Extracted from:

Portable Python Projects

Run Your Home on a Raspberry Pi

The Pragmatic Bookshelf

Raleigh, North Carolina

The Pragmatic Programmers

Portable Python Projects

Run Your Home on a Raspberry Pi

Mike Riley

edited by Jacquelyn Carter

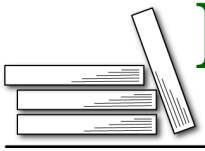
Portable Python Projects

Run Your Home on a Raspberry Pi

Mike Riley

The Pragmatic Bookshelf

Raleigh, North Carolina



For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit *https://pragprog.com*.

The team that produced this book includes:

CEO: Dave Rankin
COO: Janet Furlow
Managing Editor: Tammy Coron
Development Editor: Jacquelyn Carter
Copy Editor: L. Sakhi MacMillan
Indexing: Potomac Indexing, LLC
Layout: Gilson Graphics
Founders: Andy Hunt and Dave Thomas

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

ISBN-13: 978-1-68050-859-8
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—February 2022

As demonstrated in our previous project, the Pi can be used as an inexpensive laptop or desktop replacement, capable of running Python scripts that can also run on other operating systems. What makes the Pi so interesting in the home automation world is its diminutive size and expansive ports and GPIO pins. Sensors and actuators can be attached to the Pi via these interfaces, giving the Pi a wide spectrum of inputs to process. One of the simplest of inputs is a switch.

For this project, we'll use a very basic water sensor that turns on (completes or closes an electrical circuit) when water is detected across its two probes and turns off (breaks or opens an electrical circuit) when water is no longer present across both probes. To make this switch more useful, we'll write a Python script that not only detects if the switch is on or off but also sends an email when water is detected.

Setup

Here's what you need to build this project.

Hardware

- Floor water sensor[1]

Software

- Gmail (or other IMAP-compliant email) account

Attach one of the water sensor's white wires to pin 39, which is associated with ground on the Pi 4, and the sensor's red wire to pin 40, which is associated with GPIO 21. GPIO 21 (pin 40) provides the power source and the ground pin is the drain. When looking at a properly oriented Pi 4, pins 39 and 40 are the furthest-right bottom and top pins, respectively.

Since the water sensor doesn't have female adapters for the Raspberry Pi's GPIO pins, and since it's not a good idea to directly solder the wires directly to the Pi's GPIO pins, I recommend using female to female jumper wires. You can affix the water sensor wires into one end of the jumper wires and then hold it in place either using electrical tape or, my personal favorite, heat-shrink tubing.

If you opt to seal the female jumper wire with the exposed water sensor wiring, use a heat-generating source such as a hair dryer on high setting to shrink the tubing around the connection.

1. https://www.amazon.com/Floor-Water-Sensor-Flood-Detection/dp/B079YB1T8J?th=1

GPIOs and Pi Models

The GPIO Pinout map is different depending on which model of Raspberry Pi you are using. This book assumes you have the Pi 4 Model B+ and will reference these GPIO pin locations accordingly. Here's a useful tip from one of this book's technical reviewers and fellow Pragmatic Bookshelf author Maik Schmidt: if you're running Raspberry Pi OS, you can type in pinout in the Terminal window, and a graphical map and list of all the pinouts for the model of the Pi that you have will be displayed onscreen. Very cool!

Whenever water is detected across both of the water sensor's probes, the electrical circuit will close, meaning the switch is flipped on. When this occurs, GPIO 21 should report that it senses current flowing through it. Refer to the photo on page 7 to see where the water sensor wires should attach to the Pi GPIO pins.

With the male leads of the two water sensor wires converted into female ends, you can easily attach and remove the connection. This will come in handy when adding additional sensors or other attachments that you want to test, either in conjunction with or isolated from the water sensor connection.

Test Script

Before we flesh out a full-blown Python script to handle our email notification workflow, let's first test the water sensor to be certain that it is working as expected.

We'll be calling upon a custom Raspberry Pi library called RPi.GPIO to interrogate the status of the GPIO pins. Like many other Pi-centric/Pi-optimized software, the RPi.GPIO Python library is already pre-installed in Raspberry Pi OS. So there's no need to use Pip to install it.

Create an empty Python file called watersensortest.py. Open the file using your preferred text editor and enter the following Python code:

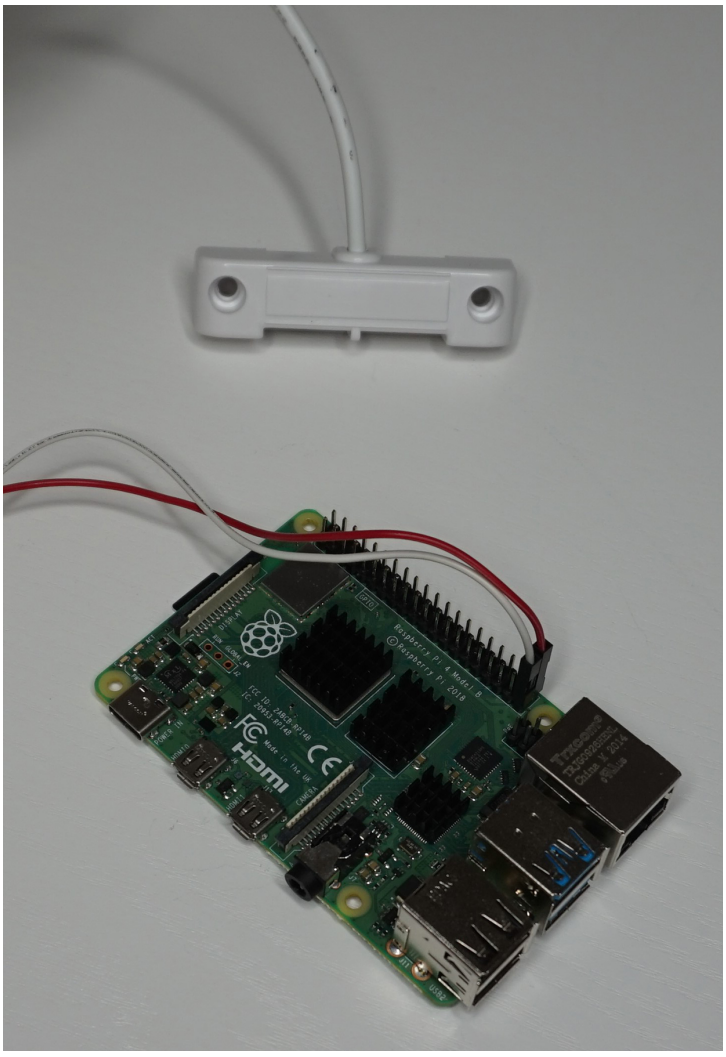
watersensor/watersensortest.py

```
1 import RPi.GPIO as GPIO
  import time

2 GPIO.setmode(GPIO.BCM)
  GPIO.setup(21, GPIO.IN, pull_up_down = GPIO.PUD_UP)

  alert_trigger = False

3 while True:
      if GPIO.input(21):
4         if alert_trigger != True:
```



```
            print("No water detected.")
❺           alert_trigger = True
    else:
        if alert_trigger != False:
            print("Water detected.")
            alert_trigger = False
    time.sleep(1)
```

❶ Import the RPi.GPIO library that we'll use to poll the status of GPIO pin 21.

❷ Initialize the GPIO library and set the status of pin 21 to UP via the GPIO.setup parameter pull_up_down = GPIO.PUD_UP.

❸ Run an infinite loop to poll the status of GPIO 21 every second.

❹ If GPIO 21 is pulled down (that is, water is detected and the circuit is closed), then print that water has been detected. Otherwise print no water detected.

❺ To prevent a constant stream of water/no water messages being printed to the Terminal window, set a variable called alert_trigger to True or False depending on whether the state change message has been outputted already to the Terminal window.

Save and execute the script with python3:

```
$ python3 watersensortest.py
```

If everything is coded error free and your water sensor is working and correctly connected to your Pi, you should see the output of a single line, No water detected. Dip the two metal probes of the water sensor into a small cup of water and check your Terminal window. A new Water detected. message should appear. Remove the sensor from the water and another No water detected. message should be displayed.

You can continue to repeat this test as many times as you prefer. Press the CTRL+C keys on your keyboard when you're satisfied with the accuracy and consistency of the results to interrupt and escape the forever looping While routine in the script.