**Automated Plant Watering with a Raspberry Pi**  
  
After reading about how well things can grow indoors, I started thinking that maybe automation was my path to healthy plants. So, I decided to build the bare minimum - get a plant, a pump, and a water sensor. When the water sensor says "no water here", use the pump to put water there.  
  
I also decided to run it all through a Raspberry Pi to as an excuse to interact with the RPi GPIO.  
  
Here's how I did it!

**Materials:**

1. [Raspberry Pi 3](https://amzn.to/2CSUgTC)
2. [Soil Moisture Sensor](https://www.amazon.com/gp/product/B00ZR3B60I/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B00ZR3B60I&linkCode=as2&tag=cyberomelette-20&linkId=747cf37750038d3b8bc9c4b7fbc4f227)
3. [Flexible Water Line](https://www.amazon.com/gp/product/B0002AQI9A/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1&tag=cyberomelette-20)
4. [5V Relay](https://www.amazon.com/gp/product/B0057OC5O8/ref=as_li_tl?ie=UTF8&tag=cyberomelette-20&camp=1789&creative=9325&linkCode=as2&creativeASIN=B0057OC5O8&linkId=79128c0fb9efa41048a1490b5007a164)
5. [3-6V Mini Micro Submersible Pump](https://amzn.to/2r93Smi)
6. [TOLI 120pcs Multicolored Dupont Wire](https://www.amazon.com/gp/product/B06Y2B48RK/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B06Y2B48RK&linkCode=as2&tag=cyberomelette-20&linkId=75009e2d26426b98b9ff6ef66dc76917)
7. 5v Power Supply ([Any USB Cable](https://www.amazon.com/gp/product/B00MG317YC/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B00MG317YC&linkCode=as2&tag=cyberomelette-20&linkId=a3e0faea00eaeb3132ecaa871ce09585)+ [USB Wall Charger)](https://www.amazon.com/gp/product/B01LCDJ7LG/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=B01LCDJ7LG&linkCode=as2&tag=cyberomelette-20&linkId=95c38a3fc2b9e7e43592ebe3c4a3cc81)

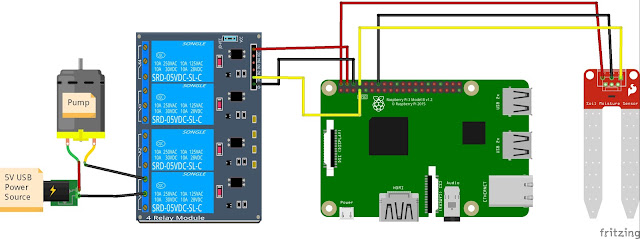
**Wiring:**

The first thing I did was make my 5V power supply from a USB cable. Using an old phone cable, I cut the phone side off and fished out a red and black wire. I soldered some sturdier wires to this, and plugged it into a wall adapter. Checking with a voltmeter, this gave me 5V output.

[](https://4.bp.blogspot.com/-zOMpiAZrmQE/WcL2eOErfbI/AAAAAAABNGc/oII16bKR-3sMRRX_V6vowjgRIgsGw8N4wCK4BGAYYCw/s1600/IMG_5373.JPG)

Now time for the GPIO.

**RPi Wiring:**

[](https://4.bp.blogspot.com/-IPGUpAUBVCg/WcLxKim1tKI/AAAAAAABNGM/Ecyxqoqe_wIUQkHeAqYHGoIEkDznOqoZQCK4BGAYYCw/s1600/GPIO.jpg)  
  
Following [this GPIO layout](https://www.element14.com/community/servlet/JiveServlet/previewBody/73950-102-12-339300/pi3_gpio.png):  
  
**Water Sensor** -  plug the positive lead from the water sensor to pin 2, and the negative lead to pin 6. Plug the signal wire (yellow) to pin 8.  
  
**Relay** - Plug the positive lead from pin 7 to IN1 on the Relay Board. Also connect Pin 2 to VCC, and Pin 5 to GND on the Relay board.  
  
**Pump**- Connect your pump to a power source, run the black ground wire between slots B and C of relay module 1 (when the RPi sends a LOW signal of 0v to pin 1, this will close the circuit turning on the pump).  
  
This diagram should capture the correct GPIO so long as you are using Raspberry Pi 3. Not shown is another power source to the RPi.

**Hardware Setup:**

Once the wiring has been completed, attach the flexible hose to the pump (I used electrical tape), and drop it into a jar of water. Attach the other end of the hose to your plant.  
  
Now plug in all power sources (and make sure your Raspberry Pi is running some version of an operating system).

**Software**

Note: If you get the wiring exactly as described above, my code in the next section will work with no modifications.

There are two parts to this setup. One file controls all the GPIO and circuit logic, and the other runs a local web server.  
  
All files:  
[water.py](https://gist.github.com/benrules2/6f490f3a0e082ae6592a630bd7abe588)  
[web\_plants.py](https://gist.github.com/benrules2/c4f3db455f4f2dfbe7d5b825b0b4ee36#file-web_plants-py)  
[auto\_water.py](https://gist.github.com/benrules2/c4f3db455f4f2dfbe7d5b825b0b4ee36#file-auto_water-py%EF%BB%BF)  
[temp.py](https://gist.github.com/benrules2/c4f3db455f4f2dfbe7d5b825b0b4ee36#file-auto_water-py%EF%BB%BF)  
[main.html](https://gist.github.com/benrules2/e43c469b2c1263237dc67010fca18b53)

**GPIO Script**

Let's start with the code for controlling the GPIO. This requires the RPi.GPIO python library which can be installed on your Raspberry Pi as follows:  
  
$> python3.4 -m pip install RPi.GPIO

With that installed, you should be able to use the water.py script.

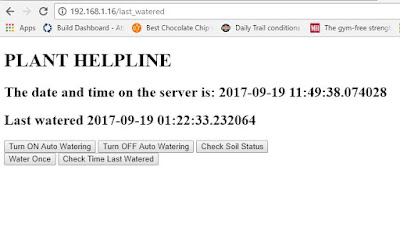
You can test this is working correctly by running an interactive python session as follows:  
  
$> python3.4  
>>> import water  
>>> water.get\_status()  
>>> water.pump\_on()  
  
This should print a statement about whether your sensor is wet or dry (get\_status()), and also turn on the pump for 1s. If these work as expected, you're in good shape.  
  
At this point you can also calibrate your water sensor. If your plant status is incorrect, try turning the small screw (potentiometer) on the sensor while it is in moist soil until the 2nd light comes on.

**Flask Webserver**

The next aspect of this project is to setup the web server. This code can be found in the file called web\_plants.py. This python script runs a web server enabling various actions from the script described above.  
  
You will need to keep web\_plants.py in the same directory as water.py described above, as well as auto\_water.py. You will also need a sub-directory called "templates" containing the html file called main.html.  
  
You will need to install flask, and psutil as follows:  
  
$> python3.4 -m pip install flask  
$> python3.4 -m pip install psutil

You will also need to create a sub-directory called templates, and place main.html in the templates directory.

Now run the following command command to start your web server:  
  
$> sudo python3.4 web\_plants.py  
  
Now if you navigate to the ip address of your RPi, you should see a web dashboard something like this:

[](https://4.bp.blogspot.com/-3pYjaUEkQ-w/WcMQVFnihNI/AAAAAAABNIM/DvKFgdGkWKMbPgjpEsryzb9Qdbz9ORLFgCK4BGAYYCw/s1600/web.JPG)

Try clicking the buttons to make sure everything works as expected! If so, you're off to the races.

**Run Website Automatically**

Finally, you probably want the website to auto start when the RPi gets turned on. This can be done using a tool called cronjob, which registers your website as a startup command.

To do so, type:

$> sudo crontab -e

This will bring up a text editor. Add a single line that reads (and make sure to leave one empty line below):

@reboot cd <your path to web\_plants>; sudo python3.4 web\_plants.py

Now when you reboot your pi, it should auto start the server.