Tim Poulsen

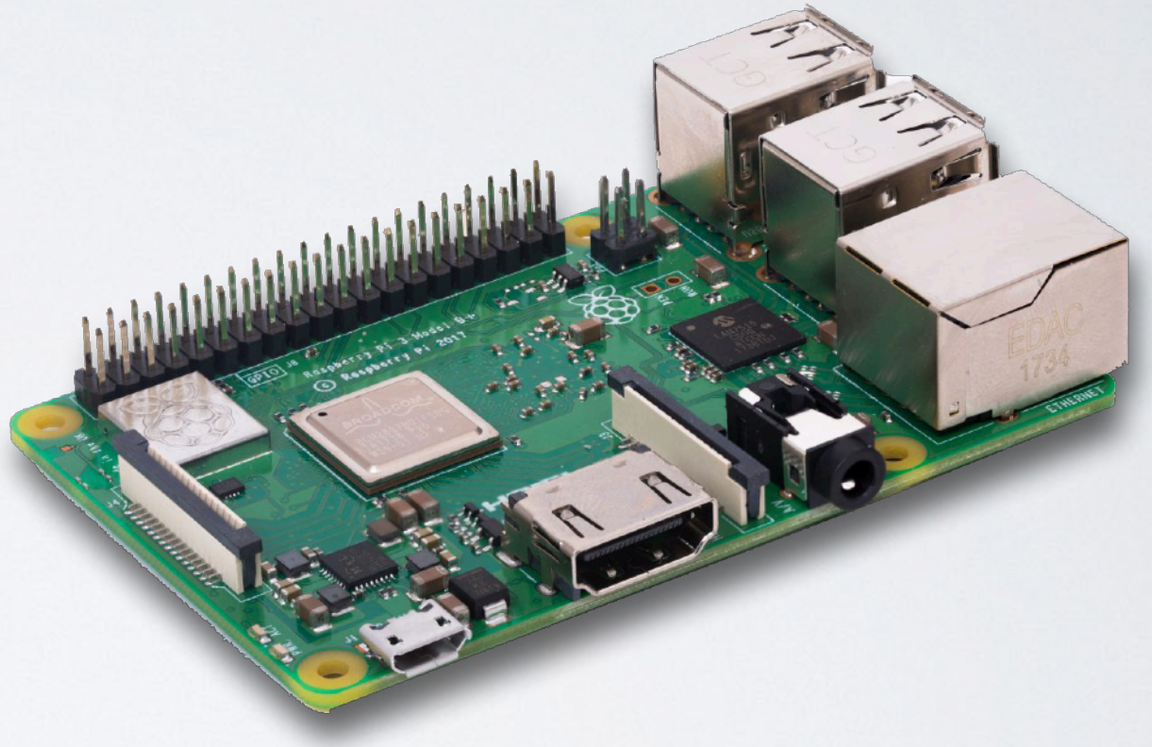
Software Engineer

R&D Team

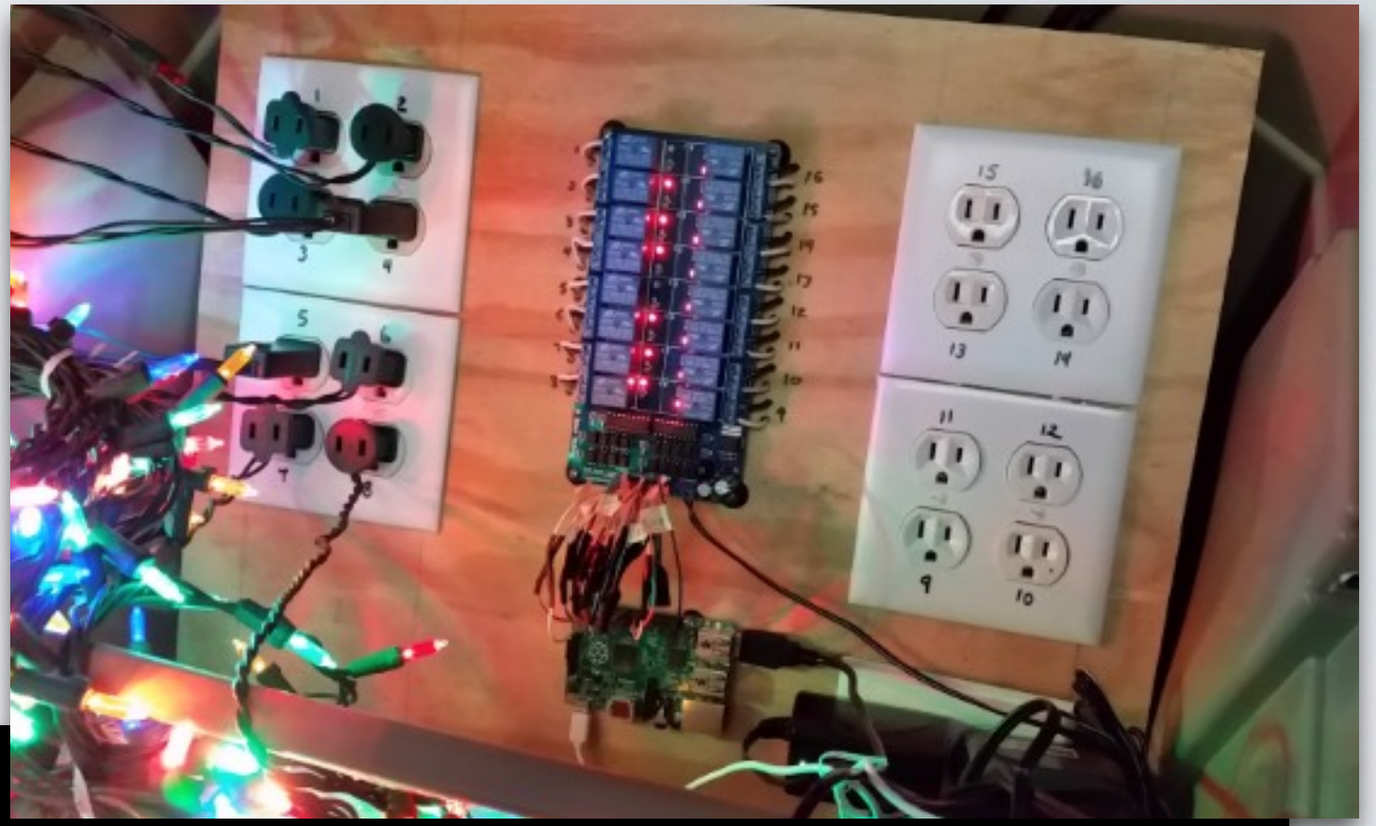
ACV Labs

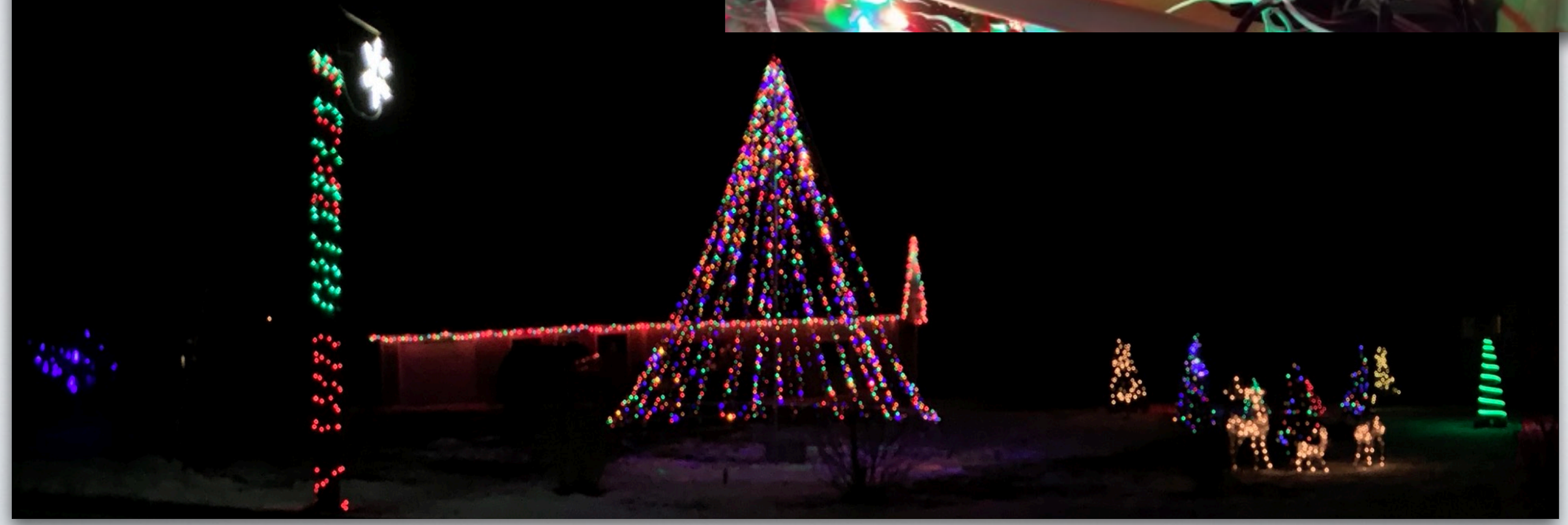
Raspberry Pi

Overview & Examples
Board components
Basic setup
Python dev
Libraries
Resources



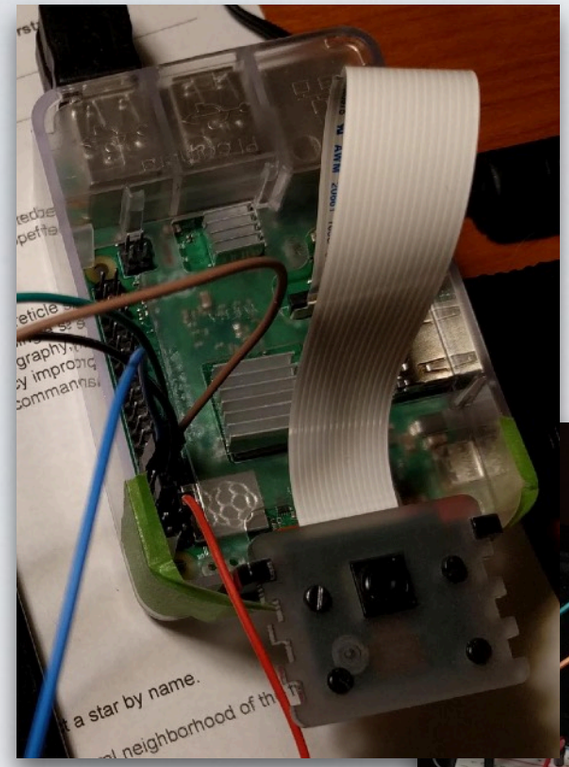
https://github.com/skypanther/PiLit

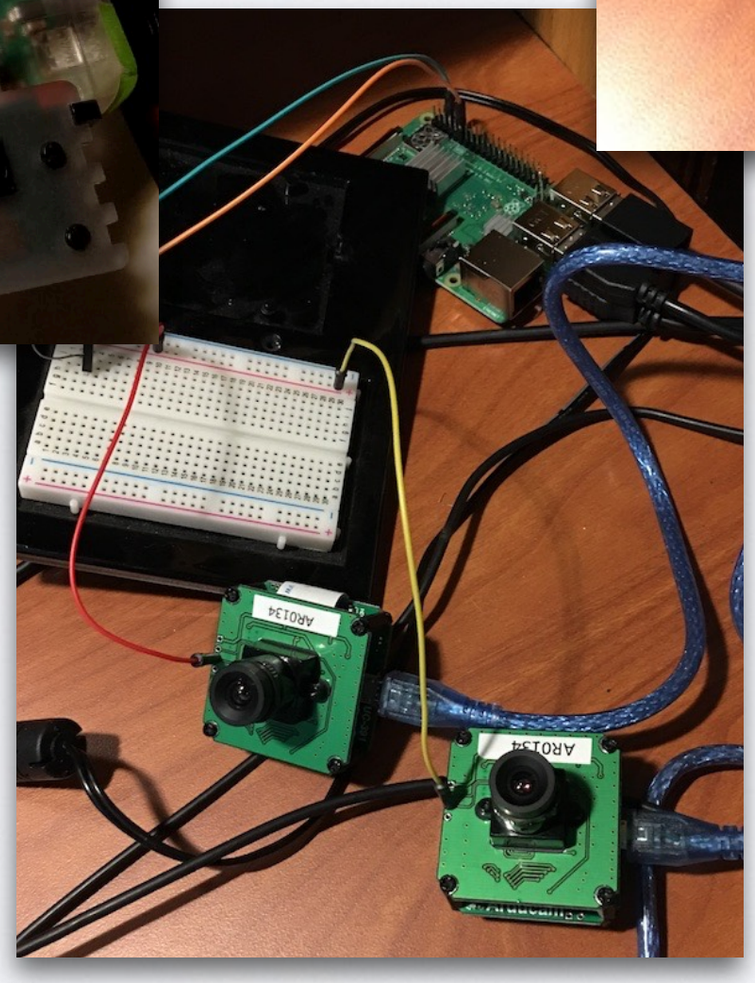


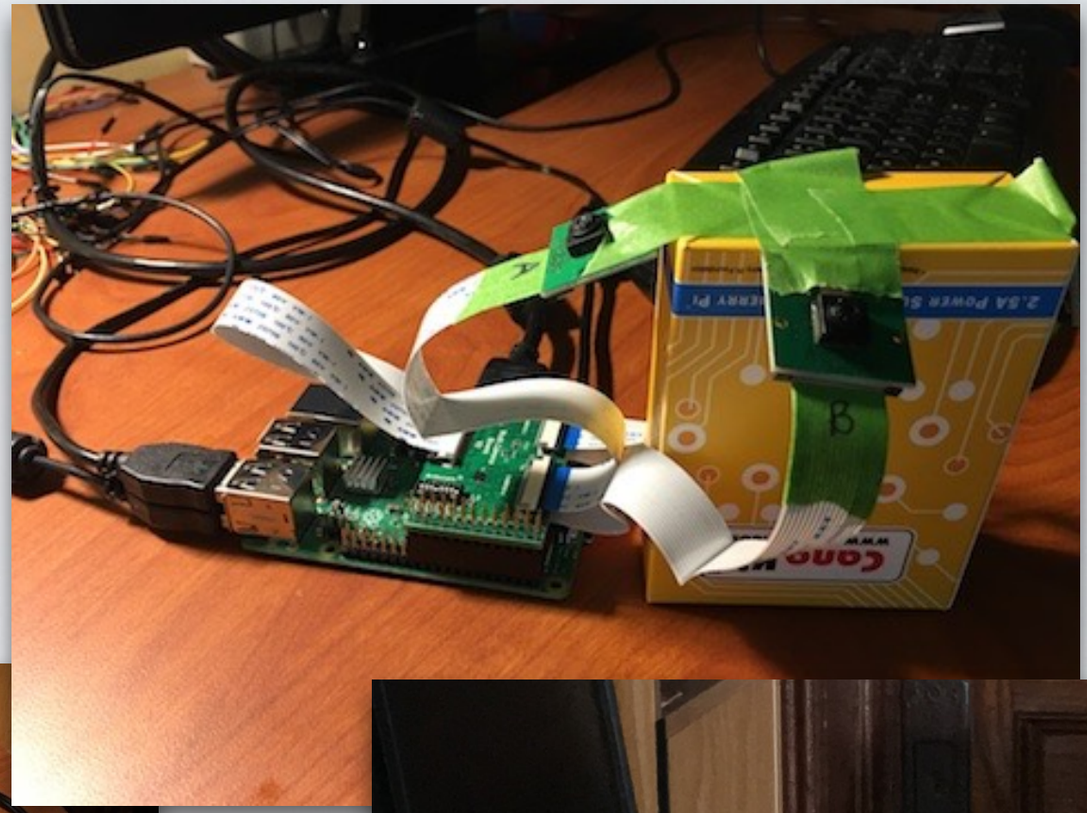


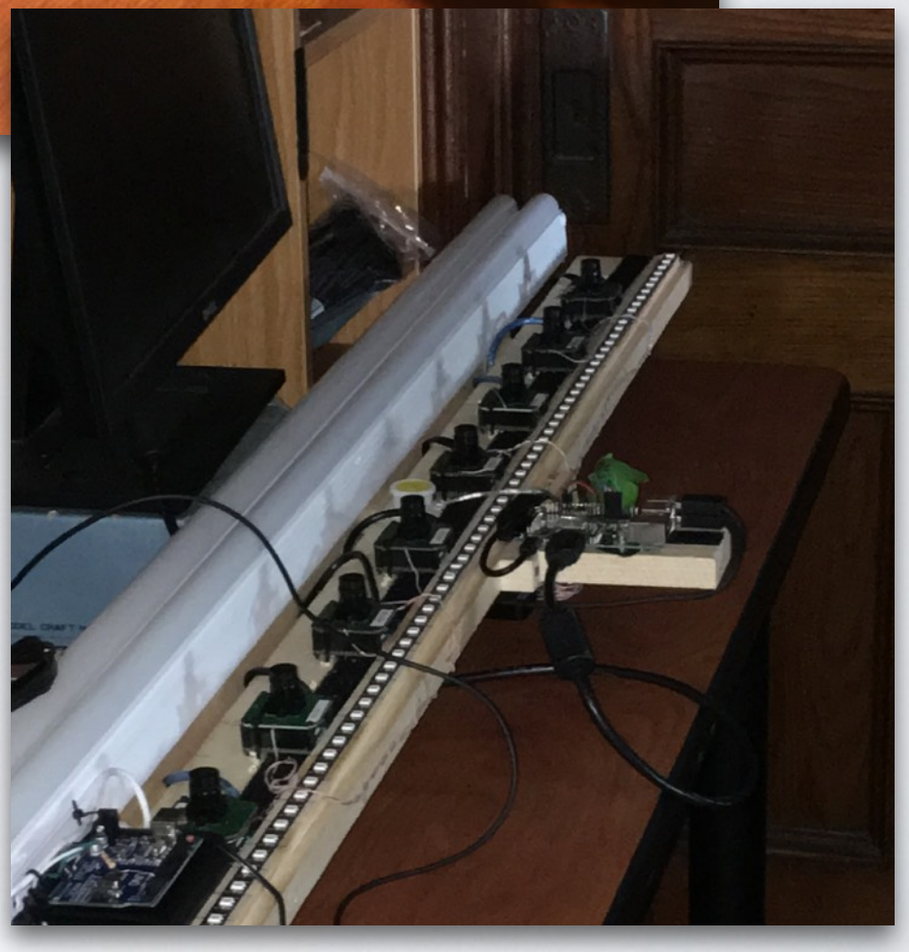
https://www.timpoulsen.com/2018/pi-birdcam.html

Camera controller board









PyImageSearch blog

www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/

PI 3B+

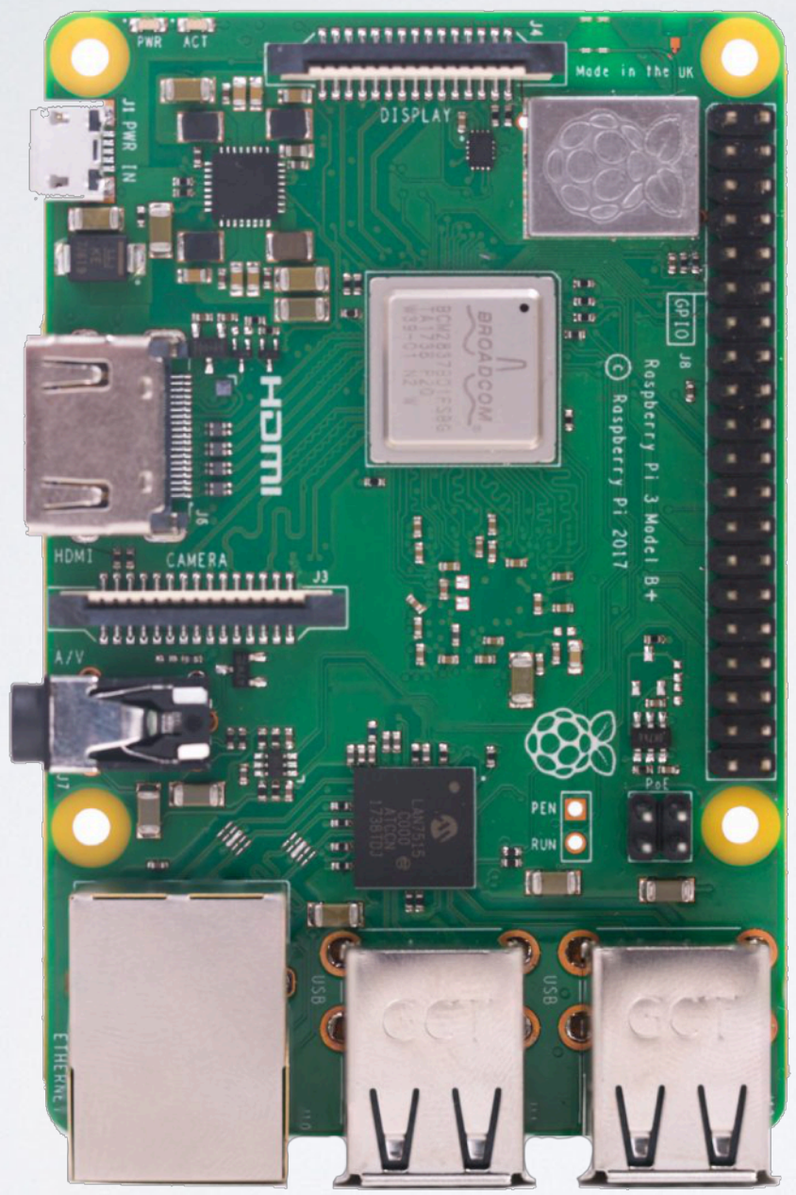
Micro SD slot (back side)

Power in (micro-USB)

HDMI

CSI camera

Stereo audio out



Gigabit Ethernet

4 x USB 2.0

Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz

1GB LPDDR2 SDRAM

2.4GHz and 5GHz IEEE 802.11.b/g/n/ac

Bluetooth 4.2, BLE

GPIO, I2C, serial

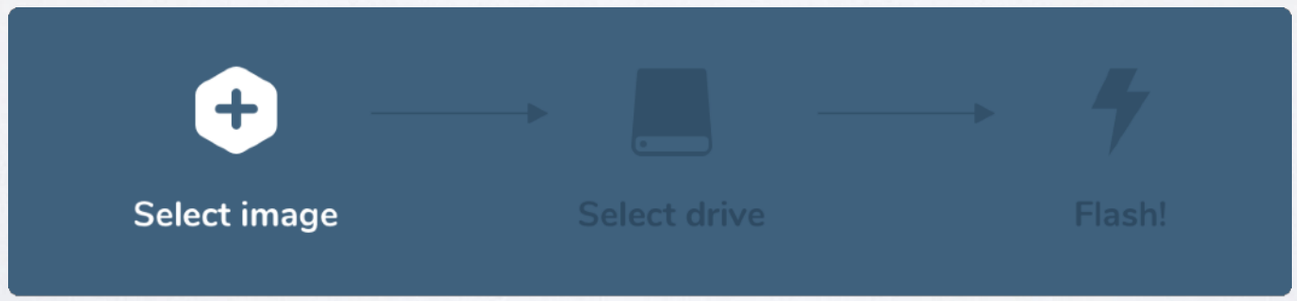
OS OPTIONS

- Raspbian (customized Debian distro)
  - *NOOBs - an installer, not an OS*
- Third-party provided Ubuntu, Windows 10 IoT Core, RISC OS, etc.

OS SETUP

- Download OS “image” .img file
- Copy to SD card
- Insert in Pi and boot

Etcher app - https://www.balena.io/etcher/



DEV ENVIRONMENT

- Python 2.x / 3.x
- C
- C++
- Java
- Scratch
- Ruby
- Wolfram language
- SonicPi
- Git

IDEs

Pre-installed:

- Thonny (Python)
- Geany (multiple)
- BlueJ (Java)
- Greenfoot (Java)
- Mathematica
- Node-RED
- Scratch

Installable:

- VS Code - https://code.headmelted.com/
- Spyder
- Ninja-IDE
- Lazarus
- lots more…

PYTHON LIBRARIES

Pre-installed:

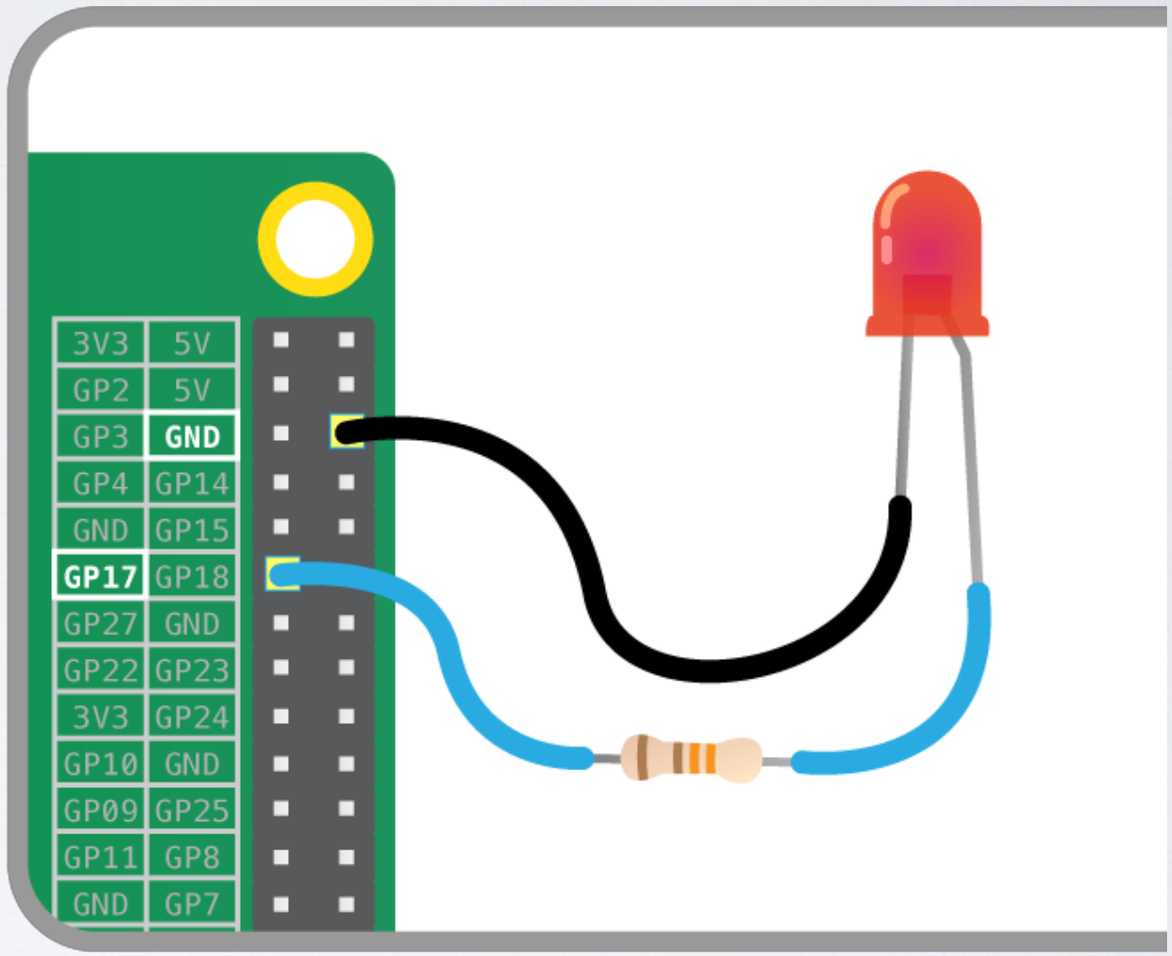
- RPi.GPIO
- GPIO Zero
- PiCamera

Installable:

- Any Python-only library
- Or, any that will compile for ARM
- www.piwheels.org - precompiled packages
  - e.g. OpenCV, Tensorflow, WiringPi, etc.

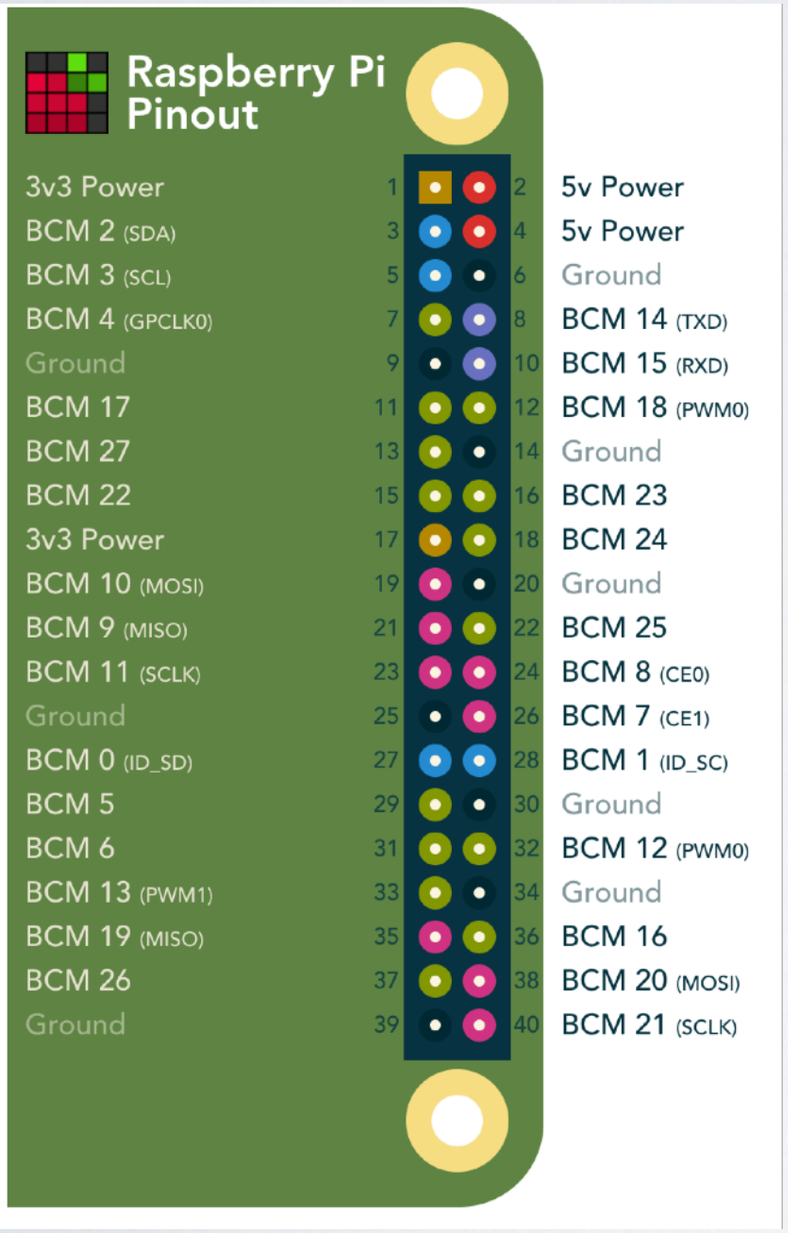
GPIO BASICS

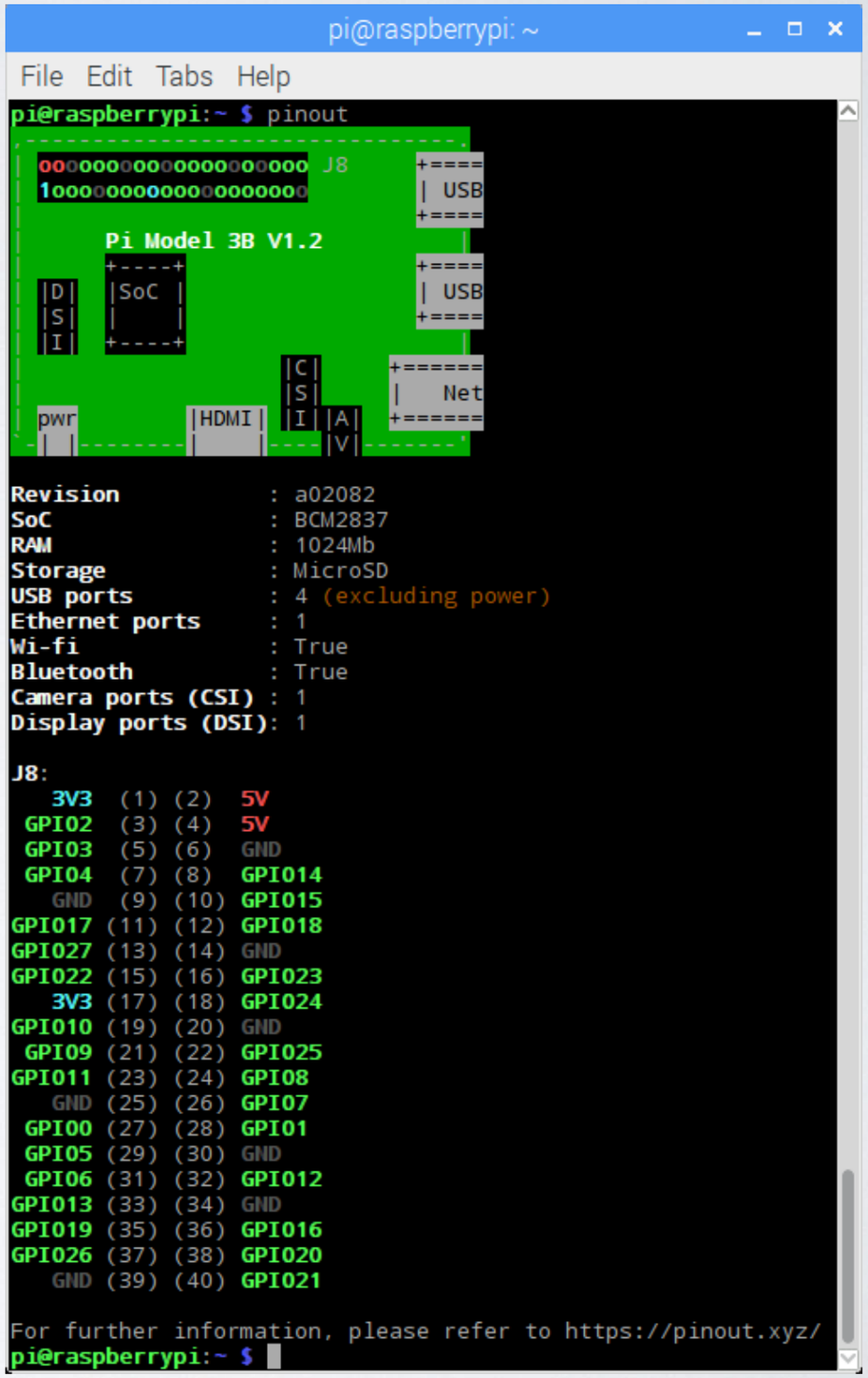
```
from gpiozero import LED
from time import sleep

red = LED(17)

while True:
    red.on()
    sleep(1)
    red.off()
    sleep(1)
```



GPIO / I2C / SERIAL

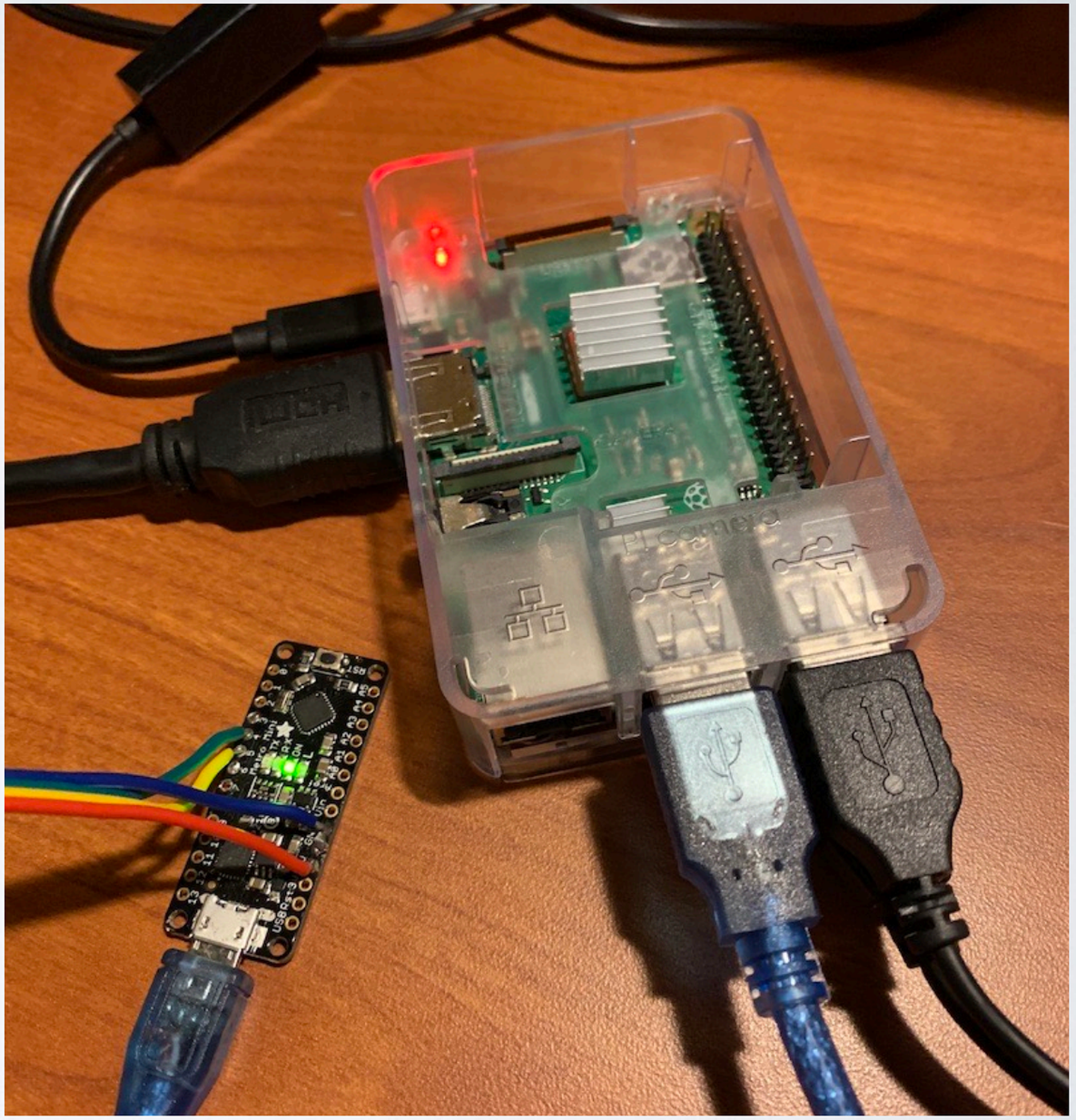
https://pinout.xyz/





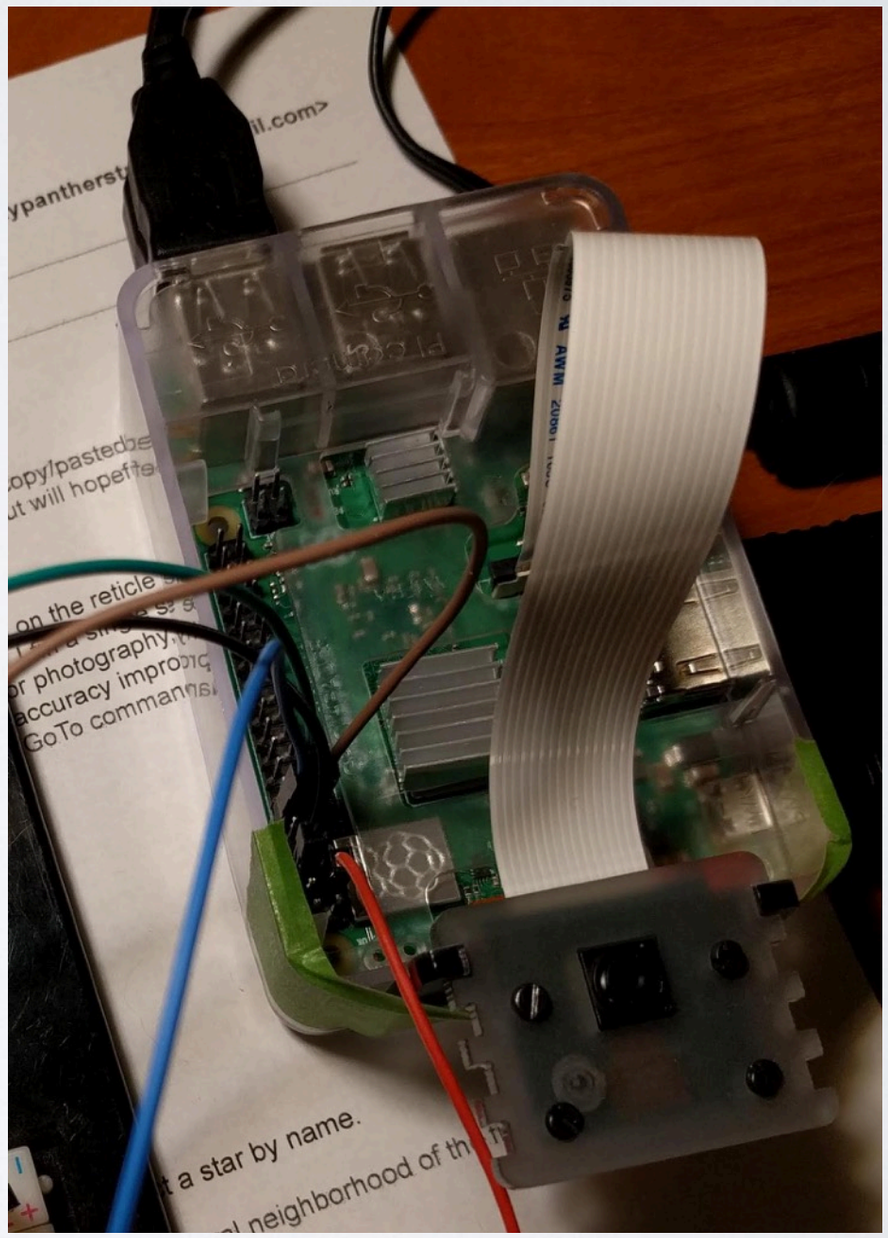
SERIAL OVER USB

```
import serial

# open serial port
ser = serial.Serial('/dev/ttyUSB0')

# write a string
ser.write(b'hello')

# close port
ser.close()
```



PiCAMERA

```
from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.resolution = (1024, 768)
camera.start_preview()

# Camera warm-up time
sleep(2)
camera.capture('foo.jpg')
```



THANKS!

- timpoulsen.com
- github.com/skypanther
- @skypanther
- www.linkedin.com/in/timpoulsen