Automated Model Farm Report

Travis Hyland A00231113
Computer Engineering 3 – Project 3
4/20/18

Table of Contents

Introduction

Summary

The main aim of this third-year project is to build and implement an Automated Model Farm system which collects data from various sensors and uploads the data to a website. This data would be monitored constantly allowing the farmer to check up on the farm while on the go elsewhere.

These sensors will be connected to multiple Arduino microcontrollers which will all send data to a central hub for processing. This data will then be sent to the Raspberry pi which will upload it to the website. This will be accomplished by hosting a website on the Raspberry pi and using port forwarding to make the website available anywhere. To control the sensors, multiple subsystems will be created to accomplish this wirelessly. These systems will be communicating wirelessly with the Arduino Mega using wireless connectivity. There will also be some independent systems which will perform other tasks elsewhere. These systems are listed below:

1. Shed Scraper System
2. Greenhouse
3. Hay Shed Alarm system and electric fence monitor
4. Tractor body detection and Burglar Alarm
5. Lighting control system
6. Water Trough monitor

Shed Scraper System

This system will be used to clean the cattle shed automatically every 4-6 hours. This would reduce the work of the farmer and keep the cattle content as they are not standing in filth. This system would be connected to an Arduino mini and use 2 motors to control the movement of the scraper.

Greenhouse

This system will be used to monitor the conditions of a greenhouse. This will be done using two sensors; a soil moisture monitor and a temperature sensor. Both systems will continuously take readings and send them wirelessly to the website. This will allow the farmer to know when to water the plants or if to increase the heat with a heat lamp.

Hay Shed alarm system and electric fence monitor

This system will be used to monitor the CO_2 levels in the hay shed. If they rise above a certain amount, an alarm will be set off which will alert the farmer of a fire. There will be a gas sensor used to detect the change in CO_2 levels in the shed. There will also be an electric fence monitor to detect if the fence is broken or not.

Tractor body detection and Burglar Alarm

This system will work independently to the rest of the network as to allow it to be used wherever the tractor might be. It will have 2 modes; body detection and burglar alarm. The body detection will work when the tractor is turned on and in use. It will alert the farmer if a body is detected near the tractor while it is in use. The burglar alarm will work when the tractor is turned off and locked. The farmer will have to arm it first and then it will monitor for any movement near the tractor when not in use.

Lighting Control System

This system will be used to control the lighting on the farm wirelessly though the use of Bluetooth. It will allow the farmer to turn on and off any of the lights on the farm from anywhere they would be in Bluetooth range. This would allow the farmer more time to accomplish tasks without having to go and turn on lights from a fixed location.

Water Trough Monitor

This system will wirelessly monitor the level of water in the cattle trough and when it drops below a certain level, a pump will be activated to refill the trough. This will the farmer to identify broken or frozen pipes more quickly and minimise the risk of cattle being dehydrated.

The use of these systems on modern farm could help reduce waste and costs which would help modern farmers concentrate on more important tasks. This system, although a model, could be implemented relatively easily on a farm and would help monitor the farm from wherever you might be whether it’s in bed or on holidays, you could be constantly updated and have more piece of mind that everything is working correctly.

Objectives

For our third year Project 3 module, we must complete a project based on wireless communication. These projects must meet three main objectives:

1. **Must incorporate wireless communication**

In the project, there must be a wireless component to allow certain parts of the project to communicate wirelessly. This means using means such as Wi-Fi, Bluetooth or ZigBee. Data must be sent and received in the project to show the wireless connection.

2. **Must use Arduino in project**

There must an Arduino used to control the sensors or mechanisms incorporated in the project. The Arduino can be used to control sensors, motors, communicate with a pc or server and is able to be altered indefinitely. The Arduino is a cheap and very easy to use microcontroller which would work excellently to control the projects being built.

3. **All persons in groups must contribute equally to the project**

For making the project, each group has two people. Each person in the group must contribute equally to the overall project. This means then that all marks are awarded fairly and all work done is split evenly.

Our idea for the project is to make an Automated Farm. To make this project, we have many personal objectives:

1. **Will use raspberry pi**

The raspberry pi must be used in the project to host a website remotely. We would use the pi to host a website and to be the hub which receives the data from different sensors and meters around the farm. This would also be used to run the Cattle feed management program which would allow us to track feed usage and purchase more if needed.

2. **Will make a website to monitor project**

We will make a website to monitor the projects sensors and motors. This will allow us to remotely check and see if there is enough feed, if there is water in the troughs and if the electric fence is working. This website will be running on the raspberry pi and will be accessible anywhere.

3. **Will make a farm app**

We will make an app which will allow us to view the website on a mobile device easily and use the features. This app will be made for android phones as so to be able to install it onto any android phone easily. This app will also allow you to turn off the electric fence remotely to make repairs or adjustments.

4. **Will make a feeding program**

We will make a program that will allow us to track the usage of cattle feed in the shed. It will consider the amount of meal bought, used and left in the storage. This program will run on the raspberry pi and will be used to measure the meal used daily and estimate when more will have to be ordered.

Health and Safety

With the connecting of header pins, wires and other components, a considerable amount of soldering will be performed. There will also be moving parts on the project which could cause harm. To attach the parts to the model, a hot glue gun will be used which could cause bodily harm. These tasks could cause bodily harm and precautions will be taken to prevent any accidents. These precautions are listed below:

Soldering Iron

- A soldering irons heating element can reach temperatures of up to 400^0c, you must place the soldering iron in a stand which will prevent it from cause harm to anything or anyone. See Figure 1 below.

Figure 1 Soldering Iron in Stand

- Never hold wires or components with your hands while soldering, always use a clamp. See Figure 2 below.

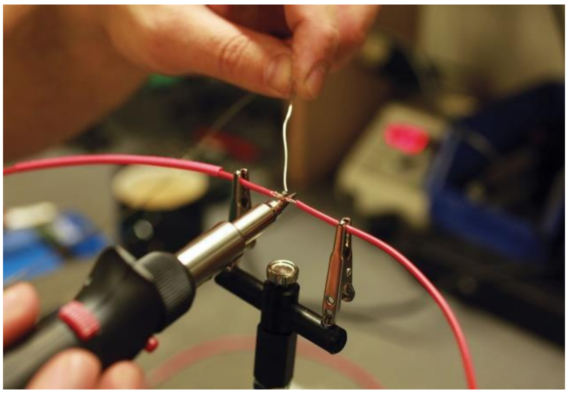


Figure 2 Clamp in Use

- Regularly clean the tip of the soldering iron in the sponge. See Figure 3 below.

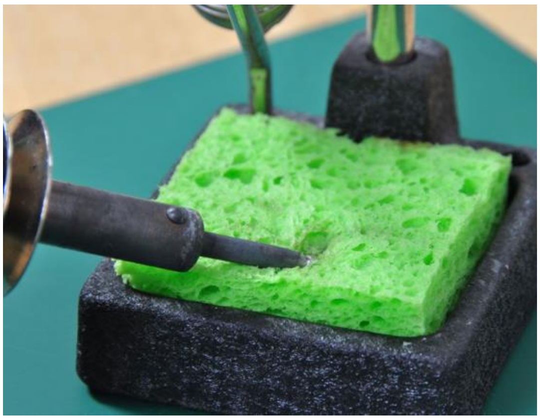


Figure 3 Wiping iron tip

- Solder can splash, be sure to wear appropriate eye protection. See Figure 4 below.

Figure 4 Safety Goggles

- Rosin is a resin which is contained in soldering flux. This is what generates the fumes seen when soldering. Long term exposure to this can lead to health issues. When soldering, you should always be in a well-ventilated area.

Moving parts

- Keep all long hair tied back to prevent it getting caught in moving parts. Also, don’t wear loose clothing or any ties when working near moving parts.
- Keep your hands back when a part is in motion. This will help reduce the likelihood of getting injured from moving parts.

Glue Gun

- A glue guns heating element can reach temperatures of up to 120^0c, you must place the glue gun in a stand which will prevent it from cause harm to anything or anyone.
- After a glue gun is used, some melted glue may still leak out of the nozzle. To prevent injury, keep a glue gun placed upright on a piece of wood or paper to catch any leaking glue.

Background Research

For the project, background research had to be completed to further understand what tasks had to be undertook and what parts were to be used to build and program the project.

Firstly, the individual ideas had to be researched to understand how they will work and what parts are required to make them. Then the parts will have to be examined to know if they will work correctly. Once all of this is complete, the project build can then commence. The ideas being used are listed below:

1. Shed Scraper
2. Water level Monitor
3. Greenhouse temperature and soil moisture measurement
4. Tractor alarm system and body detection
5. Feed Monitoring Program
6. Electric Fence Breakage Monitor
7. Hayshed Monitor
8. Farm lighting system

Shed Scraper

This tool will be used for the clearing of animal waste from cattle sheds. This system will be controlled wirelessly from the main Arduino mega. This scraper will use a rack and pinion method of movement. The pinion will be connected to a motor which will be controlled by the Arduino. The shed scraper will work on a routine where the shed will be cleaned every 3 hours.

Inspiration for the inclusion of this idea came about from the following video: https://www.youtube.com/watch?v=cTLtG6WpqCY

Water Level Monitor

This system will be used to monitor the cattle drinking troughs and keep them full consistently. This will work by using a water level sensor and a small water pump. The water level sensor will be placed in the tank to monitor and the water pump in a separate water storage container. Once the level of the water drops below a certain point, the water pump will trigger and start pumping water into the tank. The status of the water level will be uploaded constantly to the website to help log any breakages or blocks.

Greenhouse temperature and Soil moisture measurement

This system will allow the user to check the current state of the greenhouse temperature and soil moisture by using the website or app. A soil moisture sensor will be used to take readings of the soils water content. This then will pass information to the main Arduino which will be connected to the Raspberry Pi, which will then display the relevant information on the website.

The same principle applies to the temperature sensor which will take readings of the current temperature in Celsius in the greenhouse. This information will then be passed to the main Arduino wand onto the Raspberry Pi, which in turn will upload it to the website.

Tractor Alarm System and body detection

- **Tractor alarm system**
  This will incorporate the use of magnet switches, PIR sensors and sirens to alert the farmer of any harmful activity. There will be magnet switched attached to the locks on the doors of the tractors. When the lock is closed, the alarm can be armed. The alarm is armed using a NFC card or keyring which the farmer possesses. When armed, the doors will be monitored but there will also be PIR sensors placed around the tractor. If these detect any movement, they alarm will trigger. Once the alarm is triggered, it will sound the alarm for 5 minutes and if movement is still detected, the timer alarm will start again. To silence the alarm or disarm it, the farmers NFC tag or card will have to be used.
- **Body Detection**

  The body detection system will work like cars parking sensors except they will only detect movement, not stationary objects. When movement is detected around the tractor, a small buzzer will sound in the tractor which will alert the driver of potential accidents. This system will use the same PIR's used in the alarm. This addition will aid in decreasing farming accidents.

Feed monitoring program

This program will serve the purpose of monitoring fee usage on the farm. It will allow the farmer to purchase more feed from the local agri co-op and to track usage over a certain amount of time. The application will work by having the farmer input usage and this information is the processed and recorded accordingly. The total amount of feed on the farm will be updated depending on how much is used daily and how much there is after a delivery.

The application will be created using java. The data will be input using a touchscreen which will be mounted in the feed shed. There will be a button that will allow the farmer to edit the amount of feed in storage after they enter the correct login information.

When the feed amount has dropped below a certain level, the farmer will be able to order more feed using a button at the bottom of the screen. The program will then email the local co-op with the order details.

Electric fence breakage monitor

This system will provide a method for the farmer to be alerted instantly when the electric fence is broken down. This could have happened because of an animal breaking it or from the wire being grounded by overgrown foliage. The fence will be connected to the Arduino mini which will continuously monitor it for any fault. Once current is flowing through the circuit, the fence will be active and this will then send a signal to the website updating that it is working. If the fence is broken, a message will be sent to the website notifying that something has broken down the fence. The farmer will be able to check the status of the fence by viewing the website or through the app.

Hayshed monitor

This system will grant the farmer a method to help prevent fires occurring in the hay shed. There will be a MQ-2 gas sensor used to detect the $CO2$ in the air. When the gas sensor detects $CO2$, it sends a signal to the website which tells the farmer that there is a fire in the shed. Fires could occur from hay being wet when being put in the shed and it drying to rapidly or by someone starting one purposely.

Farm Lighting system

This system will be used to control the lights on the farm It will allow the farmer to turn on and off the lights using the mobile app and a Bluetooth module. There would also be a way for the farmer to turn on sperate lights so areas where light isn't needed wouldn't be lit. This would help reduce electrical cost and the lights wouldn't disrupt the cattle at night if they are needed elsewhere.

Neopixels will be used to light the farm as they use less power and provide better light than normal bulbs.

Wireless Background Research

In this project, there will be a wireless method of communicating from one Arduino to another and vice versa. There are various methods that can be implemented.

These methods are listed below:

1. **Bluetooth**

   Bluetooth is a very simple to use wireless connectivity method. It allows serial communication over a 10-meter radius. The module which we would use for this method is the BT06 Bluetooth module. This method has advantages and disadvantages though. These are listed below:

   <u>Advantages</u>

   - Ease of Availability: These days, Bluetooth modules are found in all smartphones and most laptops which makes it a very easy method to use as you could connect another device whenever needed with little tweaking required.
   - Cheap: You can purchase Bluetooth modules for very little on Aliexpress which would make the use of them much more viable.

   <u>Disadvantages</u>

   - Range: Bluetooth has a very short range of only 10 metres. This means that the sender and receiver would have to be located very close to each other which would limit the system dramatically.
   - Data Rate: Bluetooth data transfer rate is very slow and as the distance get further, the rate gets slower. This would limit the amount of data being able to be sent at once over the network.
   - Signal Interference: Over a small area, only a certain amount of Bluetooth modules can communicate as if there are many, interference occurs which might delay the transfer of vital information.
   - Security: The security standards of Bluetooth are very low. The Bluetooth network is very easy to get into from an outside source which makes the system very unsafe for use.

2. Wi-Fi

Wi-Fi is a very good method of wireless connectivity. It allows a connection of over 100m. The module chosen for this method is the NRF24L01 wireless transceiver. Like Bluetooth, this method has advantages and disadvantages. These are listed below:

<u>Advantages</u>

- Expandability: It is very convenient to add more wireless devices to a network. All you need to do is connect a module and enter some simple configuration settings. This would allow a person to add more sensors and mechanisms to the system very quick and easily.
- Efficiency: Wi-Fi modules require very little power to run which make them very energy efficient. They can also communicate with multiple devices at once which makes them very useful in this project.
- Security: Wi-Fi modules are much more secure to use in a project as the network will only allow a new device to communicate if they enter the correct configuration settings. It makes the system harder to break into and manipulate.
- Data transfer rate: Wi-Fi can transfer large amounts of data over a network which makes them a very plausible application for use in this project

<u>Disadvantages</u>

- Reliability: Even though Wi-Fi signals are very strong at close to medium range, many different factors can decrease its strength. These factors include concrete walls, electrical wiring and galvanised sheeting.

3. ZigBee

ZigBee is another wireless communication method. The module can communicate up to 100 feet indoors or 300 feet outdoors (with line-of-sight). Like Bluetooth and Wi-Fi, this method has advantages and disadvantages.

<u>Advantages</u>

- Ease of use: Setting up a network using ZigBee is very simple and easy to do. You can get two Arduino boards with Xbee shields talking to each other without any configuration, using just the standard Arduino serial commands.
- Low power usage: ZigBee uses lithium batteries which last for a long time without needing replacement.

<u>Disadvantages</u>

- Security: ZigBee is not secure like Wi-Fi. It is more open to outside intruders to alter the system. This makes ZigBee very unreliable to use in a public system.
- Coverage: ZigBee has a very limited outdoor coverage. It can do 300 feet but only with line of sight. In this project, the transceivers are going to be placed in sheds and away from line of sight altogether so this system would not work.

In the end, the chosen method of wireless communication to be used in the project is Wi-Fi. It would allow the entire system to be more secure, more flexible and more reliable. Even though Bluetooth is an easier method to implement, the disadvantages outweigh the advantages which would limit the projects capabilities.

Components

In this project, many different types of components will be used to complete the system. The components range from sensors and Wi-Fi shields to pumps and motors. The components that will be used in this project were all thoroughly researched beforehand to be sure that they are best suited for this project. These components are listed below:

1. **NRF24L01+ Wi-Fi Module**

This is a Wi-Fi transceiver that works on the 2.4GHz ISM band. It has a low operating voltage of 1.9 to 3.6 volts. It has a high transfer rate of 2Mbps and has 125 frequency points to meet any needs of multi-point communication and frequency hopping communication. It also has a built-in antenna which makes it very compact.

This will be used to allow the Arduino's and the Raspberry pi to communicate efficiently. There will be one mounted on each Arduino which will allow them to communicate wirelessly.

This product was purchased at the following link:

https://www.aliexpress.com/item/2pcs-lot-Free-shipping-NRF24L01-wireless-data-transmission-module-2-4G-the-NRF24L01-upgrade-version-New/32682504145.html?spm=a2g0s.9042311.0.0.KnBrXe

2. MQ2 MQ-2 Gas Sensor Module

The MQ2 MQ-2 gas sensor is suitable for detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane. Its fast response time and high sensitivity makes it well suited for the project. The sensitivity of the sensor can be adjusted using the potentiometer. It has a wide detecting scope and long, stable lifetime.

This will be used in the hay shed on the farm. It is so at night when the farmer is not at the farm, this will monitor for any fire starting in the bales which will then alert the farmer.

This product was purchased at the following link: https://www.aliexpress.com/item/MQ-2-Smoke-liquefied-flammable-gas-sensor-module-methane-gas-for-arduino/32549378289.html?spm=a2g0s.9042311.0.0.Kl77rz

3. Submersible Water Pump

This is a self-priming miniature mute submersible pump. It can be powered with 3v or 5v connection with a 100mA current. It can pump 1.2 to 1.6 litres per minute which is more than enough for the project.

This will be used to water the flowers in the greenhouse and to fill the cattle water troughs on the farm. It will be turned on and off using a transistor and water level sensor in the trough and with a transistor and soil moisture detector in the greenhouse.

This product was purchased at the following link:

https://www.aliexpress.com/item/1pcs-Mute-Submersible-Pump-Water-Pump-DC-3V-5V-For-PC-Cooling-Water-Circulation-DIY/32816435741.html?spm=a2g0s.9042311.0.0.Hht6dW

4. ATMEGA168 5V 16MHz Arduino Pro Mini

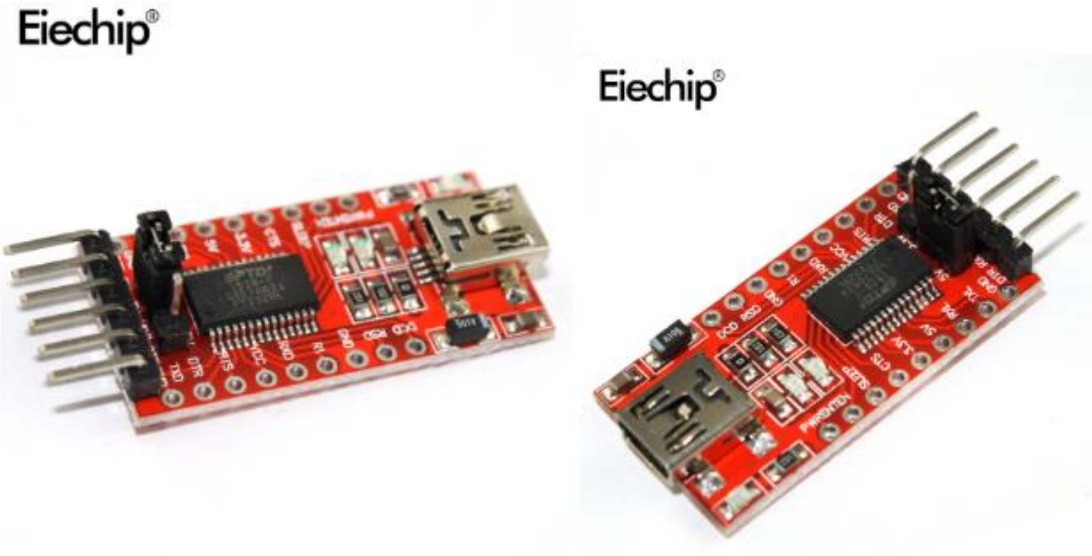
This is a microcontroller board made based upon the ATMEGA168 chip. It has 6 analogue I/O, 14 digital I/O, an on-board reset button and header pins for serial connection. It runs at 5V and 16MHz.

This is what will be used to control all parts of the project. They are small which allows them to be hidden easily and still offer the same power as a normal Arduino Uno board. These will use the Wi-Fi module to communicate wirelessly to each other.

This product was purchased at the following link:

https://www.aliexpress.com/item/10pcs-Pro-Mini-168-Mini-ATMEGA168-5V-16MHz-For-Arduino-Compatible-With-Nano/32250395153.html?spm=a2g0s.9042311.0.0.ldykDG

5. FT232RL USB to Serial Adapter Module



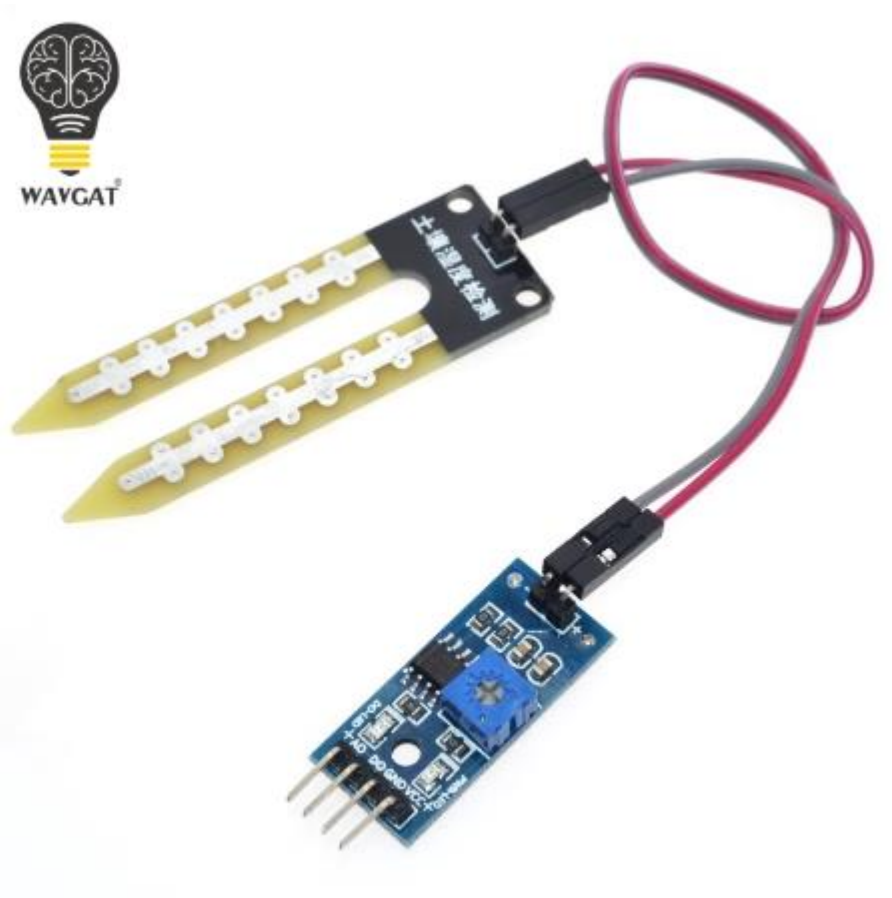
This is a serial to USB adapter that allows the Arduino mini to be programmed through USB. It has built in current protection using a 500MA self-restore fuse. It supports 3.3V and 5V power which suits this project.

This will be used to program the Arduino mini boards as they have no on-board connector for USB. Jumper cables are needed to connect this properly to the board.

This product was purchased at the following link:

https://www.aliexpress.com/item/Free-shipping-1pcs-lot-New-FT232RL-FT232-USB-TO-TTL-5V-3-3V-Download-Cable-To/32645814447.html?spm=a2g0s.9042311.0.0.ldykDG

6. Soil Moisture Hygrometer Detection Sensor



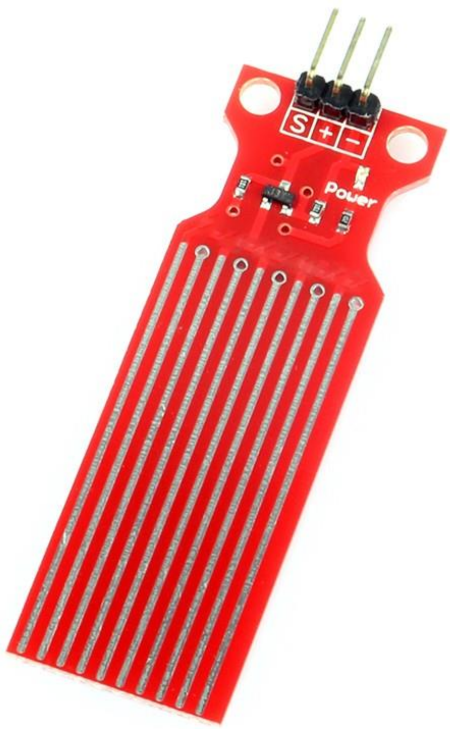
This is a soil moisture detection sensor. It can be used to detect the moisture content of soil and send a signal to the Arduino. Whenever the soils become dry, the output is high. The sensitivity can be adjusted using the on-board potentiometer. It operates at 3.3V to 5V.

This will be used in the greenhouse to detect the moisture content every 5 minutes. This will allow the plants to be grown in the best soil conditions.

This product was purchased at the following link:

https://www.aliexpress.com/item/Smart-Electronics-Soil-Moisture-Hygrometer-Detection-Humidity-Sensor-Module-For-arduino-Development-Board-DIY-Robot-Smart/32562744759.html?spm=a2g0s.9042311.0.0.ldykDG

7. Water Level Sensor



This sensor essentially describes itself as it records the level at which a liquid is at, based upon the liquids level measurement on the probes of the sensor. This component usually supplies a small amount of current, when the liquid reaches a certain point on its probes it will come into contact with a longer and shorter electrode. This then completes a circuit and sets off an internal switch which determines the level that the liquid has reached. The sensor has an operating voltage of 3-5v and has a detection area of 40mm x 16mm.

This component will be used in the water trough monitor system. It can be purchased at the link below:

http://irishelectronics.ie/epages/950018241.sf/en_IE/?ObjectID=3841457

8. PIR

The PIR sensor is made up of a pyroelectric sensor which can detect levels of infrared radiation up to 10 metres away. They have a low power consumption, low cost and have a wide lens range which is needed for the tractor movement alarm.

When motion is detected the sensor outputs a high digital signal on the output pin and waits three seconds. If movement is still detected after 3 seconds, the sensor continues to output high. This sensor can use 3v or 5v to power it.

This sensor can be purchased at the link below:

https://www.aliexpress.com/item/Free-shipping-1PCS-LOT-HC-SR501-HCSR501-SR501-human-infrared-sensor-module-Pyroelectric-infrared-sensor-imports/32700407854.html?spm=2114.search0104.3.2.ec76ce96bK52pt&ws_ab_test=searchweb0_0,searchweb201602_3_10152_10151_10065_10344_10068_5722815_10342_10343_10340_5722915_10341_5722615_10697_10696_10084_10083_10618_10304_10307_10302_5722715_10059_10534_308_100031_10103_441_10624_10623_10622_5722515_10621_10620,searchweb201603_19,ppcSwitch_5&algo_expid=ce8f8f2b-490b-478e-9072-085138c900fc-0&algo_pvid=ce8f8f2b-490b-478e-9072-085138c900fc&transAbTest=ae803_2&priceBeautifyAB=0

9. Bluetooth Module

The BT06 Bluetooth module is designed for quick and efficient data transmission. This Bluetooth module supports the UART interface and supports the SPP protocol which has the advantages of low power consumption, small size, high sensitivity and low cost. It uses 3.3v to power it which suits our project perfectly. It also has a built in PCB radio frequency antenna which will give better connection to devices. It operates over the 2.4 – 2.48GHz unlicensed ISM band.

The module can be purchased at the link below:

https://www.aliexpress.com/item/Bluetooth-Serial-Port-Wireless-Data-Module-Compatible-SPP-C-With-HC-06-Arduino-Bluetooth-3-0/32827603914.html

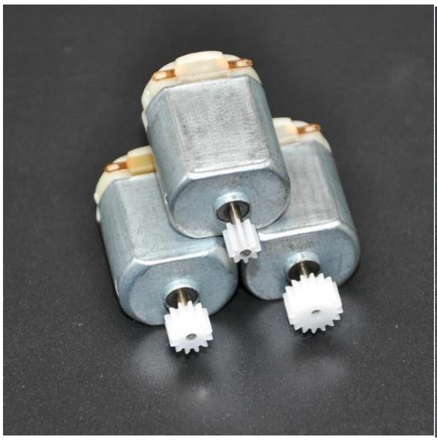
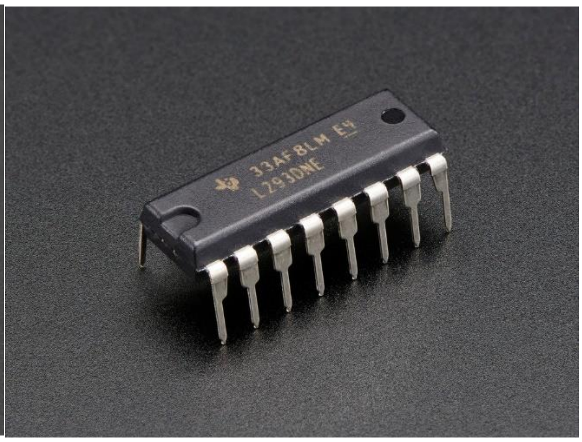
10. DHT11 basic temperature and humidity sensor

A DHT11 is a low-cost, basic digital temperature and humidity sensor. It uses a thermistor to measure the air temperature and a capacitive humidity sensor to detect the humidity. It sends out a digital signal that requires careful timing to grab correctly. The only downfall is that the sensor has a rate of one reading every two seconds which in some situations might be too slow and inaccurate.

This sensor can be purchased at the link below:

https://www.aliexpress.com/item/Smart-3pin-KEYES-KY-015-DHT-11-DHT11-Digital-Temperature-And-Relative-Humidity-Sensor-Module-PCB/32730774914.html?spm=2114.search0104.3.27.5c214fafYdeOWa&ws_ab_test=searchweb0_0,searchweb201602_3_5722916_10152_10065_10151_10344_10068_5722816_10342_10343_10340_10341_10698_10697_10696_5722616_10084_10083_10618_10304_10307_10302_5722716_5723016_5711215_10059_10534_308_100031_10103_441_10624_5722516_10623_10622_5711315_10621_10620-10622,searchweb201603_15,ppcSwitch_5&algo_expid=48e3b03f-0d9e-4931-a77b-b83032bdc7c7-6&algo_pvid=48e3b03f-0d9e-4931-a77b-b83032bdc7c7&transAbTest=ae803_2&priceBeautifyAB=0

11. DC Motor and L293D Motor driver chip

The DC motor runs on 3-5v. These can be speed controlled with PWM with an Arduino but a separate power source is needed to power them as the Arduino digital pins cannot provide enough power to run the motor. Therefore, the L293d motor driver chip must be used.

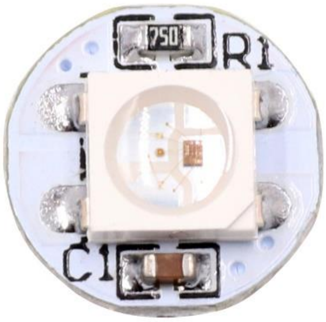
The L293d motor driver chip can be used to drive two sperate DC motors with up to 600mA per channel. The chip contains two full H bridges which means you can drive 4 solenoids.

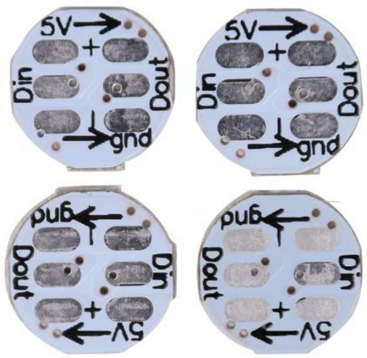
The dc motors and l293d chips can be purchased at the following links respectively:

https://www.adafruit.com/product/711

https://www.aliexpress.com/item/Free-shipping-5pcs-lot-L293D-L293-DIP16-IC-Best-quality/32835396694.html?spm=2114.search0104.3.22.60804151A8Inb7&ws_ab_test=searchweb0_0,searchweb201602_3_5722916_10152_10065_10151_10344_10068_5722816_10342_10343_10340_10341_10698_10697_10696_5722616_10084_10083_10618_10304_10307_10302_5722716_5723016_5711215_10059_10534_308_100031_10103_441_10624_5722516_10623_10622_5711315_10621_10620,searchweb201603_15,ppcSwitch_5&algo_expid=52fb1225-28a2-47ad-9357-70adc669d734-6&algo_pvid=52fb1225-28a2-47ad-9357-70adc669d734&transAbTest=ae803_2&priceBeautifyAB=0

12. SK6812 RGBW Neopixels





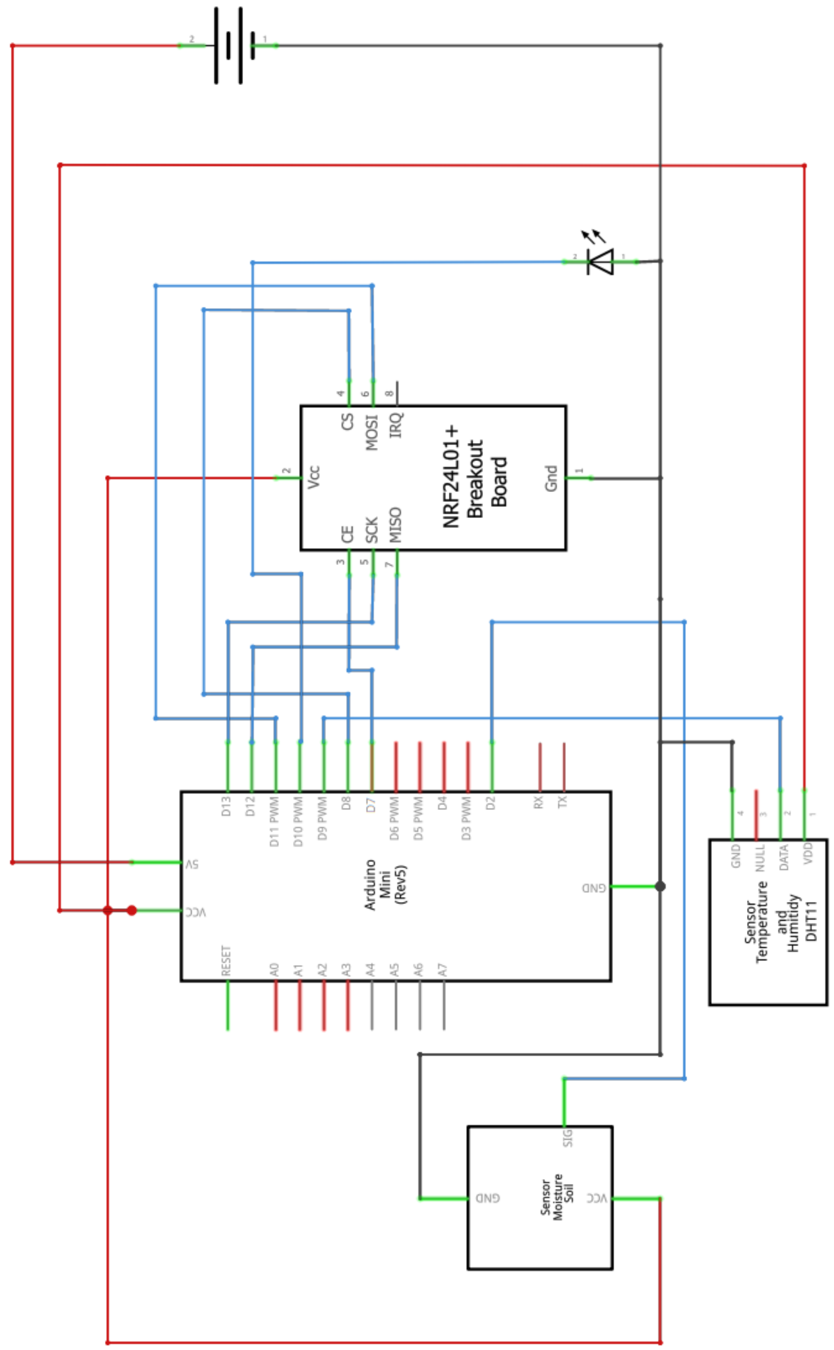
The SK6812 Neopixels are small full-colour LED’s that are small, relatively cheap and easy to implement and use. They have red, blue and green LED’s integrated alongside the driver chip onto a tiny surface mount-package controlled by a single wire. They can be turned on individually or all at once. They required a microcontroller to use them and a coding library which allows them to work.

The Neopixels can be purchased at the following link:
https://www.aliexpress.com/item/SK6812-RGBW-warm-white-led-addressable-strip-60leds-m-with-60pixels-m-non-waterproof-4m-long/32664325219.html?spm=a2g0s.9042311.0.0.k14yyz

Hardware Design

This section of the report is focused on the physical side of the project, here there will be a description of the individual circuits and their operation. For each circuit created in this project there will be a detailed description explaining how they operate. A circuit diagram for each system will be shown and the connections that are used will also be shown on the diagrams.

Greenhouse



The Greenhouse incorporates the use of the following components; Arduino mini, DHT11 sensor, Soil moisture sensor, RF24 wireless module, LED and a battery pack.

The greenhouse circuit is powered using two AA batteries in a battery pack which each provide 1.5v of power each which in total provides 3v output. Ideally the Arduino should be powered by a 5v power supply but they were not low cost and as easily accessible so these were used instead.

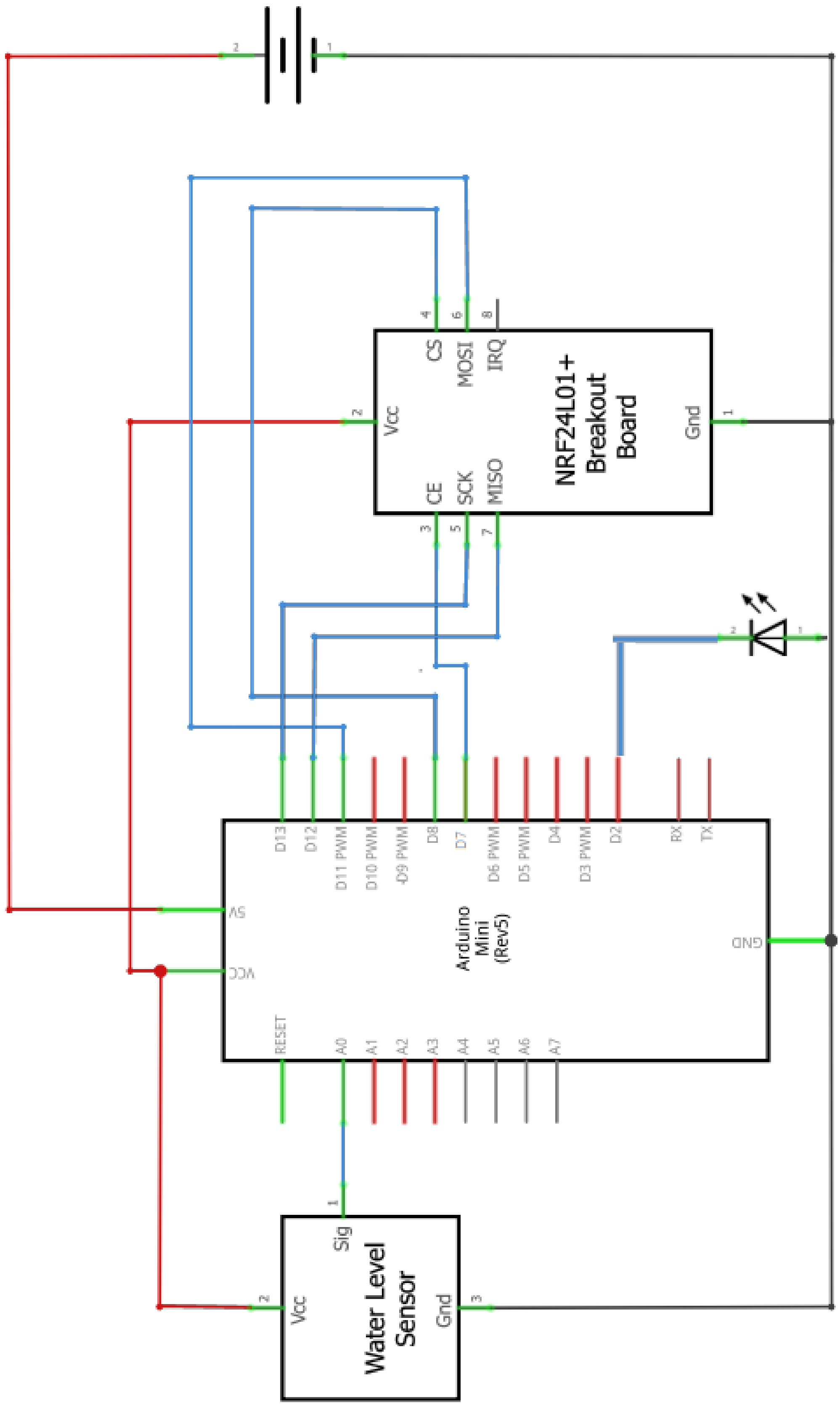
Initially the plan was to have a small pump provide water to the greenhouse that would allow the system to water the plants as required but the pumps never arrived in the post which led to the use of an LED in its place. When the LED is lit, the pump should be pumping water to the plants. The LED works based upon the readings of the soil moisture sensor.

The RF24 card is used to provide the wireless connection to the Mega which is the main "Data hub" of the project as all the data is collected there. A more detailed explanation of the RF24 modules are available in the Background Research above and the Software Design section below.

The DHT11 module reads in data using a thermistor to measure the air temperature and a capacitive humidity sensor to detect the humidity. It sends out a digital signal that requires careful timing to grab correctly. The only downfall is that the sensor has a rate of one reading every two seconds which in some situations might be too slow and inaccurate. In this circuit, the system updates every 10 seconds so the slow read time does not affect the readings or accuracy.

Once the data is sent from this circuit to the Arduino Mega, it is uploaded to the website and saved into the MariaDB in a readings table. The readings are displayed under the Temperature column and under the Soil Moisture column.

Water Trough



The water trough system uses the following components to receive and send readings; Arduino Mini, RF24 wireless module, water level sensor, an LED and a battery pack. The circuit is powered using two AA batteries in a battery pack which each provide 1.5v of power each which in total provides 3v output. Ideally the Arduino should be powered by a 5v power supply but they were not low cost and as easily accessible so these were used instead.

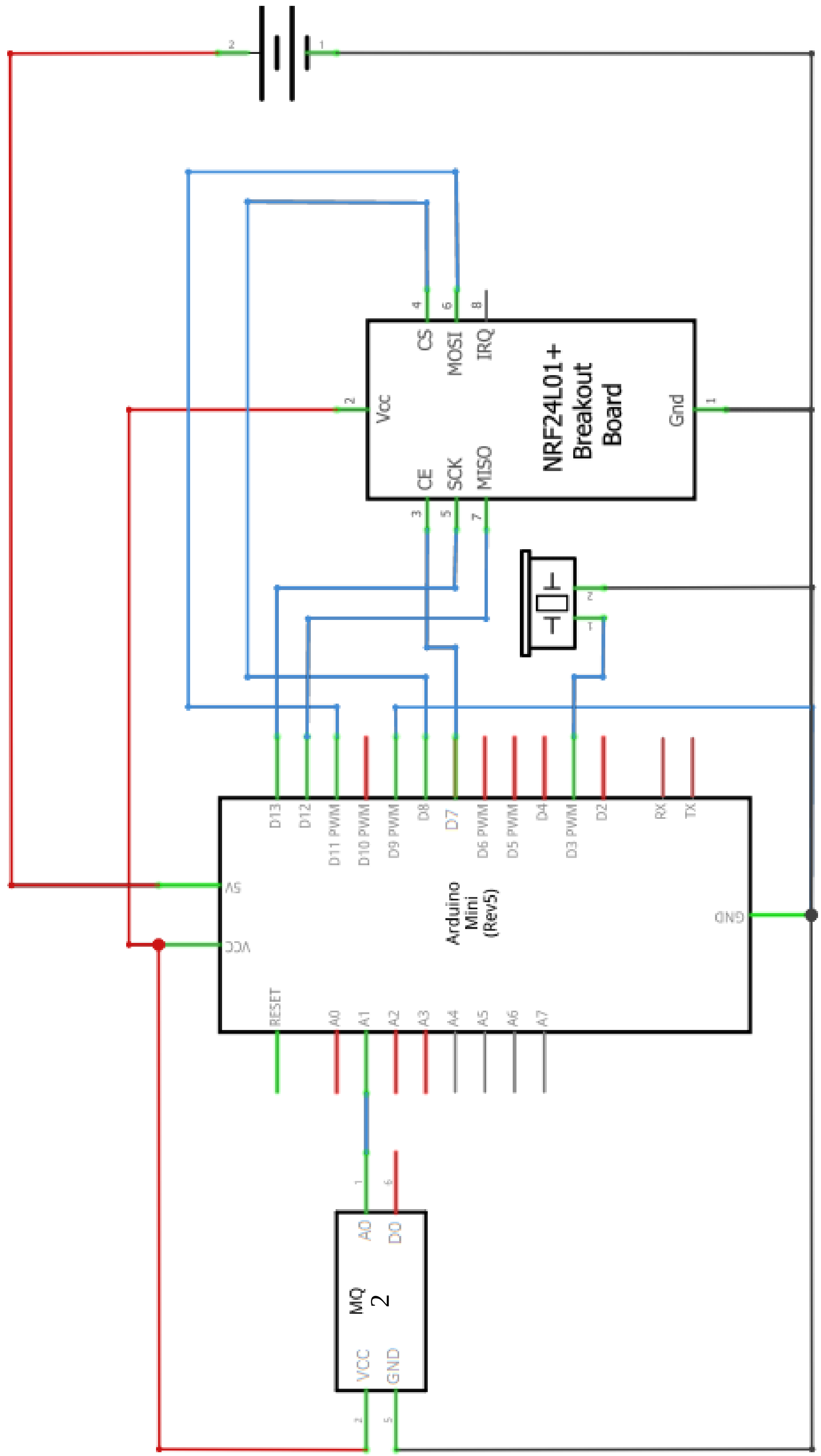
Initially the plan was to have a small pump provide water to the trough that would allow the system to refill the water in the trough as required but the pumps never arrived in the post which led to the use of an LED in its place. When the LED is lit, the pump should be pumping water to the trough. The LED works based upon the readings of the water level sensor.

The RF24 card is used to provide the wireless connection to the Mega which is the main “Data hub” of the project as all the data is collected there. A more detailed explanation of the RF24 modules are available in the Background Research above and the Software Design section below.

Once the data is sent from this circuit to the Arduino Mega, it is uploaded to the website and saved into the MariaDB in a readings table. The readings are displayed under the water trough column and would display full when the trough is full or filling when the pump is on and trough isn’t full.

In practical use the sensor would have to be surrounded by a protective shroud to stop the sensor from getting damaged or broken off by the cattle. The trough would also have to be cleaned constantly to stop dirt from building up on the sensors contacts.

Hayshed and Electric Fence



The Hayshed circuit incorporates the use of the following components; Arduino Mini, RF24 wireless module, MQ-2 gas sensor, a buzzer and a battery pack. The circuit is powered using two AA batteries in a battery pack which each provide 1.5v of power each which in total provides 3v output. Ideally the Arduino should be powered by a 5v power supply but they were not low cost and as easily accessible so these were used instead. Unfortunately, since the sensors aren't receiving enough power, they give false readings sometimes but this can be remedied by connecting a better power supply.

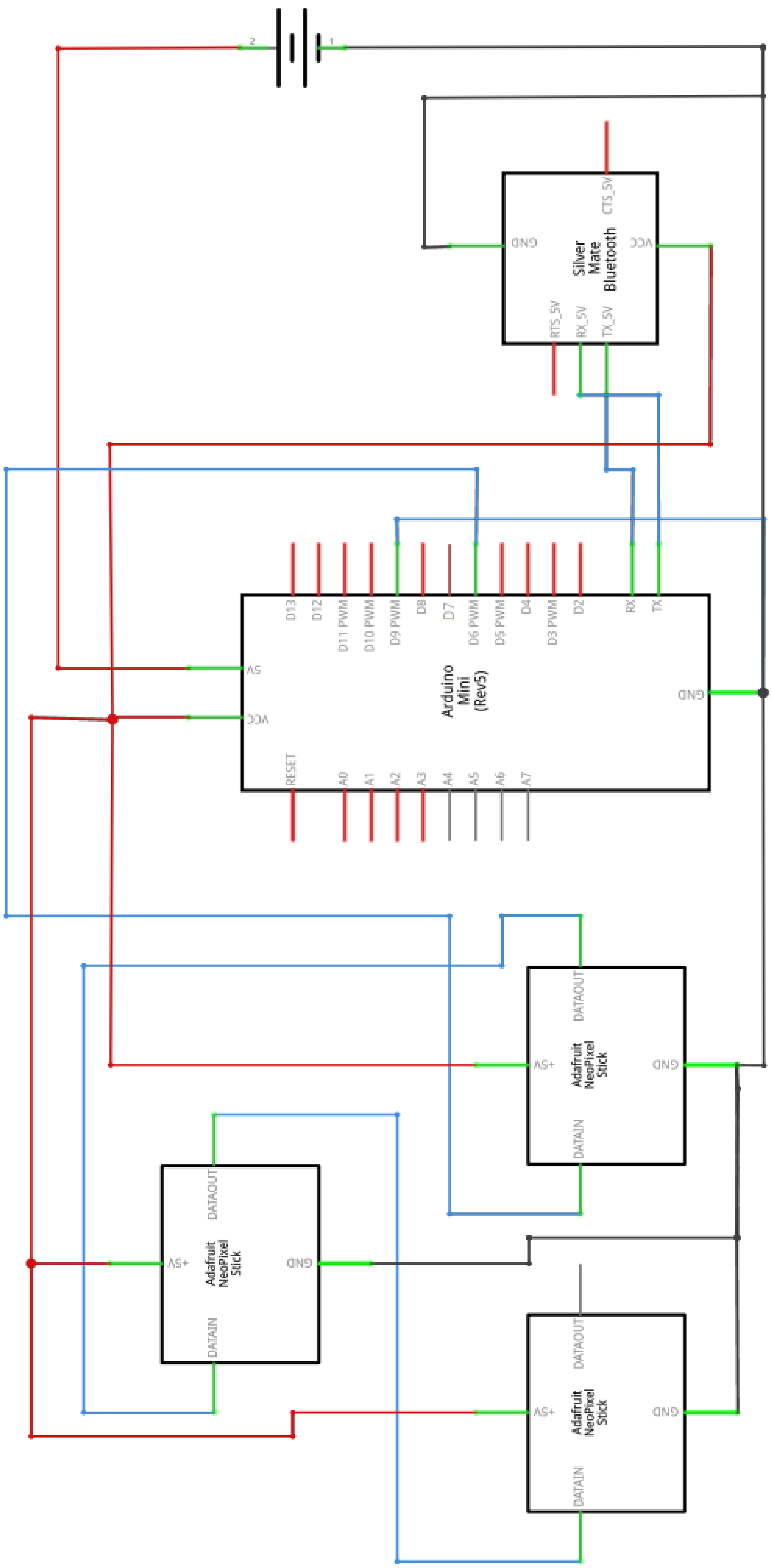
The gas sensor can read multiple different gas values. These are as follows; Hydrogen gas, LPG, CH_4(Methane), CO_2, Alcohol, Smoke, Propane and air. The sensor is an analogue output which means it must be connected to an analogue pin for to receive readings. The output voltage of the gas sensor increases when the concentration of the gas detected rises. Sensitivity of this sensor can be adjusted by changing the potentiometer onboard.

The electric fence part of the circuit is the wire going from pin 9 to ground. When this wire is broken, the system reads LOW from the pin and so sends the alert to the website.

The RF24 card is used to provide the wireless connection to the Mega which is the main "Data hub" of the project as all the data is collected there. A more detailed explanation of the RF24 modules are available in the Background Research above and the Software Design section below.

Once the data is sent from this circuit to the Arduino Mega, it is uploaded to the website and saved into the MariaDB in a readings table. The readings are displayed under the hayshed column and the electric fence column. If the sensor detects a gas concentration higher than a pre-coded value, FIRE is displayed in the readings table but if not, Clear is shown. If the electric fence is broken, it displays BROKEN and if not, Working is displayed.

Shed Lighting system

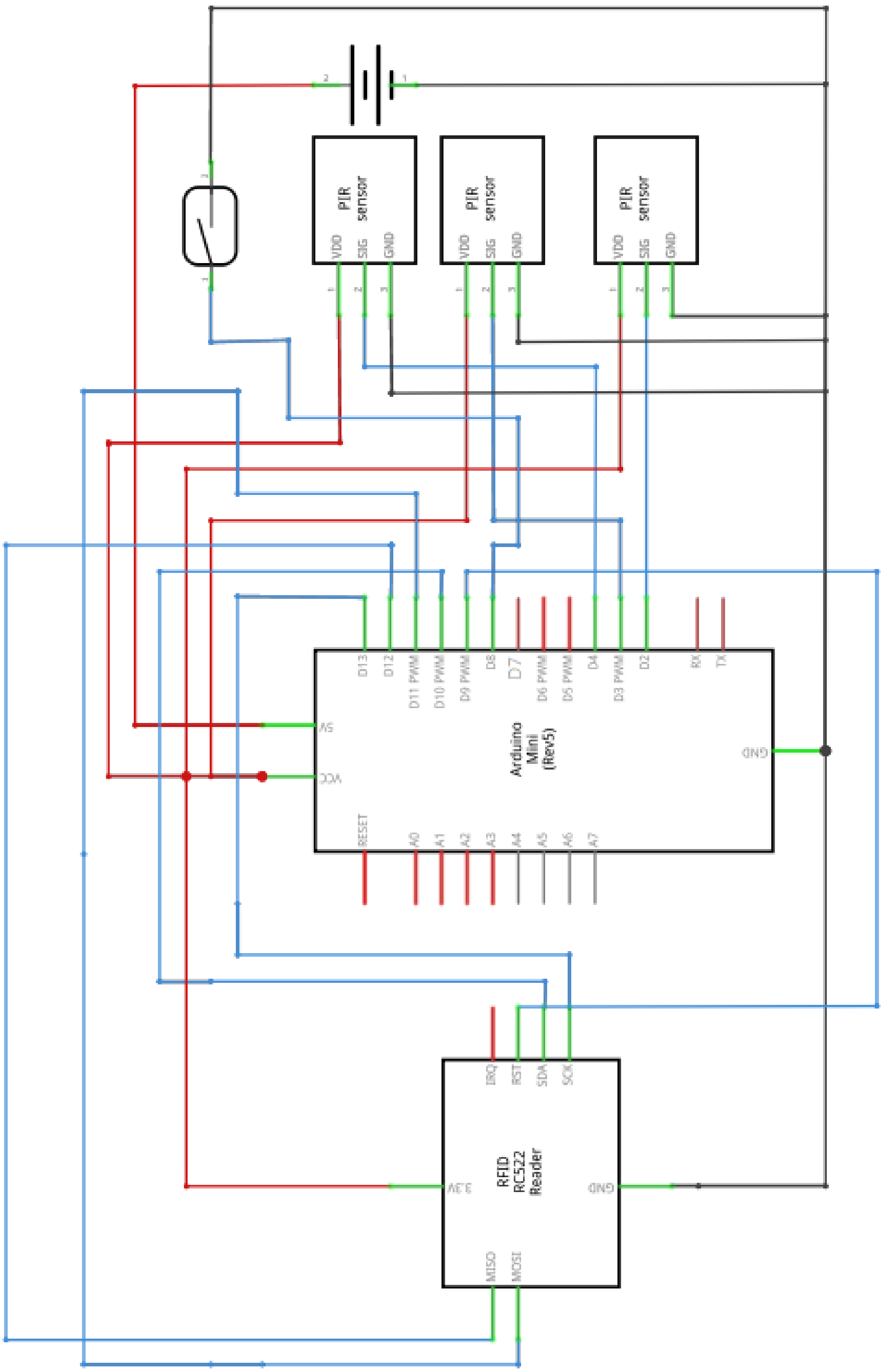


The shed lighting system incorporates the use of the following components; BTO6 Bluetooth module, Arduino Mini and three Neopixels. The circuit is powered using two AA batteries in a battery pack which each provide 1.5v of power each which in total provides 3v output. Ideally the Arduino should be powered by a 5v power supply but they were not low cost and as easily accessible so these were used instead.

The system works by having a phone use Bluetooth and pair with the BT06 module. Values are sent to the system from the project mobile app. These different values turn on and off different Neopixels in the circuit.

If this system was being implemented in an actual farm, Bluetooth connectivity is not a viable option as the range is limited. A better method of communication would be using WIFI. The app could send commands to the RF24 cards using a separate local network.

Tractor Alarm System



The tractor alarm system incorporates the use of the following components; RC522 RFID card reader, Arduino Mini, 3 PIR’s, a magnet switch and a buzzer. This systems purpose is for body detection during the day and to act as a burglar alarm at night. The system has 2 modes which are enabled two different ways.

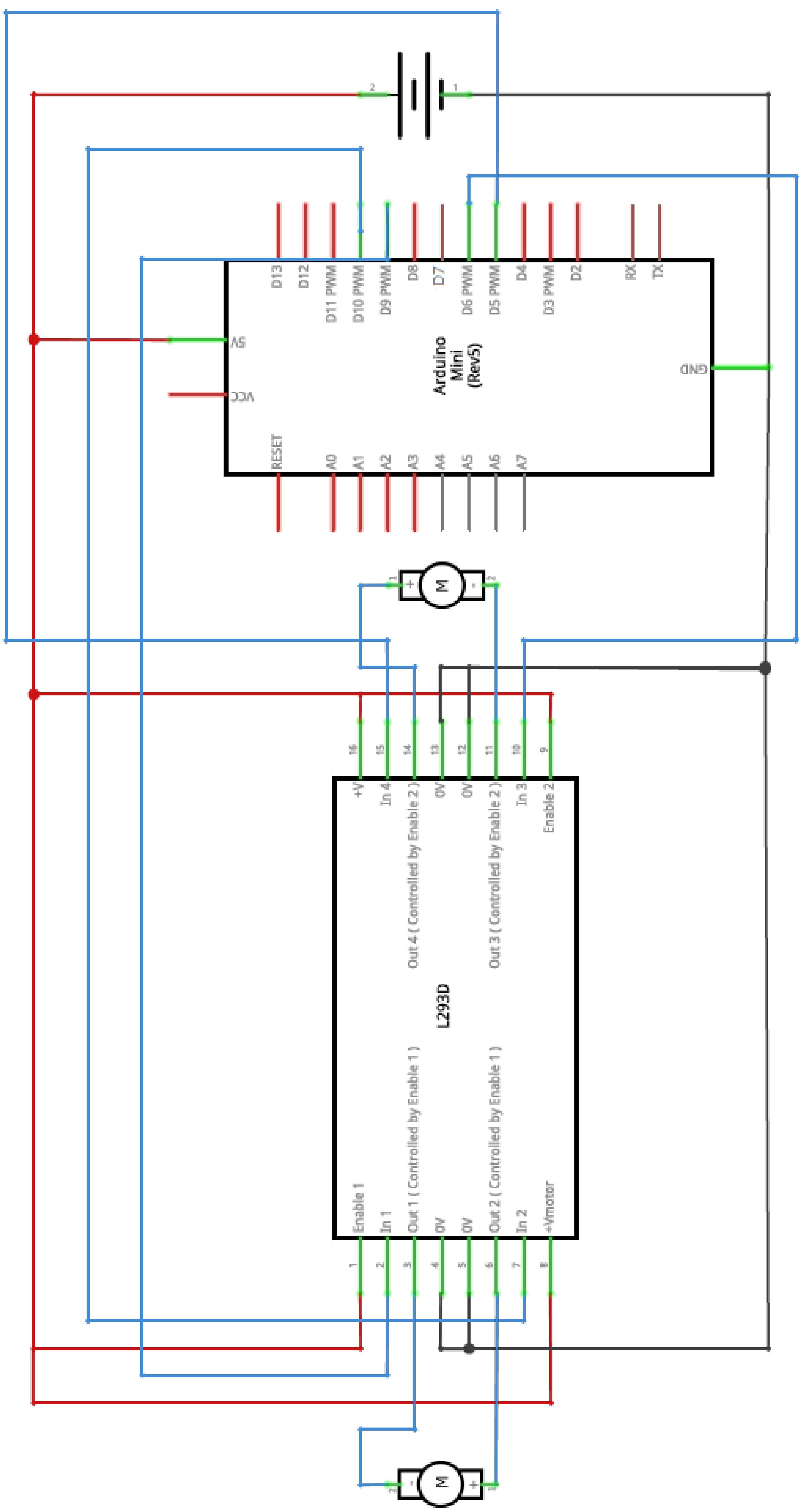
The first mode is the body detection. This is enabled when the tractor is started and the door is unlocked. When both criteria are met, the system will then monitor for people near the tractor while in use. If a person is detected, a buzzer starts beeping in the tractor to alert the driver.

The second mode is the burglar alarm. This is activated once the door is locked, and the RFID tag is scanned. When armed, the system then monitors for anyone near the tractor. If movement is detected, the alarm is triggered. This time a siren sounds outside the tractor cab.

The circuit is powered using two AA batteries in a battery pack which each provide 1.5v of power each which in total provides 3v output. Ideally the Arduino should be powered by a 5v power supply but they were not low cost and as easily accessible so these were used instead.

If this system was implemented in an actual farm, a solar panel would have to be attached to the roof of the tractor and the battery would have to be changed to a rechargeable one. This would allow the system to work better and be more reliable when the tractor is not near the farm.

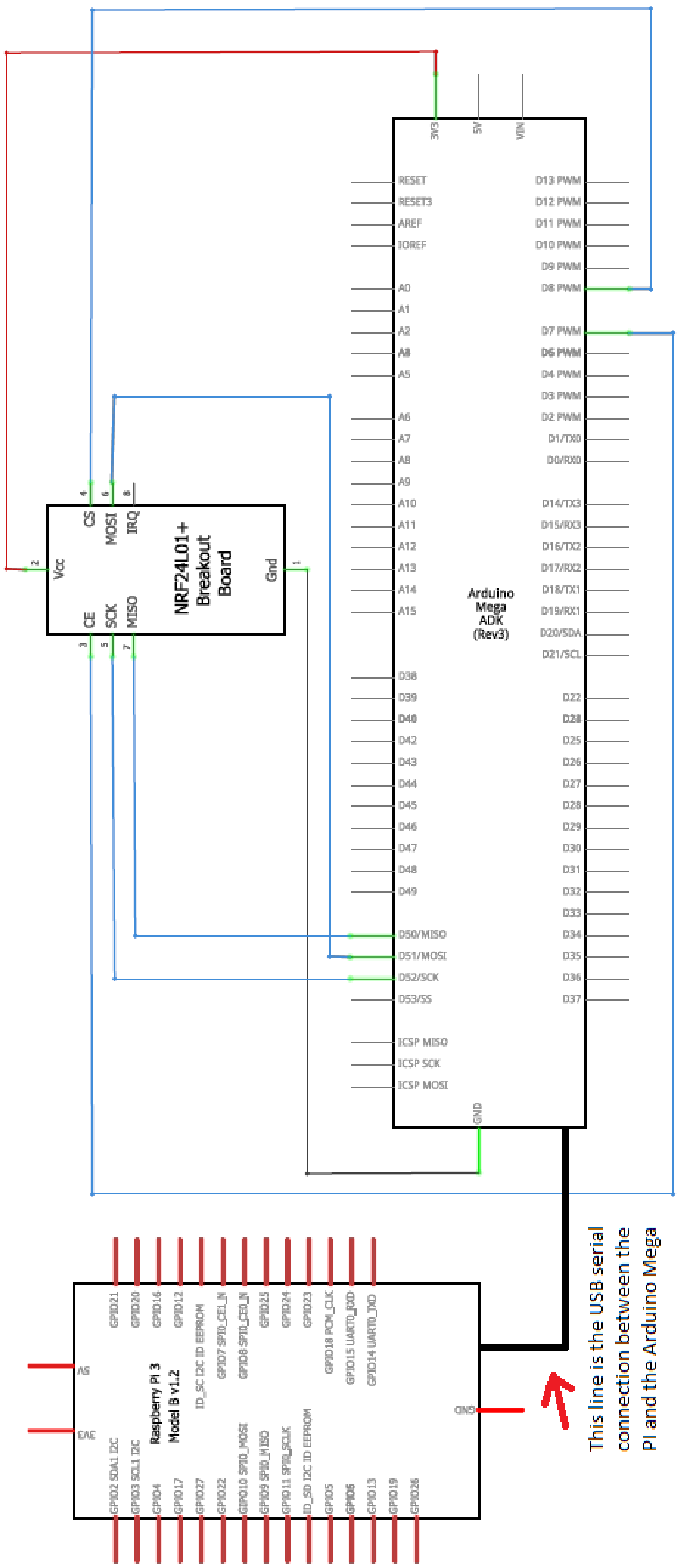
Shed Scraper



The shed scraper system incorporated the use of the following components; Arduino Mini, two DC motors and a L293D motor driver chip. Arduino Mini, RF24 wireless module, MQ-2 gas sensor, a buzzer and a battery pack. The circuit is powered using two AA batteries in a battery pack which each provide 1.5v of power each which in total provides 3v output. Ideally the Arduino should be powered by a 5v power supply but they were not low cost and as easily accessible so these were used instead. But if you wanted to use motors with more torque, you would need to get a better power supply.

The system works on a timer that has the shed scraper activate every 15 seconds where as in a real scenario, the scraper would activate every 4-6 hours.

Data Hub System



The circuit above is that of the “Data hub” for this automated farm. It incorporates the use of the NrF24L01 2.4GHz wireless module to send and receive data to the relevant systems mentioned previously. This system is connected to the Raspberry Pi 3 model B, which hosts the website and receives all the information on the serial port of the MEGA.

The reason that the Arduino Mega is using the Nrf24L01 module is because this was simpler to accomplish than directly connecting the wireless module to the Raspberry Pi. The NrF24L01 module in this system receives the information from each sub system sending data to the website, this is checked one by one. The Arduino Mega is then connected to the Raspberry PI using the USB serial cable. Furthermore, the sensor data is being received constantly by the Mega and is being printed onto the serial monitor. The Raspberry Pi reads the information being printed to the serial monitor of the Mega. A python program on the Raspberry Pi sorts the data being printed to the serial monitor then adds the information to a PHP table.

For the shed scraper system only the “Data hub” acts as a transmitter sending data to that sub system after a certain amount of time has passed.

Software Design

The Automated Model farm is created by connecting multiple subsystems in a network and make them talk and send data to and from each other. The microcontrollers that were used in this project were Arduino Mini’s, an Arduino Mega and the Raspberry Pi. These systems are programmed in different ways. These are explained below.

Arduino Mega and Mini’s

The Arduino microcontrollers use C and C++ programming language to be programmed. This can be done using a program called Arduino IDE (Integrated Development Environment). The Arduino IDE contains a text editor for writing code, a text console, a message area, a series of menus and a toolbar with buttons for common functions such as save, upload, etc... It connects to the Arduino microcontrollers to upload programs and communicate with them. Programs written here are called sketches. These sketches are written in the text editor and saved with the file extension .ino. The editor has features for searching/replacing and cutting/pasting text. The message area give feedback from the program while uploading or saving ad also displays errors in the process. The console displays all text output by the Arduino IDE including error messages and other information. The toolbar allows you to verify and upload code, create a new sketch, open a sketch, save a sketch and open the serial monitor.

Before uploading a sketch, you need to choose the correct port which the Arduino is plugged into. This can be done though the menu bar option Tools > Ports. You also must select the correct board and version you are using. This can be done though Tools > Board and Tools > Processor. Once you have selected the correct serial port, board and version, you upload the cod by pressing the upload button on the toolbar or by using the keyboard shortcut Ctrl + U. When the board starts uploading, you will see the TX and RX LED’s blink. When the upload is completed, the amount of memory the code has used will be displayed and the total memory available.

Raspberry Pi

The Raspberry Pi uses Raspbian as its operating system. Raspbian is a Debian based Linux operating system. The pi could use also use Ubuntu, Windows 10 IoT Core or other arm based operating systems. It can run the Arduino IDE which is used to program the Arduino's and it can be used to do what a fully-fledged desktop can do, but in a tiny form factor and at a relatively cheap price point. In this project, the raspberry pi is being used to receive information from Arduino boards, sort and upload it to a website which it is hosting. The website hosing server application being used is called Apache2 web server. It is a free and open-source cross-platform web server. The website then used MySQL as a database to store values received. The MySQL database is then maintained using a tool called phpMyAdmin. This is a free, open source administration tool for MySQL and Maria DB. It is written primarily in PHP.

The Arduino mega will be used to receive data from all the separate sub systems, sort it into one string and then send this string to the Raspberry Pi. When the Raspberry Pi receives the string, a python program splits the string into single values and adds the values to a table in the MySQL database. This table is then visible on the website at the following link: https://amfarm.eu.ngrok.io/test2.php

The Apache web server is a local web server and to make it live, port forwarding would have to be used. A problem occurred though when the college internet would not allow port forwarding. To bypass this problem, a program called Ngrok was used (https://ngrok.com/). Ngrok is a program that allows local website to be visible anywhere. It connects to their cloud service which accepts traffic on a public address and relays that traffic through the Ngrok process running on the Raspberry Pi and onto the specified address. The process will continue to work even if the Pi changes networks.

To create the website, a basic template was downloaded from https://onepagelove.com which allows entire websites to be created in one html file. Information on the project was then added to this template and the links to the readings table was added.

The Arduino Mini's were used to create all the small subsystems in the project. They were used because of their small form factor, low cost and low power consumption. Each of the subsystems had their own mini and were powered by a battery pack. They all had to be coded separately and tested. The subsystems coding is explained below.

RF24 communication

This connection takes place in each of the subsystems. After any current values are read from the sensors, they are sent to the Arduino Mega via the rf24 modules. These modules use the 2.4GHz wireless band and communicate using addresses and tunnels. To begin using the code you must import certain libraries. These libraries gave access to pre-coded methods which are used in conjunction with the rf24 cards. The libraries used are as follows:

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <printf.h>
```

SPI.h

This library is used to communicate with SPI devices while using the Arduino as the master device. SPI (Serial Peripheral Interface) is a synchronous serial data protocol used by microcontrollers for communicating quickly over short distances with one or more peripheral devices. It can be used to have two microcontrollers communicate which is what is happening in this system. One of these devices is classed as the master device and the other, the slave device. In this case the Arduino mini is the master and the rf24 card is the slave device. There are three main lines of connection that are needed to use this library:

1. **MISO (Master in Slave Out):** This is used by the slave device to send information to the master device.
2. **MOSI (Master Out Slave In):** This is used by the master device to send data to the slave device(s).
3. **SCK (Serial Clock):** The clock constantly pulses which synchronize data transmission generated by the master.

nRF24L01.h + RF24.h

These libraries are used to allow the Arduino to communicate effectively with the nrf24 wireless modules. They include the methods which are used to initialise the rf24 module, connect to a certain address and send data. The methods which can be called are as follows:

- **RF24 (uint8_t _cepin, uint8_t _cspin):** This is the main method used to create the rf4 object using a constructor.
- **Begin ():** This is used to enable the rf24 card.
- **StartListening ():** Start listening on the piped opened for reading
- **StopListening ():** Stop listening for incoming messages
- **Write (const void *buf, uint8_t len):** Write the data to the open writing pipe.
- **Available ():** Test whether there are byes available to read off the pipe
- **Read (void *buf, uint8_t len):** Read the available data from the reading pipe.
- **OpenWritingPipe (uint64_t address):** Open a certain pipe for writing
- **OpenReadingPipe (unit8_t number, uint64_t address):** Open a certain pipe for reading

There are many other methods which can be called such as methods to set data payload size, set data transmission rate and set power amplifier level.

Printf.h

This is a library incorporated with the rf24 modules. It allows you to view information about the module and its settings on boot. This can help to troubleshoot any problem that would occur.

After importing all the required libraries, you must create the rf24 object and assign certain pins. An array is then created that saves the addresses that will be accessed by the rf24 methods later in the code. A variable of fixed size then must be created to store the incoming data from the reading tunnel.

```
RF24 radio(7, 8); // CE, CSN
const byte addresses[][6] = {"hayrd", "haysd"};
char charBuf[50];
```

In the setup () method, you must start the rf24 object created earlier. Then the power level must be set as to make sure that connections are accessible at a longer distance.

The reading and writing pipe are then opened using the address array to allow sending and receiving data from other systems.

```
radio.begin();
radio.setPALevel(RF24_PA_MAX);

radio.openWritingPipe(addresses[1]); //00003
radio.openReadingPipe(1, addresses[0]); //00002
```

In the main loop, the program first starts listening and then waits until it gets sent data from the Arduino Mega. When it receives the data from the Mega, the system splits the received information into the required readings and saves them into variables. After the data is saved to a variable, the radio is told to stop listening and open a writing pipe to send data to the Mega. The code writes the data, which is first converted to a char array, to the open writing pipe. This process then loops back and repeats.

```
message.toCharArray(charBuf, 50);
radio.stopListening();
radio.openWritingPipe(addresses[1]); //00003
```

Hayshed

The hayshed subsystem has two subsystems built into one; the hayshed system and the electric fence system. This had to be done as the rf24 card that sent data from the electric fence subsystem blew and so the systems were incorporated into one. The system is reading two values, the gas sensor and the electric fence status.

The addresses used for the hay shed are shown in the screenshot below:

```
const byte addresses[][6] = {"hayrd", "haysd"};
```

The MQ-2 gas sensor has a digital and an analogue output but the analogue was used to give a more accurate reading. The digital output only returned a 1 when the gas level rose above a pre-set value. The analogue output constantly sends values of the current gas level which allows for an exact reading. Before using the gas sensor, it must be set as an input or it will not read values correctly. When the sensor detects a reading over a present variable, which in this case was 400, the message which is being sent to the Mega changes from “Clear” to “FIRE”. This information then gets uploaded to the website and sounds a buzzer attached to the subsystem. The buzzer will keep sounding until the gas sensor reading drops below 400.

Once the code checks the gas sensor, the electric fence is then checked. This is done by checking if the input pin reads HIGH or LOW. If low, the fence is broken and the variable being sent changes from working to broken. Once the fence is fixed, this value returns to working.

Water trough

The water trough subsystem is used to measure the amount of water in the cattle's water trough. This is done using a water level sensor and a pump. The code begins by importing the libraries to send data via rf24. The water level sensor or pump does not need a certain library to work. The addresses which are used for the water pump are shown below:

```
const byte addresses[][6] = {"wtred", "wtsed"};
```

The water sensor uses an analogue input pin as it is more accurate to read values from 0 to 1024 then 0 to 255. The sensor must first be set as an input to be used with the code. Unfortunately, the water pump did not arrive before the project deadline so it could not be used but in its place an LED was added to show that when it was lit, the pump would be turned on. Once the water level sensor reads a value below a certain level, the message "Filling" is sent to the website and when the water rises above this value, it sends "Full".

Greenhouse

The greenhouse subsystem is used to monitor the temperature of the greenhouse and the soil moisture level. This is done using a DHT11 sensor and a soil moisture sensor. The code begins by importing the libraries for the rf24 and then another library shown below.

```
#include <dht.h>
```

This library is used read values correctly from the sensor. Once the libraries are imported, the DHT object is then created and after which the sensors are both set as inputs.

```
dht DHT;
```

```
  pinMode(soilval, INPUT);
  pinMode(dht_apin, INPUT);
```

In the main loop, the soil moisture sensor is read first as digital inputs. This is because the board attached to the soil moisture sensor has a potentiometer which can be used to change the moisture content required. This will then send "1" if the soil moisture content has fallen below the set level. When this happens, the rf24 sensor sends "Dry" to the website. If the soil moisture level is above the set level, the website receives "Watered".

After the soil moisture sensor is read, the DHT sensor is then read. To do this, the sensor must first be checked to see if it's working. It does this by checking if the readings are accurate using a read test. This test result is saved into a variable called test, as shown below.

```
 int test = DHT.read11(dht_apin);
```

If this test could return three possible values/errors:

1. 0 (DHTLIB_OK): This means that the test was ok and the sensor is working accurately.
2. -1 (DHTLIB_ERROR_CHECKSUM): This means the check sum has failed and that data received may be incorrect. In this case the temperature value is left unchanged as it is impossible to determine which byte failed in the checksum.
3. -2 (DHT_ERROR_TIMEOUT): A timeout has occurred and communication has failed. In this case the temperature will receive the value DHTLIB_INVALID_VALUE.

After the test has been ran and come back positive, the temperature is then measured using `String temps = String(DHT.temperature);` Both this value and the soil moisture level are then sent to the Mega.

Tractor Alarm and body detection

This subsystem is made up of three PIR's, magnetic switches and a RFID card reader. It is to be used for two purposes:

1. During the day, when the tractor is in use, the PIR's are used as a body detection system. This is implemented as so to help reduce accidents on farms caused by not paying attention to other people while operating the tractor.
2. During the night, when the tractor is turned off and the cab is locked, the PIR's are used for alarm motion detection. If a person goes near the tractor at night, the PIR's will detect them and alert the farmer.

The code begins by importing the required libraries.

```
#include <SPI.h>
#include <MFRC522.h>
```

The rf24 libraries are not required here as this system is run separately to the wireless network. This is so if the tractor is being used away from the farm, the alarm system is still usable.

The MFRC522.h library is used in conjunction with the RFID card reader. This library allows you to read and write to and from different types of RFID cards or fobs using a RC522 based reader connected via an SPI interface. This library provides useful methods which are used to work with the reader.

To begin using the reader you must define what pins it is connected to and create the reader object.

```
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.
```

Once this is done, you must define which pins the PIR's and magnet switch are connected to.

```
int PIRPin = 2;
int PIRPin2 = 3;
int PIRPin3 = 4;

const int sensor = 7;
int state;
const int buzzer = 8;
```

A buzzer is also connected to sound when the alarm is triggered. In the setup loop, you first initiate the SPI bus and then the reader object.

```
SPI.begin();       // Initiate  SPI bus
mfrc522.PCD_Init();   // Initiate MFRC522
```

The PIR's must be set as inputs and the magnet switch must be set as an input with and internal pull-up resistor attached. Once the setup loop is finished, the main loop is created. In this main loop, there are two types of loop.

1. **Cab Unlocked and Tractor started:** When this loop is running, the PIR sensors are reading constantly looking for any movement and if there is movement detected, a high-pitched beeping will commence. To begin, the PIR's are read using a digital pin and the values are saved into a Boolean variable which converts the input to TRUE or FALSE.

```
val = digitalRead(PIRPin);  // read input value
val2 = digitalRead(PIRPin2);  // read input value
val3 = digitalRead(PIRPin3);  // read input value
```

If any of the sensors read HIGH, the alert beeping will commence.

When the magnet switch is closed, this indicates the cab being locked. When this happens the second loop begins:

2. **Cab Locked and Tractor off:** When this loop is running it first checks if an RFID card is detected. This is done by running the unArmed() loop. The code first used a method from the mfrc522 library to check for a RFID card.

```
if ( ! mfrc522.PICC_IsNewCardPresent())
{
  return;
}
```

If a tag is not detected, the loop breaks but if a card is detected, the loop continues to read the card/tag. It does this by saving the card id to a string called content first, then checking if it matches the required card/tag.

```
String content = "";
byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
  Serial.print(mfrc522.uid.uidByte[i], HEX);
  content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
  content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();
Serial.print("Message : ");
content.toUpperCase();
if (content.substring(1) == "80 2F 8F 4D") |
{
  Serial.println("Authorized access");
  Serial.println();
  delay(3000);
  Armed = true;

}

else
{
  Serial.println("Error");
  delay(3000);
}
```

If the card that was read matches the required ID, the armed variable is set to true which runs the Alarm loop.

```
 if (!Armed) {
   UnArmed();
 }
 else if (Armed) {
   noTone(buzzer);
   Alarm();
 }
```

The alarm loop is the same as the body detection loop but when it sounds it makes a loud alarm noise instead of just a beeping sound. It first checks if any of the sensors are reading HIGH and if not continues to loop until an RFID card is scanned again. Once an RFID card/tag is detected again, the disarm() loop runs. This loop works the same as the unarmed loop earlier except this time the armed value is being set to false if the card/tag scanned is correct.

```
if (content.substring(1) == "80 2F 8F 4D")
{
  Serial.println("Authorized access");
  Serial.println();
  delay(3000);
  Armed = false;

}
```

Shed Scrapper

This subsystem is made up of a L293D motor driver chip and two dc motors. These work together to drive the shed scraper. This system requires no libraries to run as it’s not reading values or communicating to other devices. The code begins with defining which pins the motors are connected to, as shown below.

```
const int aright  = 9;
const int aleft  = 10;
const int bleft  = 4;
const int bright  = 5;
```

After setting the pins, a timer must be created to run the system alongside. The timer will count in milliseconds and the scraper system will activate every 15 seconds. This just for demonstration purposes, in a real-life scenario, the time would be every 4-6 hours.

Once the pins to the motors are set to output in the setup void, the system is ready to go. Every 15 seconds, the if loop will be true and run the shed scraping routine.

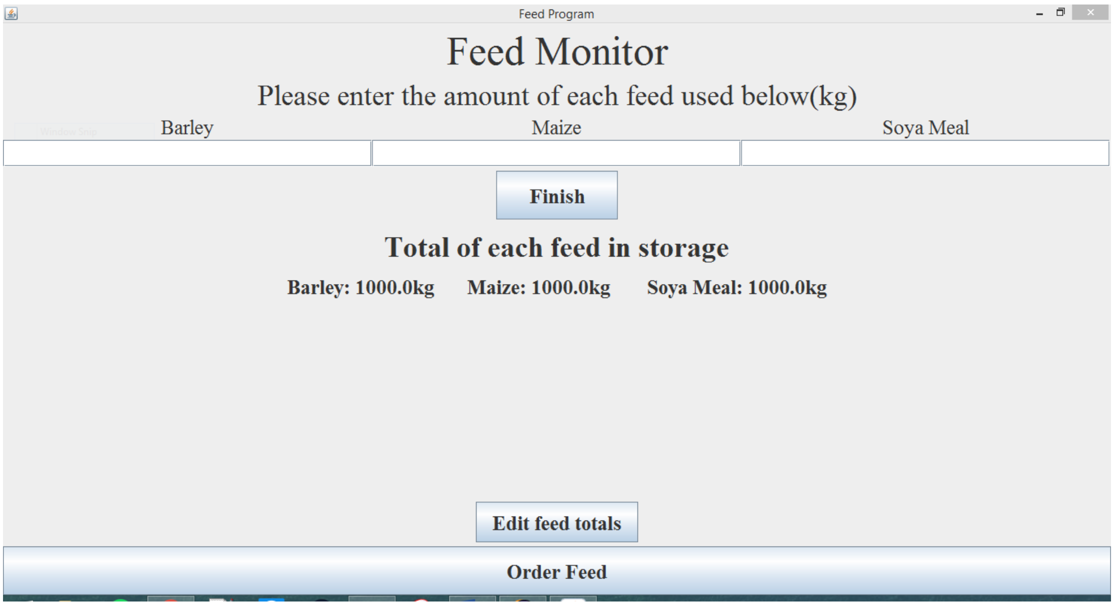
```
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
  previousMillis = currentMillis;
  digitalWrite(aleft, HIGH);
  digitalWrite(aright, LOW);
  digitalWrite(bleft, HIGH);
  digitalWrite(bright, LOW);
  delay(195);
  digitalWrite(aleft, LOW);
  digitalWrite(aright, LOW);
  digitalWrite(bleft, LOW);
  digitalWrite(bright, LOW);
  delay(2000);
  digitalWrite(aleft, LOW);
  digitalWrite(aright, HIGH);
  digitalWrite(bleft, LOW);
  digitalWrite(bright, HIGH);
  delay(195);
  digitalWrite(aleft, LOW);
  digitalWrite(aright, LOW);
  digitalWrite(bleft, LOW);
  digitalWrite(bright, LOW);
  delay(2000);
}
```

Java Meal Program

The java meal program is a simple program which was designed to monitor the meal usage. The program consists of three separate java classes; Email, MainGUI and RunFarm.

Email.java is a class that is used to send an email to the local co-op when the meal supplies are low. It makes use of Google's mail API which is used for logging into a Gmail account and sending emails. To use this API, we had to include the java repository javax.mail.jar. This repository provides classes that model a mail system. The package defines classes that are common to all mail systems in use. It is based on internet standards such as MIME, SMTP, POP3 and IMAP.

MainGUI.java is a class that creates a GUI that handles all the user interface. This GUI is displayed below.



From this GUI, you can input the amount of each feed used daily and it will show what is left in storage below. Then the farmer can order feed by pressing the order feed button which will order 1 tonne of each feed type from the co-op. If the farmer didn't use all the meal that they entered, they can use the edit feed totals to change the value by whatever they wanted. This is also used so that when more feed is delivered, the farmer can add the new amount to the old stock easily.

RunFarm.java is the runner which creates the user GUI from the MainGui class.

Android Application:

This app is used in conjunction with the HC-06 Bluetooth module to control the NeoPixel LED's in the Bluetooth light system. The App was created using the software called Android Studio 3.0.1. The app also includes a method of viewing current weather status based on your location using an API key from https://openweathermap.org/ . Also, a method for viewing the website built for this project is also included in the Apps simple design.

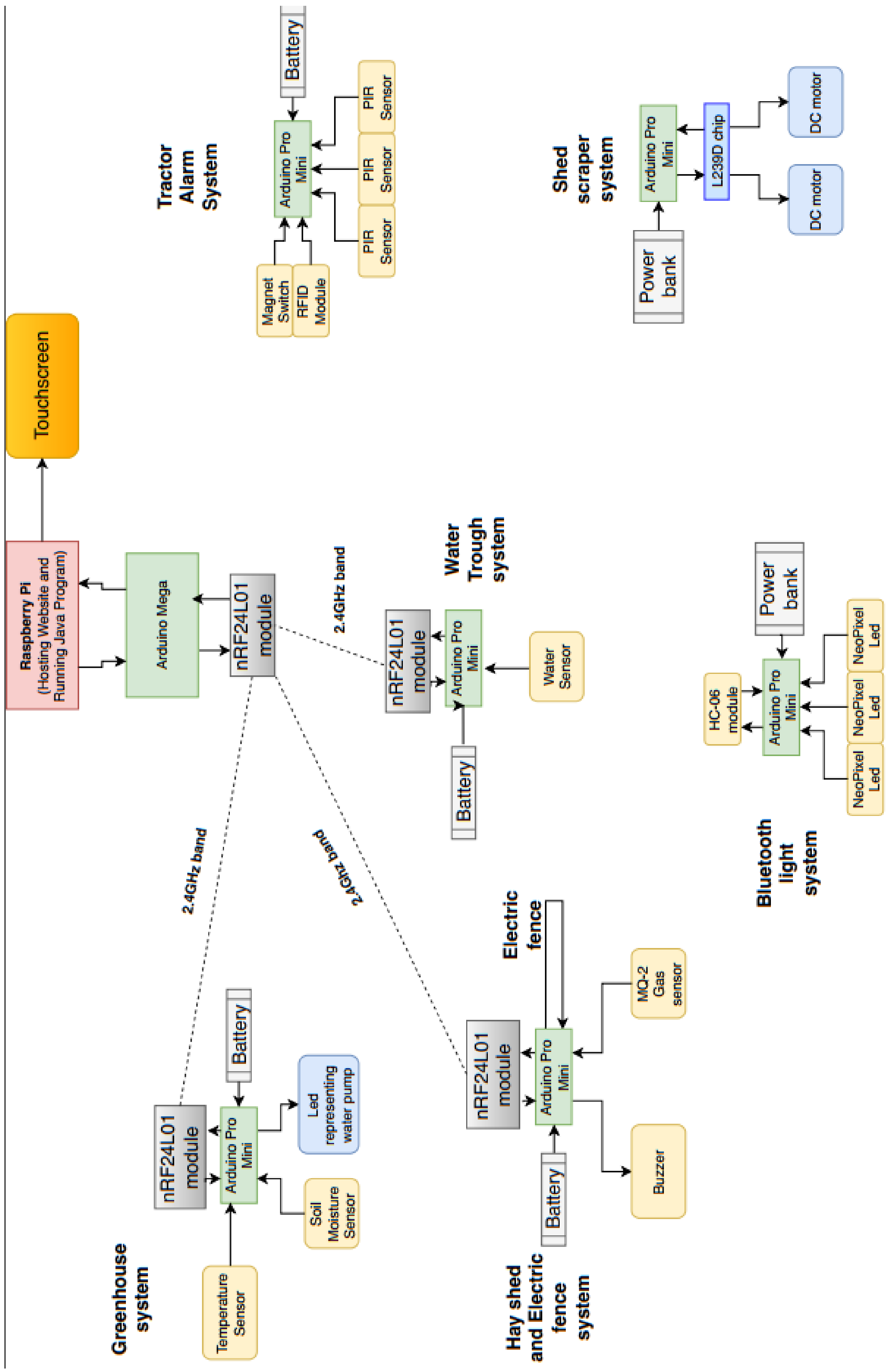
To initiate a connection between the Android App and the HC-06 module, the device which the app is on must pair with the HC-06 module. The Bluetooth pairing code in the "Device List" screen of the app creates the connection by storing the address of the HC-06 module as a variable. To complete pairing the devices, on the Android devices side a user will have to type in the password "0000" in the pairing password box. Once a device is paired with the HC-06 module when it is chosen from the device list it will automatically run the LED control screen when connected.

How this app works is via the sending of data over serial communication. Once a button is pressed on the Android App a piece of data is sent to the HC-06 module. This data is sent through the TX pin of the HC-06 module which is connected to the RX pin of the Arduino. The code on the Arduinos side of things checks the data that is incoming on the serial monitor, if the code matches the certain piece of data received it will control one of the lights states.

The core Bluetooth code was provided by using the link here:

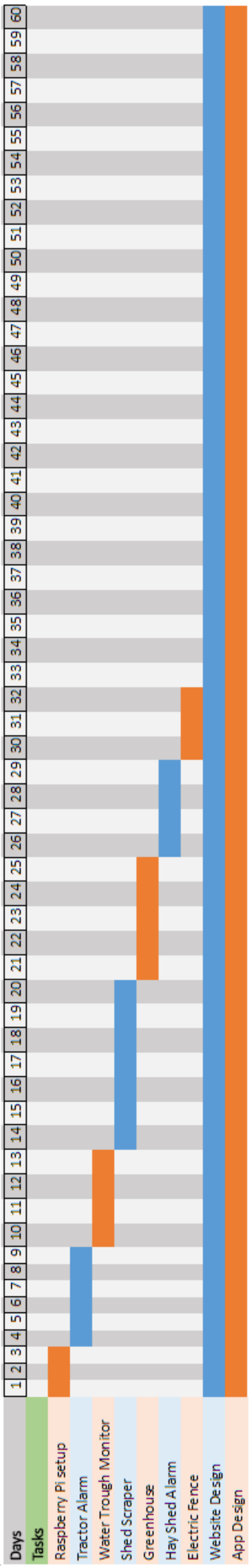
https://maker.pro/education/bluetooth-basics-how-to-control-an-led-using-a-smartphone-and-arduino-2

Project Flowchart

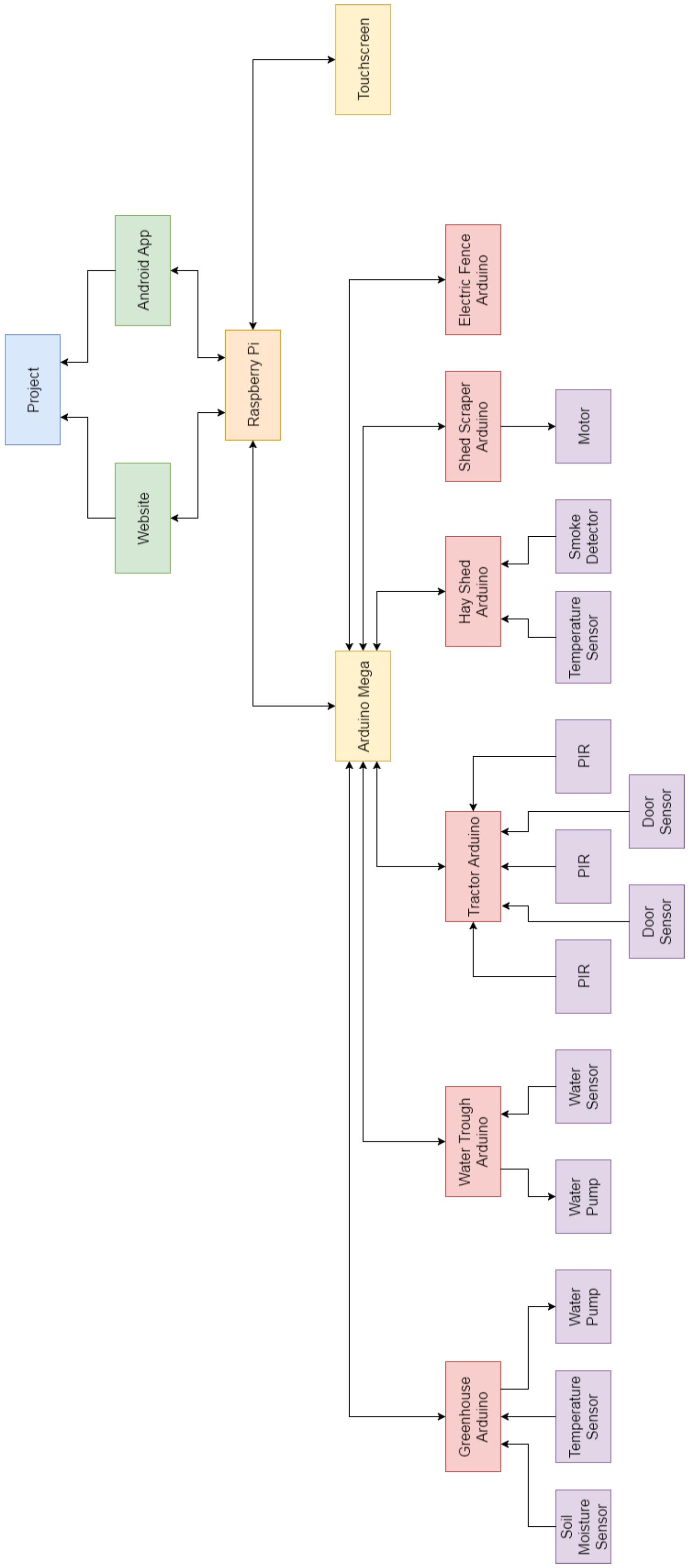


Gantt Chart

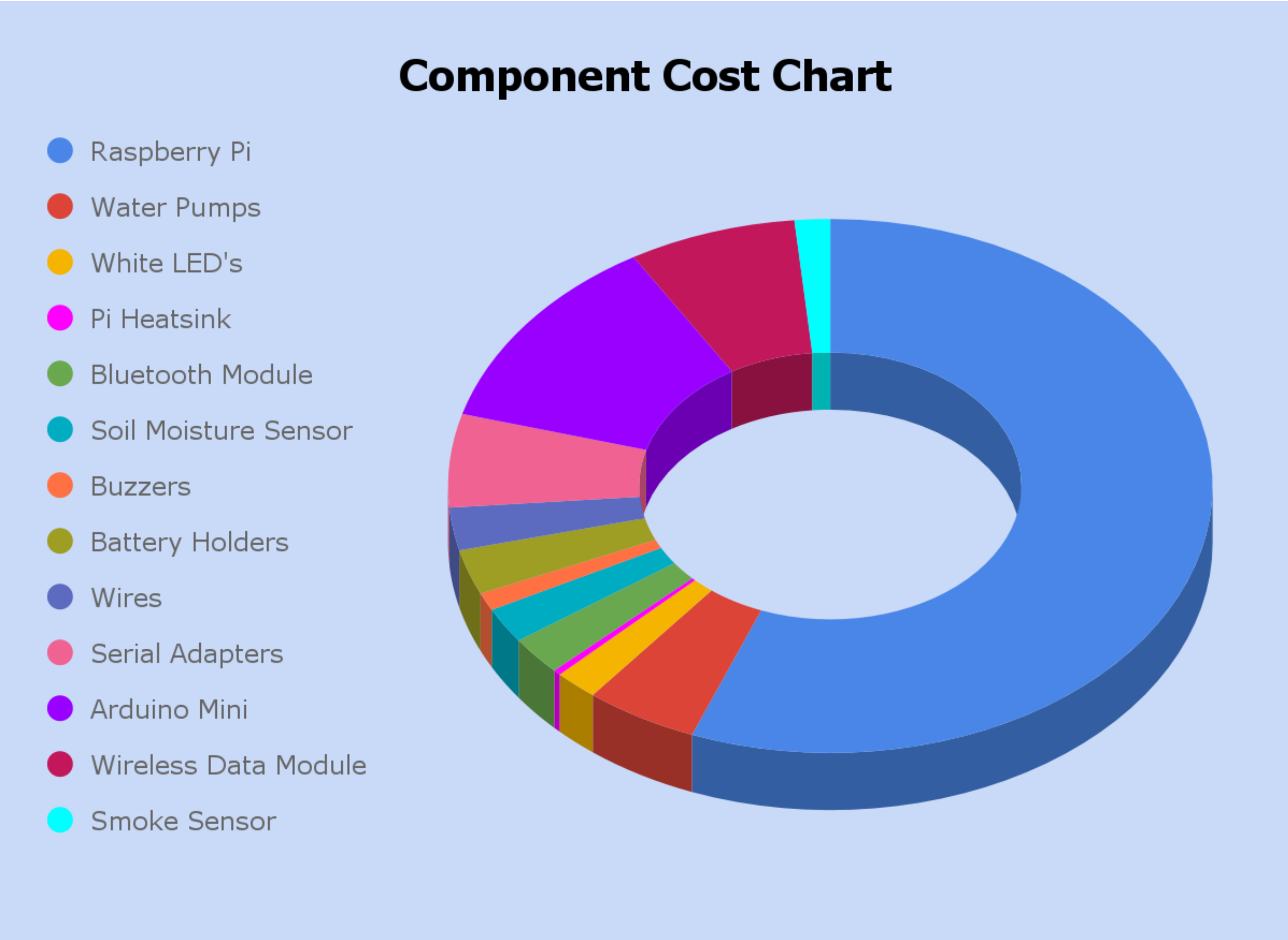
Task	Task Start Date	Task Duration
Raspberry Pi setup	1/15/2018	3
Tractor Alarm	1/18/2018	6
Water Trough Sensor	1/24/2018	4
Shed Scrapper	1/28/2018	7
Greenhouse	2/4/2018	5
Hay Shed Sensors	2/9/2018	4
Electric Fence	2/13/2018	3
Website Design	1/15/2018	60
App Design	1/15/2018	60



Work Breakdown Structure



Cost of Components



This pie chart represents the total expenditure for the automated model farm so far. The total cost of the project sums to €98.83 As can be seen above the Raspberry Pi makes up more than the total of the sum of the other components used in this project.

Implementation

In this section, the work undertaken by this group member (Travis Hyland) will be discussed. The project that was built was split up into 7 subsystems originally. Over time that count dropped to 6. This was because of certain parts bowing and not being able to source spares. The system build was split into many different parts. These parts are listed below:

1. Understand each component
2. Solder header pins
3. Water Trough System
4. Electric Fence
5. Hay shed Alarm
6. Greenhouse
7. Tractor Alarm
8. Set-up raspberry pi
9. Create server on pi for website
10. Create website
11. Shed Lighting System
12. Build App
13. Shed Scraper
14. Mega Data hub
15. Build Model
16. Create Feed Monitor Program

I will now explain each of these headings in detail below.

1. Understand each component

Once all the components had been received for the project, each component individually. This process took about a week to understand how each component worked. To connect to the Arduino minis, a FT232RL USB to serial adapter module had to be used. The rf24 wireless modules were complicated to understand at first. The modules worked using tunnels and address and each had to be assigned a separate address for reading and writing. These modules were also had to use 3.3v to be powered. If 5v was used, they are unreliable to connect accurately. To change the output voltage on the Minis from 5v to 3.3v, you connect the vcc output to the raw input. This then changes the output vcc voltage to 3v which is that is required for the rf24 cards.

2. Solder header pins

After running some test programs with the Arduino minis, it was decided to solder on a header pins onto the Arduino minis to allow a better connection between the mini and components. These pins were soldered along both sides and then the end for programming was soldered with pins on top to allow the boards to sit into a breadboard correctly. There were then right-angled pins soldered into the RFID module to allow a more secure connection for when in use in the project.

#######PIC OF MINI#######

3. Water Trough system

The water tough system was then built. This system incorporated the use of a water sensor, Arduino mini, rf24 card and LED. The original plan was to use a small pump to refill the tank when the water level had dropped. Unfortunately, these components never arrived in the post so they could not be included in the project. In the pumps place, an LED was added to represent the pump. When the LED is on, the trough is detected not to be full and the pump should be active. When the trough is full again, the LED will turn off.

Originally to connect the water level sensor to the mini, a makeshift female to female header connection was used These connections were made using two header pins pushed together and taped. This connection was very unreliable as the pins were not soldered together, just touching. These used as the ordered male to female jumpers that were ordered never arrived. These headers led to may problems with the project. They blew 4 of the rf24 modules as the pins in the header were all touching. In the end, a pack of male to female jumpers were bought locally an extremely increased price but they were essential to the project’s completion at this point.

This system was then powered using a 2 AA battery pack. This was enough power as the water level module could run using 3-5v.

4. Electric Fence

The electric fence module was a module that was going to monitor the electric fence on the farm for any breaks in the circuit. This system just used two wires for the detection system; ground and D9 on the mini. These wires were connected and programmed to use a pullup internal resistor. When the wire was broken, this read as HIGH which meant that the system would send BROKEN to the serial. This system was the powered using a 2 AA battery pack as it would not require much power to read if the wire was broken and the rf24 ran on 3v.

5. Hay shed alarm

The hay shed system is used to monitor the hay shed and detect any smoke coming from the bales. This will help alert a farmer if the hay shed is smoking or is on fire. This could happen from trespassers purposely starting a fire or from bales drying to quickly in the shed after baling. This system uses a MQ-2 gas sensor module to detect the smoke. The sensor is placed above the bales and the system will be run using a USB power bank. This is because the gas module requires 5v to detect smoke accurately. When building the system, the analogue pin was used at first to output readings to the mini but it was discovered that the analogue connection was faulty as the values were not constant at all. In a 10 second period, it would constantly fluctuate detecting values ranging from 100 to 800. To fix this problem, the digital output was used to receive values. Now the sensors detection level can be adjusted using the on-board potentiometer. These readings were very reliable and allowed the system to work much more efficiently.

While testing the hay shed system, the rf24 module blew as a bad makeshift header pin adapter was used for connection the rf24 to the mini. Because of this setback, the electric fence system had to be incorporated into the hay shed system as to use the one rf24 module to transmit data about both detections. The rf24 now transmits two values instead of one.

6. Greenhouse

The greenhouse system is used to monitor the temperature and soil moisture content within the greenhouse. It uses a DHT11 sensor to monitor the temperature and a soil moisture detection sensor to measure the soil moisture content. The Arduino mini is used as the main controller and an rf24 module is used to send and receive data from other systems. The system is powered using a 2 AA battery holder which provides 3v of power to the system. Unfortunately, this is not enough to power the DHT11 module and the led to the system printing false values occasionally. For every 5 correct values, 1 incorrect value would print. Ideally a power bank should have been used to power the system but there was only 1 available to be used, which was needed for the hay shed alarm.

7. Tractor Alarm

The tractor alarm system has two modes, day mode and night mode. The day mode is used as a body detection system to help farm accidents. The night mode is used as a burglar alarm which must be enabled to be active. The system incorporates the use of 3 PIR sensors, a magnetic switch, an Arduino mini and an rf24 module. The magnetic switch would be attached to the tractors door lock. When the door is unlocked, the switch is open. This the enables the day mode. Until the tractor is turned off or the door the is locked, the system is monitoring for any movement around the tractor while it is in use. This is in hope to help prevent farm accidents relating to being hit by a tractor or run over. When the door is locked, the system is sitting in standby. The farmer must scan the RFID tag they have on their keys to enable the night mode. Once in night mode, the PIRs are still used to monitor for any movement but now they will sound a different alarm. When the day alarm sounds, a buzzer makes a beeping sound in the tractor but when the night alarm is triggered, a loud siren noise is made to signal that there is a trespasser. When the system was originally built, false values were constantly being read. This fault was then traced back to the same makeshift headers used elsewhere in the project. Once this error was fixed, the system worked perfectly.

8. Set-up raspberry pi

Once a few of the basic systems were built, it was decided to create the server which the website will be stored on. This was going to be done on a Raspberry pi. The Pi was ordered as part of a kit with Raspbian pre-installed on an SD card. While creating an account on the OS, the system crashed. This error continued occurring at the same point of set up. To fix this error, the SD card had to be formatted and a new version of Raspbian Installed. The OS was downloaded from https://www.raspberrypi.org/ and flashed onto the SD card using Etcher. Etcher is a software that is used to flash OS's to memory cards such as an SD card and a Micro SD card.

9. Create server on pi for website

Once Raspbian was installed correctly and an account was made, the software required for to host a server was installed. This software is called Apache Web Server. Once Apache was installed, MySQL had to be installed. MySQL is a relational database management system based on SQL. It is used for creating web databases and logging applications. The database was managed using PhpMyAdmin. To then have the Arduino communicate with the Pi, the python-serial module was installed. To have data sent from the Arduino to the database, a python program (datalogger.py) had to be written which takes the data coming in from the Arduino, sorts and then uploads it to the database.

10. Create Website

Once the database was ready data to be saved to it, a website had to be created to display the website. To create the website, a basic HTML website template was downloaded from https://onepagelove.com . The website is used to display information on the project and has links to view the Results table and to download the AmFarm apk for android. To make the results from the database visible on the website, a PHP file was created to take the data and display it correctly and in the correct order. This file is a mix of PHP, which is used to take the data from the MySQL database, and HTML, which is used to make the webpage the data is displayed upon.

11. Shed lighting system

Once most of the main subsystems were created, the standalone systems could then be built. The shed lighting system is one as it does not communicate with any other part of the project. This was created using a Bluetooth module, three Neopixels and an Arduino mini. The Neopixels are like normal LED's except they controlled using a single wire. Since they have digital driver chips built in, they must be programmed when to turn on. They won't just light when given power. They have three inbuilt LEDs (red, green and blue) which all must be assigned a value to light correctly. The system connects to an android phone which must be running the AmFarm app which sends different signals to the system on what LEDs to turn on and off.

12. Build App

This part of the project was completed using by the other member in this group, Sean Grenham.

13. Shed Scraper

The Shed scraper system is the second standalone subsystem in the project. It is used to clear the cattle shed of dung every 4-6 hours. This stops the build-up of dung which leads to having to taking a long time to clean out the sheds weekly. This system works with two motors that pull a bar across the shed to scrape the floor clean. This system need two motors as the motors we had had very little torque and more was needed to pull a bar. In a real scenario, a sensor would be added on to check for any animal standing in front of the bar to prevent any injuries.

14. Mega data hub

This subsystem is used as the main data hub in the project. This system is used to collect data from all the separate subsystems and send the data to the Pi. The Mega is plugged into the Pi using a serial cable which allows for the python code to real the serial input. The Mega talks to all the subsystems one by one to get data. This ensures that the data is not mixed or corrupt when it arrives at the Pi. The Mega opens and closes alternate addresses for the separate subsystems to talk over. All these subsystems must then open or close certain reading pipes at certain intervals to ensure correct data transmission.

15. Build Model

To build the model, a large piece of plywood with a hardwood veneer top and bottom was used to make the sheds. This was cut by the other project group member, Sean Grenham, but unfortunately the guide on the circular saw was not correct which lead to uneven cuts. These pieces were then given to the other group member, Travis Hyland, to build the model. A chisel was used to square up the mis-cut timber and make the sheds square and level. One the timber was squared, the sheds were built and glued together and placed on the base board.

Holes were then cut in the sheds to allow wires to be passed through and for sensors to be mounted on the inside of the separate sheds. The roofs of the sheds were not glued as to allow easy access to the interior of the shed.

16. Create Feed Monitor Program

When the project was built, the final piece of the puzzle was to include a feed monitoring program to allow the farmer to monitor the feed usage and to order more feed if needed from the local co-op using an email api provided by google which allows the program to send emails from a premade Gmail account.

Summary

To summarise the implementation section of the project, the workload wasn’t split exactly even as the project was more software technical. Also, because the project was only a single semester build, this put a great deal of pressure on getting all the subsystems and programs working correctly as time was of the essence. There were multiple setbacks but these were all overcome through further research and error checking.

Finished Project Pictures

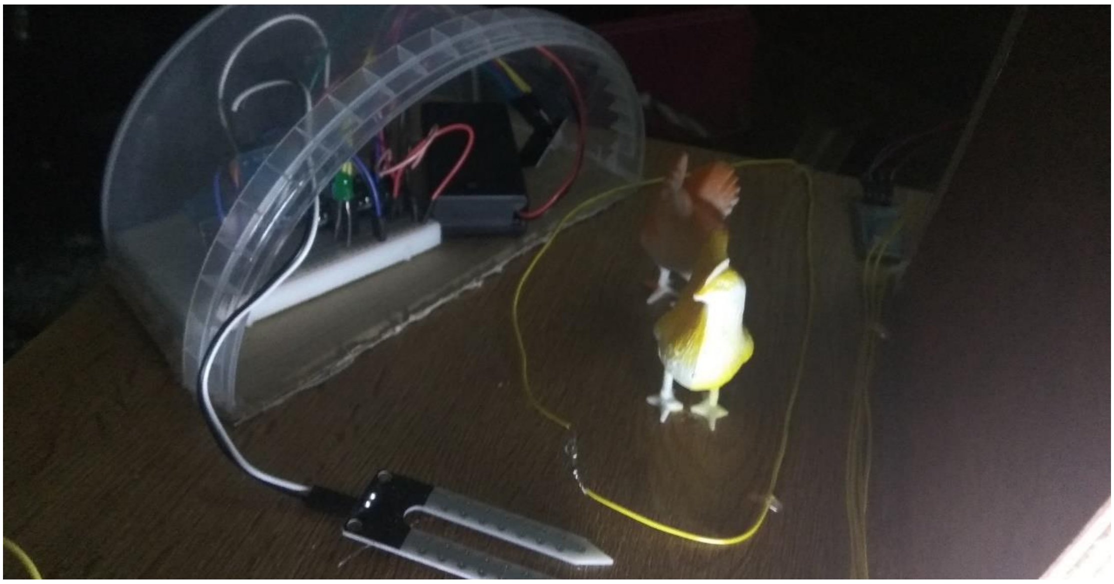


Figure 5: Completed Greenhouse

Figure 6: Completed Shed Scraper

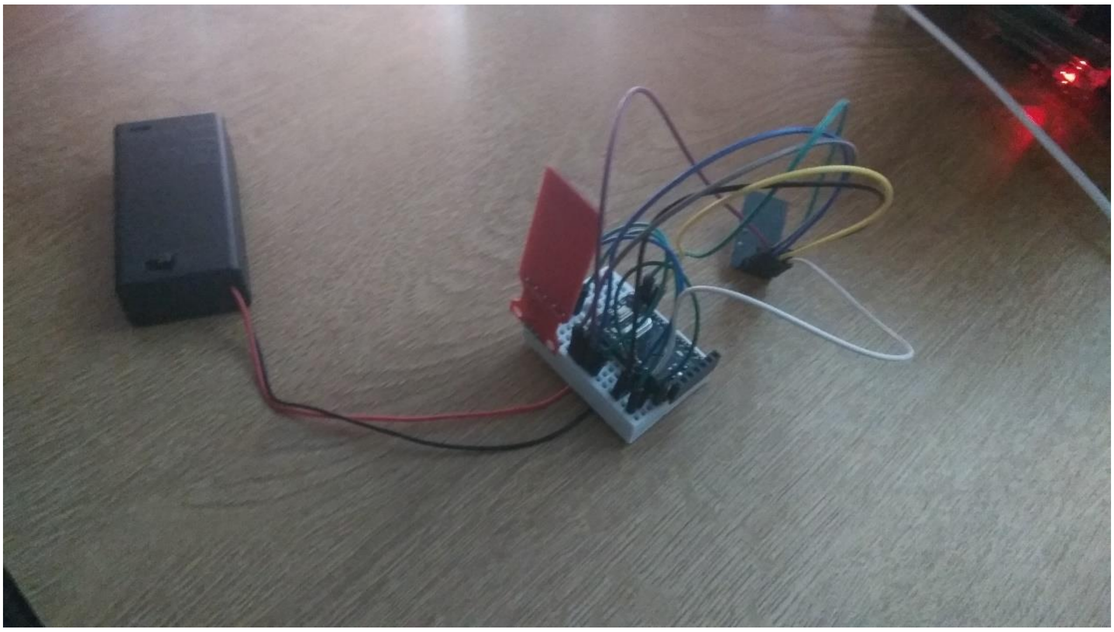


Figure 8: Finished Water Trough System

Figure 7: Completed Hayshed and Feed Shed

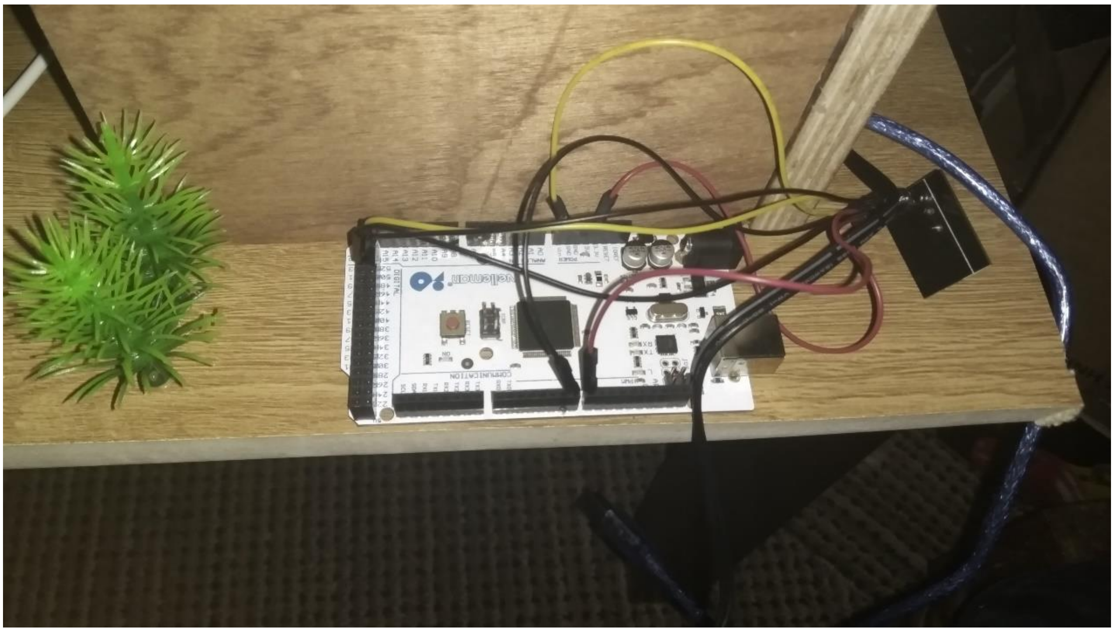


Figure 10: Finished Data Hub

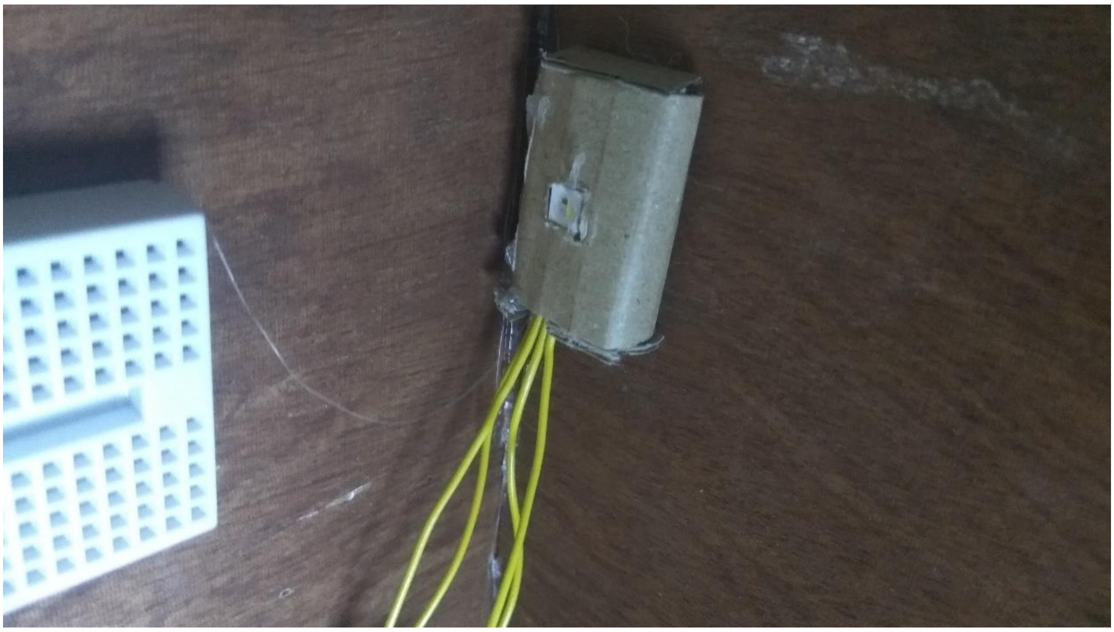


Figure 9:One of the finished farm lights

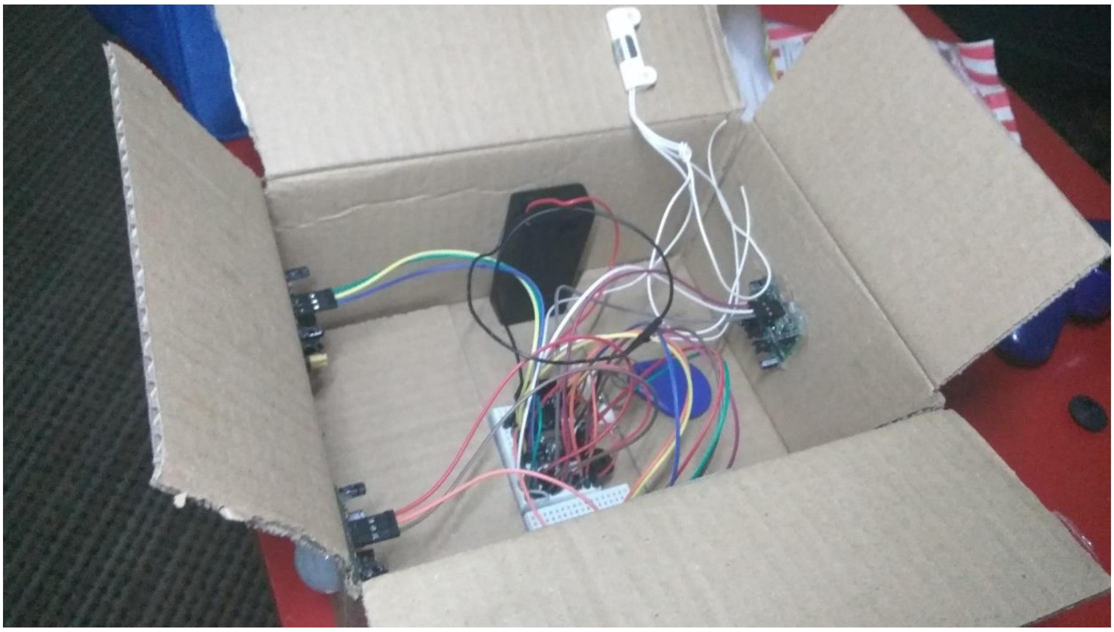


Figure 12: Finished tractor alarm

Figure 11: Finished Model

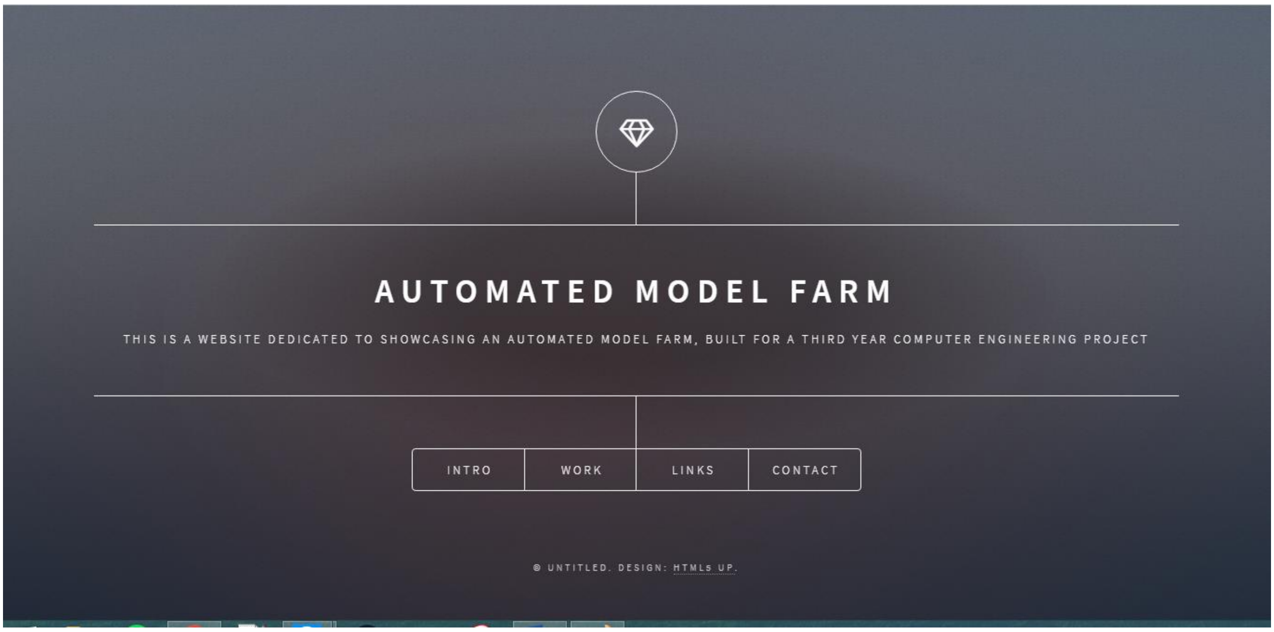


Figure 14: Website homepage

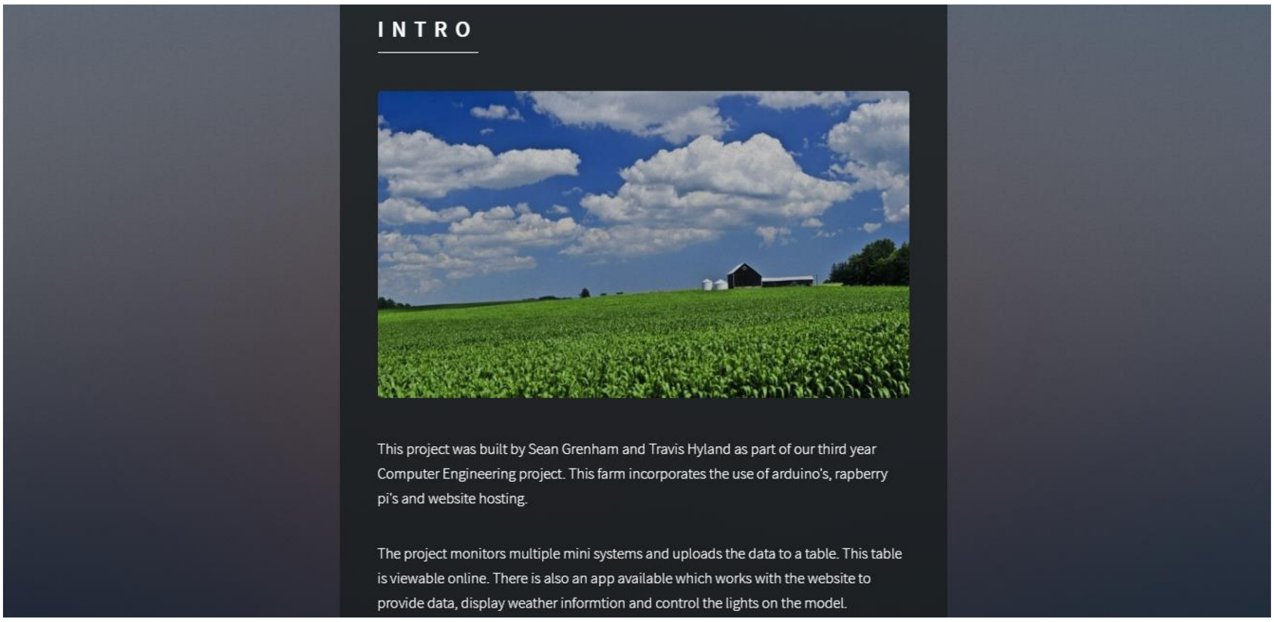


Figure 13: Website Info Screen

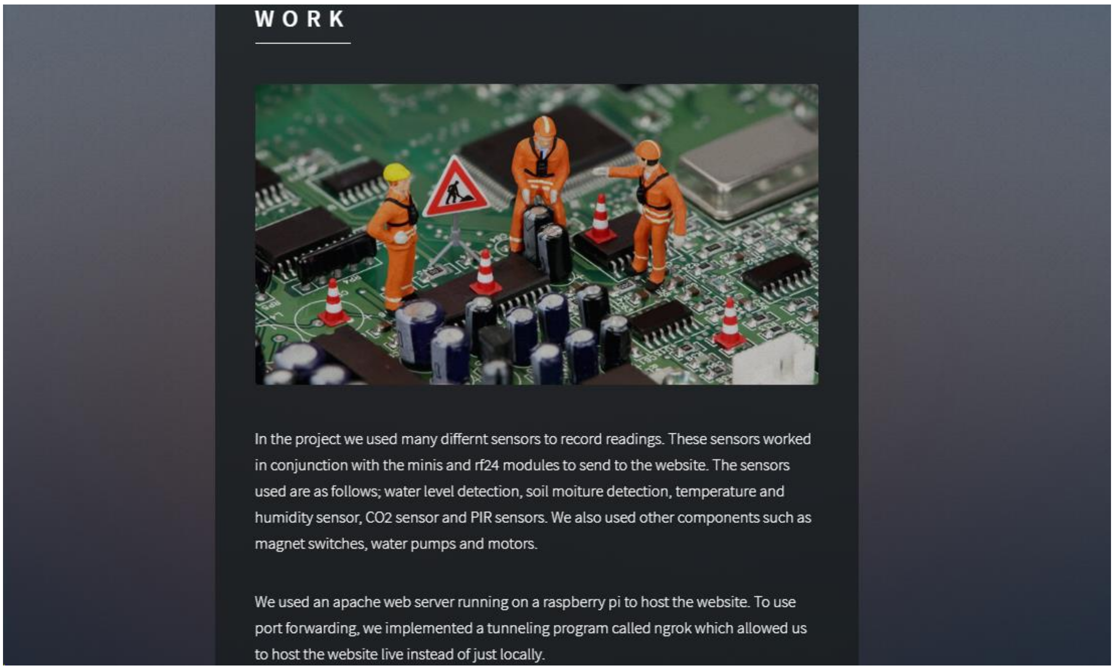
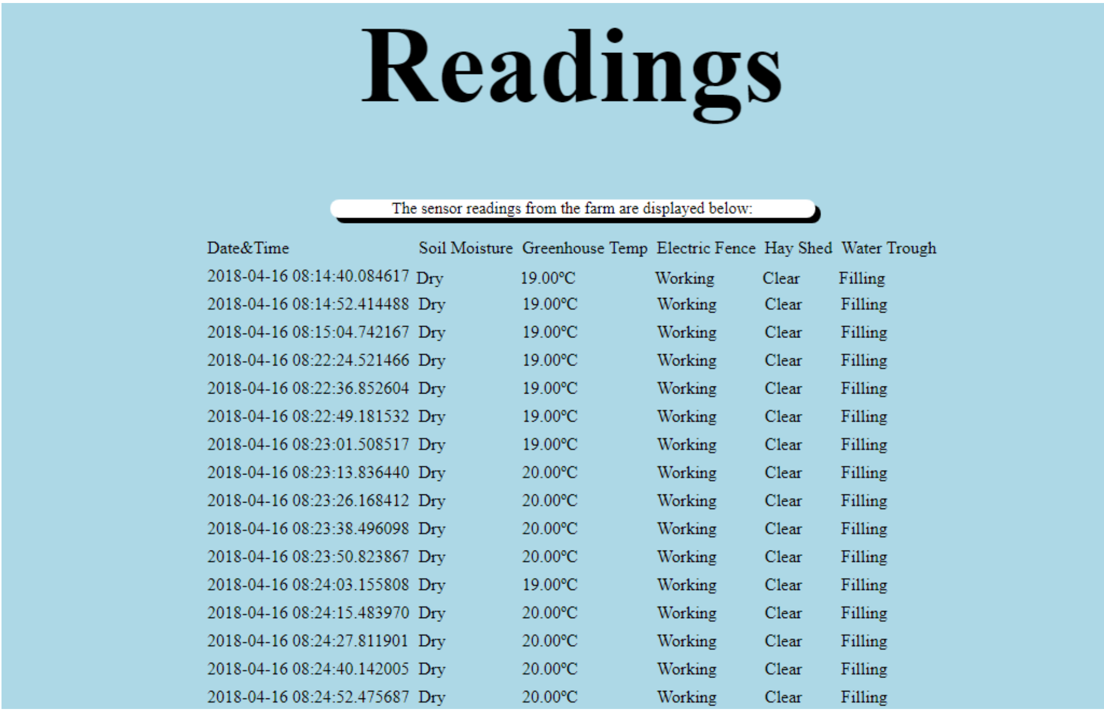


Figure 16: Website Info Screen

Figure 15: Website links page

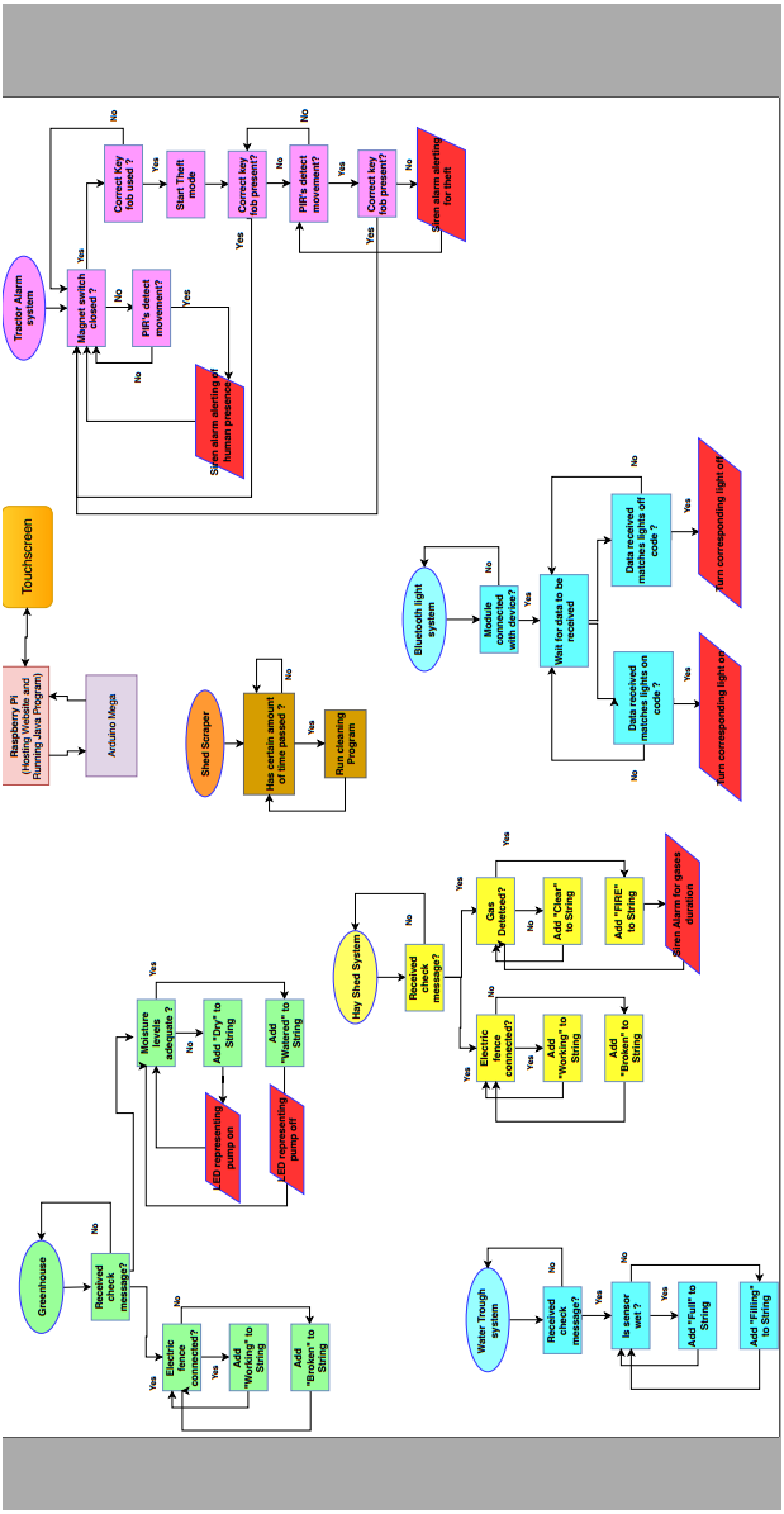
Figure 18: Website Info page



Date&Time	Soil Moisture	Greenhouse Temp	Electric Fence	Hay Shed	Water Trough
2018-04-16 08:14:40.084617	Dry	19.00°C	Working	Clear	Filling
2018-04-16 08:14:52.414488	Dry	19.00°C	Working	Clear	Filling
2018-04-16 08:15:04.742167	Dry	19.00°C	Working	Clear	Filling
2018-04-16 08:22:24.521466	Dry	19.00°C	Working	Clear	Filling
2018-04-16 08:22:36.852604	Dry	19.00°C	Working	Clear	Filling
2018-04-16 08:22:49.181532	Dry	19.00°C	Working	Clear	Filling
2018-04-16 08:23:01.508517	Dry	19.00°C	Working	Clear	Filling
2018-04-16 08:23:13.836440	Dry	20.00°C	Working	Clear	Filling
2018-04-16 08:23:26.168412	Dry	20.00°C	Working	Clear	Filling
2018-04-16 08:23:38.496098	Dry	20.00°C	Working	Clear	Filling
2018-04-16 08:23:50.823867	Dry	20.00°C	Working	Clear	Filling
2018-04-16 08:24:03.155808	Dry	19.00°C	Working	Clear	Filling
2018-04-16 08:24:15.483970	Dry	20.00°C	Working	Clear	Filling
2018-04-16 08:24:27.811901	Dry	20.00°C	Working	Clear	Filling
2018-04-16 08:24:40.142005	Dry	20.00°C	Working	Clear	Filling
2018-04-16 08:24:52.475687	Dry	20.00°C	Working	Clear	Filling

Figure 17: Website Readings Page

Project Logic Chart



Testing and Evaluation

Once the project was built, it was time to start running tests to make sure all the subsystems were reading values correctly, uploading them properly and displaying the on the website. During this testing, multiple problems were found and had to be found and fixed before the deadline. These problems are discussed below

Bad makeshift female to female headers

Earlier in the project, female to female headers were required to connect sensors and rf24 cards to the project. These connectors consisted of two female to male 8 pin headers pushed together to make a single female to female 8 pin block.

These headers were meant to be temporary but when the female to male jumpers that were ordered didn’t arrive, they became permanent. But while testing, 4 of the rf24 cards blew and each one had been using these connectors. These connectors were then taken from the project and scrapped. Eventually female to male jumpers could be sourced which allowed the project to then be tested properly.

Incorrect sensor readings

Each of the sensors in this project are powered using 3v AA battery packs, bar the MQ-2 gas sensor which had a 5v battery bank. Most of the sensors ideally require a 5v power supply to measure exact readings but unfortunately more 5v power supplies could not be sourced. This can be seen on the website where the temperature sensor reads -999^{o}C every 10 or so readings. This was factored in for in other systems by giving them a high tolerance range for all readings but for temperature, readings are coming straight from the sensor so they can’t be altered.

In the future, the code could be altered to save the previous reading and if it is drastically different to the one before, repeat the previous value.

Faulty Gas Sensor

During the testing phases, it was discovered that the gas sensor being used is faulty and is giving false values constantly. It was discovered by printing the analogue values that were being read from the gas sensor as they were received. They fluctuated constantly from 100 up to 800 within a 10 second test period. Different jumpers, Arduino mini and power supply was used but the values were still false.

To remedy this problem, the output was changed from analogue to digital. When this was done, the values were not checked on the mini. All that the mini received was one’s and zero’s. The detected limit was now changed via the onboard potentiometer which would make getting the exact readings impossible to get but for this project, the digital reading didn’t have to be an exact value. As long as when smoke is detected the output value changes from 0 to 1, it is all that needed.

Summary of Testing

Apart from the previous errors, the project all tested correctly and printed values to the website which is the final goal. The minis all wait until the mega asks for information to check sensors and send data which will help keep power usage down and keep data from getting corrupted. When the project was all tested correctly, the code was commented out as so if people try and remake the project, they will understand how the code works.

Future Development

If more time was given to complete this, there could be a great amount more extras added to make the project more complete and able for practical use. These include:

- Connecting the rf24 modules to a WIFI network and having each one separately uploads data. If this method was done, there would be less of a chance of data being corrupted while going from the sensor to the database. The rf4 modules can now be coded to work with any WIFI router and upload data the same as the Arduino Wi-Fi shield. This would keep costs down and allow systems to be controlled separately from online as they are all sending and receiving data separately.
- Use a different system for to control the shed scraper system. The system currently implemented is not practical and just for demonstration purposes. One system that could be used is having the motor mounted on the scraper and run the scraper on a track. This would allow no cables going across the yard and a battery and charging station method could be used to power the system. If the scraper was run often enough, there wouldn't be enough dung built up to put much pressure on the motors.
- A watering system could be installed into the greenhouse to allow the plants to be watered on a timed schedule and this could be adjusted depending on the temperature inside the greenhouse. The watering system could be suspended from the roof of the greenhouse which would allow adequate cover of all the plants. There could also be a system installed that feeds the plants with certain plant food which could be attached to the same timer as the water. This would help the plants thrive more and produce more crop when they are finished growing. A PIR could also be fitted in the greenhouse to detect any intruders such as rabbits and sound a horn to scare them away.

- The water level sensor will have to be placed inside a protective box. Cattle like to scratch off any rigid surface they can find so the sensor would have to be placed into a protective box to stop it getting damaged. Also, there is a sensor called the SBT Aqua sensor that checks water for contamination in real-time which could be used to test the water for any contamination. Cattle often stand near the water trough after drinking and often dung into the trough which makes them not want to drink the water (not the smartest animals at times). This sensor could help the farm know when this happens so they can clean it out before the cattle get stressed from no water which makes them break out.

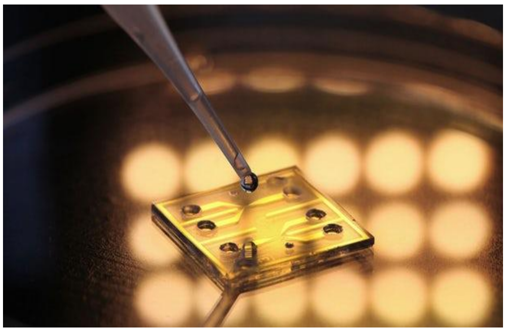


Figure 19: SBT Aqua sensor

- The tractor alarm could include a vibration sensor to detect if glass is broken or if the tractor is being moved. This would work with the alarm better as the intruders could just block the PIR sensors and they would never alarm. Another addition could be a text message system. A GSM module could be attached to the tractor alarm as so whenever movement is detected, the farmer would receive a message alerting them of the activity. A GPS module could be attached which could update the tractors location every 10 minutes to an online database so if it is ever stolen, there is tracking for the tractors location.
- All the systems bar the shed scraper could be fitted with solar panels and rechargeable batteries to allow the system to be totally self-sufficient. This would reduce all energy costs of the project to a minimum.
- The electric fence system could be fitted with multiple mini systems that could have one placed per field. Usually a farms electric fence is all interconnected and this would help track where the fence is broken. A mini with a rechargeable battery, which could be charged from the fence’s current, could be placed and use a GSM module to upload each fields status to a database. If a field is broken down, the system wouldn’t detect current and so alert the farmer to approximately where it has broken down.

These advancements could be added to the project to improve the systems reliability and make it seem like a more viable option for monitoring a farm. Every subsystem in the project could be altered to include many more functions and the overall system could work by having a base product for sale and all the extra systems are sold as separate modules. As more advancements are made in different sensors and ways of recording data, more modules could be released offering these systems.

Conclusion

The overall project turned out much better than expected, being able upload the data to a live website was an added extra which was discovered during the process of building the project. I think it adds hugely to the expandability of the project and makes it much more likely to be used in practical use.

The components suited their jobs very well and after some careful tinkering, all were mostly reliable. If power wasn’t a problem to the project, the readings would be very exact and would allow a farmer to keep track of many different processes on their farm from anywhere and trust in its exactness.

If I was remaking this project in the future, I would purchase components off a more reliable retailer rather than Aliexpress who are known to not have the best quality items for sale. Also, if all the ordered components had arrived, the project could have had more depth and technically than it presently has. If more rf24 cards were available to use, there could be many other subsystems that could be created easily and would suit the whole farm system perfectly such as a weighing system to instead of having the farmer input used feed, the system would automatically track this by monitoring the weight of feed left in each bag.

In relation to the projects objectives, I will summarise each conclusion separately below:

1. **Must incorporate wireless communication**
   In the project, there was three methods of wireless communication used to connect and interact with the subsystems; Wi-Fi, Bluetooth and NFC. The wireless cards that were used for the project suited the project perfectly and could definitely be used in a practical build of this project due to their range and data transmission rate. The Bluetooth module for the lighting system worked very well but believe that if

this system was going to be implemented on a farm, Wi-Fi would be an easier to use method. The NFC tag for the tractor alarm worked very well and it would be very practical as the tag would sit on the tractors keys and make it a very easy system to use.

2. **Must use Arduino in project**

   In the project, there was many Arduino systems used to interact and control the system. The use of Arduino minis was a very good idea as they are cheap, small and can be run off 3v or 5v power supplies which makes them very power efficient. The Arduino mega used for the main data hub wasn't really needed as there was only basic code running on it and it used very little memory onboard for which the mini had a sufficient amount for.

3. **All persons in the group must contribute equally**

   For the project, I believe each team member worked to the best of their ability. The project was split hardware and software, but both members had to work together for some sections as it would take too long to accomplish alone. The group worked well together, each member having their own strengths and weaknesses but all in all they complemented each other well.

In relation to the personal objectives, I believe all the personal objectives were met by the end of the project. A raspberry pi was used which hosted a website that displayed all the required project data. A farming app was created that allowed the control of the farm lighting system, displayed current weather for the farmer current location and had a link which showed the website data. The feeding program was created and was able to email the co-op about any orders which was an extra to the goal.

All in all, the project went excellently and it has left both members of the group with a yearning to do more with Arduinos and the Raspberry Pi. The group was a good match and the project was a great topic as both members were from farming backgrounds.

All code that is used is located at the following website:
https://github.com/THETHYLAND/AutomatedModelFarm

References

Figure 1: https://startingelectronics.org/beginners/tools/soldering-iron/

Figure 2: http://www.lasertools.co.uk/product/7019

Figure 3: https://www.pinterest.com/pin/478859372852788587/

Figure 4: https://www.seeedstudio.com/Soldering-Safety-Glasses-p-1489.html

Figure 5: https://newatlas.com/bacteria-water-sensor/39794/