

Robot Gardener

Control Unit Documentation

Version 3

design by
Samuel Bagot

Table of Contents

What is the OuterBabylon Control Unit.....	3
Chapter 1: Warning and Disclaimer.....	4
Chapter 2: Possible Community Directions.....	5
Chapter 3: Overview.....	6
Chapter 4: Electrical Box Preparation.....	7
Chapter 5: Build the Heavy Current Router.....	9
Chapter 6: Build the Relay Box.....	13
Chapter 7: Build the Arduino Project Box.....	21
Chapter 8: Build the GIF Power Box.....	26
Chapter 9: Modify the Power Strip.....	28
Chapter 10: Build the Sensor Patch Bay.....	34
Chapter 11: Building the Sensors.....	37
Chapter 12: Splicing the Solenoids Onto Plugs.....	40
Chapter 13: Completing the Unit.....	44
Chapter 14: Loading the Arduino Software	45
Chapter 15: Helpful Tools.....	47
Chapter 15: Parts List.....	48
Builder Notes.....	65

What is the OuterBabylon Control Unit

The Control unit is simply a closed box system that controls a list of electrical peripherals by providing a web interface and computer automated programing. We will use this unit to control the watering of a garden with water solenoids and moisture sensors and the temperature with fans and temperature sensors. The control unit is designed to be installed near an outside garden and be weather proof.



Chapter 1: Warning and Disclaimer

Supplemental video: “Chapter 1: Disclaimers and Warnings” at www.YouTube.com

Know:

- Work in progress so documentation is going to be more up to date
- The videos are to help get people started and provide general guidance
- This project deals with dangerous electrical power and should be taken seriously
- This project is for adults only
- All work should be double checked by someone with advanced electrical knowledge of dangerous electrical systems
- Advanced knowledge of wiring and electrical safety is required to complete this project safely
- This project should not be attempted by anyone who is unfamiliar with electrical safety.
- Read the entire document before starting on anything mentioned inside this document.

Chapter 2: Possible Community Directions

A control unit like this one can be used in a variety of environments and has many possible directions. It would be nice to see some future projects based off this one or a part of this implementation. It's versatility comes from it's simple core of functionality. It's most basic accomplishments are the following simple tasks:

- Remotely turn devices on or off manually through a web interface
- Turning devices on or off on preprogrammed intervals
- Turn devices on or off after analysis of analog sensor data
- Provide data output of analog sensors through a rest web interface

Peripherals - This unit can actually be altered to control any number of different types or peripherals. It can be used to control several fluid pumps or more water solenoids. It doesn't have to be used in an outside or garden environment. The concepts and ideas in this project can even be used to control a series of decorative lights or refrigerators in your house. It can be used to automate mechanisms to feed cattle on controlled intervals or remotely on demand through web interfaces or rest APIs.

Sensors – Any number of different sensors can be used. For example, Ph levels are important to gardening and soil. A PH sensor can be utilized to control the injection of chemicals to balance the soil during watering.

Software - Better software can be written with more options on tolerances or to incorporate more sensors. These control units can be made to talk to each other as well to share data or distribute operations. There can also be a central software that can talk to several units. They can be made to scan your home network and identify control units and help conform their configurations as well.

Documentation – This document is periodically updated and revised as typographical errors and unclear sections are identified. Any community help along these lines will be greatly appreciated. Both pdf and doc formats are available for people to download and edit.

Chapter 3: Overview

In this document, we will outline and discuss the basic building of the garden control unit. Below is a big picture outline of what we will detail in this document. There is also companion video to much of this build process at www.RobotGardener.com or www.OuterBabylon.com.

- Building the control unit
- Building the sensors
- Building the solenoids
- Installing the software into the Arduino
- Mounting control unit
- Running the electronics into the garden

Supplemental video: An additional video “Chapter 3: Work Area and Setup” is available on www.YouTube.com which provides information about our workspace setup and how to apply the project template to the project board.

Chapter 4: Electrical Box Preparation

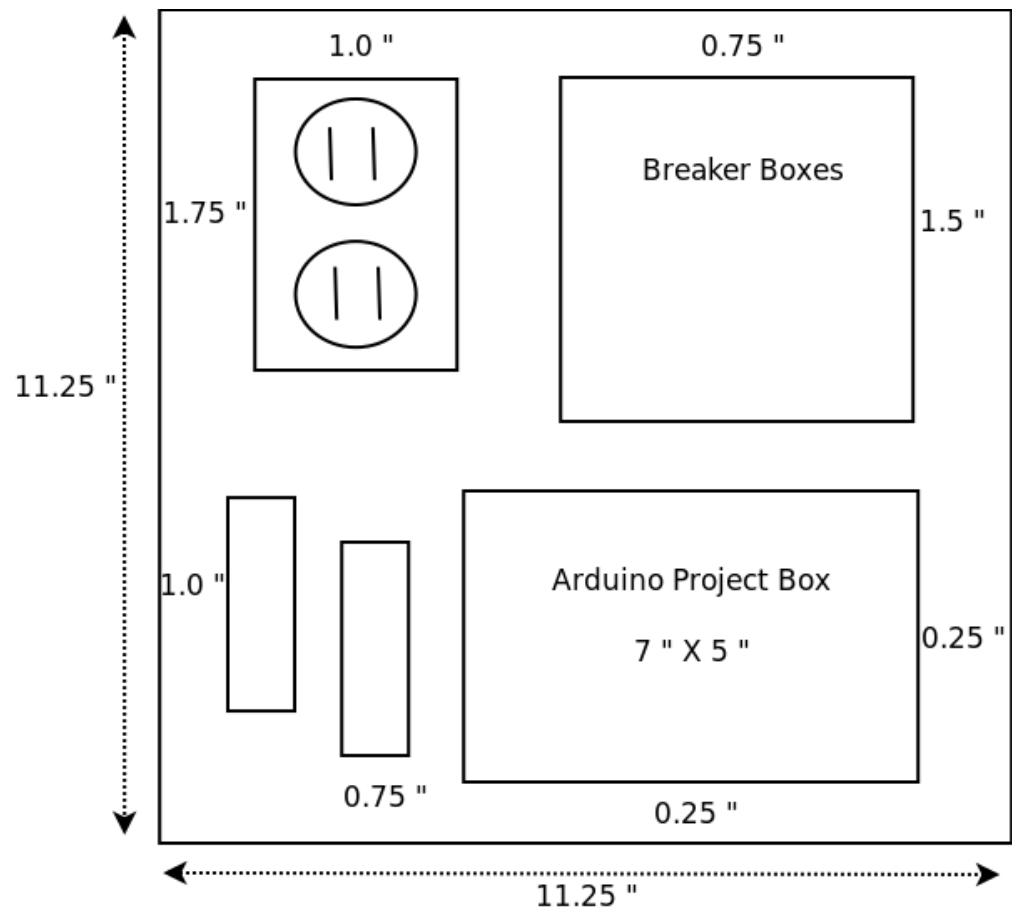
Parts list:

- 1' x 1' Waterproof Electrical Box
- 21/32x12x48 Paint Grade Edge-glued Panel

Cut the board to 11.5" X 11.5".



Draw the template onto the cut board.



Keep in mind that the above template is not completely drawn to scale so that the text measurements can be larger and more readable. The measurements are simply suggested distances from the side of the project board so that all the components will fit in the end.

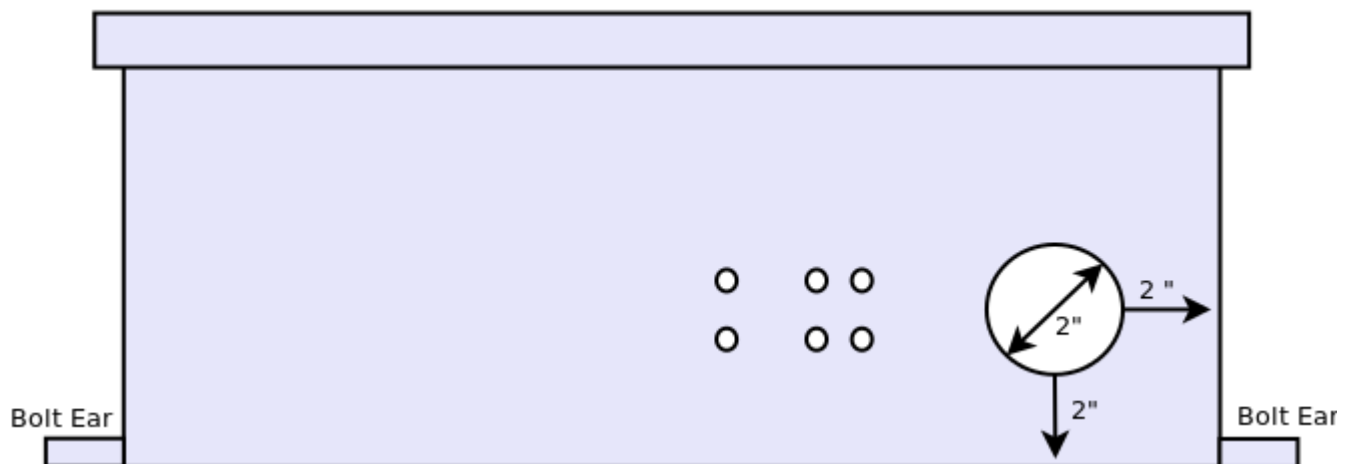
Lay the cut board into the large electrical box once to make sure that every thing is going to fit in the end.

Drill the holes in the box. For this we will need to drill a hole in the bottom of the box that is approximately 2" wide. This will most likely take a special drill bit. All external cables for power and sensors will run through this water-resistant hole.. I positioned the hole so that it has a margin of 1" from the eventual backside of the box and a margin of 1" of the eventual right side.

Also, it is best to make sure that you choose the correct side to be the bottom. So choose a side of the big electrical box that doesn't have a plastic ear for bolt mounting to be the bottom. We will want the two sides with bolt ears to be the left and right sides so it can be mounted to a couple of stakes later.

The smaller holes just need to be in two parallel columns as centered as possible. These smaller holes will be how we will zip tie the incoming and outgoing cables.

Eventual Bottom View



The final box should look similar to the picture. Note the bolt mount ears on sides other than the side you chose to be the bottom.



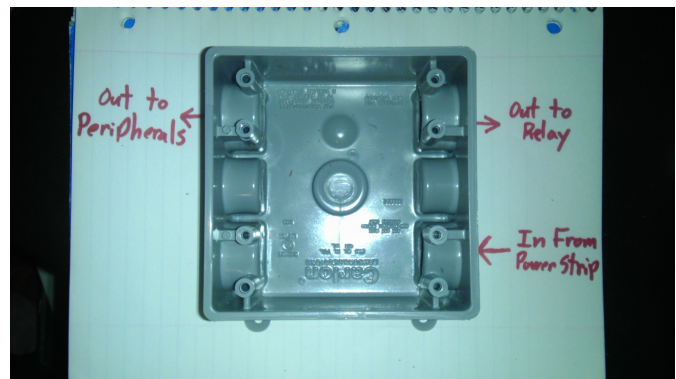
Chapter 5: Build the Heavy Current Router

Supplemental video: “Chapter 5: Build the Heavy Current Router” at www.YouTube.com

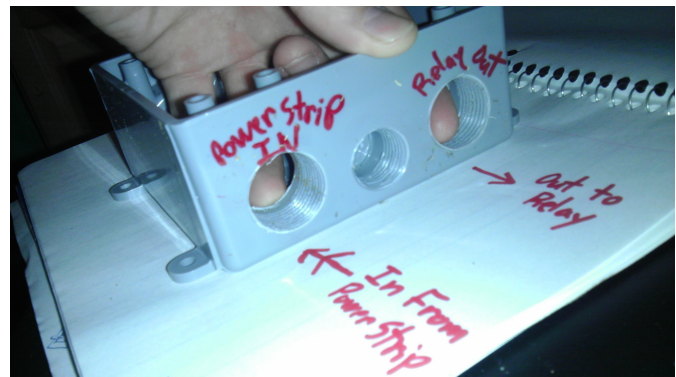
Parts list:

- Breaker Box
- Breaker Box Lid
- Wire Screws
- 4 x Extension cords 15'
- 3 x NM/SEU Cable Connectors 3/4"
- 24V .65A power supply from the garden timer

Mark your breaker box.



Break out the proper holes.



Insert and screw in the cable connectors.



Cut the 4 extension cords:

Chop the four 15' extension cords into three lengths. These three lengths will be an 11' length with the female terminal still attached, a 2' length with two cut wire ends, and a 1' length with the male terminal still attached. Below is a description of the length and what they will be used for.

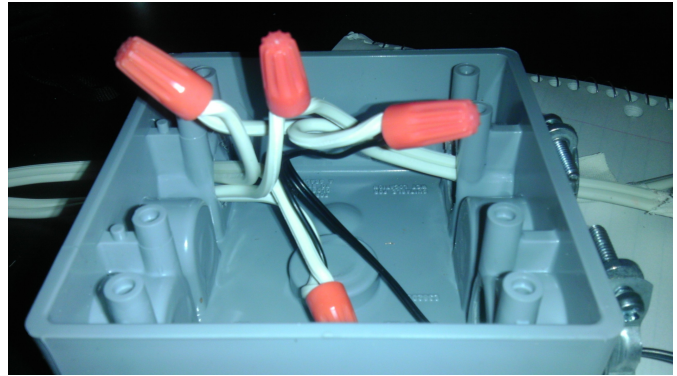


For solenoid 1 peripheral - chopped 11' female length to go to the solenoid 1 outside of the control unit.
 For solenoid 2 peripheral - chopped 11' female length to go to the solenoid 1 outside of the control unit
 For fan peripheral - chopped 11' female length to go to the fan outside of the control unit
 For light/pump peripheral - chopped 11' female length to go to the light or pump outside of the control unit

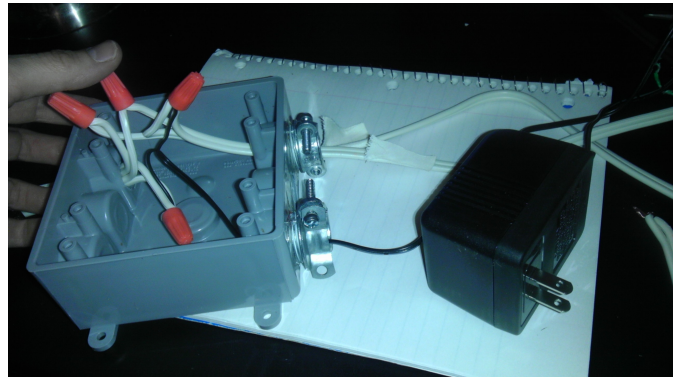
For solenoid 1 relay- chopped 2' cable from high voltage router to relay
 For solenoid 2 relay- chopped 2' cable from high voltage router to relay
 For fan relay- chopped 2' cable from high voltage router to relay
 For light relay- chopped 2' cable from high voltage router to relay

For solenoid 1 splice - chopped 2' male extension to go from power strip to the high voltage router
 For solenoid 2 splice - chopped 2' male extension to go from power strip to the high voltage router

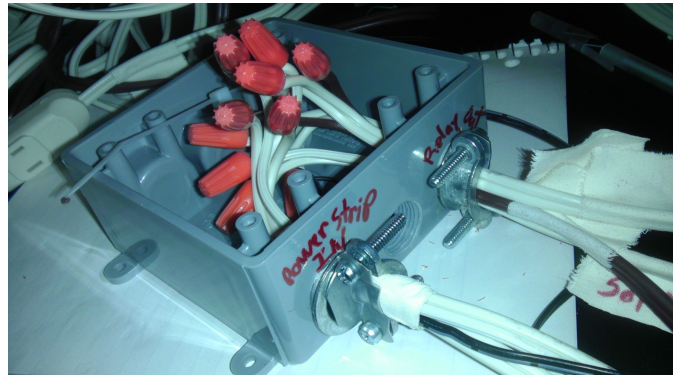
For fans power - chopped 2' male extension to go from power strip to the high voltage router
 For light/pump power - chopped 2' male extension to go from power strip to the high voltage router
 For solenoid power- the 24V .65 Amp converter to go from power strip to the high voltage router



Pull the wires through the appropriate holes in the high voltage router and connect them using the wiring diagram. Start with the solenoids. Remember that some wire nuts involving the solenoids will have three wires and all others will have two. This is because both solenoids will branch off each of the wires from the power supply. Make sure that the wire nuts are tight and that no bare wire is visible around the bottom edges of the wire nuts once they are on. Shorts can not be risked in this portion of the construction.



Continue with the same methodology of connection the wires with wire nuts to accomplish the wiring diagram for the high voltage router. Complete the light circuit and then the fan circuit.



Now to test the high voltage router, you must temporally cap off each of the four cable pairs coming out that will go to the relay shield from the port labeled "relay out". This will simulate all of the relays being turned on and closing their corresponding circuits. Again this allows all electricity to flow through the circuit and this needs to be done on a clean non-electrically conductive table with hand and tools away from the high voltage router.

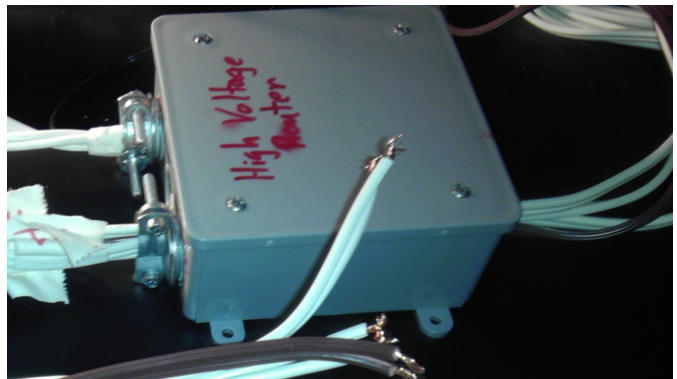


This test requires that you have a power strip with the power control in the off position and the high voltage router completely closed with its breaker box lid. You will plug in a light bulb into a peripherals female extension coming out of the router and plug the male counter part into the power switch. With your hand away from the work area, you will turn the power strip on to see if electricity goes to the light bulb. If it does then repeat the test for the next peripheral until they are each tested.

If one peripheral route doesn't produce the light turning on, then everything must be unplugged from power sources and male and female ends and the high voltage router must be opened and the circuit rebuilt. Remember that the solenoid tests should produce a light from the bulb that is significantly weaker since the amperage of the solenoid power supply is much lower than the other peripherals.



Screw close the breaker box.



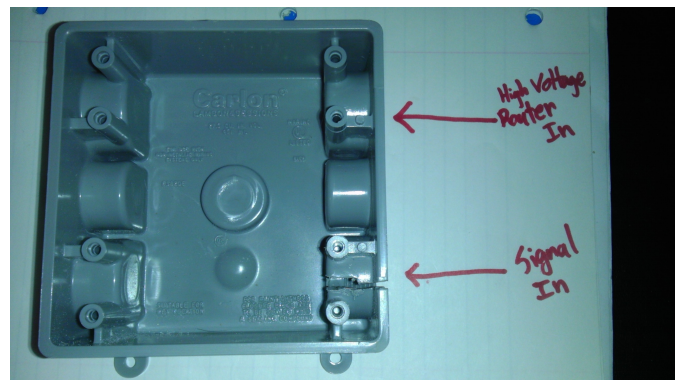
Chapter 6: Build the Relay Box

Supplemental video: “Chapter 6: Build the Relay Box” at www.YouTube.com

Parts list:

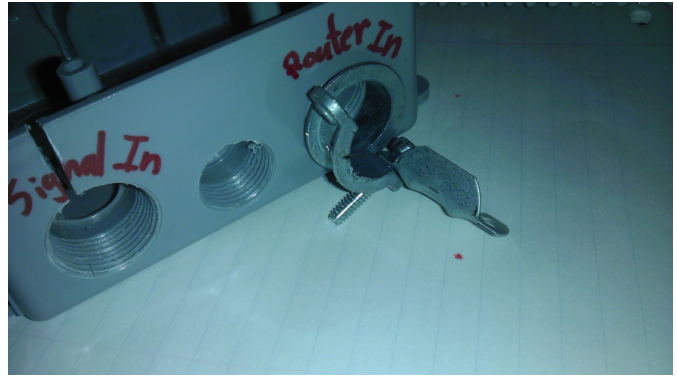
- Breaker Box
- Breaker Box Lid
- Seed Studio Relay Shield
- Signal Wiring
- 3 x NM/SEU Cable Connectors 3/4”
- 2 x longer wood screws

Prepare the router box by labeling one port “high voltage in” and one port “ signal in” as indicated in the picture below. Break out these two port holes so that you can later route wire through them.



Insert and screw in the NM/SEU cable connectors into the port marked “high voltage router”.



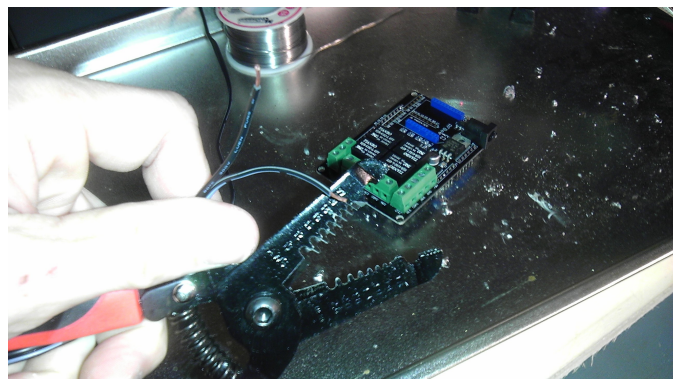
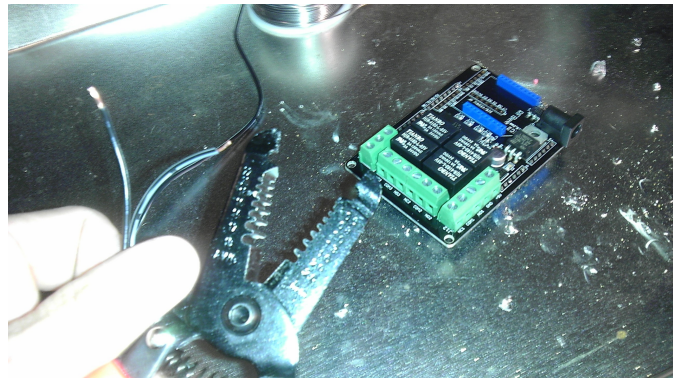


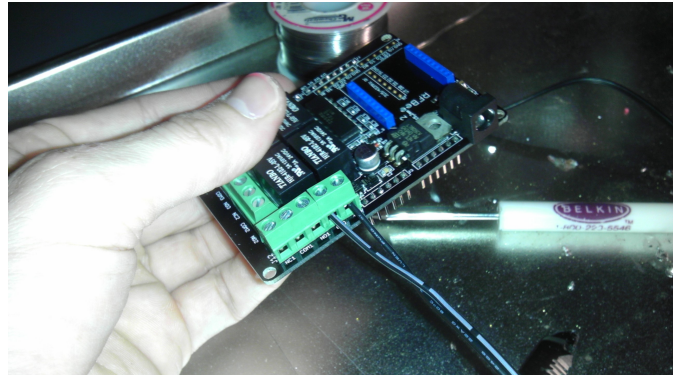
Set the breaker box aside.

Strip off a two small wire lines that can carry a 9V and Ground current.

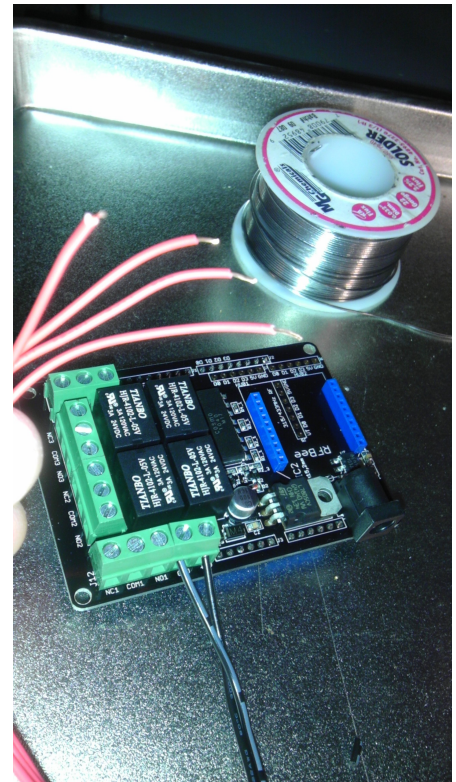
Attach one, preferably a wire with a line or dashed line down it's side indication ground, to the Ground terminal on the relay shield.

Attach one, preferably a wire without a line or mark on its side, to the 9V terminal. This will power the relay shield. The other end of this wire will run to the signal patch bay in a future step to supply power from the Arduino power supply, through the Vin power circuit, so that it doesn't have to be pulled from the power being used to run the Arduino itself.





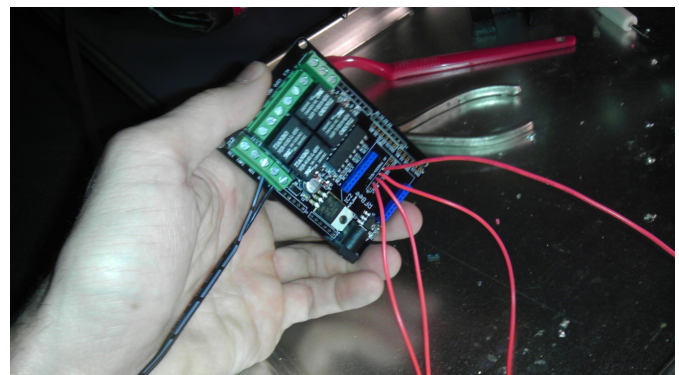
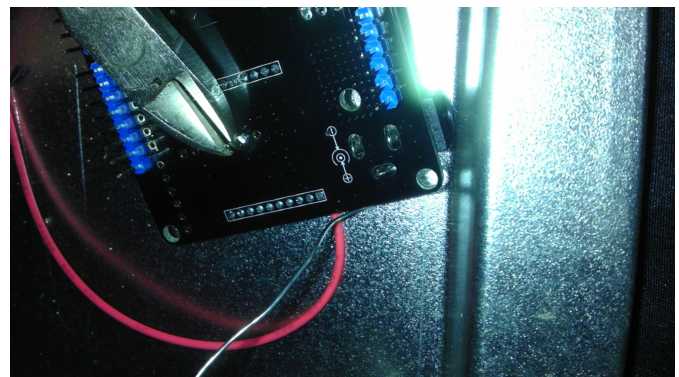
Strip four more signal cable wires so that they can be soldered to the relay shield on one side and soldered onto the Arduino on the other side.



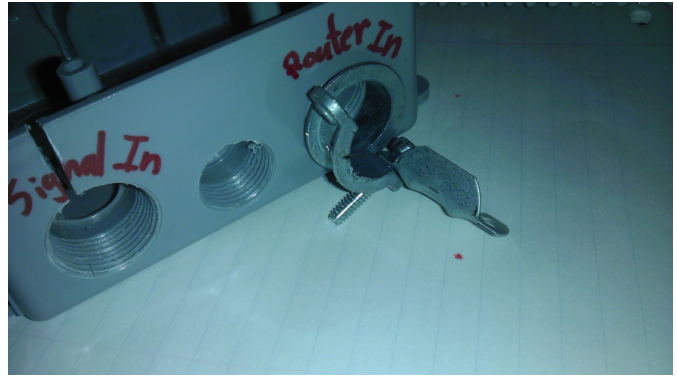
Attach the signal cables by soldering four signal wires onto the relay shield board as show in the positions labeled D0, D1, D2, and D3. Make sure that these are about a foot long each because they will need to go out of the relay shield box and into the Arduino and have some space to be soldered onto the Arduino.



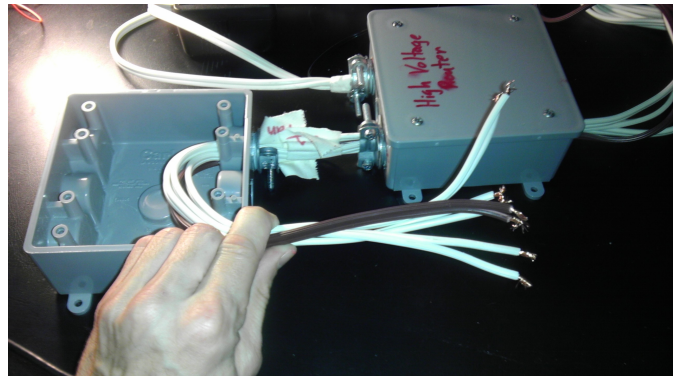
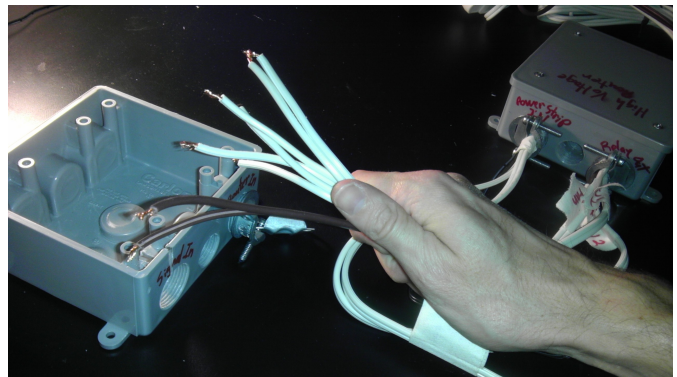
Clean up any tall or misshapen solder connections on the back to prevent any shorts or misdirection of electricity.



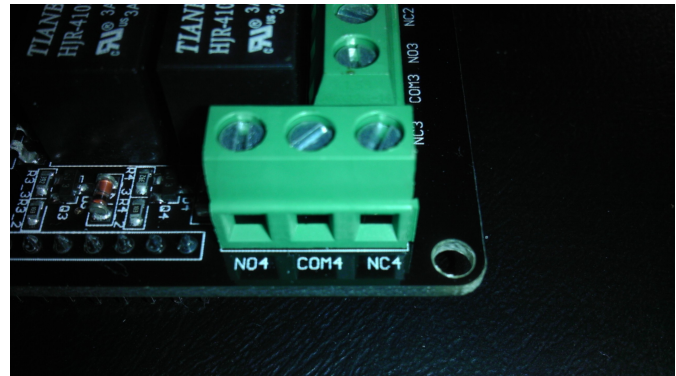
In my example I cut a small groove through the breaker box into the port for the signals so that I can have more options on routing these signal wires and power wires out to the Arduino and the patch bay. Just keep it in mind.



Route the high voltage relay lines from the high voltage router through the appropriate port on the relay shield breaker box. Do not tighten down the cable connector clamp on the relay shield breaker box until the relay lines are later secured in a relay terminal.



This is a relay terminal port. Observe it closely and note what each port on the terminal is labeled as. We are going to fully avoid any port with “NC” in the label.

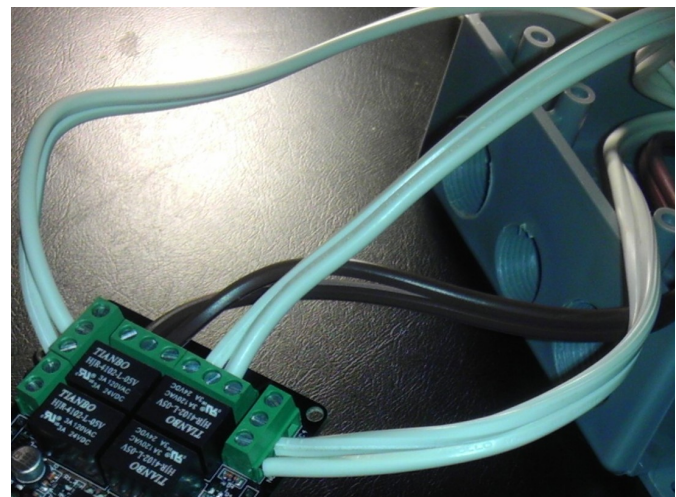


Below is how we will route the peripherals into the relay shield terminals.

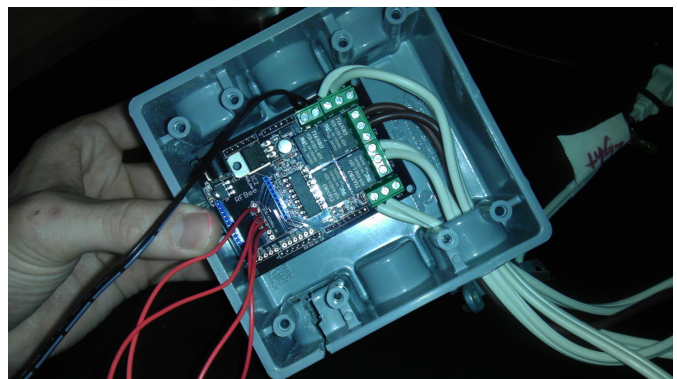
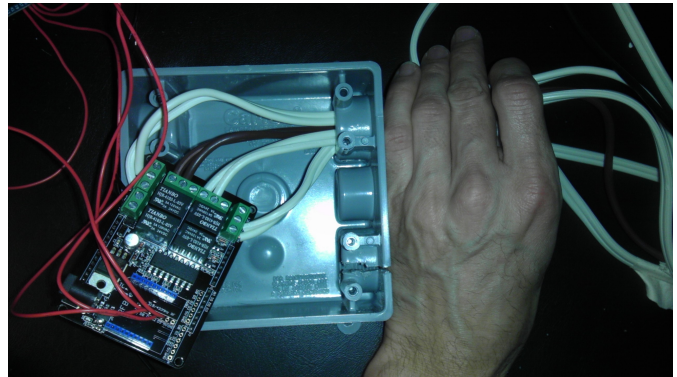
	Arduino Digital Pin Out	Relay Shield Signal Port	Relay Shield Terminal
Sol 1	31	D0	COM1
Sol 2	33	D1	COM2
Light	35	D2	COM3
Fan	37	D3	COM4

Secure each of the four pairs of wires from the high voltage router into a terminals as indicated by the table above..
IMPORTANT: NEVER CONNECT ANYTHING TO ANY RELAY TERMINAL PORTS HAVING “NC” IN THEIR LABEL. NC means “normally closed” and allows electricity to travel through the circuit when the relay is not engaged. This is definitely not what we want and must be insured not to happen. Once each of the four wire sets is secured into a terminal set, go back through and look closely to make sure that you didn't accidentally attach any wires to any ports having “NC” in their labels. The proper two ports to use on any given terminal are the “COM” and “NO”. NO means “normally open” and only allows electricity to travel through the circuit when you have properly engaged its corresponding relay. This is what we want.

Once all four of the wire pairs from the high voltage router are properly secured into “NO” and “COM” ports of relay terminals, check to see the on “NC” terminal ports were used and make sure that each of the terminal port screws are as tight as you can get them without breaking the port.



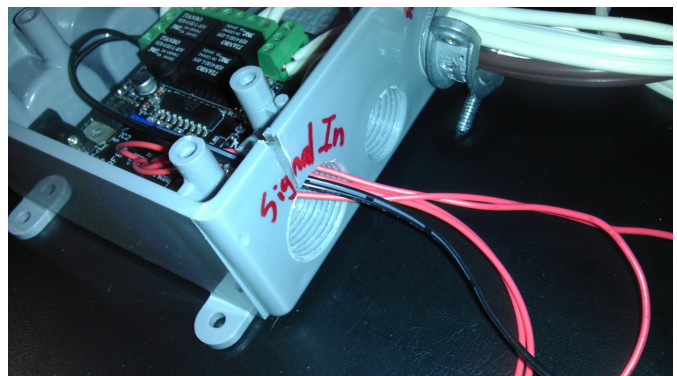
Then, lightly pull the wire pairs back through the breaker box port until a comfortable amount of wire is left inside the relay breaker box. Then tighten down the cable connector clamp to keep the wires in place inside the relay shield breaker box.



You may want to put a small screw into the relay shield through one of the proper holes so that it doesn't move around but I leave mine floating.

Now route the signal and power wires out of the relay shield breaker box and close the lid.

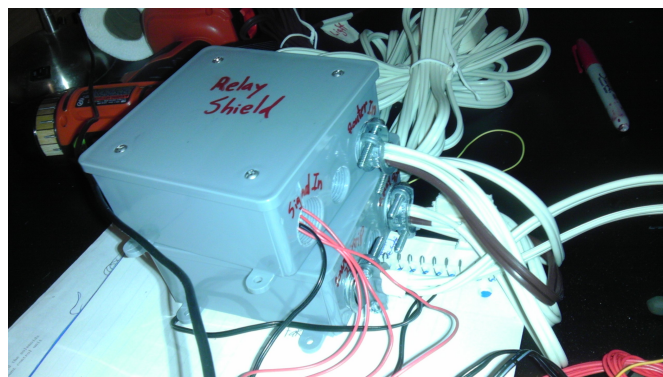
TODO: write more on the testing of the relay shield.



Test the relay shield with the high voltage router and the Arduino plugged in. Make sure that your signal paths and high voltage routing through the relay shield is proper to produce the proper relay reactions listed in the chart below.

Once everything checks out. Mount the relay shield on top of the heavy current router and to the project board. Do this by placing 2 screws through opposing corners of the breaker boxes. The same screw will go through the mounting hole on the top relay shield breaker box, through the same mounting hole on the heavy current router breaker box, and then into your

project board into it's appropriately marked spot from the template.



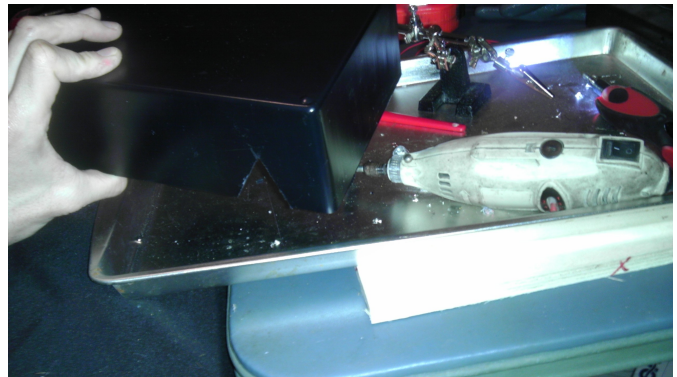
Chapter 7: Build the Arduino Project Box

Supplemental video: “Chapter 7: Build the Arduino Project Box” at www.YouTube.com

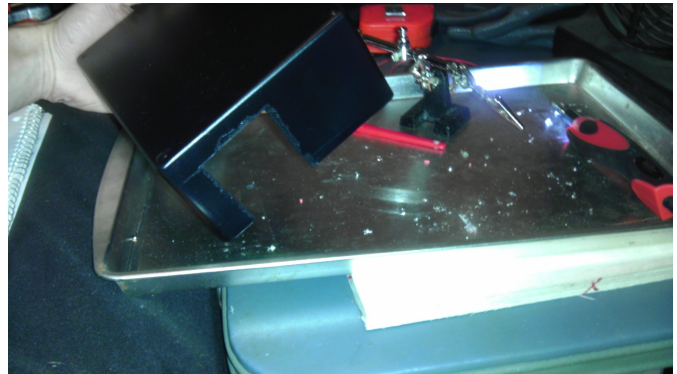
Parts list:

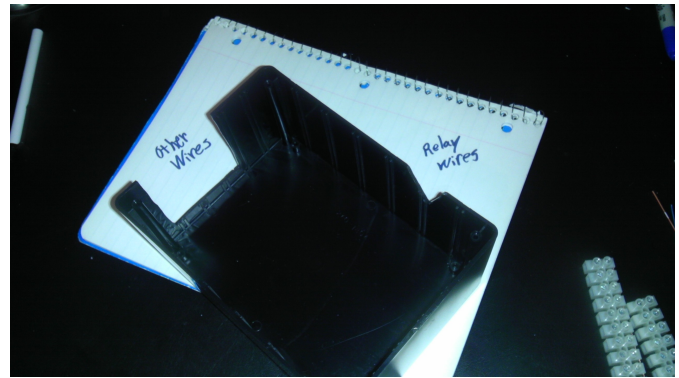
- Electronics Project Box
- Arduino
- Signal wire
- Spacer for Arduino
- 2 x small wood screw
- 9V .65A Arduino power supply

Cut a hole towards one side for the relay signals to be pulled through. The suggested cutting pattern is depicted below.

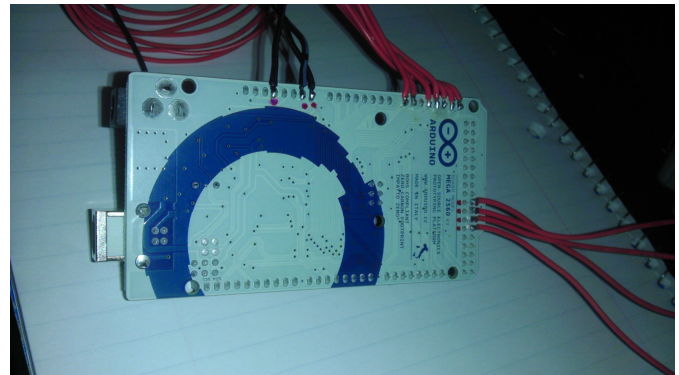


Cut another larger hold at one end for the power wires, sensor wires, Ethernet cable, Arduino power cord, and USB computer cord. The suggested cutting pattern is depicted below.



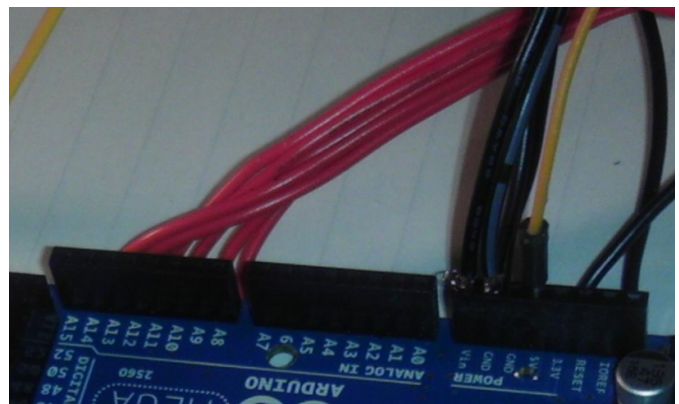


Next we are going to solder the sensors wires, power wires, and the digital out wires that go to the relay shield. We'll end up with something like this:



So, next, you should cut 7 one foot lengths of the stranded signal cable for the sensors. This will include 5 signals for the moisture sensors and 2 signals for the temperature sensors. These wires are going to be ran to the signal patch bay in the future once it's completed.

Also, you will want to cut 3 more one foot lengths of the stranded signal cable for routing some low amperage electricity to the signal bay as well. These wires include a 3.3V to fuel the temperature sensors, a 5V to fuel the moisture sensors, and a ground for many components.

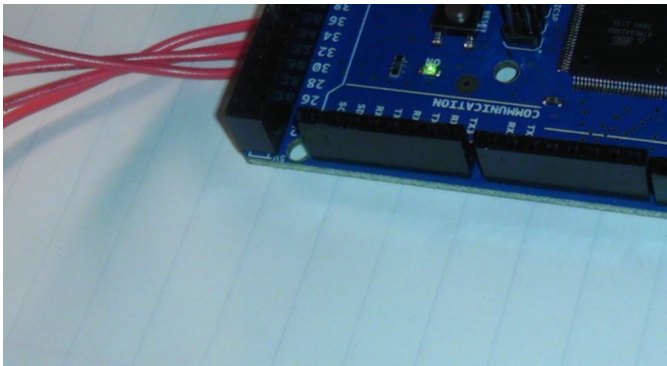


Begin by heating up your soldering iron and soldering the 7 sensor signal wires to the bottom of the Arduino. The pins that this project uses are analog output pins. The 2 for the temperature sensors are A8 and A9. The pins for the moisture sensors are pins A11, A12, A13, A14, and A15. Don't hold the soldering iron to the metal for more than three or four seconds without pulling it away and letting the board cool. It's pretty easy to burn a PCB board like an Arduino.

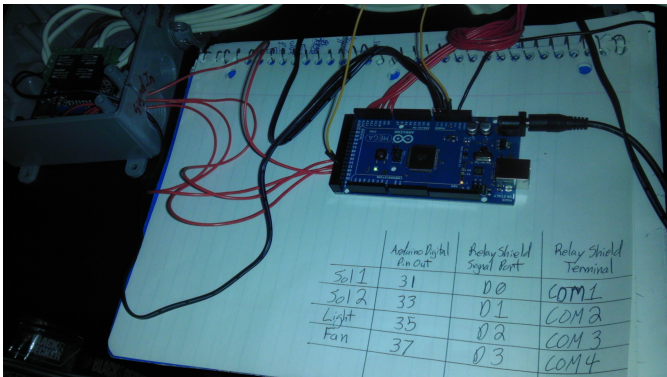
If you are uncomfortable soldering onto your actual Arduino board then you can also place the wires into the top slots of the Arduino. I did this for some of my first models, but the connections can sometimes wiggle out and are generally less stable.

Repeat this process to attach the 3.3V, 5V, and ground wires onto the Arduino board.

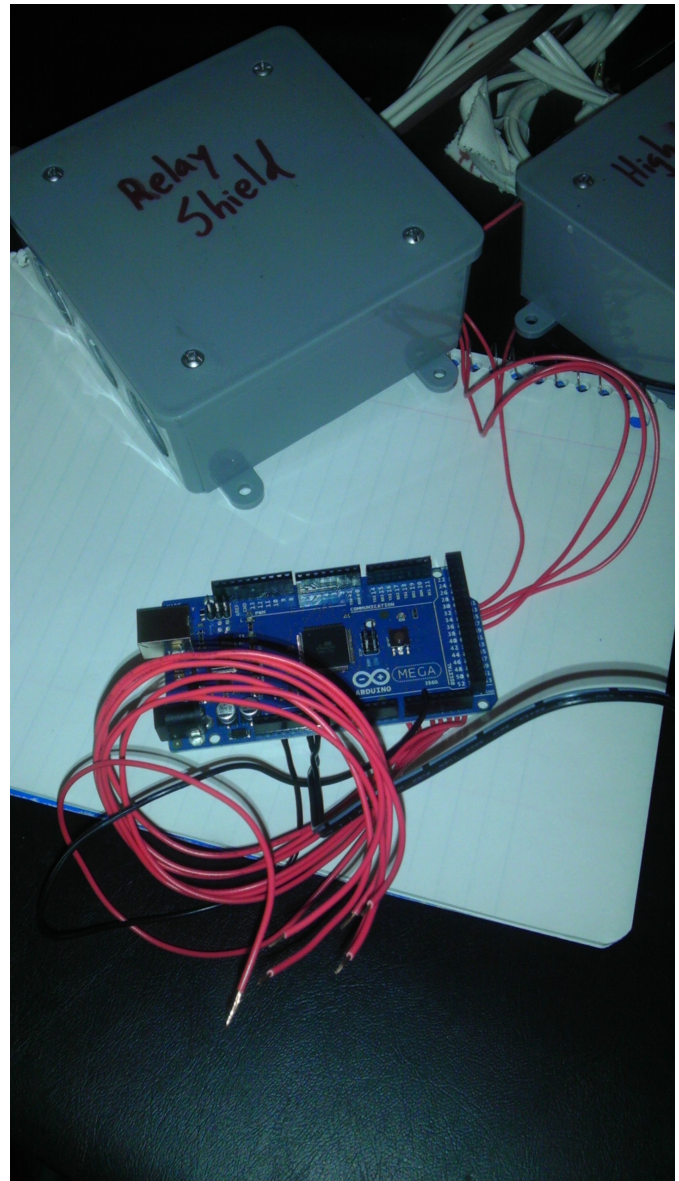
Next you will want to take our previously completed relay shield with its four relay signal wires and attach them in the same way to your Arduino board. The attachment pattern is shown in a table below. Attaching them in this configuration will ensure that community code can control your smart garden correctly.



Make sure that you run the relay shield signals out of the appropriate port on the relay shield breaker box. This is hard to reverse once the signal lines are soldered to the Arduino board. If you have cut a small slot from the top of your relay shield breaker box down into the relay signal port on that box, then you can later slide these new cables into place.

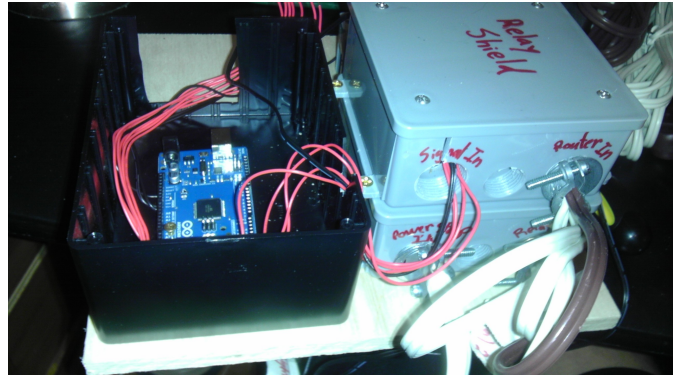


Peripheral	Arduino Digital Input Pin	Relay Shield Signal Port
Solnoid 1	31	D0
Solnoid 2	33	D1
Light	35	D2
Fan	37	D3

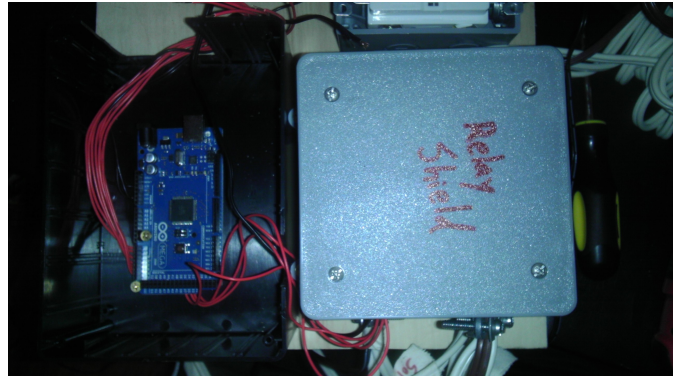


Put spacers of some sort under the Arduino mega so that it doesn't rest on the bottom of the project box.

Take a small wood screw and tack down the Arduino to the project box and the project board. Make sure that you do this through the small mounting hole that is cut in the back of the Arduino board closest to the analog signal ports. This is because there are more critical components close to some of the other mounting holes on the Arduino Mega, and we don't want to short or disrupt those components.



TODO: Test the Arduino and the relay shield setup.



Keep track of the lid to the box and the screws that will hold it in place.

Chapter 8: Build the GIF Power Box

Supplemental video: “Chapter 8: Build the GIF Power Box” at www.YouTube.com

Parts list:

- GFI power socket
- small one socket electrical box
- Long outdoor extension cord
- NM cable Connector Clamp 3/8” trade size 1/2” knockout
- small wood screw

Some how I didn't get decent pictures of this process through any of my builds, but this will be the final product. There is nice video help with this if you want more direction of course.



Chop the big extension cords female side off with a foot left on the female end. This should produce a really long cord with a male end. The GFI outlet will become the new female end of this extension cord.

Break out a port hole on the single socket electrical box for the chopped end of the extension cord to enter. Preferably the port that you want to break out is one on the top or bottom so that the cord will have a little more space to enter the single socket electrical box. Add the NM cable connector clamp into the port that you just broke out.

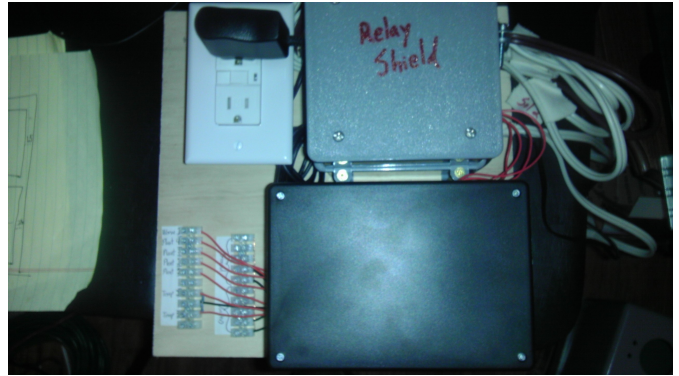
Run the cut end of the long extension cord into the power box through the broken out port hole. Add GIF to the end of the cut cord by inserting the colored individual wires from the cord into their appropriately marked ports on the back of the GFI unit.

Mount this single socket electrical box to the board with a small wood screw according to the board template that we have drawn onto the board. I found it pretty easy just to drill the screw through the back of the socket box and then into the board.

Mount the GFI into the single socket electrical box with the two screws that are based in the GFI. Then add face plate to the front of the GFI unit and it should start looking more like a wall socket.

Now secure the input power cord in the input port by clamping down the NM cable connector. The point of this of course is to make sure that nothing that pulls on the cord will pull on the GFI connections into the single socket electrical box.

Press the test button on the front of the GFI.



Chapter 9: Modify the Power Strip

Supplemental video: “Chapter 9: Modify the Power Strip” at www.YouTube.com

Parts list:

- Power Strip

Warning:

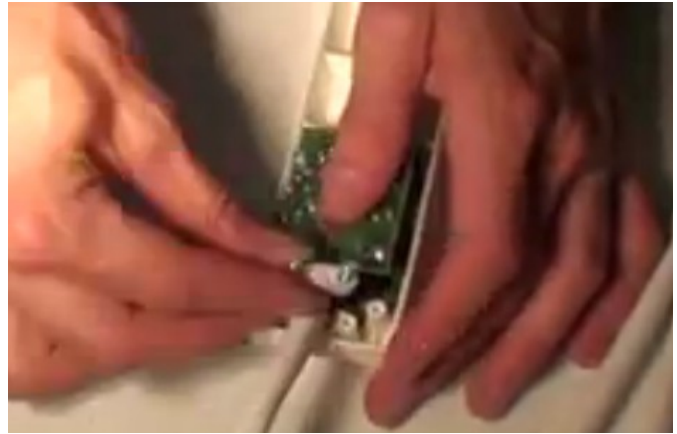
This part of the project is simply to make the cord shorter on the power supply. It is not completely necessary as the extra cord can hang outside the box through the wire input port on the bottom of the box. This modification must be done correctly. Please read through the section of this document called “Chapter 1: Disclaimers and Warnings” and review the supplemental video that goes along with that chapter.

Make sure the power strip is not plugged in for this section.

Open up the power strip by unscrewing the backing screws and popping the backing plate off.



Remove the screws and collar that hold the input power cord in place



Pull out the input power cord and remove the guts of the power strip.



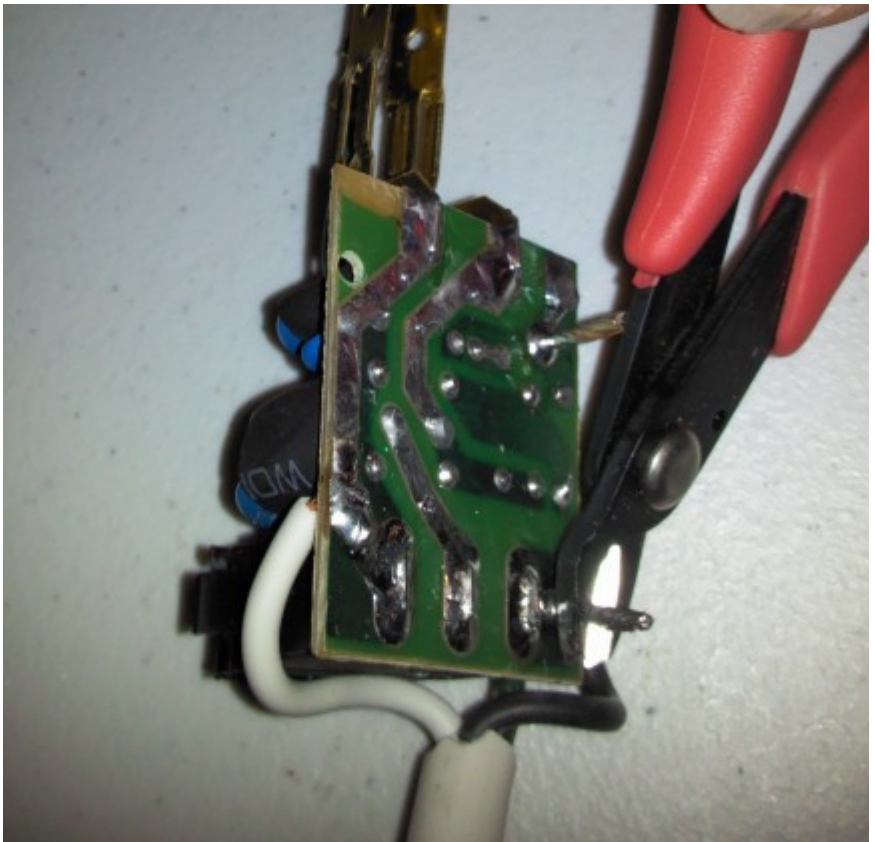
Make sure to take a picture of all angles of the soldered connections inside the power strip so that you can solder new connections appropriately for your model of power strip. This is important and confusion here will produce shorts and potentially dangerous results.



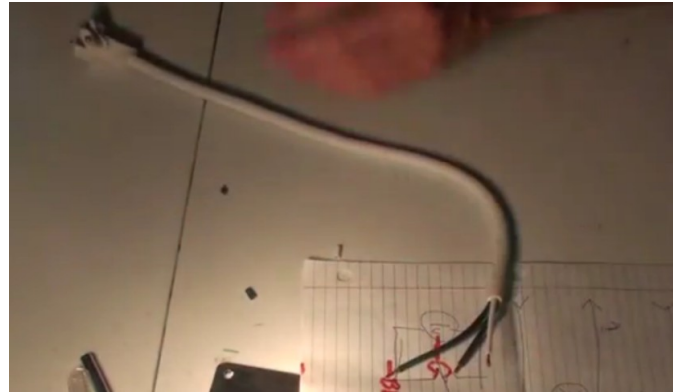
With a heated soldering iron, remove the soldered leads that connect the power cord to the power strips internal circuitry.



Trim any extra solder from the back of the guts to prevent shorts.



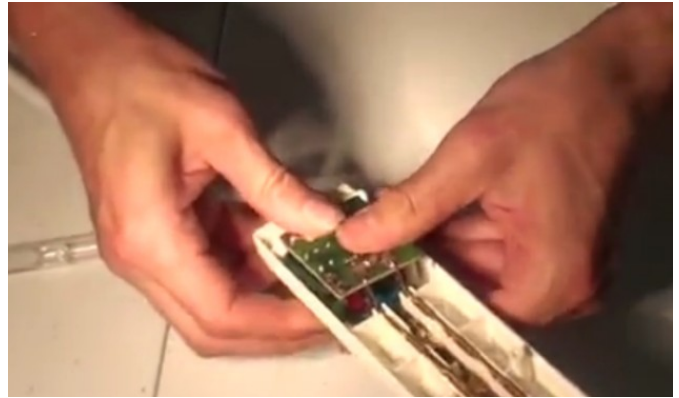
Once the cord is removed, cut it to about a length of two feet with the male connection end still attached to the two foot length.



Consult your picture of the complete unmodified power strip to reattach by soldering the shorter cord.



Replace the guts into the power strip frame.

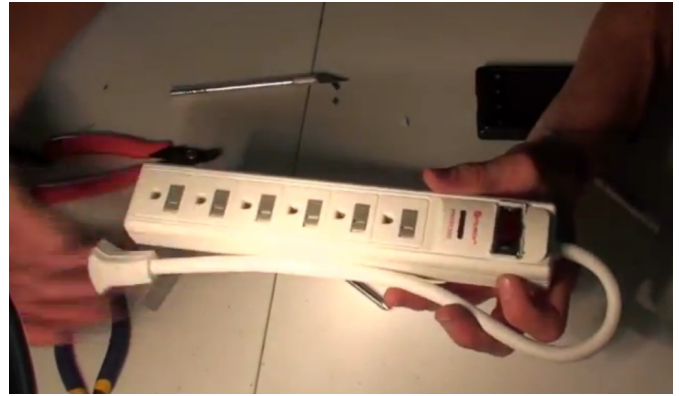


Replace the screws and collar that held the input power cord into place.



Replace the back and screw in the backing screws to the power strip.





Chapter 10: Build the Sensor Patch Bay

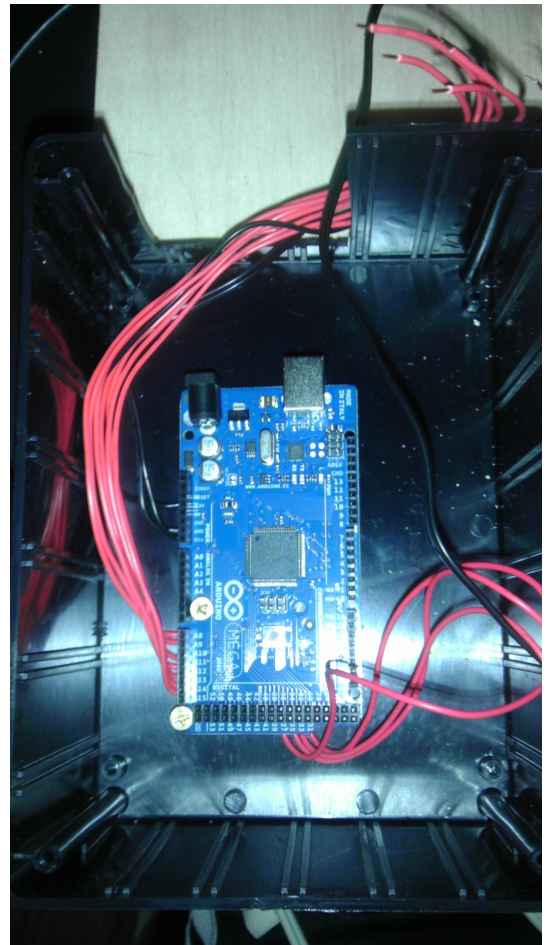
Supplemental video: “Chapter 10: Build the Sensor Patch Bay” at www.YouTube.com

Parts list:

- 2 X 12 Position Barrier Strip Euro Terminals
- 1 X 10K Ohm Resistors ¼ Watt

First, check out the wiring diagram included in this project to see how we are going to wire this component up. The wiring diagram can be found at www.RobotGardener.com or www.OuterBabylon.com.

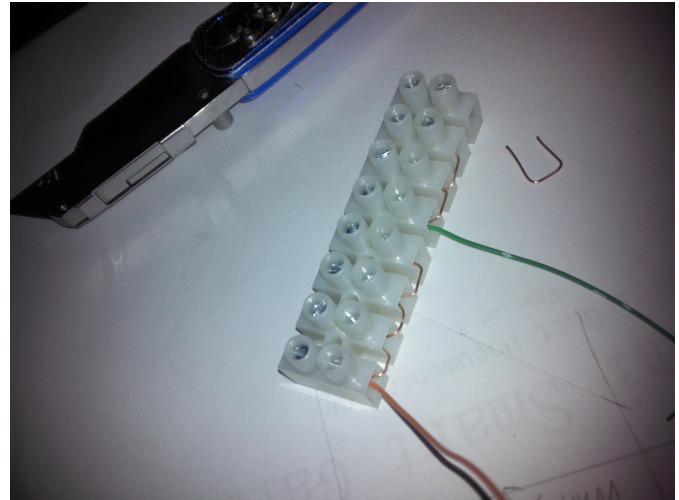
I will refer to the back of the sensor patch bay as the side that all the wire will plug into which come from the inside of the control unit. These wires include the wires coming from the Arduino.



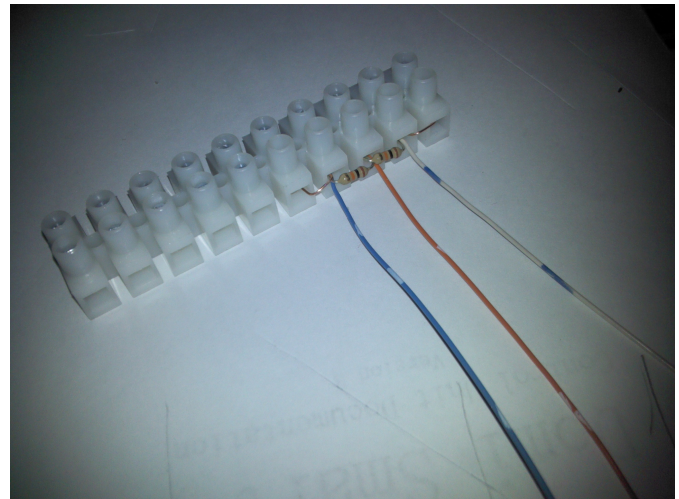
I'll refer to the front of the sensor patch bay as the side that includes all the inbound wires like the sensor signal wires from outside the control unit.

Note how the ground and Vin lines are looped through several back end port holes in one of the Euro terminals on the wire

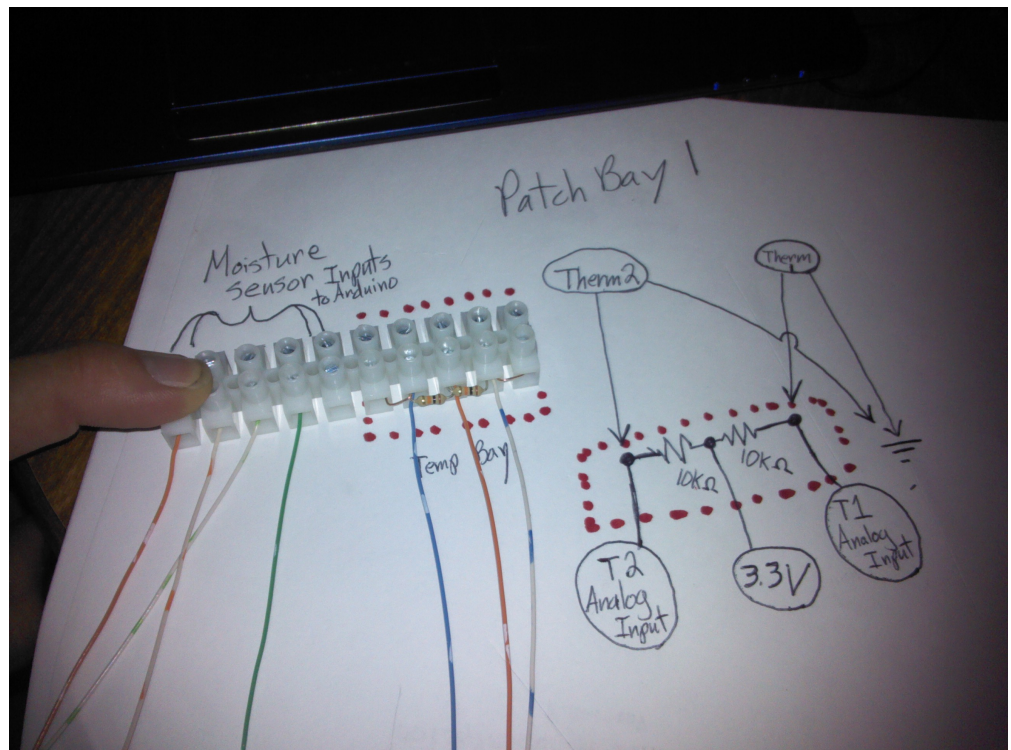
diagram. This is so that there are more spots on the front of the patch bay to connect things that require th power and ground. For this, we will need to make some little metal connectors for the back of the sensor bay like the following.



Next, wire up the voltage divider according to the wiring diagram. This is the part of the sensor patch bay that requires the two 10 ohm resistors and the 3.3V input power.



Then, run the five analog sensor lines into the back of the sensor patch bay for the five analog ports that will read the moisture sensors.



Mount the Euro terminals onto the board where the template specifies and we'll be done with the work in this section. Later, we will run the sensors cords into the box and into the front of the sensor patch bay. For now though, we will wait to finish the unit because it's easier to add in the sensors once the unit is complete.

Chapter 11: Building the Sensors

Supplemental videos:

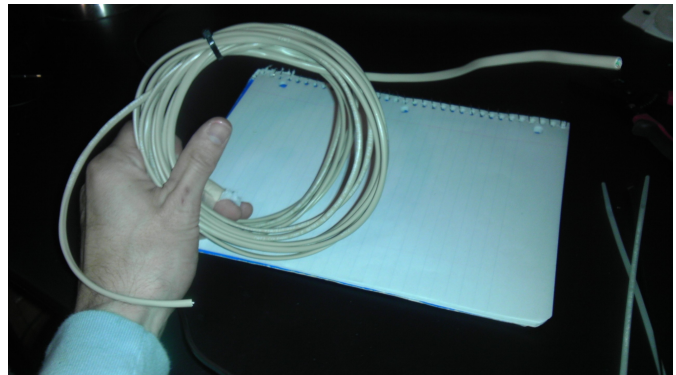
- “Chapter 11 - A: Building the Sensors” at www.YouTube.com
- “Chapter 11 - B: Wire Stripping” at www.YouTube.com
- “Chapter 11 - C: Building the Temperature Sensor” at www.YouTube.com
- “Chapter 11 - D: Hooking Up A Sensor” at www.YouTube.com

Todo: Make sure that parts list items match the master parts list items titles

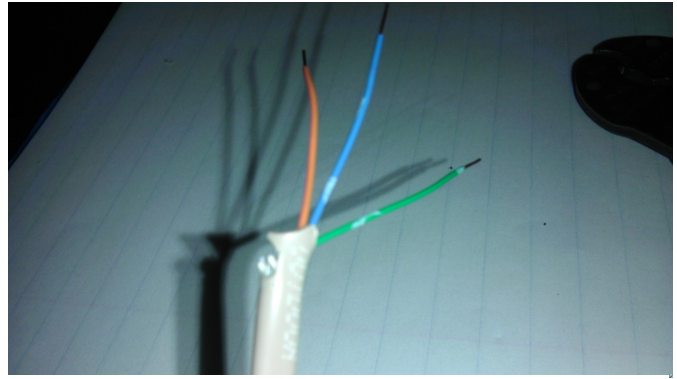
Parts list:

- 2 X 1/4" - 1/8" Heat Shrink Tubes
- 2 X 10K Precision Epoxy Thermistor
- 5 X Vegetronix Soil Moisture Sensor - 2 meter cable
- 1 X Cat3 Cable – 250'
- 1 X Roll of Electrical Tape
- 1 X Liquid Electrical Tape

Cut 7 lengths of Cat3 cord and wrap them neatly so that they are manageable.



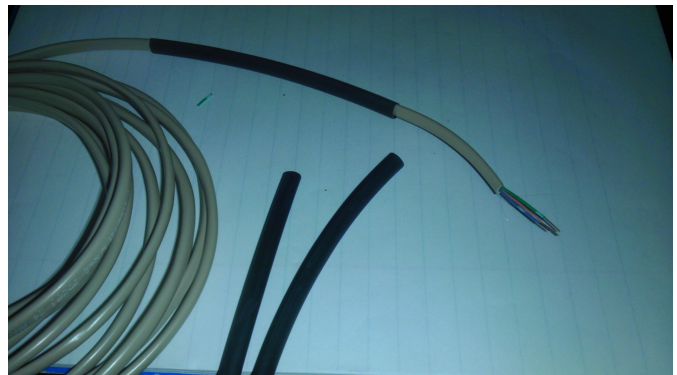
Strip the ends of five of them and cut out some of the inner wires so that there are only three wires left. We will use these five cords to extend the moisture sensors. When you do this, you will want to choose the same three colors of inner wires to preserve so that we can map these colors to the colors on the moisture sensors and temperature sensors. For instance, possibly chop the cord down to a red (Vin), green (ground), and blue (signal) inner wires so that they can easily be mapped to the actual sensor wires that are red (Vin), uninsulated (ground), and black (signal).



Strip the ends of the other two of them and cut out some of the inner wires so that there are only two wires left. We will use these five cords to extend the temperature sensors. Choose the same colors and do the proper color matching as you did with the moisture sensors above.

Go through each exposed wire on each of these seven cords and strip the wires down to metal. Expose about a third of an inch of metal on each wire.

One end of each of these seven cords will be soldered to a sensor so we can go ahead and add a piece of shrink wrap now to each in preparation for when the soldering is done and we seal the new cord.



Carefully go through each of the seven cords and solder the proper sensors onto the exposed wires. Seal each individual wire with a small wrap of electrical tape.



Once your done adding all the sensors to the cords, move the heat shrink wrap into place and heat it with a lighter until it securely seals the new cord.



Put these seven extended sensors to the side for when we are ready to do the install later. They will be one of the last things that we add to the control unit.

Chapter 12: Splicing the Solenoids Onto Plugs

Supplemental video: “Chapter 12: Splicing the Solenoids Onto Plugs” at www.YouTube.com

Parts list:

- 2 X 1” FNPT In-Line Jar Top Water Solenoid
- 2 X Male cord ends that were saved from chapter called “Build the heavy current router”
- add electrical tape
- liquid electrical tape
- 1/4” - 1/8” Heat Shrink Tubes

In this section, we are going to splice the water solenoids into a male plug so that we can plug them into the peripheral extension cords when we are doing the final install. This is just a simple splice job where we will just strip the ends of the solenoid wires and soldering them onto a male plug.



The reason that these water solenoids are 24V 0.65A and require a special wall adapter is because electricity is less dangerous when the current is low. The typical setup for household electrical circuits is 110V and 15A - 20A. Since these solenoids are going to be around water and the ground around them may be wet, we have chosen water solenoids that will be much lower amperage and therefore less dangerous if they short out or arc into a user. If it happens to shock you, it will be a small buzz and there is no harm involved where as a higher current may hurt you.

Before you start soldering, remember to place a piece of heat shrink wrapping around one end of the cord you are about to solder. This is so that you can simply slip the shrink wrap down and secure it once the soldering job is done. If you forget this, then there will be no way to get the shrink wrap on once the soldering is done.



Now solder the two wires together but do not heat seal the shrink wrap yet.



Once you are done with the soldering, you'll have to insulate the two wires from each other with two simple wraps of electrical tape. Just make sure that there is no exposed metal from the solder job that could potentially arc out or short from wire to wire.





Now add a little liquid electrical tape on top of the electrical tape that you just wrapped around the two wires. The liquid electrical tape is so that a fairly secure seal will result when the final heat shrink wrapping squeezes the soldered wires.



So far we have soldered the two wires, wrapped them with electrical tape, and added water proof liquid electrical tape as moisture sealant. Finally, before the liquid electrical tape dries, we will slide the heat shrink tubing over the solder point and secure it by running a flame over it.



Once this is complete, you will have sealed the cord, and it should look something like the following.



This is what you should end up with. These are ready to plug into the female extension cords that will come out of the bottom of the control unit.



Chapter 13: Completing the Unit

Supplemental video:

- “Chapter 13 - A: Completing the Unit” at www.YouTube.com
- “Chapter 13 - B: Notes on Debugging” at www.YouTube.com
- “Chapter 13 - C: Simple Systems Test” at www.YouTube.com

Chapter 14: Loading the Arduino Software

Supplemental video: "Chapter 14: Loading the Arduino Software " at www.YouTube.com

Parts list:

- Arduino Mega 2560 Board
- USB Cable (classic printer cable)

The Arduino comes with a blank memory which needs to be loaded with the program that will run your Arduino setup. I've written some basic software to load into your Arduino at <http://www.RobotGardener.com>. To install the software on the Arduino you will need to do a couple of things.

Next you'll need to download the Arduino compiler/loader program. It's produced and maintained by the folks at Arduino and is a pretty nice compiler.

Go to the Arduino site and download the version that's appropriate for your computer's operating system.

<http://arduino.cc/en/Main/Software>

Unzip that and move the contained directory to some place like your home on Linux or your "My Documents" Windows. The program in this directory called "arduino" is the actual compiler up-loader. Create a folder in this directory named "sketchbook" while you are here. This is where you will save code to that you write in the Arduino IDE.

Also, notice the "libraries" directory by the compiler/uploader. We will download an additional library next and add it to the "libraries" directory.

My program relies on an MIT licensed Arduino library called Webduino. You will need to download this from

<http://code.google.com/p/webduino/>

Once you have downloaded it, you can unzip it place the contained library directory into the library directory of the Arduino compiler/uploader. If the library directory is named Webduino-Master, you will have to rename it to take the '-' out of it's name so that the Arduino compiler/loader will be able to include it. Leave it named something like WebduinoMaster.

The Arduino Compiler/uploader only searches for libraries when it starts so reopen it once you copy the library if you already have the compiler/uploader open.

Now download my basic software from

<http://robotgardener.com/open-documentation/control-unit-construction/chapter-14-loading-the-arduino-software/>

Unzip it and add the contained project to the sketchbook directory that also sits next to the Arduino compiler/uploader. This is the code that we will compile and upload.

Now you will actually be able to attach your Arduino to your computer via the USB connection cable and upload to it. Connect your Arduino now and wait some seconds.

Now start the Arduino compiler/uploader program.

To attach the Arduino board via USB, you'll need to go to Tools > Board and select the option that contains "Arduino Mega 2560" or your board model. Then, you'll have to choose the USB port that your Arduino is plugged into. Go to Tools > Serial Port and select the one listed. This is either really simple or you'll have to look through the forums on the Arduino site to find out how to find the port that your Arduino is attached to or why all the options are gray in color.

Sometimes the serial port selection in the tools menu is grayed out because your user doesn't have permission to the “dialout” group. I'm on Linux Mint 14, so I fixed that by adding my user to the “dialout” group with the following command:

```
sudo usermod -a -G dialout [my_username]
```

Now you can go to File > SketchBook and the RobotGardener code should be there for selection. Once the code appears on the screen, you'll then go to File > Upload and the compiler will compile and the uploader will upload to the Arduino! If all goes well then there will be a success message in the text dialog at the bottom of the compiler. Now your Arduino is ready to work in your RobotGardener.

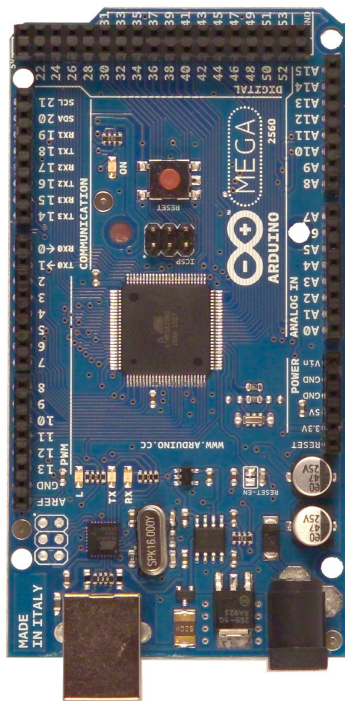
Chapter 15: Helpful Tools

Wire strippers
Soldering Iron
Solider
Solider stand with magnifying lens
Wet sponge for cooling the soldering iron
Silicon or liquid electrical tape to seal any connections from moisture like the water solenoids
Drill
Drill bit for Phillips screws
Drill it for small pilot holes in wood and electrical box
Power screw driver
Power screw driver charger
Exacto knife
Small screw driver for Phillips
Needle nose pliers
Exacto knife
Dremil tool with circular disk bit
Tool box

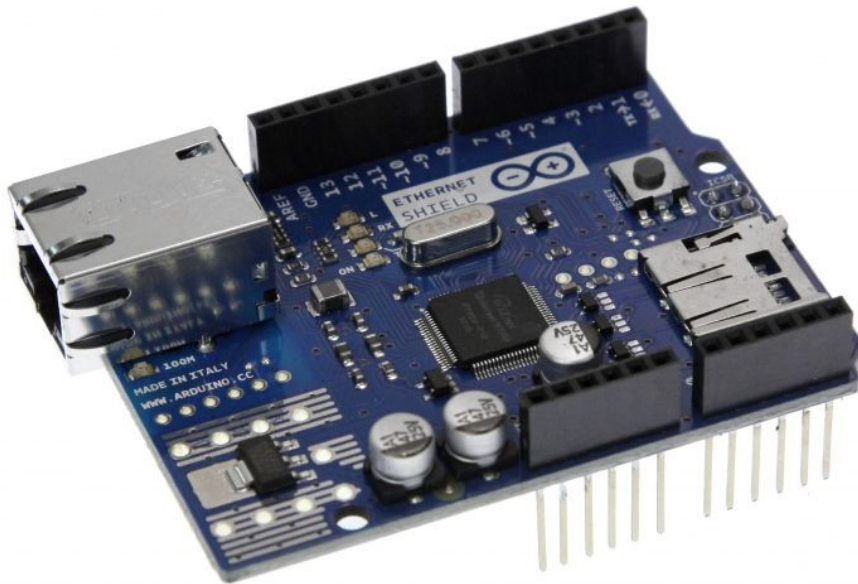
Chapter 15: Parts List

This is the suggested parts list. My parts mostly came from on-line sources, Lowe's, and Radio Shack.

1 X Arduino Mega ADK Board Rev3 – Online \$50.42



1 X Arduino Ethernet Shield R3 – Online \$34.02



1 X Seeed Studio 4-Channel Relay Shield for Arduino – Online \$19.95



1 X 21/32x12x48 Paint Grade Edge-glued Panel – Lowe's \$6.85



1 X 16 Gauge Lawn & Garden Cord – Lowe's \$22.97



1 X 1000 Joules 6 Outlet Power Cord + Surge Protector – Lowe's \$17.95



4 X 16 Gauge 13 Amp Brown Indoor Light-Duty Cord – Lowe's \$3.97



1 X 16 Gauge 13 Amp Indoor Light-Duty Cord – Lowe's \$4.47



1 X Wire Twist Nuts 25pk #18 - #14 – Lowe's \$3.95



1 X Ground Fault Circuit Interrupter – Lowe's \$15.98



1 X 12CI 1G Romex Handy Box Old Work Breaker Box – Lowe's \$0.83



1 X NM Cable Connector 1/2" Knockout – Lowe's \$0.36



2 X 2 Gang Blank Breaker Box Cover – Lowe's \$2.55



2 X WP 2G T Breaker Box - Lowe's \$8.17



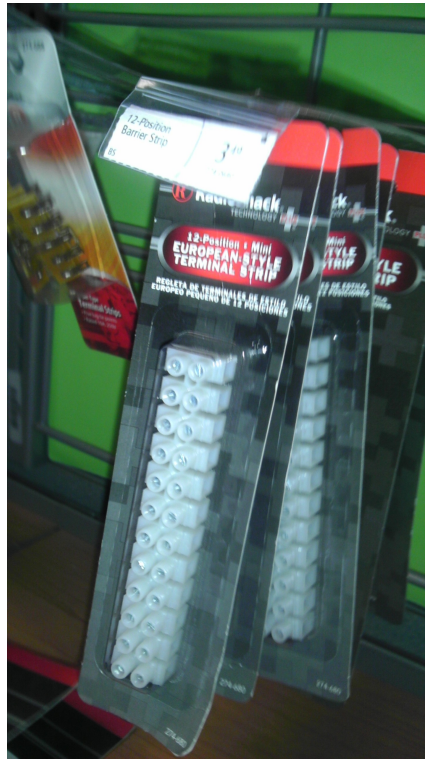
4 X ¾ NM/SEU Connector – Lowe's \$1.02



1 X 22 Gauge Stranded Hookup Wire – Radio Shack \$8.49



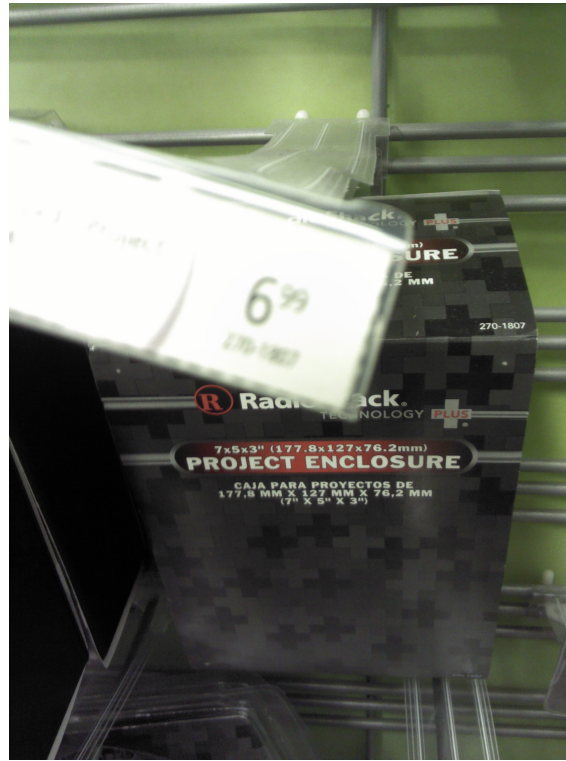
2 X 12 Position Barrier Strip Euro Terminals – Radio Shack \$3.49



1 X 10K Ohm Resistors ¼ Watt – Radio Shack \$1.19



1 X 7x5x3 Project Enclosure – Radio Shack \$6.99



2 X 1/4\" - 1/8\" Heat Shrink Tubes – Radio Shack \$2.16



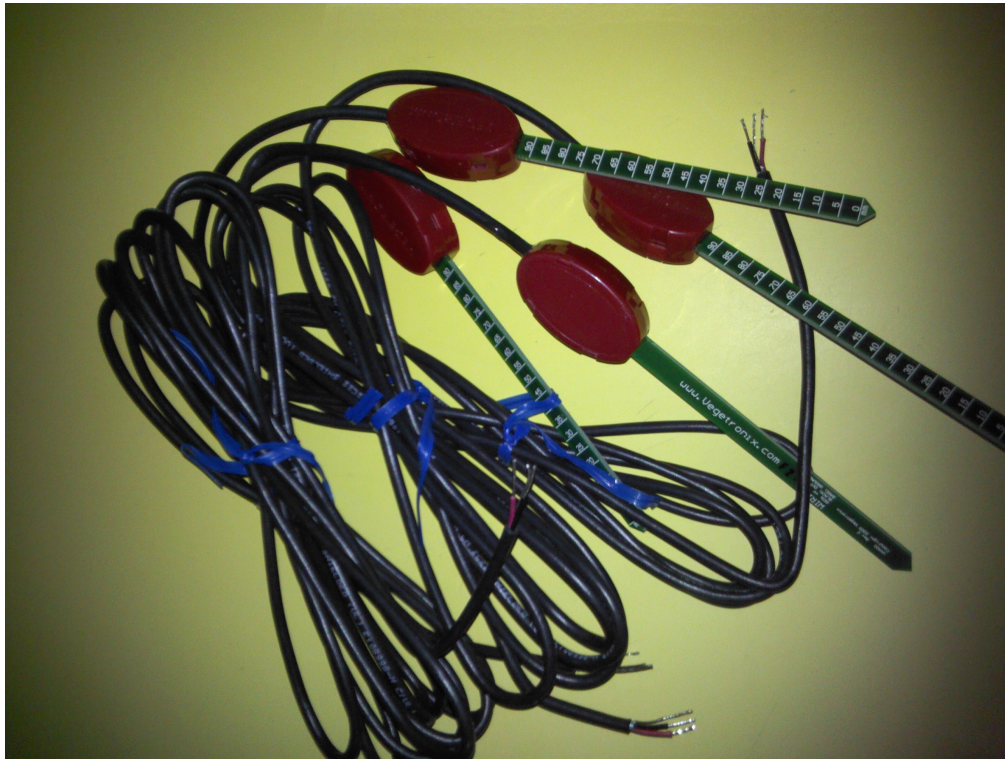
1 X Cat3 Cable – 250' – Lowe's \$31.47



2 X 10K Precision Epoxy Thermistor – Online \$4.00



5 X Vegetronix Soil Moisture Sensor - 2 meter cable – Online \$34.



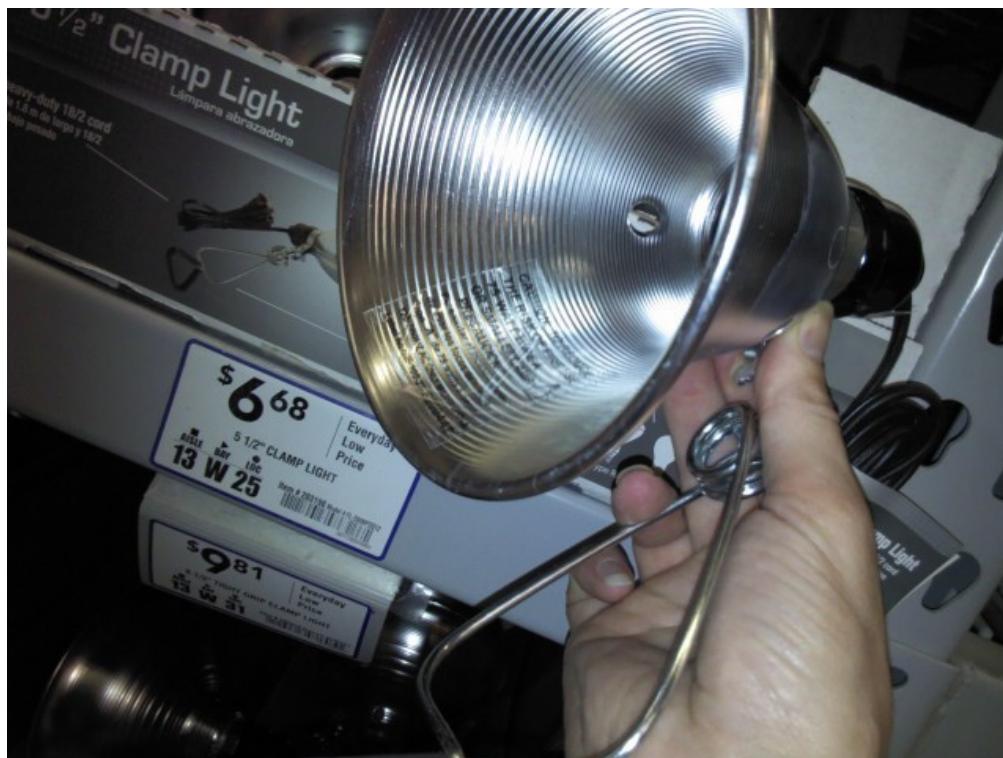
2 X 1" FNPT In-Line Jar Top Water Solenoid – Lowe's \$10.26



1 X 4 Station Dual PGM Timer – Lowe's \$19.97



2 X 5 1/2" Clamp Light – Lowe's \$6.68



1 X 75W Light Bulbs – Lowe's \$4.98



1 X 1' x 1' Waterproof Electrical Box – Lowe's \$33.60



1 X Roll of Electrical Tape – Lowe's \$0.69



1 X Liquid Electrical Tape – Lowe's \$5.99



Builder Notes

This is a section for the user to record notes about their specific build and implementation for future reference.