Reporting Water Sensor Values, Step by Step With Raspberrypi & EC2

Detailed instructions to parts and steps here: github.com/kathrynthegreat/rasberrypi

Katya Vasilaky*, Ryan Considine^

*Cal Poly, ^M Squared Lasers

Parts needed

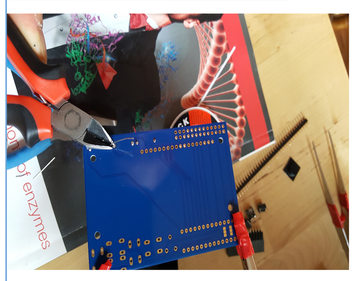
Total Cost: $129

- Raspberry Pi Model B ($38)
- Half size breadboard ($3)
- Female to male jumper wires ($10)
- Male to male jumper wires ($8)
- MCP3008 ($10)
- Octopus soil moisture sensor brick ($7)
- RGB Negative 16x2 LCD+Keypad Kit ($25)
- Stacking header raspberry pi 2x13 extra tall ($2)
- Android USB cable ($5)
- Soldering Iron ($16)
- Ethernet Cable ($5)

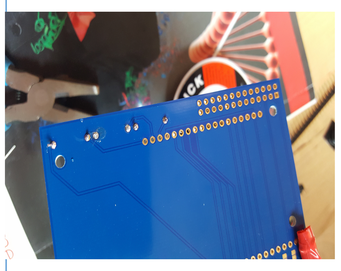
Assembly

The tools. Fan is used to exhaust the fumes from the solder. Soldering gun, soldering tool kit…

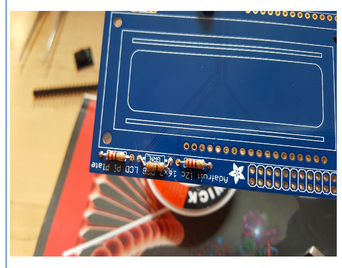
The large Red arrow is the LCD display which displays raspberry pi's output text. The Light Blue arrow is the 16pin I2C port expander chip - MCP23017-E/SP . The Green arrow is the Potentiometer which controls the brightness of the display. The four Red arrows are control buttons up, down, left, right, and select.



LCD Printed Circuit Board (PCB). Has pre-drilled holes. Here we are cutting a resistors wire before soldering it to the PCB board.



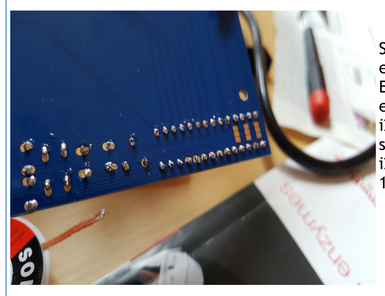
Soldered resistors on the PCB board.



Resistors soldered into the PCB board. The colored stripes on the resistors. These stripes tell you about the resistance and tolerance of the resistor. *The middle resistor: 330 ohms which has orange, orange, brown, gold bands on it.

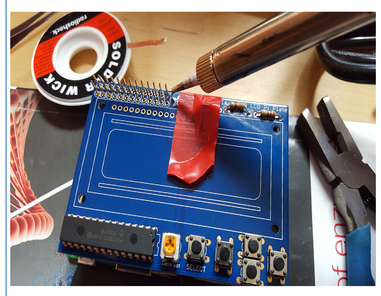
This resistor acts as the backlight control resistor for the green backlight pin. The two outside resistors: are two 220 ohm resistors - they are the backlight series resistors for the RGB backlights on the LCDs. These resistors are colored Red Red Brown Gold. (*info from Adafruit)

Soldering in LCD Buttons. *These buttons are useful to send a signal to the Pi (say if you have a basic menu system). We have a 4-way 'direction pad' for up/down/left/right input and a button to the right called SELECT. These 5 buttons should be able to make 'talking' back to your project easy. These are connected to the I2C port expander chip so they require no extra pins on the Pi, our library does the work of reading whether they are pressed. (*info from Adafruit)

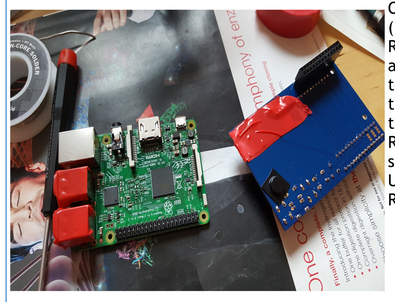


Soldering in the I2C port expander chip - MCP23017-E/SP - .*This is a 16-pin expander chip, that uses the i2c bus. That means you can send it commands using the i2c pins on an Pi and control 16 more digital pins!

5 of those pins go to the buttons, 6 go to the LCD control and 3 are used for the backlight (the remaining 2 are unused). (*info from Adafruit)

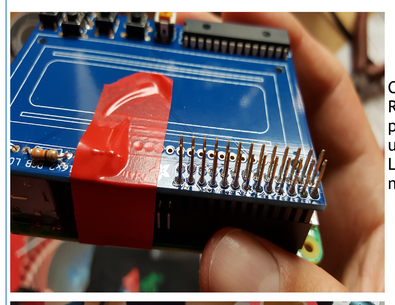


Soldering in the 26 pin header. The header will connect to the raspberry pi *this will send power and data between the two boards and also provide a mechanical stability. (*info from Adafruit)

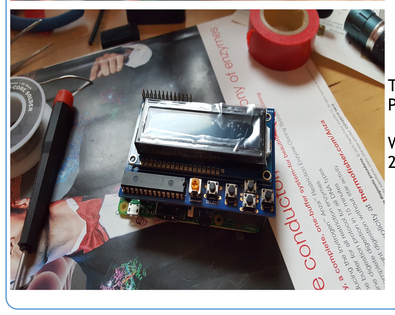


Connect the LCD header (red arrow) to the pins on Raspberry Pi (green arrow). This will snap together fitting the LCD to the Raspberry Pi. The red tape it used on the Raspberry Pi B+ to avoid short circuiting from the USB ports on the Raspberry Pi.

*WARNING: If you're using the new Raspberry Pi model B+ with 4 USB ports you will need to take some extra steps to ensure the backlight resistor leads do not touch the USB ports and short out. See the steps below for more details. (*info from Adafruit)



Connected LCD Board to Raspberry Pi. Note the pins sticking up can be used for add-ons to the LCD display but are not need for this project.



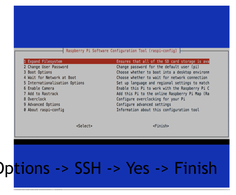
The LCD snaps onto the PCB and then you'e done!

We followed http://bit.ly/2qhbcuK for wiring.

Configuring your Pi

First you'll need to install the Raspbian OS. Use a keyboard and monitor with usb to access the pi's command line; download raspbian to the SD card and enable ssh.

$ sudo raspi-config

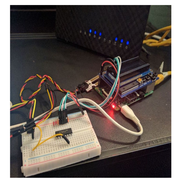


Select Advanced Options -> SSH -> Yes -> Finish

```
kathryns-Air-2:~ kathrynvasilaky$ ifconfig
        inet 192.168.1.216
```

Find your computer's IP address with ifconfig. Turn your pi on and connect it directly to the router.

Turn your pi on and connect it directly to the router.



```
kathryns-Air-2:~ kathrynvasilaky$ nmap -sn 192.168.1.216/24
Nmap scan report for raspberrypi (192.168.1.134)
```

sudo apt-get install nmap. Look for pi's IP address with nmap.

Ssh in to your pi!
$ ssh pi@pi@192.168.1.134
Default password is "raspberry"

Configure wifi on your Pi, by storing network information.
pi@raspberrypi:~ $ sudo nano /etc/network/interfaces

```
GNU nano 2.2.6                File: /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d


auto lo
iface lo inet loopback
iface etho inet dhcp

allow-hotplug wlan0
auto wlan0


iface wlan0 inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

pi@raspberrypi:~ $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

```
GNU nano 2.2.6                File: /etc/wpa_supplicant/wpa_supplicant.conf

country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="mynetwork"
    psk="mypassword"
}
```

Restart wlan0, and the pi.

```
pi@raspberrypi:~ $ sudo wpa_cli reconfigure
pi@raspberrypi:~ $ sudo ifdown wlan0
pi@raspberrypi:~ $ sudo ifup wlan0
pi@raspberrypi:~ $ ifconfig wlan0
pi@raspberrypi:~ $ reboot
kathryns-Air-2:~ kathrynvasilaky$  sud
pi@raspberrypi:~ $ ssh pi@192.168.1.2
```

```
18:16:01 > Discovery progress 25%
18:16:01 > Host is up:   192.168.1.221
           HW Address:   B8:27:EB:4B:38:32 (Raspberry Pi Foundation)
           Hostname:     raspberrypi
```

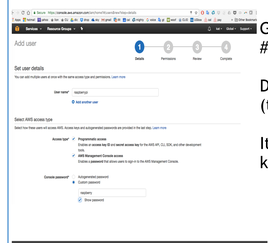
```
pi@raspberrypi:~ $ sudo apt-get install build-essential python-dev python-smbus git
pi@raspberrypi:~ $ git clone https://github.com/adafruit/Adafruit_Python_MCP3008.git
pi@raspberrypi:~ $ cd Adafruit_Python_MCP3008
pi@raspberrypi:~ $ sudo python setup.py install
# Test the readings out
pi@raspberrypi:~ $ cd /Adafruit_Python_MCP3008/examples
pi@raspberrypi:~/Adafruit_Python_MCP3008/examples $ nano simpletest.py
# Comment out the software part, keep the Hardware SPI section
# Hardware SPI configuration (taken from pi@raspberrypi:~/Adafruit_Python_MCP3008/examples/simpletest.py)
    SPI_PORT   = 0
    SPI_DEVICE = 1
    mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE))
    #test the water level out
    mcp.read_adc(5)
```

```
pi@raspberrypi:~/Adafruit_Python_MCP3008/examples $ sudo python simpletest.py
Reading MCP3008 values, press Ctrl-C to quit...
|   0|   1|   2|   3|   4|   5|   6|   7|
-----------------------------------------------------------------
|   4|   3|   3|   4|   3|   0|   3|   2|
|   1|   1|   2|   2|   1|   0|   0|   0|
|   0|   2|   2|   2|   2|   0|   0|   0|
|   0|   1|   2|   2|   2|   1|   0|   0|
|   2|   2|   2|   2|   2|   1|   0|   0|
|   3|   2|   3|   3|   3|   0|   1|   2|
|   4|   3|   3|   3|   3|   0|   2|   3|
```

ctrl+o to exit, enter, ctrl+x to

```
pi@raspberrypi:~/Adafruit_Python_MCP3008/examples $ sudo python simpletest.py
```

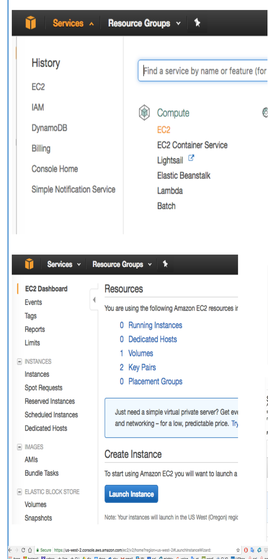
EC2, Amazon Web Services, Dynamo DB



Go to console.aws.amazon.com/iam/home?#/users

Download the security credentials .csv file (that's your key)

It contains Access key ID, and Secret access key

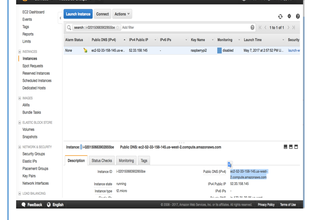


Start an EC2 instance
Create a new key pair
Download the .pem file
Place that file in a folder
CD to that folder from the command line

$ cd /Users/kathrynvasilaky/Documents/OneDrive/Python/raspberrypiagain

Protect a file against accidental overwriting

$ chmod 400 raspberrypi.pem

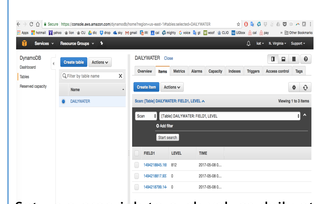


Instal AWS CLI
$ pip install --upgrade --user awscli
Use your keys to configure your account
$ aws configure
Sign into your account
$ ssh -I yourpem.pem ec2-user@ec2-52-33-158-145.us-west-2.compute.amazonaws.com


```
[ec2-user@ip-172-31-29-85 ]$ mkdir venv
[ec2-user@ip-172-31-29-85 ]$ cd venv
[ec2-user@ip-172-31-29-85 venv]$ virtualenv -p /usr/bin/python3.4 python34
[ec2-user@ip-172-31-29-85 venv]$ pip install boto3
(python34)[ec2-user@ip-172-31-29-85 venv]$ touch table.py
# Copy paste the table code from table.py
(python34)[ec2-user@ip-172-31-29-85 venv]$ nano table.py
# Create the database
(python34)[ec2-user@ip-172-31-29-85 venv]$ python table.py
```

Back to the PI

```
pi@raspberrypi:~ $ touch update.py
pi@raspberrypi:~ $ nano update.py
# Copy past the code from my update.py file. Replace with your AWS keys & email in
# pip install smtplib (command that sends you the email)
pi@raspberrypi:~ $ chmod +x /home/pi/upload.py
pi@raspberrypi:~ $ sudo python update.py
# Check your database on AWS to see if a reading got pushed, and if you received an
```



Set up a cron job to push values daily at 10:15 AM

```
pi@raspberrypi:~ $ crontab -e
    15 10 * * *  /home/pi/upload.py
```

ctrl+o, enter, ctrl+x# pip install smtplib (command that sends you the email)

```
pi@raspberrypi:~ $ sudo /etc/init.d/cron start
pi@raspberrypi:~ $ sudo /etc/init.d/cron status
```