Installation and Initial set-up of Resilient Smart Garden System

Table of Contents

List of Materials

Electronic Hardware-

- Raspberry Pi B+
- BB400 Solderless Plug-in BreadBoard
- Phantom YoYo High Sensitivity Moisture Sensor
- Electric Solenoid Valve 1/2” DC 12V normally closed inlet flow switch
- 12VDC 1A Regulated Switching Power Supply
- MCP3008 8-channel 10-bit ADC with SPI Interface
- OctagonStar DHT11 Digital Humidity/Temperature Sensor
- Adafruit Power Relay FeatherWing
- USB Mouse/Keyboard
- Micro SD card with at least 4GB of memory
- Micro USB power supply @ 5v 1.2A (minimum)

Cables/Connectors-

- Raspberry Pi breakout board with pin numbering
- GPIO Ribbon Cable for Raspberry Pi (40 pin)
- Male to Male jumper wires
- Female to Male jumper wires
- 12-10 AWG 10/Clam Female disconnect insulated (2)
- Standard wire connectors (2)
- ½ Vinyl tubing (required Length) (2)
- Female to Female PVC ½” to ½” (2)
- Barbed Male 90* Elbow ½” (2)
- Low voltage copper wire (required Length)
- HDMI Cable
- Ethernet Cable

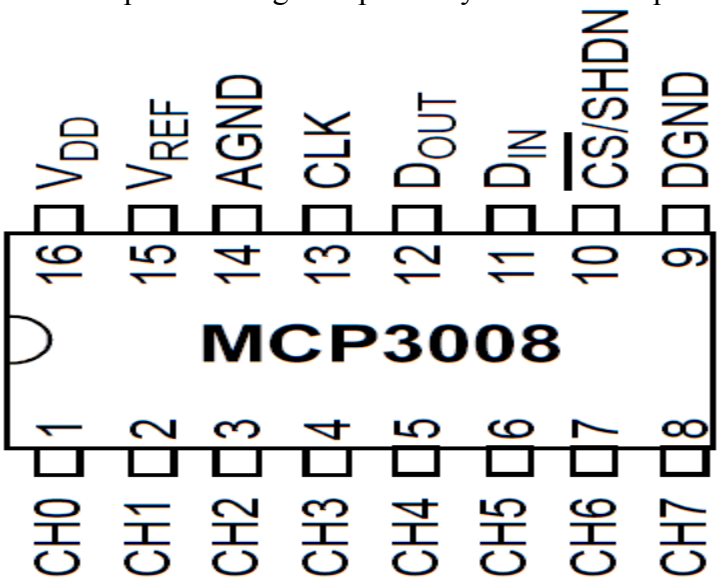
Software-

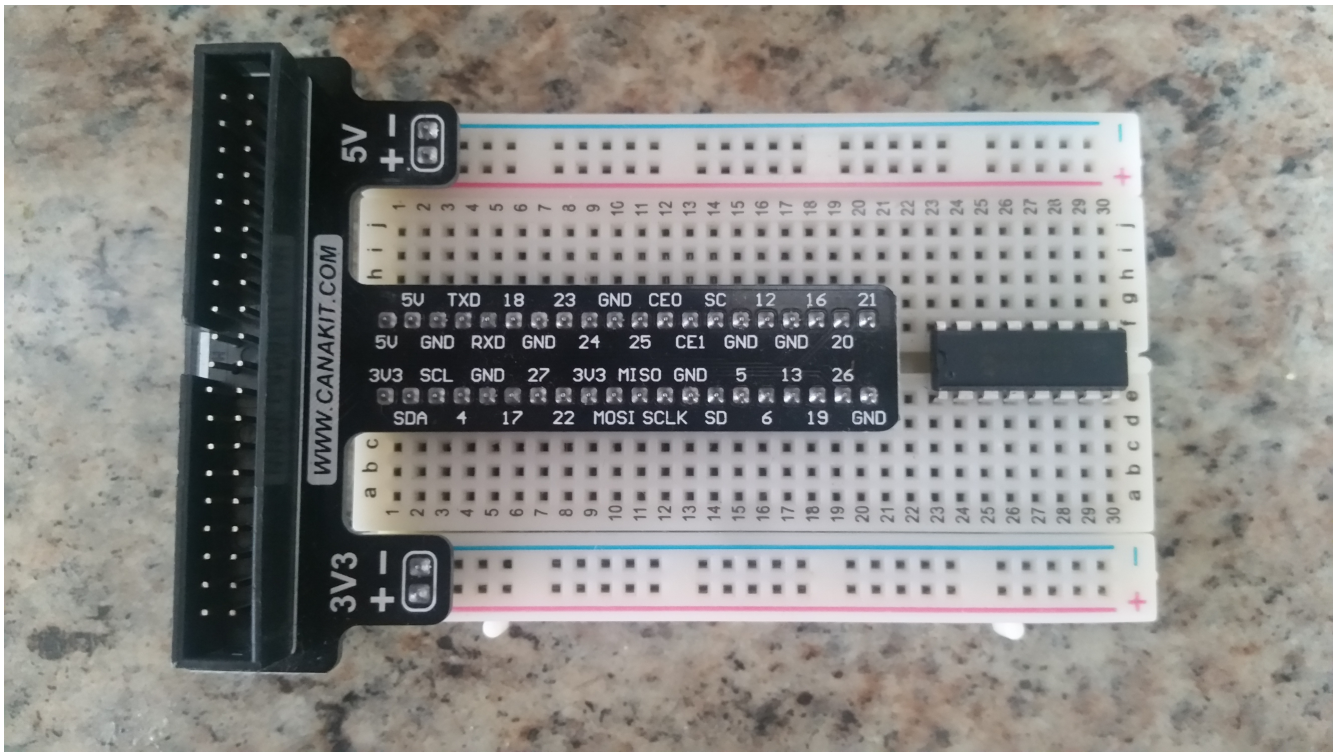
- MCP 3008 Library
- DHT 11/22 library
- Rasbian linux OS for Raspberry PI
- Smart Garden Setup Python File
- Putty SSH and Telnet client for Windows
- Android IOS for data archiving, and visualization (optional)
- Win32DiskImager utility

Initial setup and Wiring for MCP3008

MCP3008

The MCP3008 is a 8-channel 10-bit analog to digital converter (ADC). This ADC will allow the analog soil moisture sensor to send data to the Raspberry Pi. The MCP3008 connects to the Raspberry Pi using a SPI serial connection. There are two methods to use the MCP3008 with a Raspberry PI, a software SPI and a hardware SPI. This guide will use a software configuration. Place the MCP3008 on the breadboard so the chip resides on each side of the board, the half circle indicates the orientation with CH0 pin and VDD on the top left and right respectively. Place the chip below the breakout board.

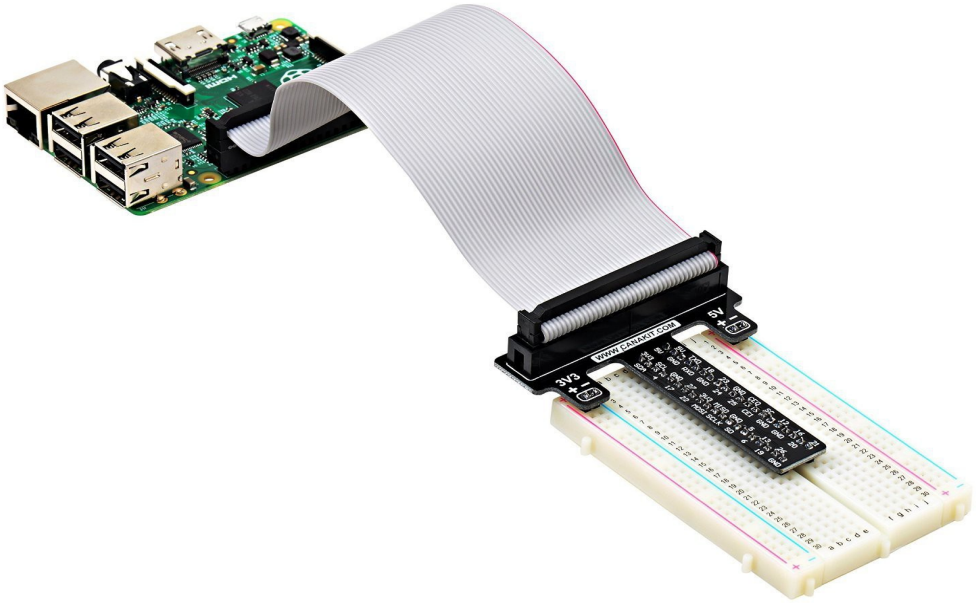




Software SPI

To connect the MCP3008 to the Pi using a software SPI configuration use the male to male jumper cables to connect the terminals from the chip to the Pi breakout board. The included code will use this pin numbering. The breakout board will be attached to the breadboard as shown below.

- MCP3008 VDD to Raspberry Pi 3.3V
- MCP3008 VREF to Raspberry Pi 3.3V
- MCP3008 AGND to Raspberry Pi GND
- MCP3008 CLK to Raspberry Pi pin 18
- MCP3008 DOUT to Raspberry Pi pin 23
- MCP3008 DIN to Raspberry Pi pin 24
- MCP3008 CS/SHDN to Raspberry Pi pin 25
- MCP3008 DGND to Raspberry Pi GND
- MCP3008 CH0 to Moisture Sensor OUT
- Moisture Sensor GND to Raspberry Pi GND
- Moisture Sensor VCC to Raspberry Pi 5V



Necessary software installations

The Raspbian Operating System

The Raspbian Operating System is Linux based. Raspbian provides a fully functional operating system with a terminal and GUI interface. To get started, download Raspbian here (https://www.raspberrypi.org/downloads/raspbian/). It is recommended that you choose the RASPBIAN JESSIE LITE version of Raspbian. It is a lighter, more basic version of the OS. For detailed instructions for installing Raspbian, visit their install instruction website (https://www.raspberrypi.org/documentation/installation/installing-images/README.md). To complete the installation, you will also need at least a 4gb micro-SD card.

Insert the SD card into your SD card reader and check for a newly assigned drive letter. Once found unzip and run the Win32DiskImager, you will need to Run as Administrator, right click on the executable to find the option. Select the file extracted for the Raspbian zip file and select the SD card drive letter. Then click "Write" and wait for it to finish. Once finished exit the Imager and safely eject the drive to avoid data corruption. The SD card can now be inserted into the Raspberry Pi and booted.

Before startup, connect the USB Mouse and Keyboard. Then plug in the Micro USB power and the Pi will power on automatically, then the Pi will ask for a ID and password, the default ID is "pi" and the password is "raspberry". The Pi will begin to initialize for startup, once finished a terminal will be displayed and can be operated as a normal Linux terminal. If you prefer a Desktop view type "startx" then hit enter, you will be given access to a Desktop UI. Take some time to familiarize yourself with the OS and where files are stored, also this may be a good time to learn some simple Linux commands for using the terminal without a Desktop. The terminal will be the black box in your task bar near the top.

The Python Language and MCP3008 Library

The Python programming language is the main language that the Pi is programmed in, and so it will be the language that all the code will be compiled to. To ensure that all modules of code are up to date you must first install the latest Python updates. To do so open the terminal and type these commands.

sudo apt-get update

You should then see several prompts and packages downloaded and updated.

```
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian jessie InRelease [13.2 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian jessie InRelease [14.9 kB]
Get:3 http://archive.raspberrypi.org/debian jessie/main armhf Packages [127 kB]
Get:4 http://archive.raspberrypi.org/debian jessie/ui armhf Packages [53.6 kB]
Get:5 http://mirrordirector.raspbian.org/raspbian jessie/main armhf Packages [8,981 kB]
Fetched 9,190 kB in 34s (265 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $
```

Next, you will want to install the Smart Gardens setup package. Simply download the smartgardens.rar from the Smart Gardens website (http://jchavis.hopto.org:49180/smartgardens.rar). Once you have download the rar file, simply extract all the files to a local folder. The folder name is unimportant. It is recommended that you install it in your home directory such as /home/pi/SmartGardens/. Open the folder you extracted the package to. You should see the following file structure.

```
pi@raspberrypi:~/Desktop/piDir $ ls -l
total 52
drwxrwxrwx 9 pi  pi 4096 Oct 28 04:55 Adafruit_Python_DHT
drwxrwxrwx 5 pi  pi 4096 Nov  8 05:20 Adafruit_Python_GPIO
drwxrwxrwx 5 pi  pi 4096 Oct 28 04:55 Adafruit_Python_MCP3008
-rwxrwxrwx 1 pi  pi 7160 Nov  4 04:14 Config.py
-rwxrwxrwx 1 pi  pi 5618 Nov  4 04:14 Config.pyc
-rwxrwxrwx 1 pi  pi   56 Nov  4 01:00 job
-rwxrwxrwx 1 777 pi    0 Dec 13 05:51 log.txt
drwxrwxrwx 5 pi  pi 4096 Oct 25 20:20 RPi_GPIO
drwxrwxrwx 2 pi  pi 4096 Nov 28 07:00 setupfiles
-rw-rwxrwx 1 777 pi 2046 Dec  3 01:28 setup.py
-rw-rwxrwx 1 pi  pi 6622 Dec  8 22:00 takeReadings.py
pi@raspberrypi:~/Desktop/piDir $
```

To start the Smart Garden's installation, type the following command:

sudo python setup.py

This will load the setup screen.

```
pi@raspberrypi:~/Desktop/piDir $ sudo python setup.py
1 - Complete install, registration, and setup with the latest packages and drivers
2 - Complete install, registration, and setup with current files
3 - Complete registration and setup, without install
4 - Add a Garden
5 - Delete a Garden
6 - Add a moisture sensor
7 - Delete a moisture sensor
8 - Add a temperature  sensor
9 - Delete a temperature  sensor
10 - view configuration file
11 - view/change scheduled readings
12 - Exit the Setup Menu
Choose one of the following options
```

It is recommended you choose option 1 which will download and install all necessary files. Setup, especially for option 1, may take a few minutes. Wait until the process completes and you will now be about to enter your username and password. If the username is taken, please try again until you find an available username. Next you will be asked to enter the name and description of your garden.

Next is the temperature sensor setup. You will be asked for the name and channel number the sensor is on. Complete this loop until all temperature sensors have been entered. Enter the name "exit" to stop the temperature sensor setup.

```
Enter the user name you want, 'exit' to quit: emptyName
Enter password: password
Enter the password again: password
"emptyName has been registered"
Enter the name for this garden
my garden
Enter a description for this garden
This is my first garden
Please enter information for the temp sensors

enter the sensor name, 'exit' to quit
temp1
Enter the channel numbertemp1 is on
19
enter the sensor name, 'exit' to quit
exit
exiting sensor input
```

The next step is the moisture setup. You will be asked for a sensor name, pin input, and moisture limit. A moisture reading below the given limit will release water. After entering all the moisture sensors, type “exit” at the name prompt. The software will then register your garden on the Smart Garden database. You will then be asked if you wish to add another garden. Complete this process until all gardens are setup.

Once the gardens are completed, you will be asked to schedule your daily readings. Smart Gardens will record only 2 readings per sensor per day. This process will guide you through setting up the automatic readings. In the example below, we have set our readings for 5:30am and 5:30pm. Times are local, military times. This is the last step in the setup.

```
Scheduling automatic readings twice daily
For this reading, enter the hour in military time (0-23)
5
For this reading, enter the minute (0 - 59)
30
scheduling readings for daily at 5:30
setting the following command: /home/pi/Desktop/piDir/job
*************************Scedule saved*************************
For this reading, enter the hour in military time (0-23)
18
For this reading, enter the minute (0 - 59)
30
scheduling readings for daily at 18:30
setting the following command: /home/pi/Desktop/piDir/job
*************************Scedule saved*************************
Goodbye
pi@raspberrypi:~/Desktop/piDir $
```

Congratulations, you are done!

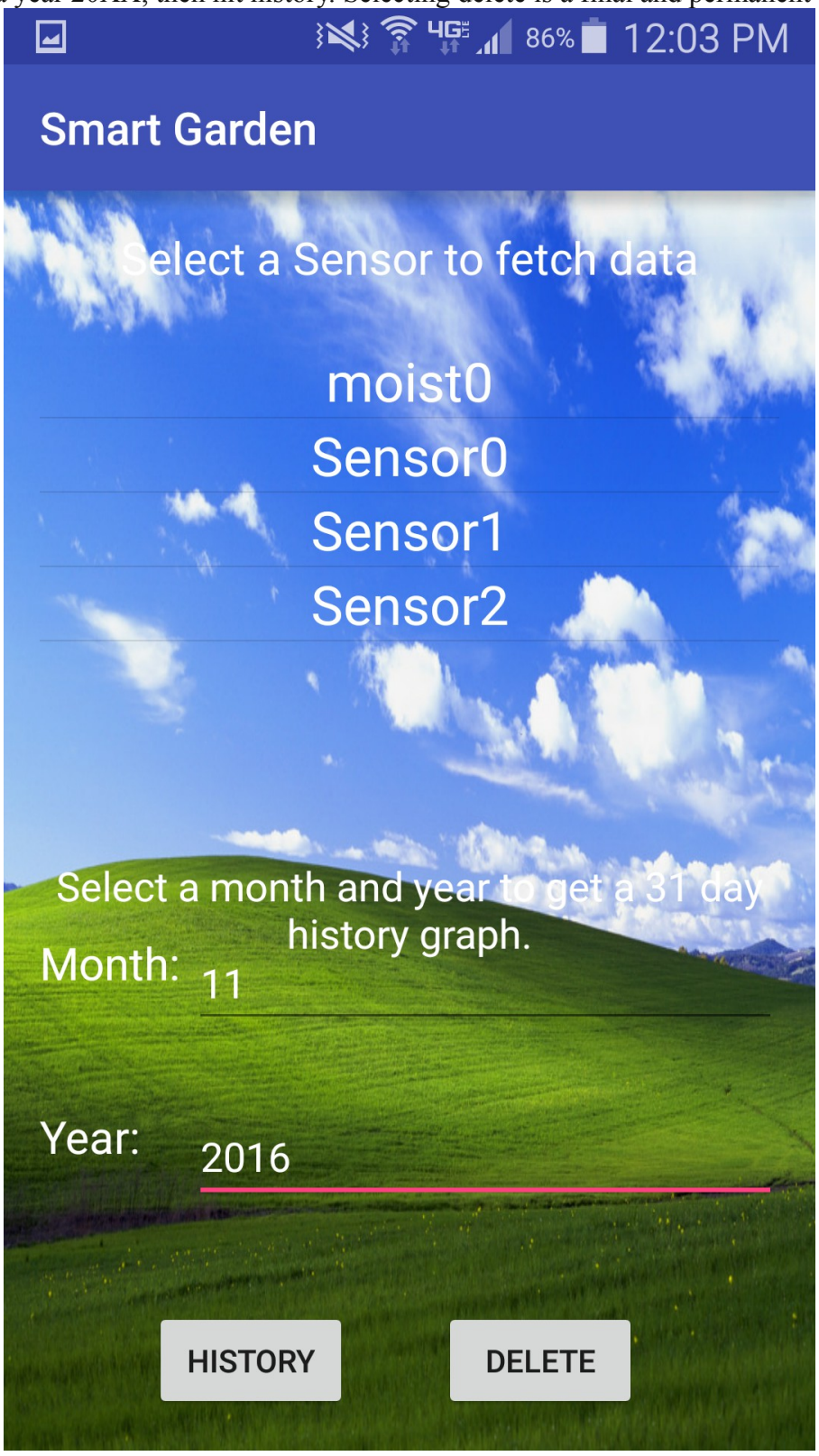
Android OS Application setup

The Smart Garden app is used for creating a profile to store garden metrics to allow users to view the history of the garden's moisture, ambient temperature and humidity. The data is displayed as a 30 day graph that is viewed by month. To use these features an account must first be created. To begin the setup download and install the Smart Garden app, available at this link: *link to .apk*

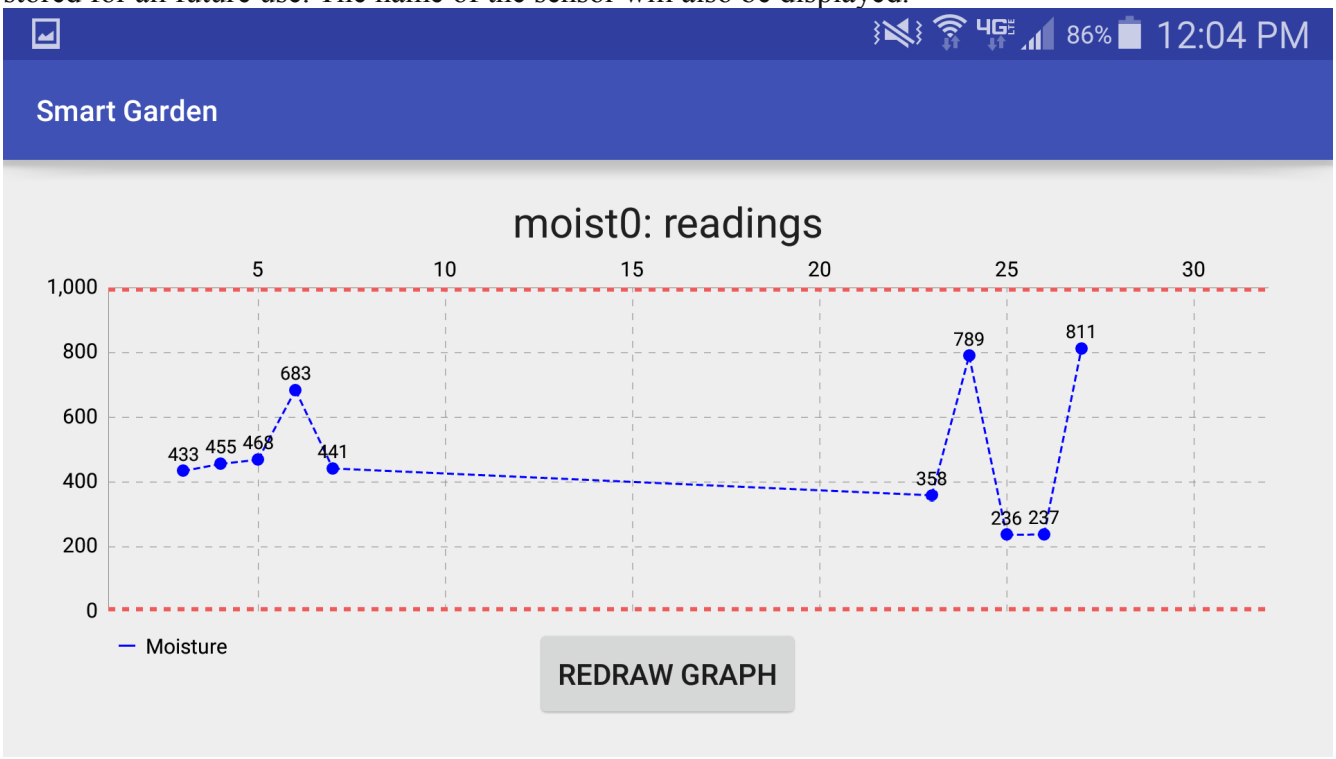
The first screen will be the Login and User Registration. If you are a new user enter the username you desire and password, then click Register. If the registration is successful then you can continue to login with the same information. There is also a link to the Smart Garden website with other helpful material.

Upon successful login you will be brought to the main screen that will display a greeting and the list of currently registered gardens with an add and delete option. There is also a history selection, to select a garden to view sensors for click on a garden name and then click history.

After selecting a garden to view you will be provided with all registered sensors in that garden. Since sensor names will not always reflect which type of sensor they are the graph will either display a data set of the temperature and humidity or just the moisture. Input a month as a number, 11 being November, and a year 20XX, then hit history. Selecting delete is a final and permanent change.



After selecting a month with readings to view a graph will be displayed and have humidity and temperature or moisture readings. It also displays the legend for the line colors. These readings will be stored for all future use. The name of the sensor will also be displayed.

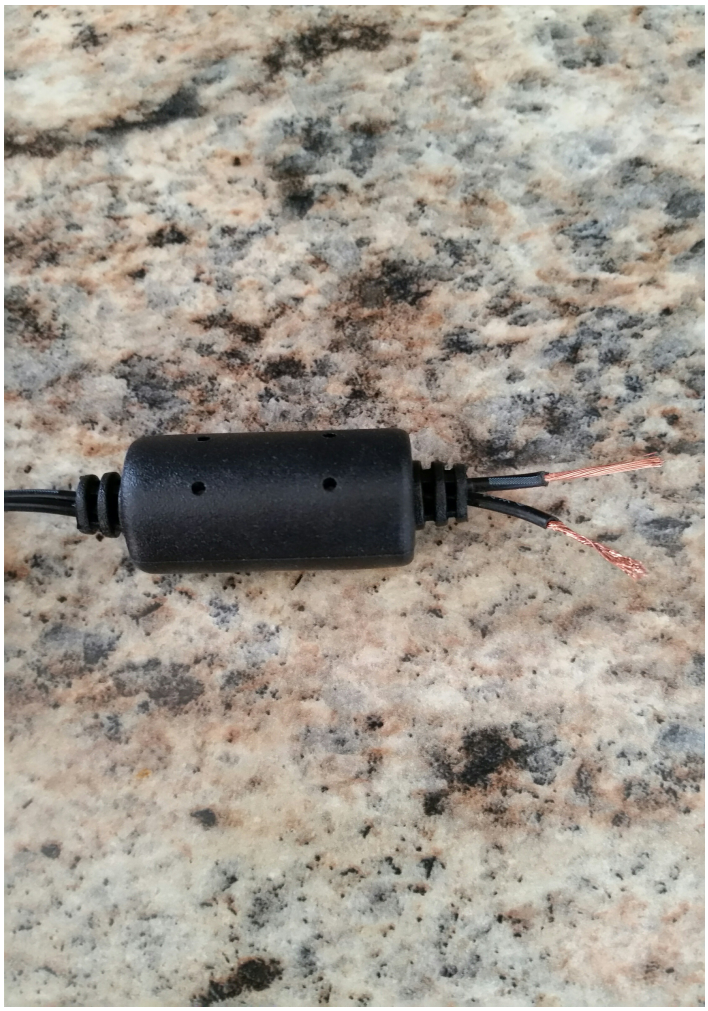


Wiring and Powering Pump

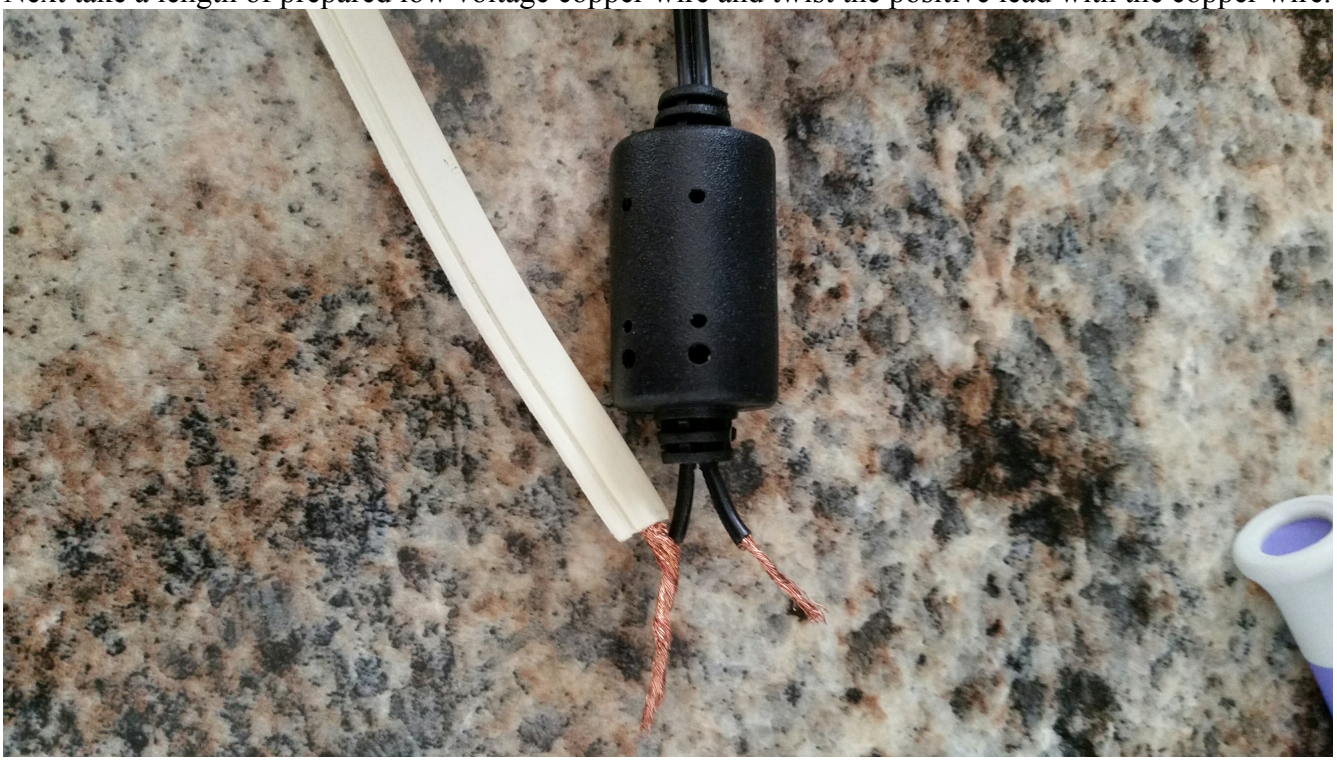
The watering system of the Smart Garden consists of an Electric Solenoid Valve 1/2” DC 12V normally closed inlet flow switch. This solenoid allows a current from a relay to be sent on command from the Raspberry Pi. Here is a full list of required parts for wiring, powering and delivering water to your garden.

- Electric Solenoid Valve 1/2” DC 12V normally closed inlet flow switch
- 12VDC 1A Regulated Switching Power Supply
- Adafruit Power Relay FeatherWing
- Male to Male jumper wires (x)
- Female to Male jumper wires (x)
- 12-10 AWG 10/Clam Female disconnect insulated (2)
- Standard wire connectors (2)
- ½ Vinyl tubing (required Length) (2)
- Female to Female PVC ½” to ½” (2)
- Barbed Male 90* Elbow ½” (2)
- Low voltage copper wire (required Length)

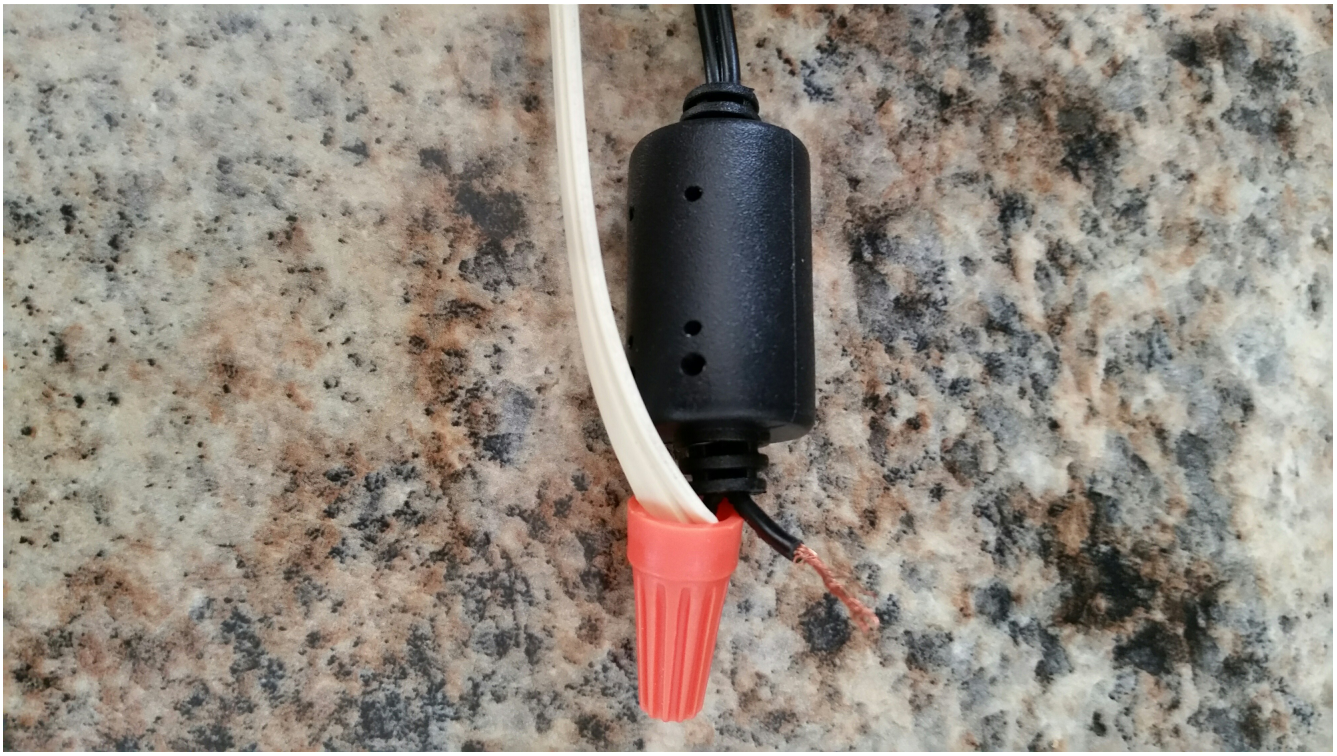
First begin by preparing 3 lengths of low voltage copper wire by stripping both ends on each wire until copper wire is exposed. Next take the 12VDC 1A regulated switching power supply and cut off and strip the end with the cylindrical piece. The side with a grey stripe is the positive end, make note of this.



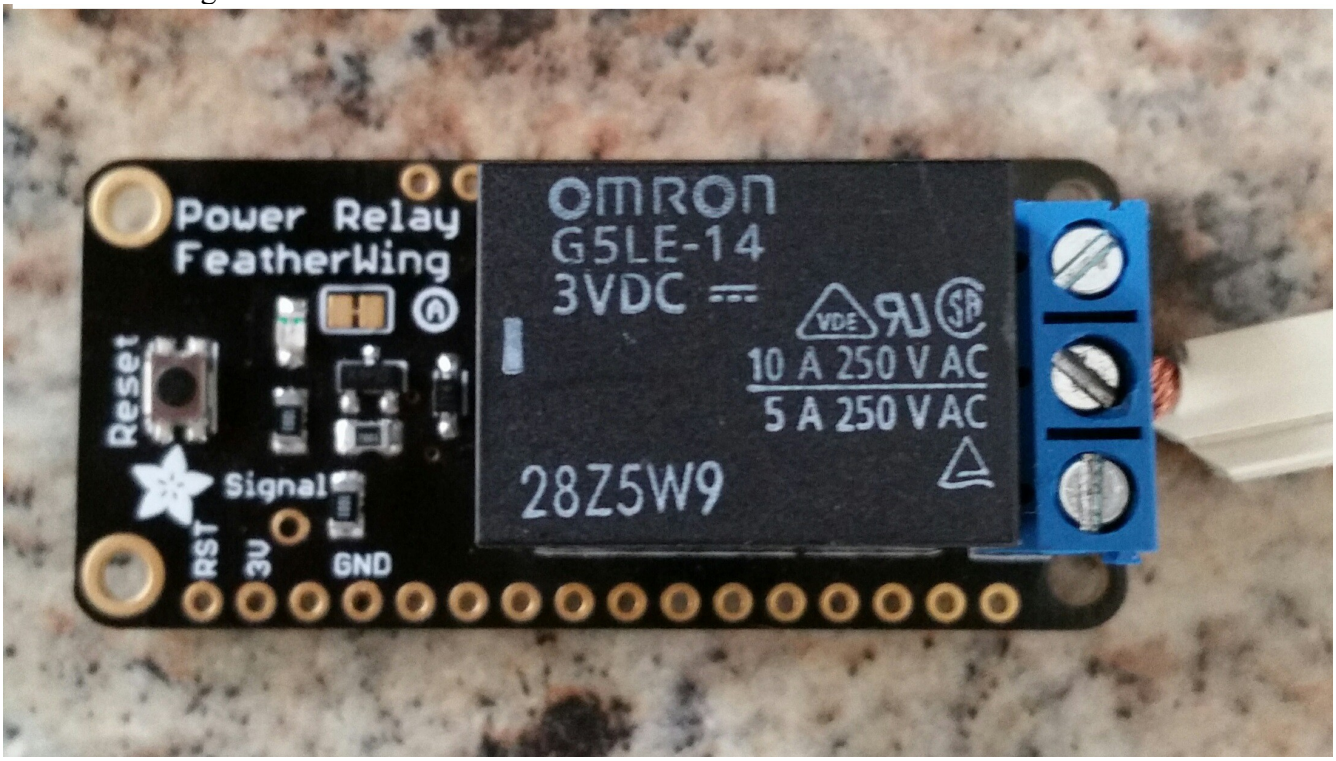
Next take a length of prepared low voltage copper wire and twist the positive lead with the copper wire.



Once that is secure take a standard wire connector and place the exposed wire inside and twist until it feels secure.

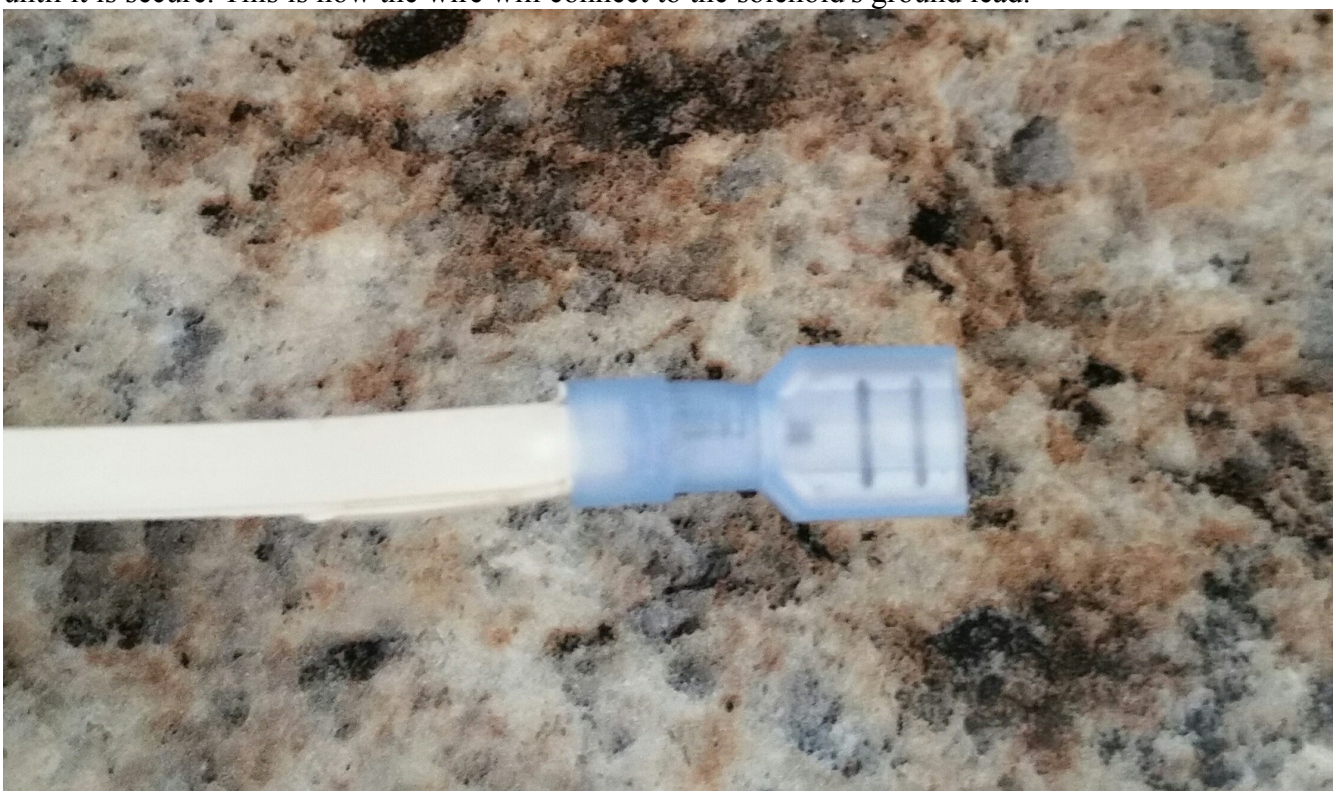


After that you are going to connect the opposite end of the connected wire to the middle section of the Ada Fruit Relay. Place the exposed length of copper inside the middle port of the relay labeled "COM" and secure using a screwdriver.

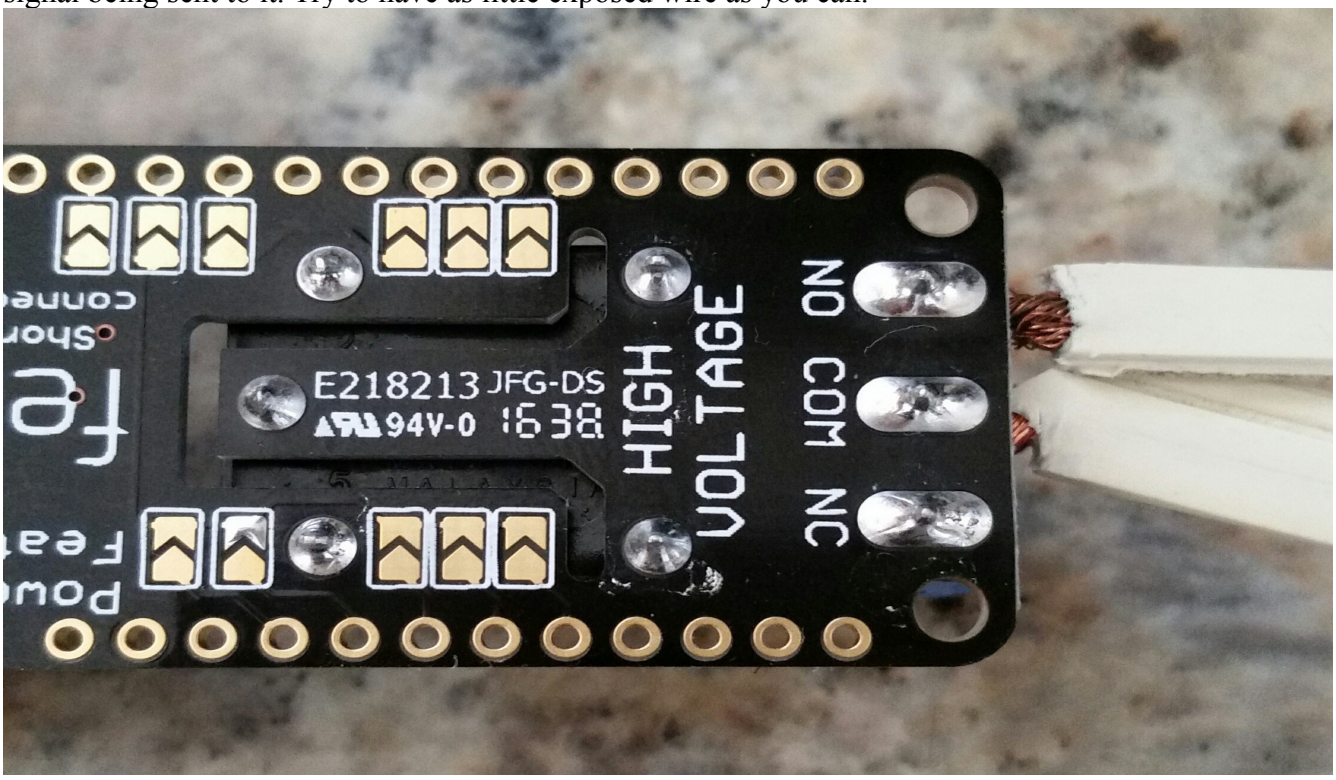


Once that is done take another prepared length of copper wire and twist it in the same fashion as you did with the positive lead on the power supply to the negative lead. Secure this with a standard wire connector. Making note that this will connect to our solenoid's negative lead.

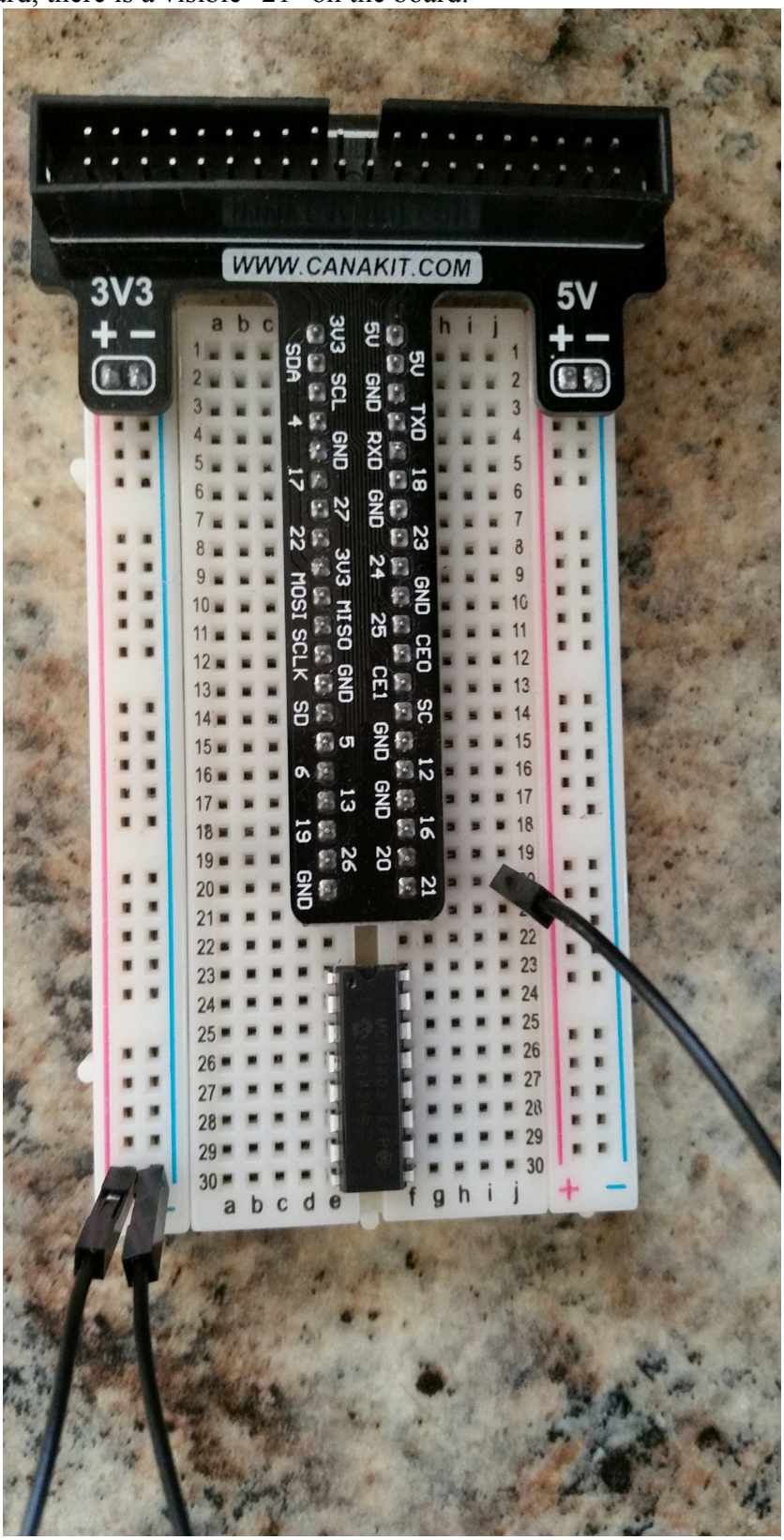
Now, take the copper wire that is connected to the negative lead of the power supply and attach a insulated clam shell disconnect. Place the exposed wire inside of the circular side and twist the wire until it is secure. This is how the wire will connect to the solenoid's ground lead.



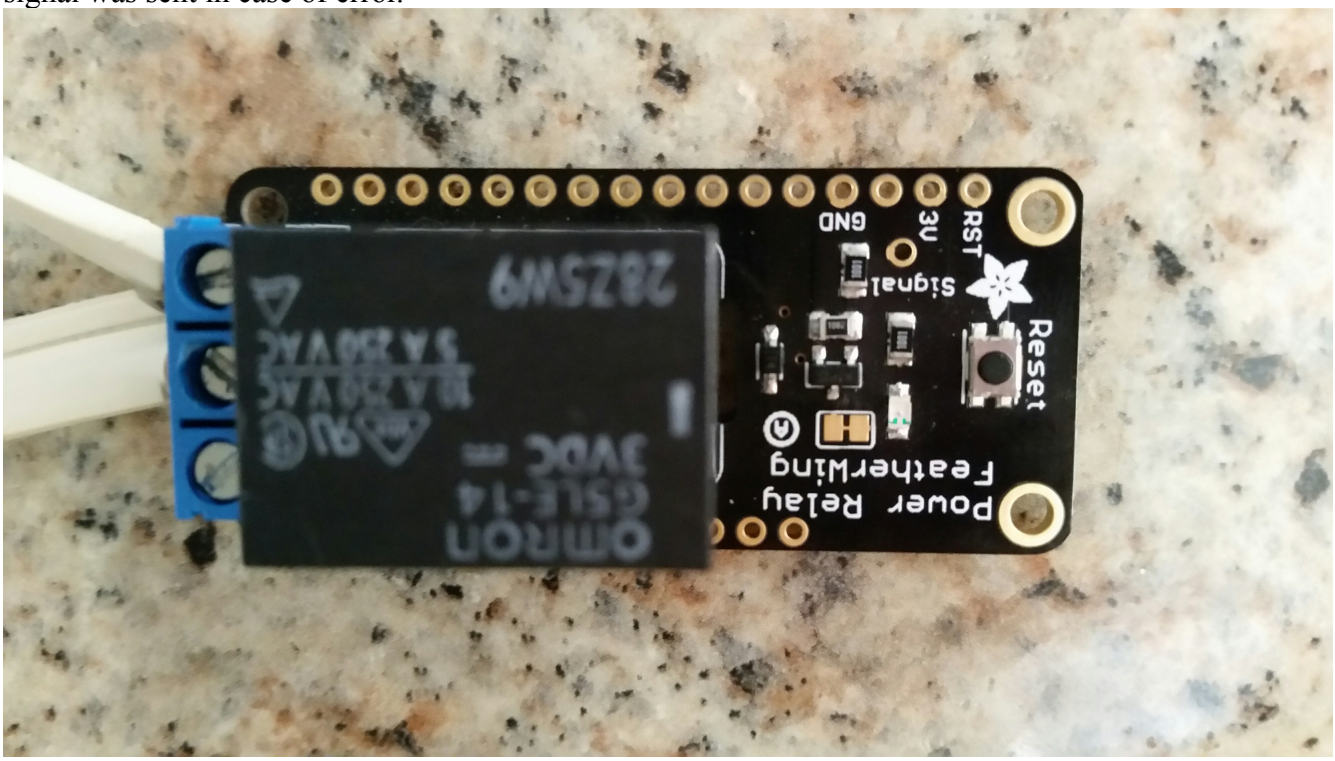
Once that is finished take the last length of prepared copper wire and secure it into the relay port marked "NO", this stands for normally open. This switch will only engage when there is an active signal being sent to it. Try to have as little exposed wire as you can.



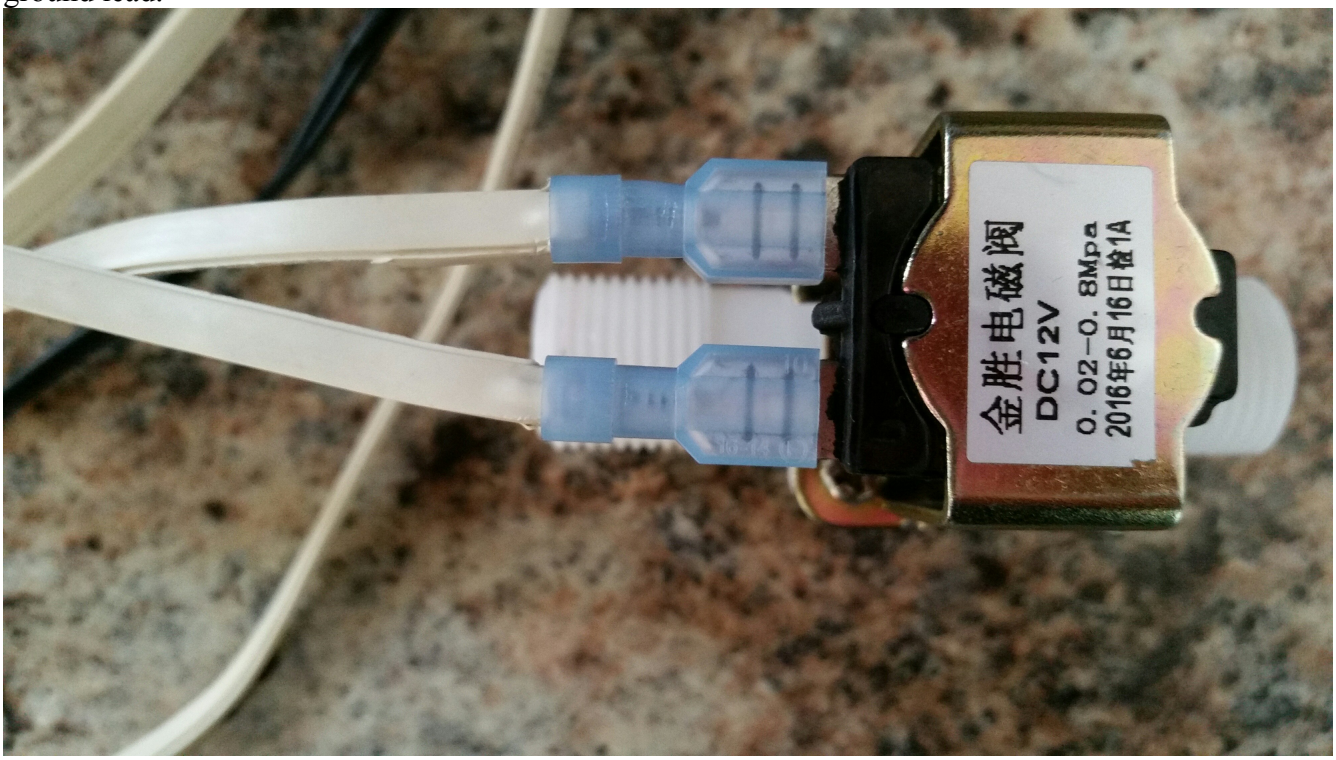
Now we can connect the relay to the breadboard attached to the Raspberry Pi. The relay has a 3.3v positive lead, a ground and a signal pin that need to be connected to the breadboard. The 3.3v wires will be on the left rail of the breadboard. The signal wire will be connected to pin number 21 by default on the breakout board, there is a visible "21" on the board.



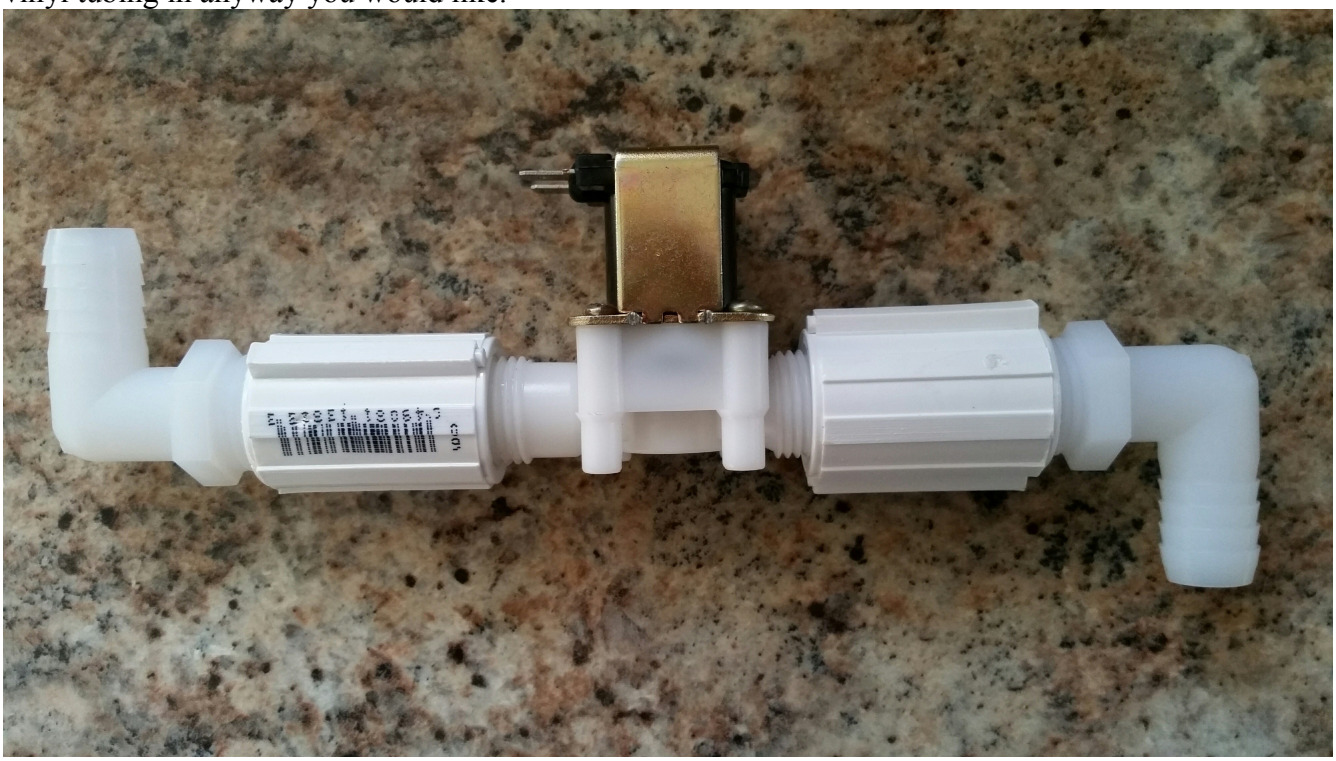
These three wires will connect to the relay to the indicated pins on the relay. You may also choose to solder the jumper wires to the relay if you would like. The reset button will return the relay as if no signal was sent in case of error.



Now prepare this wire's other end with an insulted clam disconnect, this will connect to the positive lead of the solenoid. Make note that there is a barely visible "G" on the bottom terminal indicating the ground lead.



Once that is done you can secure the tubing using the elbow joints and PVC pipes. The solenoid has a indicator arrow on the bottom of the stand which marks which direction water will flow. I have place the elbows in opposite directions for a reservoir and to allow the water flow with gravity. Attach the vinyl tubing in anyway you would like.



Appendix of links

Win32 Disk Imager - **https://sourceforge.net/projects/win32diskimager**.

Raspian Operating System - **https://downloads.raspberrypi.org/raspbian_latest**

Pre-loaded micro SD card with Raspbian - **https://www.adafruit.com/products/1121**.

MCP 3008 Library - https://github.com/adafruit/Adafruit_Python_MCP3008

DHT22/11 Library - https://github.com/adafruit/Adafruit_Python_DHT

Table 1 Pin Function Table MCP3008

PDIP, SOIC	Symbol	Description
1	CH0	Analog Input
2	CH1	Analog Input
3	CH2	Analog Input
4	CH3	Analog Input
5	CH4	Analog Input
6	CH5	Analog Input
7	CH6	Analog Input
8	CH7	Analog Input
9	DGND	Digital ground
10	CS/SHDN	Chip Select Shutdown input
11	Din	Serial Data in
12	Dout	Serial Data out
13	CLK	Serial Clock
14	AGND	Analog Ground
15	Vref	Reference Voltage
16	VDD	+2.7 to 5.5v Power

Digital Ground (DGND)
Digital ground connection to internal digital circuitry
Analog Ground (AGND)
Analog ground connection to internal analog circuitry
Analog Inputs (CH0 - CH7)
Analog inputs for channels 0 - 7, respectively, for the multiplexed inputs.
Serial Clock (CLK)
The SPI clock pin is used to initiate a conversion and clock out each bit of the conversion as it takes place
Serial Data Input (Din)
The SPI port serial data input pin is used to load channel configuration data into the device.
Serial Data Output (Dout)
The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.
Chip Select/ Shutdown (CS/SHDN)
The CS/SHDN pin is used to initiate communication with the device when pulled low. When pulled high, it will end a conversion and put the device in low-power standby. The CS/SHDN pin must be pulled high between conversions.