

Raspberry Pi Pumpdown Tester

Chris Zylstra
czylstra@ophardt.com
22 April 2022

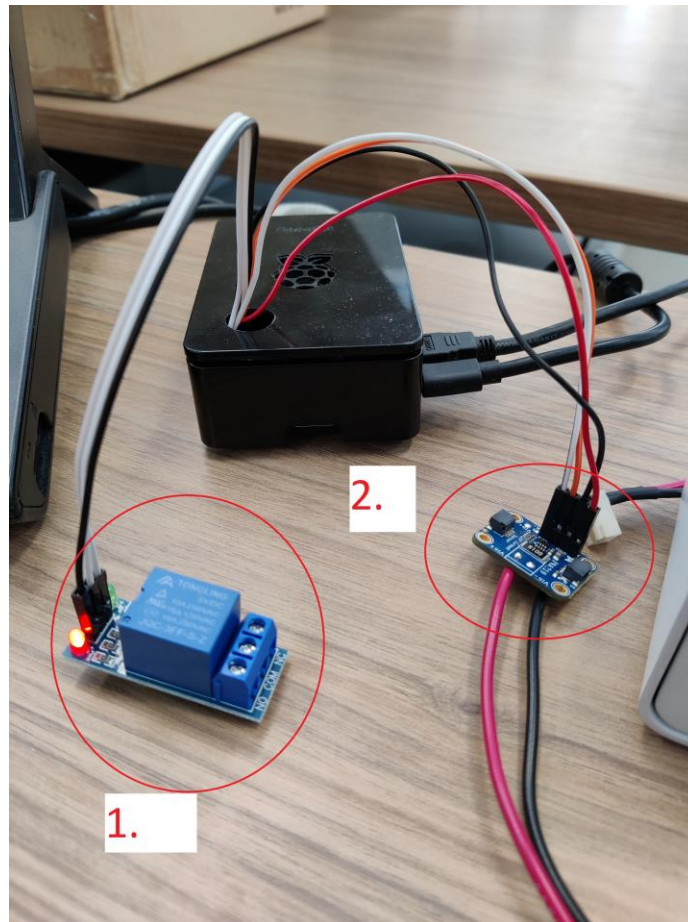
Hardware Setup

Raspberry Pi Setup:

1. Connect HDMI to monitor (micro hdmi in box)
2. Connect USB-C to power (adapter is 240/50hz compatible)
3. Connect Keyboard/Mouse

GPIO Pinout reference: <https://www.bigmessowires.com/2018/05/26/raspberry-pi-gpio-programming-in-c/>

There are two devices connected to the raspberry pi. These are already connected for your two units.



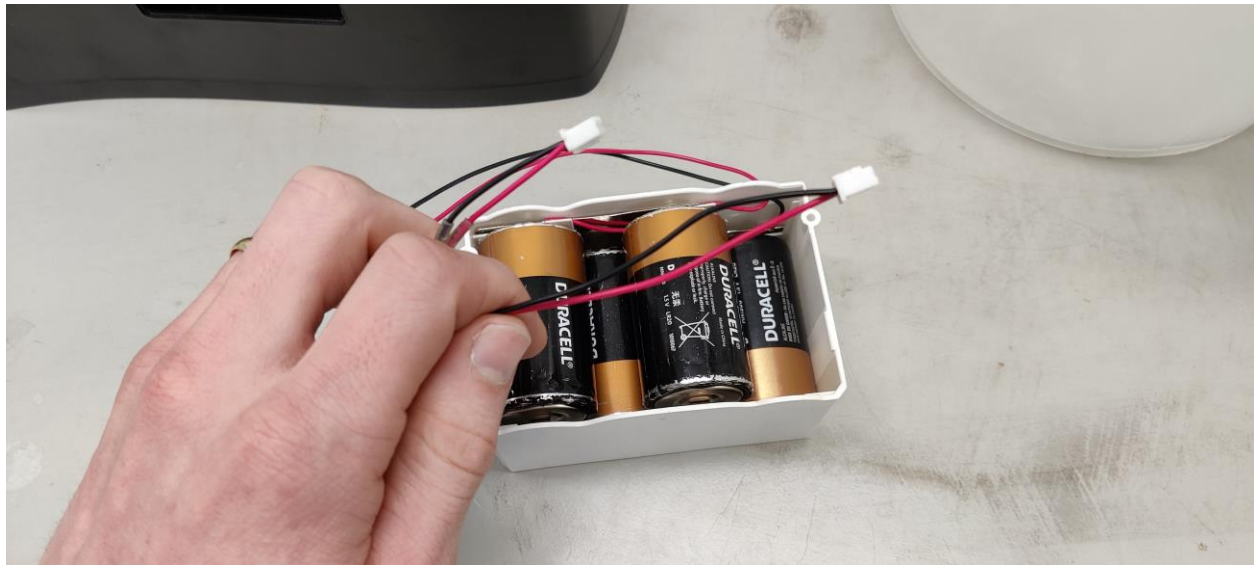
1. Relay

- a. VCC to 3.3V pin on Rpi (pin 1). Note: NOT 5V, I had problems with this.
- b. GND to gnd pin on Rpi (pin6).
- c. IN to Pin 7
- d. Com to power supply gnd (V-)
- e. NC to solenoid wire #1
- f. Solenoid wire #2 to power supply 24V (V+)

2. I2C Multimeter (only using voltage measurement for batteries)

- a. VCC to pin 11 NOT 5V otherwise it draws power from the batteries it is measuring.
- b. GND to gnd pin on Rpi (pin 9)
- c. SDA to sda pin on Rpi (pin 3)
- d. SCL to scl pin on Rpi (pin 5)

I have made split out battery packs for our measurement. One leg goes to the dispenser and one leg goes to the Multimeter so that I can measure the voltage over the lifetime of the batteries (per cycle):



Software Setup

Username: pi

Password: pumpdown

*You should not need to use the username/password. It should log in automatically.

On the desktop there is a file called *Pumpdown.sh*. Double click to run the software. This file just executes a python file called *pumpdown.py* located in Documents folder.

If it is not executing try executing in terminal (to see error messages):

```
cd Documents
python pumpdown.py
```

You can download my latest version of pumpdown.py from:

<https://github.com/chriszylstra/ophardt/blob/main/PumpDown%20Tester/pumpdown.py>
<https://github.com/chriszylstra/ophardt/blob/main/PumpDown%20Tester/Pumpdown.sh>

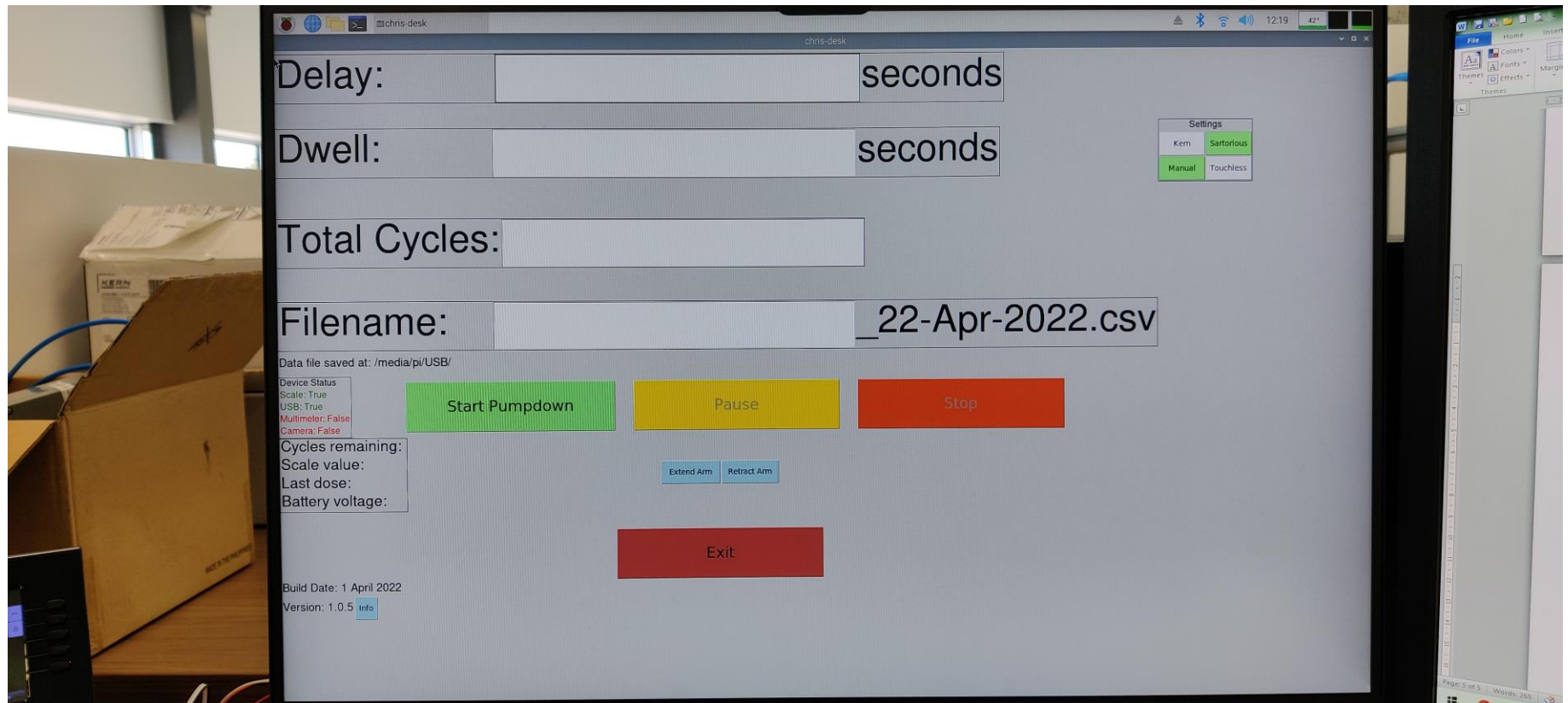
*place pumpdown.py in the Documents folder

*place Pumpdown.sh on the Desktop.

If running on a new Raspberry Pi you must execute the following command to make file executable:

```
cd Desktop
chmod +x Pumpdown.sh
sed -i -e 's/\r$//' Pumpdown.sh
./Pumpdown.sh
```

Using Pumpdown Software;



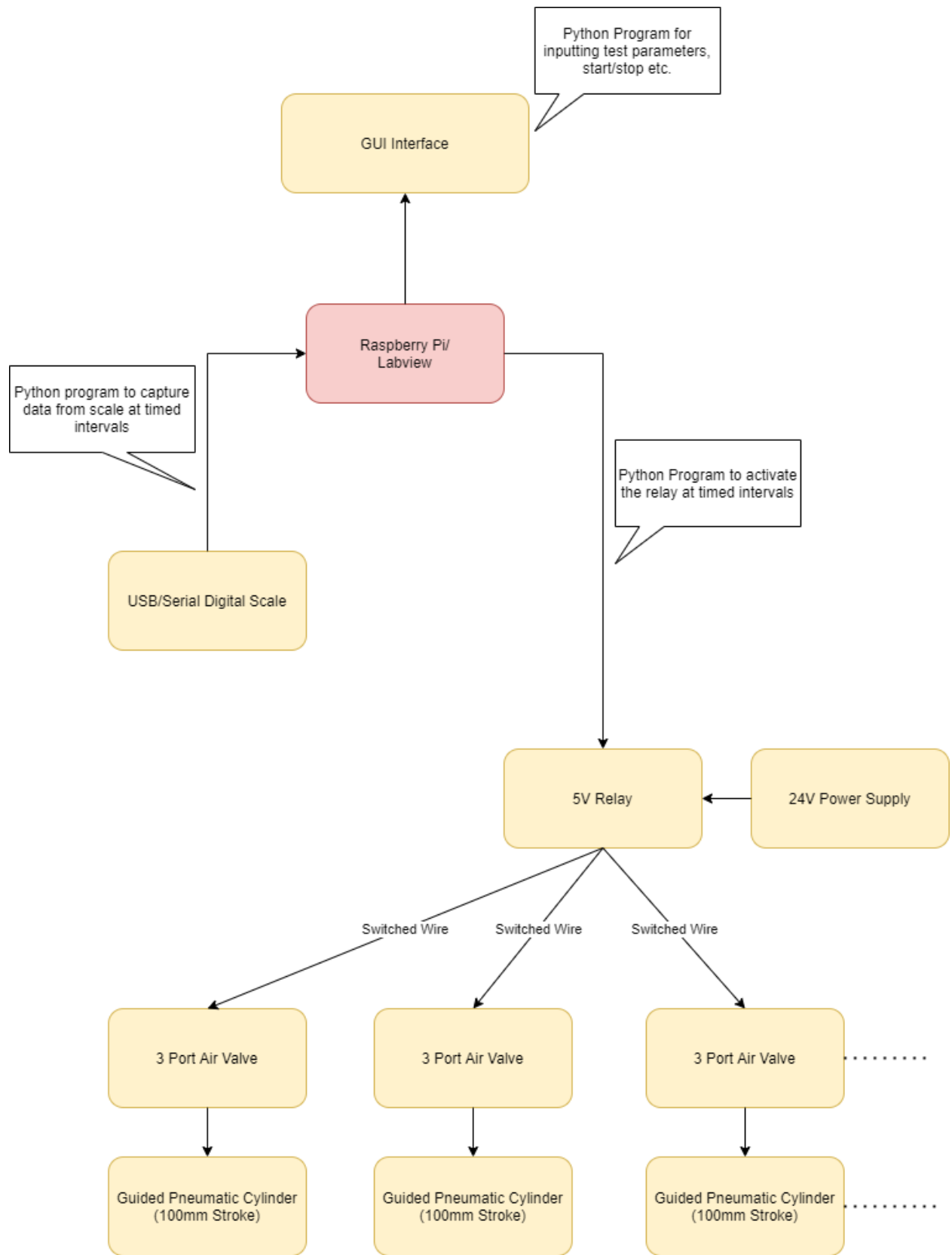
1. Delay: the number of seconds between activations (integer >1)
2. Dwell: the number of seconds to hold an activation (integer >1)
3. Total Cycles: the number of activations in a session.
4. Filename: What to call the data file. It is concatenated with the date... stored at /media/pi/USB .

-You must have a USB stick inserted into the Pi to save data. It must be named exactly “USB” otherwise data will not be saved. This could be changed in the python code if desired. *Already setup on the two units sent.

5. Device Status: Shows the status of the connected devices; Weigh Scale (Serial), USB stick, Multimeter (i2c), Camera (USB, not functioning right now)
6. Start/Pause/Stop: Used to control the test
7. Info Box shows the current data from the test. This data is stored to the CSV file on the usb every activation.
8. Extend /Retract Arm: For manual control of the relay.
9. Settings: Kern/Sartorius Scales have different communication protocols. Select the one which matches the scale you are using.
10. Settings: Manual/Touchless: Switches the dwell/delay values in software for testing Touchless dispensers. (Has no effect on 50/50 duty cycle)

Notes:

1. Battery Multimeter VCC must be connected to gpio 17 (pin 11) and not 5V otherwise it will affect the battery reading by drawing power.
2. Battery Multimeter reads random values if it is floating(not connected to anything). Just fyi.
3. Use caution with the live ac power connectors to the 24V power supply. You may prefer a safer solution such as a laptop charger that you cut into the DC side.
4. I have used VNC server for auxiliary stations here so that I can remote into it and control it. Then I only need 1 keyboard/mouse/monitor for many stations. I have not set this up on your system but I can assist if you want.
5. I am working on adding a USB webcam to the system so there is some code artifacts for this. It's not functional yet.



Images of the Frame Assembled:

