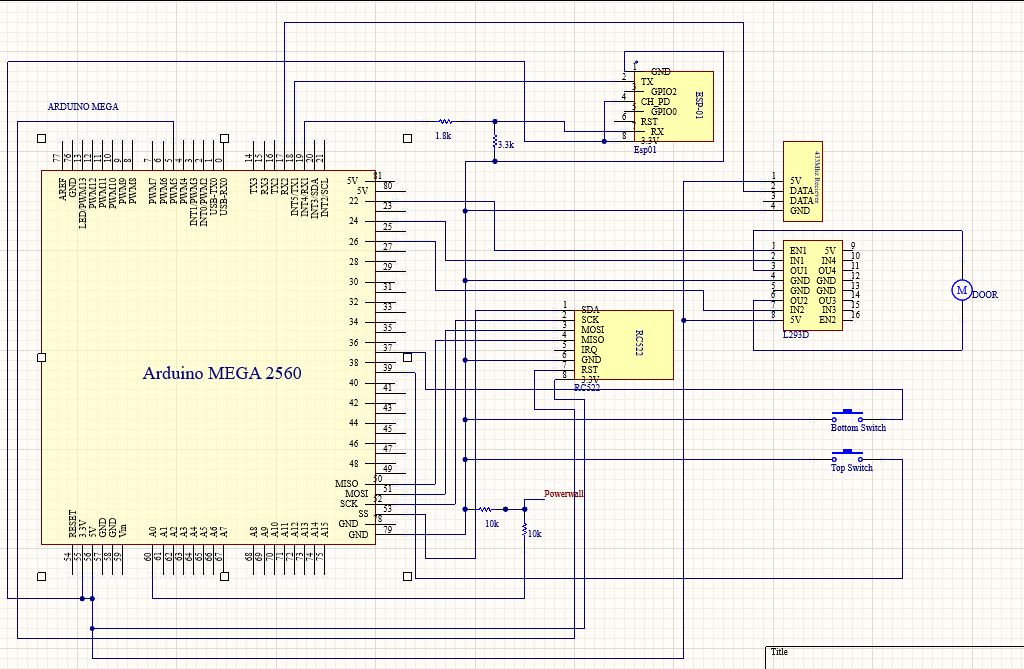
<https://media.digikey.com/pdf/Data%20Sheets/Adafruit%20PDFs/3777_Web.pdf>

Arduino Nano:

<https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>

# Appendix 4: Schematic

Arduino Mega Schematic



Arduino Nano Schematic

