

1 channel Plant Watering System with Raspberry Pi

Overview

The following guide will show you how to set up a watering system that can intelligently monitor and automatically water to the plant or flower.

Parts Required

1 x Raspberry Pi Board

1 x ADS1115 Analog Digital Converter

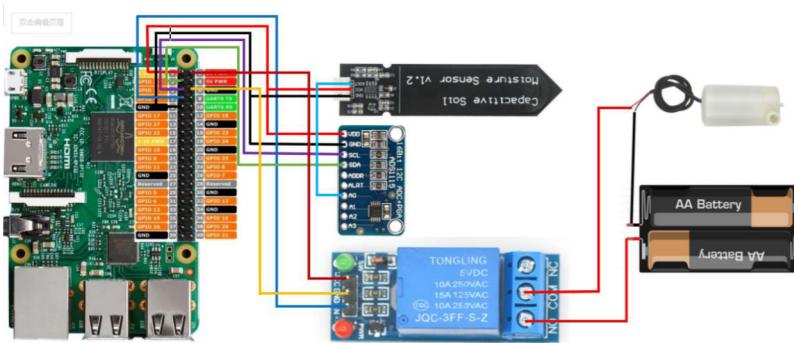
1 x Capacitive Soil Moisture Sensor

1 x One-channel Relay

1 x Water Pump

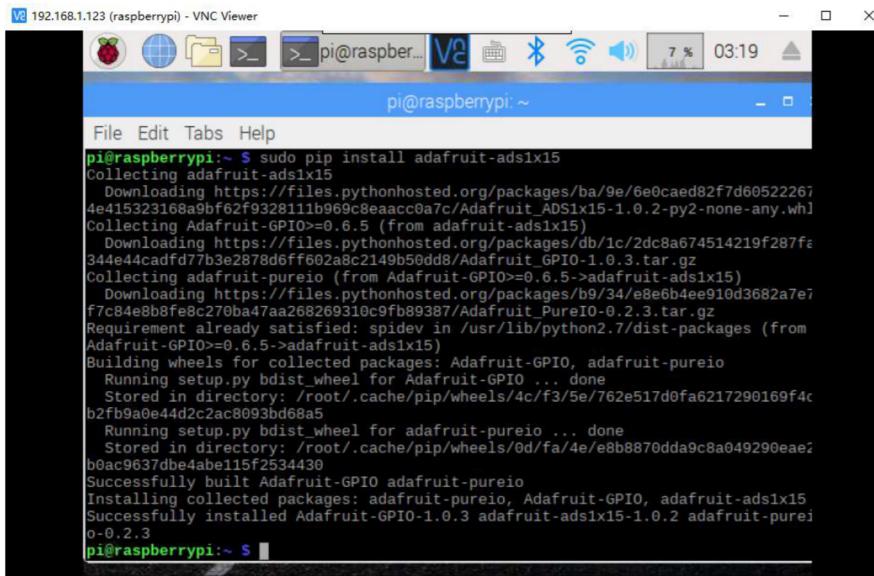
15 x Jumper Wires

Connection Diagram



How to set up with Raspberry Pi?

Step 1: Add the ADS1115 module library to the Raspberry Pi system
after connection completed, enter '[sudo pip install adafruit-ads1x15](#)'
in the Raspberry Pi terminal window, and then press Enter as
shown below.

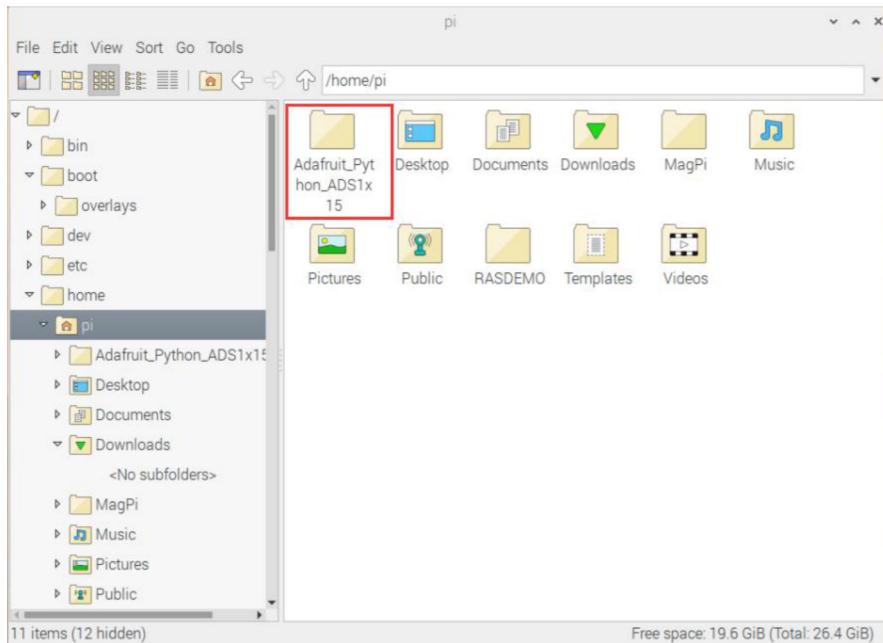


```
pi@raspberrypi:~ $ sudo pip install adafruit-ads1x15
Collecting adafruit-ads1x15
  Downloading https://files.pythonhosted.org/packages/ba/9e/6e0caed82f7d60522267
4e415323168a9bf62f9328111b969c8eaacc0a7c/Adafruit_ADS1x15-1.0.2-py2-none-any.whl
Collecting Adafruit-GPIO>=0.6.5 (from adafruit-ads1x15)
  Downloading https://files.pythonhosted.org/packages/db/1c/2dc8a674514219f287fe
344e4cadfd77b3e2878d6ff602a8c2149b50dd8/Adafruit_GPIO-1.0.3.tar.gz
Collecting adafruit-pureio (from Adafruit-GPIO>=0.6.5->adafruit-ads1x15)
  Downloading https://files.pythonhosted.org/packages/b9/34/e8e6b4ee910d3682a7e7
f7c84e8b8fe8c270ba47aa268269310c9fb89387/Adafruit_PureIO-0.2.3.tar.gz
Requirement already satisfied: spidev in /usr/lib/python2.7/dist-packages (from Adafruit-GPIO>=0.6.5->adafruit-ads1x15)
Building wheels for collected packages: Adafruit-GPIO, adafruit-pureio
  Running setup.py bdist_wheel for Adafruit-GPIO ... done
  Stored in directory: /root/.cache/pip/wheels/4c/f3/5e/762e517d0fa6217290169f4c
b2fb9a0e44d2c2ac8093bd68a5
  Running setup.py bdist_wheel for adafruit-pureio ... done
  Stored in directory: /root/.cache/pip/wheels/0d/fa/e8b8870dda9c8a049290eae2
b0ac9637dbe4abe115f2534430
Successfully built Adafruit-GPIO adafruit-pureio
Installing collected packages: adafruit-pureio, Adafruit-GPIO, adafruit-ads1x15
Successfully installed Adafruit-GPIO-1.0.3 adafruit-ads1x15-1.0.2 adafruit-purei
o-0.2.3
pi@raspberrypi:~ $
```

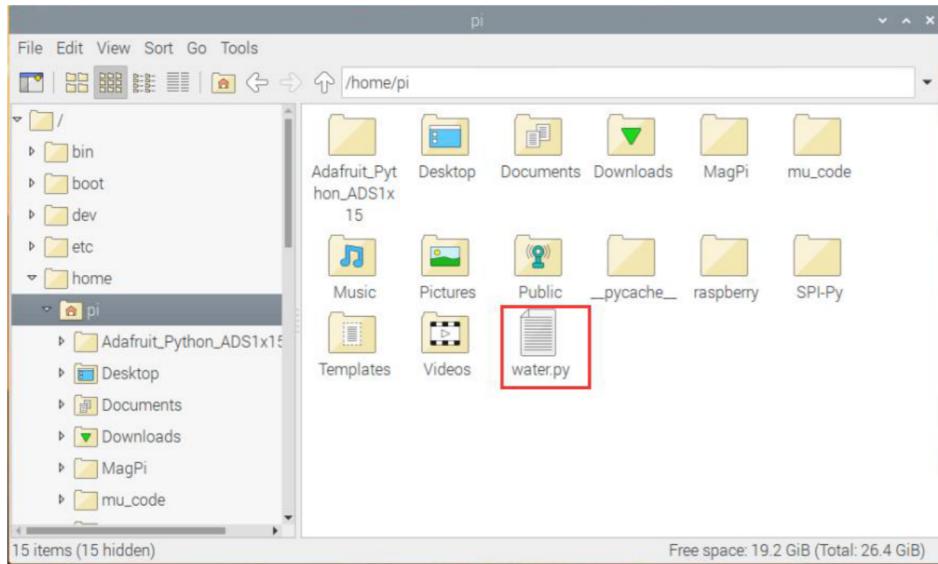
Step 2: After download completed, enter the command at the command line:

git clone https://github.com/adafruit/Adafruit_Python_ADS1x15.git

To download another file of ADS1115. After download completed, the file shown below will appear in your pi file directory.



Step 3: After the download is complete, create a new [water.py](#) file in the pi directory, as shown below.



Step 4: Copy the following code into the water.py file.

```
import RPi.GPIO as GPIO
import time
import Adafruit_ADS1x15
import math
adc =Adafruit_ADS1x15.ADS1115()
GAIN = 1
PIN=7
def setup():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, GPIO.HIGH)
    time.sleep(0.1)

values = [0]*100
def loop():
    while True:
        for i in range(100):
            values[i] = adc.read_adc(0, gain=GAIN)
    print(max(values))
```

```

if (max(values))>20000:
    GPIO.output(PIN, GPIO.LOW)
    print("ON")
    print(PIN)
    time.sleep(0.1)
else:
    GPIO.output(PIN, GPIO.HIGH)
    print("OFF")
    print(PIN)
    time.sleep(0.1)

def destroy():
    GPIO.output(PIN, GPIO.HIGH)
    GPIO.cleanup()

if __name__ == '__main__':
    setup()
    try:
        loop()
    except KeyboardInterrupt:
        destroy()

```

Step 5: Then run the program by typing the [python water.py](#) command in the terminal.

Code Interpretation:

1. We need to import the four modules, they are: Raspberry Pi GPIO module, the delay function module, the ADS1115 module, and the math module.

```
import RPi.GPIO as GPIO  
import time  
import Adafruit_ADS1x15  
import math
```

2. Instantiate an abc object using the ADS1115 module to read sensor data later; then define a position variable GAIN, and the physical pin 7 of the Raspberry Pi.

```
adc =Adafruit_ADS1x15.ADS1115()  
GAIN = 1  
PIN=7
```

3. Set the 7-pin pin mode, pin input/output mode, initialization pin 7 in the `setup()` function, and then delay the 0.1 second debounce.

```
def setup():  
    GPIO.setmode(GPIO.BEAD)  
    GPIO.setup(PIN, GPIO.OUT)  
    GPIO.output(PIN, GPIO.HIGH)  
    time.sleep(0.1)
```

4.Finally, the specific function is implemented in the `loop()` function, and the value of the sensor is judged to be greater than 20000 by the if statement. If it is larger than 20000, the relay is operated by the pump. If it is less than, the relay is turned off and the pump stop working.

```
def loop():

    while True:

        for i in range(100):

            values[i] = adc.read_adc(0, gain=GAIN)

            print(max(values))

            if (max(values))>20000:

                GPIO.output(PIN, GPIO.LOW)

                print("ON")

                print(PIN)

                time.sleep(0.1)

            else:

                GPIO.output(PIN, GPIO.HIGH)

                print("OFF")

                print(PIN)

                time.sleep(0.1)
```

Attention: The 20,000 set here needs to be set according to the moisture required by your own flowers, which is just explained with an example of 20000.

5.The last piece of code is where the function starts and ends. When the program is running, you can press ‘Ctrl+c**’ to end the code run.**

```
if __name__ == '__main__':
    setup()
    try:
        loop()
    except KeyboardInterrupt:
        destroy()
```