ISONDLO ELECTRONICS TEAM
Prototype System V1.0

Engineers Without Borders - UCT

Team Leader: Benjamin Connolly
cnnben002@myuct.ac.za

Table of Contents

Introduction

This document serves to record the progress and design process of the electronic engineering team working on the iSondlo garden project as we move into the prototype phase of the project.

The decision was made to split the electronics team into two sub-teams. This was so that two teams could work simultaneously on their parts independently of one another. This will hopefully increase overall team efficiency.

The power and solar team are responsible for the generation, manipulation, and storage of electricity generated through solar energy. They will also be responsible for the distribution of electricity for the entire electronics system and making sure that individual component's specific power requirements are met.

The control and monitoring team is responsible for the collection of data via sensors, controlling the electrical system and outputs using microprocessors, and creating an app that allows a user to view data and turn on a valve (to control the watering system).

Team Members

Control and Monitoring

- Ben Connolly (Lead)
- Heather Wimberley
- Justine Marie Pauline Bouchard
- Umutesa Munyurangabo
- Zuleigha Patel
- Bonga Njamela

Power and Solar

- Given Kibanza (Lead)
- Nozipho Sibanda
- Fhatani Netshirembe
- Antony Mason-Gordon
- Sampson Nwachukwu
- Bonga Njamela

Control and Monitoring Team

Requirements

Monitoring Requirements

- Soil moisture
- Humidity
- Temperature
- Sunlight levels
- Sensors will need to be weatherproofed

Control System Requirements

- Log sensor data
- Interface with app
- Control water valve
- Will need to have cellular connectivity (optional for this prototype)
- Components will need to be protected from the elements

App Requirements

- Display weather patterns for site
- Sensor data displays
- Control irrigation system via valve
- Notification system

Solenoid Valve Requirements

- Must be a 12V valve
- Valve must open and close (try choose normally closed valve)

Block Diagram

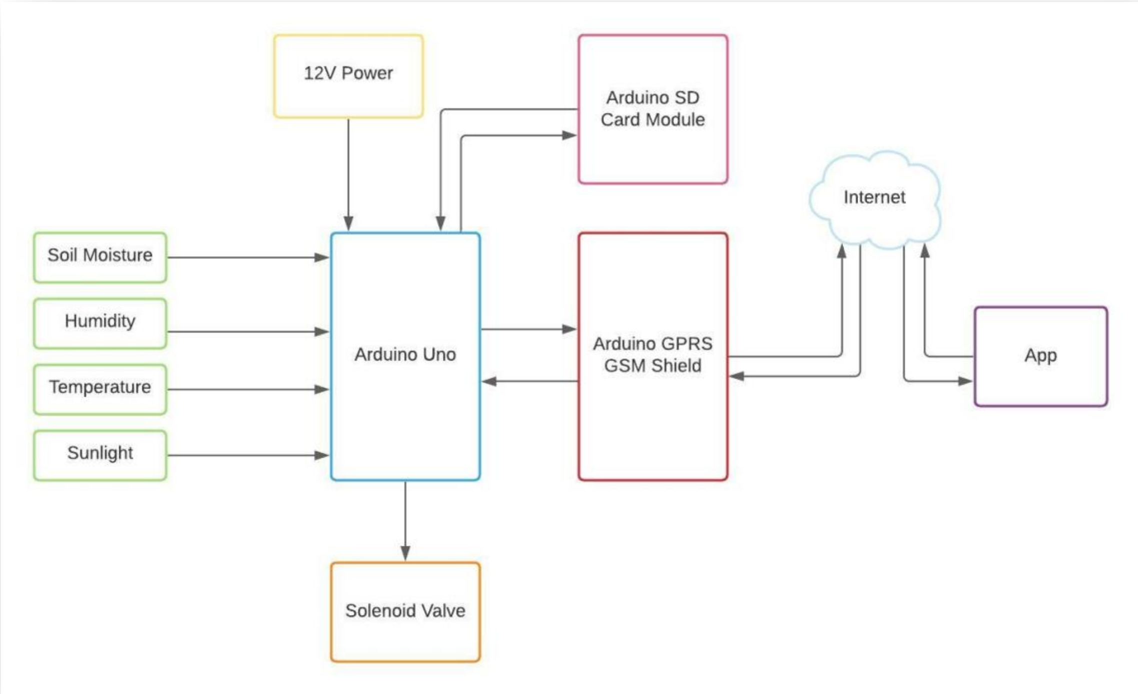

Figure 1 - Block Diagram of Control & Monitoring System

Microcontroller Design

Originally in the ideation phase, we had planned on using a combination of a Raspberry Pi and an Arduino Uno. This design was chosen so that we could expand the system if needs be by having the Raspberry Pi as the main computing chip and then Arduinos could be added onto that to extend the number of inputs/outputs that the system could handle. This was also when the site was close to Wi-fi which the Raspberry Pi could connect to so that we could send/receive data via an app. Circumstances have changed, we are now at a different site which is not close to Wi-fi which called for a new design.

From the block diagram, we can see that the new design features only an Arduino Uno as the only microcontroller. The Arduino will have a GPRS GSM Shield which attaches to it and this will give the Arduino full cellular capabilities – this helps us navigate the problem of no Wi-Fi on site.

Advantages of using this newer design:

- Simpler system as everything is Arduino Based
- A uniform programming language can be used
- Cheaper as we aren't utilising the expensive raspberry pi
- Easy to expand
- Power can come directly from 12V batteries and doesn't need to be regulated

An image of the GPRS GSM Shield that could be used is below:

Figure 2 - Arduino GPRS GSM Shield

Here is a link to a helpful tutorial explaining how to set up and use the GPRS GSM Shield - https://www.arduino.cc/en/Guide/ArduinoGSMShield

From the block diagram in Figure 2, we can see that the new design features an Arduino SD Card Module which attaches to it and this will give the Arduino full data logging capabilities – this helps us navigate the problem of having no off-site server.

Advantages of using this newer SD Card Module design:

- Simpler system as everything is Arduino Based
- A uniform programming language can be used
- Cheaper as we aren't utilising the expensive raspberry pi
- Easy to expand by purchasing an SD card that stores more data

An image of the SD Card Module that could be used is below:

Figure 3 - SD Card Module

Required Components:

- Arduino Uno
- Arduino GSM GPRS Shield
- Sim Card + Data (500 MB)
- Arduino Micro SD Card Module
- Micro SD Card
- Capacitive Soil Moisture Sensor
- Humidity Sensor
- Waterproof Temperature Sensor
- LDRs

Expansion

Important: This prototype will not be testing the expansion capabilities of the design, this is just for future reference

Expansion is important as we move forward into the fully scaled design and is relatively easy with this design. When we think of expanding our system, we think of adding a lot more sensors to our garden, the Arduino only has a limited number of ports so expansion requires us to extend the number of ports of the Arduino. The easiest way to do this is to daisy chain multiple Arduinos together via the serial port and we can then set up code so that they communicate in that way. The picture of the basic set-up will be as follows:

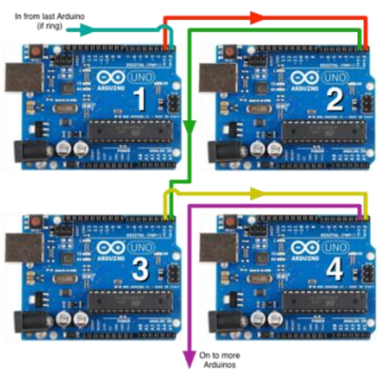

Figure 4 - Daisy Chained Arduinos

A helpful blog outlining the daisy-chaining process can be found here - http://www.wintergroundfairlands.com/2012/12/daisy-chaining-serial-connections.html

App

For the app, a node-red server was first thought of in the ideation phase. This is a good idea, but incredibly complex and time-consuming to set up, would require capital to pay for a dedicated server, and would be hard to make the display incredibly user-friendly. The solution is Blynk.

Figure 5 - Blynk Logo

Blynk is an app dedicated to prototype, deploy, and remotely manage connected electronic devices at any scale. This has a range of bonuses which include:

- Drastically more simple and easier to use
- Lots of existing documentation and support
- Offers budget-friendly and free options
- Blynk hosts the app on their servers
- Quick deployment and testing
- No need to spend hours programming an app

The Blynk app would allow us to completely customise our homepage to suit our needs. For example, we can add graphs only the sensors we need and display current data readings. An example of what our homepage could look like is below.

Figure 6 - Possible Blynk Home Screen

The figure below gives a good overview of how the overall system connects together.

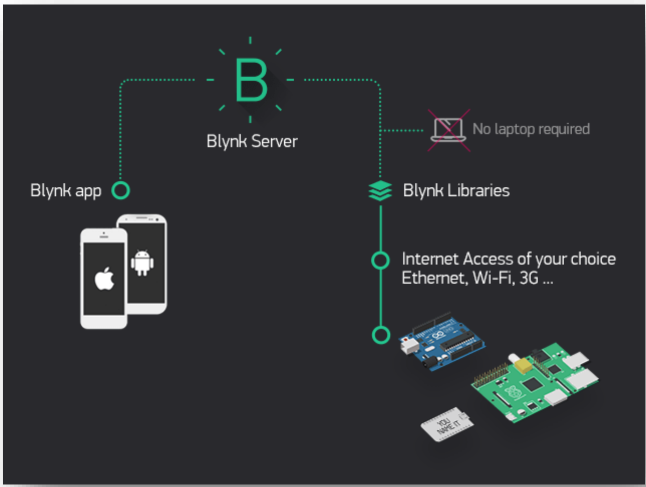

Figure 7 - Blynk System

Another great bonus of using Blynk is that it integrates perfectly with our current design. The Arduino's GPRS GSM Shield will connect directly to our Blynk app via the internet and will allow us to get real-time information about our system.

This would be a great choice because it meets the following iSondlo requirements for the app:

- It will allow us to remotely control the irrigation system using a solenoid valve
- It will display current data readings such as soil moisture/humidity/temperature etc.
- It will log previous data readings
- It has an inbuilt notification system so that users can get notifications if certain garden parameters are below a certain threshold

Required Components:

- Blynk App

Solenoid Valve System

A figure of the solenoid valve system:

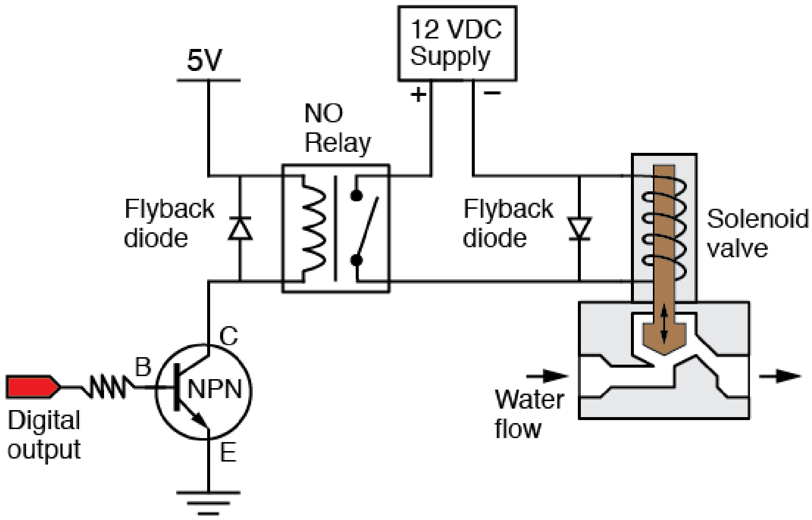

Figure 8 – Solenoid Valve System Overview

Image source:
https://web.cecs.pdx.edu/~eas199/B/howto/fishtank/wiring/img/transistor_relay_solenoid_valve_flyback.png

The logic behind the system is quite simple. The Arduino will have a dedicated digital output pin which will in turn control an NPN resistor. This resistor in turn will control a relay circuit that controls the solenoid valve. The Arduino will connect to the internet and interface via the Blynk App so that the solenoid valve can be controlled remotely.

The smaller components such as the resistors and diodes will be calculated once we know what relay and solenoid valve is going to be used exactly. This will not hinder our progress as these components are readily available from UCT's White and SASOL Electronics Lab.

Required Components:

- Relay Switch (min rating 10V, 2A, 20W)
- Solenoid Valve (DC 12V, 1A max)
- Resistors
- 2N222A NPN Transistor
- 1N4001 Diode

Weatherproofing

Soil Moisture Sensors

After doing online research, we are going to waterproof the connections and electronics of our capacitive soil moisture sensors following this tutorial: https://www.instructables.com/Waterproofing-a-Capacitance-Soil-Moisture-Sensor/

Figure 9 - Waterproofed Soil Moisture Sensor

Required Components:

- Adhesive lined polyolefin Heat-shrink tubing ¼ inch
- Adhesive lined polyolefin Heat-shrink tubing ½ inch
- Adhesive lined polyolefin Heat-shrink tubing ¾ inch
- Nail Polish
- Heat Gun/Hairdryer

Temperature Sensor

The temperature sensor chosen are waterproof ones so no waterproofing was needed.

Figure 10 - Waterproof Temperature Sensor

Light Dependent Resistors

It was quite tricky coming up with a system to waterproof the LDR because the entire component needed to be waterproofed including its connection points. Ultimately we decided to go with clear-set epoxy glue. This would allow us to cover the entire component and all its connections in one go as well as allowing it to still detect light. We could also add a stake/skewer into the epoxy before it sets so that the LDR can easily be stuck into the ground.

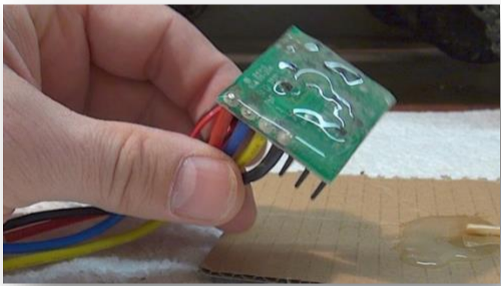

Figure 11 - Epoxy Waterproofing Electronics

Required Components:

- Clear Set Epoxy Glue
- Garden Stake/Skewer

Control Housing

The control hosing will simply be a clear plastic Tupperware. This will protect it from the elements and will provide easy access when needed. The seal provides a fully watertight seal around the container lid. The various wires coming in and out of the control housing will be through drilled holes and sealed off with clear set epoxy glue.

Required Components:

- Tupperware (420ml)
- Clear Set Epoxy

Figure 12 - Tupperware Protecting Electronics

Signal Wires & Other Connections

The signal wires (wires carrying signals and voltages between the various components) need to be waterproof rated if they are not enclosed in a watertight housing. This is extremely important because if waterproofed properly and water enters the system, there could be electrical fires and electrocution may occur damaging property, buildings and people. Similarly, all exposed connections need to be sealed off with epoxy glue.

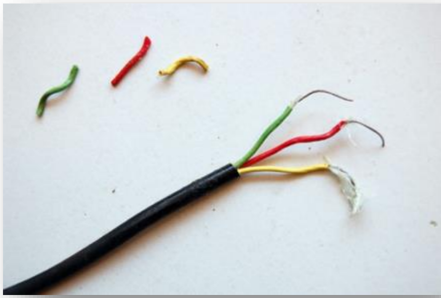

Figure 13 – Waterproofed Wires Wrapped in Double Layer of Plastic Protection

Required Components:

- Outdoor Rated & Waterproof Wires
- Clear Set Epoxy

For future reference, but not necessary for this prototype: When implementing this system into the garden it would be good to bury the cabling deep underground as not to be accidentally broken while gardening and make use of PVC piping to provide an extra layer of protection.

Testing

To test our system, we plan on setting up a miniature garden in one of the UCT labs. We will have a small pot which we can water to test our soil moisture sensor, we can simulate sun for our LDR's using a light on our phone, we can measure our temperature and humidity reading vs what the weather app is saying for our area.

The control system and app will either work or won't work so it will be easy to tell if we are on the right track or not.

The solenoid valve system will require us to connect two pipes with the solenoid valve in the middle. We can then fill the top half with water and when we active the solenoid valve, water should hopefully flow down into the second pipe and into a bucket.

We would also like to heavily test the weather-proofness of our system by extensively testing our waterproofed components and making sure that not even a drop of water manages to penetrate through to the electronics.

Required Components:

- Bucket
- ½ inch tubing

Schematics

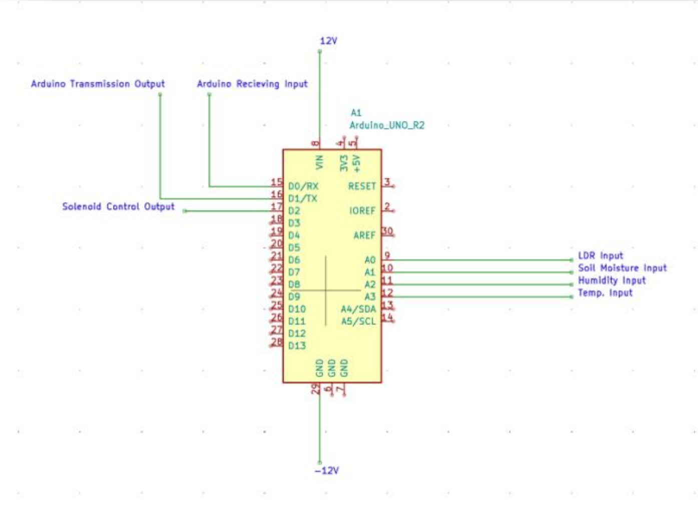

Figure 14 - Arduino Schematic

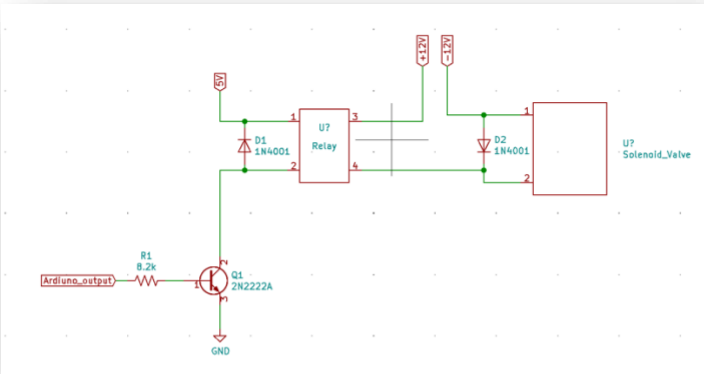

Figure 15 - Solenoid System Schematic

What We Are Aiming To Do Next

- **Get iSondlo's Feedback**
- Source our components
- Build our system
- Test our system
- Make a list of our shortcomings and find ways to fix them
- Start thinking about implementing this into the next prototype version or the iSondlo garden

Power & Solar Team

Requirements

Solar Requirements

- Perform power calculations to determine the power consumption of the electronics system.

- Design a solar system that will cater to these needs. Additionally select the components needed for the system; solar panel, charge controller, batteries, etc.

  For this prototype, the solar system needs to be able to power the entire electronics system and should store enough power to keep the system running for a few hours after the solar panel stops producing electricity (aka. It gets dark).

  When coming up with the design take into account that this is just our prototype system and that the final version will have a much greater power storage need. Therefore, start small and try to make the design modular so that we can upgrade the solar system when the time comes. It is important to outline how the expansion process will take place.

- The batteries should output 12V.

- Solar system must be weatherproof.

Power Monitoring Requirements

Important: This will not be done in this prototype.

These requirements are time permitting and may only be implemented in a future prototype.

- Design a monitoring system that provides real-time information on the solar system nodes. This should be able to link up with the Arduino and the app so that the user can see what is happening with regards to the power generation/consumption/storage.

Block Diagram

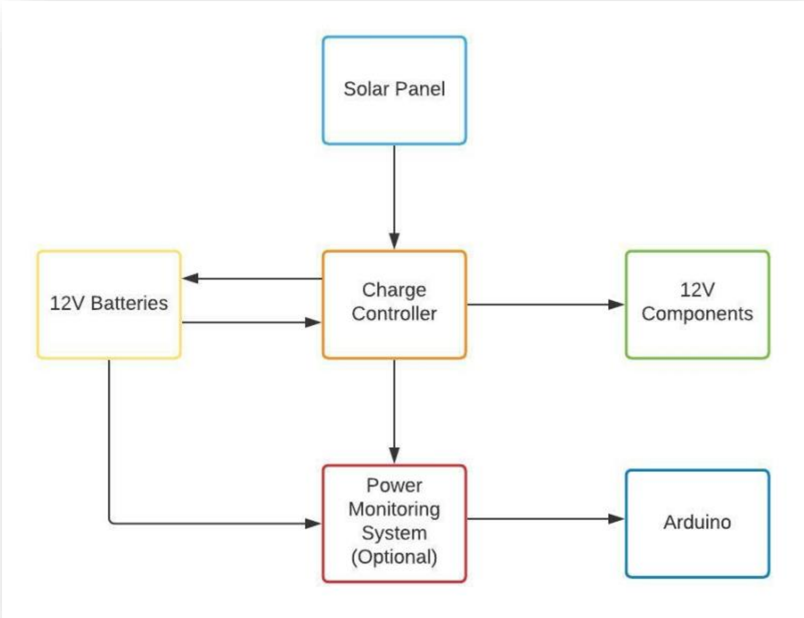

Figure 16 - Block Diagram of Solar & Power System

Power Calculations

Device	**Max Voltage (V)**	**Max Current (A)**	**Power (W)**
Arduino Uno	6	2	12
Solenoid Valve	12	1	12
		Total Power:	24

Since all the sensors/modules interface with the Arduino, we only need to take the Arduino's maximum power consumption into account. We also didn't need to take into account any of the other teams' power consumption needs for this prototype.

From the above table, we can see that the power that the system consumes is 24 Watts.

Solar Panel

From the above power calculations, we can see that we will need a solar panel that is capable of providing at least 24W to power the system. We will also need to double that power generation if we want to store enough power to last us during the night/rainy days. We have decided to go with one 50W solar panel in order to achieve this.

Figure 17 - 30 Watt Solar Panel

The solar panels will be connected to the charge controller.

Required Component:

- 50W Solar Panels (12V, 2.5A)

Charge Controller

The charge controller of the system is the 'power management system'. The electronics inside regulate the voltages and currents coming in from the solar panel and then charges the battery in a stable way. It also takes the power from the batteries and provides a stable voltage to the control and solenoid system. It also prevents overcharging and electrical spikes.

We had to choose between using a Pulse Width Modulation (PWM) charge controller or a Maximum Power Point Tracking (MPPT) charge controller. Both have their advantages and disadvantages. After doing our research, we have decided to go with the PWM charge controller and its advantages are as follows:

- Simpler design
- Cheaper
- Better suited for small solar systems
- Reduces the power entering the battery as it approaches its maximum charge (prevents overcharging)

The following figure shows how the charge controller will integrate with the rest of the design:

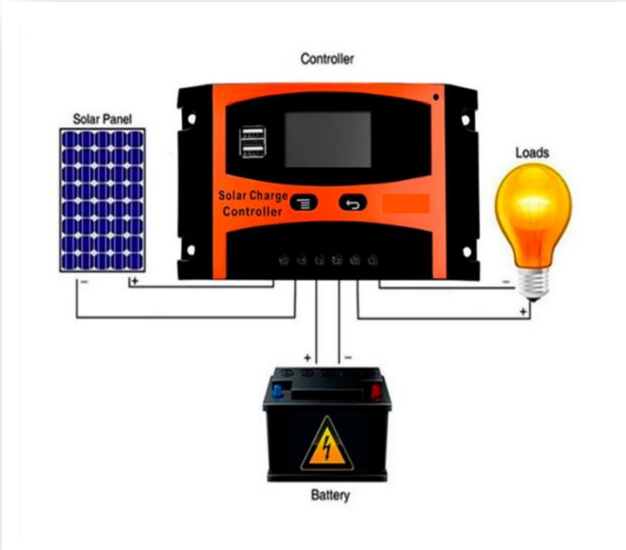

Figure 18 - Charge Controller Integration

Required Components:

- Charge Controller (12V, 5A min)

Batteries

When choosing a battery to store our power, we had two battery types to choose from (lead-acid or lithium-ion) and 3 main categories to focus on:

- Cost
- Energy density
- Charging

Taking the above into account, it seems unnecessary to be searching for a lithium ion option for early stage prototyping. The energy demands of the iSondlo electronics are significantly lower than that required in industrial applications where lithium-ion would be better suited. Additionally, the cost benefit of lead acid batteries are a significant factor at this stage of design.

Important: It must be noted however that not all batteries are self-contained/protected from weather conditions and should be housed appropriately (but not sealed).

For our prototype system, we will be making use of a 12V, 2.4Ah lead-acid battery. This should power our system for over an hour once the solar panel cuts out (from not enough sunlight).

Figure 19 - 12V Lead-Acid Battery

Required Components:

- Lead Acid Battery (12V, 2.4 Ah)

Testing

After the system's configuration, it will thoroughly be tested to ensure proper performance and safe operation. The following tests will be carried out during and after the installation of the system:

- **Continuity and resistance testing**: This process will be performed to ensure that the conductors and other terminations are well grounded.

- **Polarity testing:** At this stage, the polarity of the battery, charge controller, and solar panel will be verified.

- **Voltage and current testing:** This will be performed to ensure that the solar panel, battery, and the power inverter's operating parameters are within the required/expected specifications.

- **Performance testing:** This will also be performed to ensure that the system power and energy outputs correspond with the expected values.

- **System functional testing:** This test will be used to determine that the system is working correctly and within the required or nominal limit. It involves starting up, shutting down, and checking the nominal operating conditions of the system.

Expansion

Since the this stage of the project is a prototype, there will be scaling up the capacity of the system in the future. Therefore, the following components/parts will be scaled up when the time comes:

- Battery
- Charge Controller
- Solar Panel
- Batteries

The capacity of the battery needed in this project depends on the total load requirement of the entire system. The higher the load capacity, the higher the battery's capacity, which also depends on the number of hours the system will be ON. Since the project is at the prototype stage, it may require a low-capacity battery, but it can continually be expanded by performing a simple system load analysis/calculation and adding more batteries connected in parallel as needed.

The capacity of the charge controller will be expanded as long as there is a significant increase in the number and capacity of the panels. Suppose the number and capacity of the panels that will be installed in the future are more than the capacity of the charge controller installed initially. In that case, the capacity of the new controller will be determined by calculating the total output current and voltage of the solar panel(s). With our current charge controller, we can double the size of the system without needing to upgrade it.

What We Are Aiming To Do Next

- **Get iSondlo's Feedback**
- Source our components
- Build our system
- Test our system
- Make a list of our shortcomings and find ways to fix them
- Start thinking about implementing this into the next prototype version or the iSondlo garden
- Design and prototype weatherproofing for the system
- Design and prototype a monitoring system
- Design and prototype a mount for the solar panels

Just a note about the future prototyping of the solar mount system: This is currently being researched by the team and, considering all the factors involved, we did not feel comfortable including it in this prototype, but the following figure will provide a feel for the direction we are going in.

Figure 20 - Possible Solar Mounts for Future Prototypes

Bill of Materials and Estimated Expenses

Description	Amount Needed	Estimated Cost per Item (ZAR)	Estimated Total Cost	Subsystem	Specifications
Arduino Uno	1	167	167	Control & Monitoring	Standard Arduino Uno R3
Arduino GSM GPRS Shield	1	167	167	Control & Monitoring	SIM900 Quad Band GSM GPRS Shield
SIM Card + Data	1	49	49	Control & Monitoring	Any South African SIM Card with at least 500 MB Data
Arduino Micro SD Card Module	1	121	121	Control & Monitoring	Micro SD Card Reader Compatible with Arduino
Micro SD Card	1	96	96	Control & Monitoring	Storage Greater than 2 GB
Soil Moisture Sensor	2	46	92	Control & Monitoring	Capacitive Soil Moisture Sensor
Humidity Sensor	1	43	43	Control & Monitoring	Humidity Sensor
Temperature Sensor	1	137	137	Control & Monitoring	Waterproof Temperature Sensor
LDR	1	46	46	Control & Monitoring	Light Dependant Resistors
Solenoid Valve	1	182	182	Solenoid Control	DC 12V, 1A Max, 1/2 Inch
Relay	1	61	61	Solenoid Control	Min 10V, 2A, 20W
Resistors	1	0	0	Solenoid Control	-
NPN Transistor	1	0	0	Solenoid Control	2N222A
Diode	2	0	0	Solenoid Control	1N4001
Heat Shrink	1	91	91	Weatherproofing	1/4 inch adhesive lined
Heat Shrink	1	91	91	Weatherproofing	1/2 inch adhesive lined
Heat Shrink	1	106	106	Weatherproofing	3/4 inch adhesive lined
Nail Polish	1	67	67	Weatherproofing	Clear (preferable)
Skewers	1	136	136	Weatherproofing	-
Clear Set Epoxy	3	89	267	Weatherproofing	Clear Set
Tupperware	1	59	59	Weatherproofing	420 ml
Electrical cable	1	242	242	Weatherproofing	Wateerproof & outdoor rated
Basin Bucket	1	78	78	Testing	-
Tubing	1	119	119	Testing	1/2 inch tubing
App Blocks	1	120	120	Control & Monitoring	-
Charge Controller	1	168	168	Solar	10A, 12V
Solar Panel	1	799	799	Solar	12V, 50W
Battery	1	235	235	Solar	12V, 2.4Ah min

Figure 21 - BOM & Estimated Costs

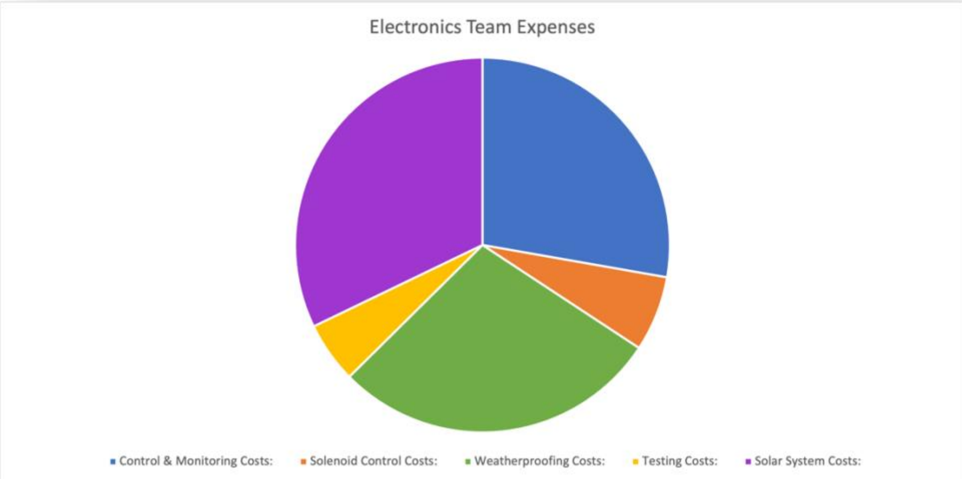

Figure 22 - Summary of Estimated Expenses

The estimated costs of our entire electronics prototype design came to a total of R3739.00. Unsurprisingly a big percentage of that cost came from the control and monitoring electronics and the solar equipment, but perhaps more surprising to the team was how much the weatherproofing is estimated to cost us.

We tried to source parts from the UCT electronics' white lab where we could. However, you will notice that most of the components come from www.banggood.com and www.takealot.com . I'm sure that I don't need to give Takealot any introduction, but banggood is a Chinese-based online retailer. I (the team leader) have used them many times, they have thousands of reputable reviews and most **importantly they are much cheaper than local stores.** I would suggest that we use the stores that are listed in the BOM to get our components. I have not taken into account the discounts, shipping costs, and insurance and therefore I would have a uncertainty percentage of 10%, bringing the total cost to R4113.00 . We would rather the team overshoots on the estimated costs and then give leftover money back to EWB than being held up by waiting for more cash to be released to us.

We would also like to point out that the prototype version for the control and monitoring team does not need a lot more capital to scale up to our full size. This initial prototype will most likely be our biggest expense and all the electronic equipment that passes our tests (which we expect all our equipment to do) in this prototype will go on into the final implementation.