

**Tim Poulsen**

Software Engineer

ACV Labs (R&D team)



# RASPBERRY PI

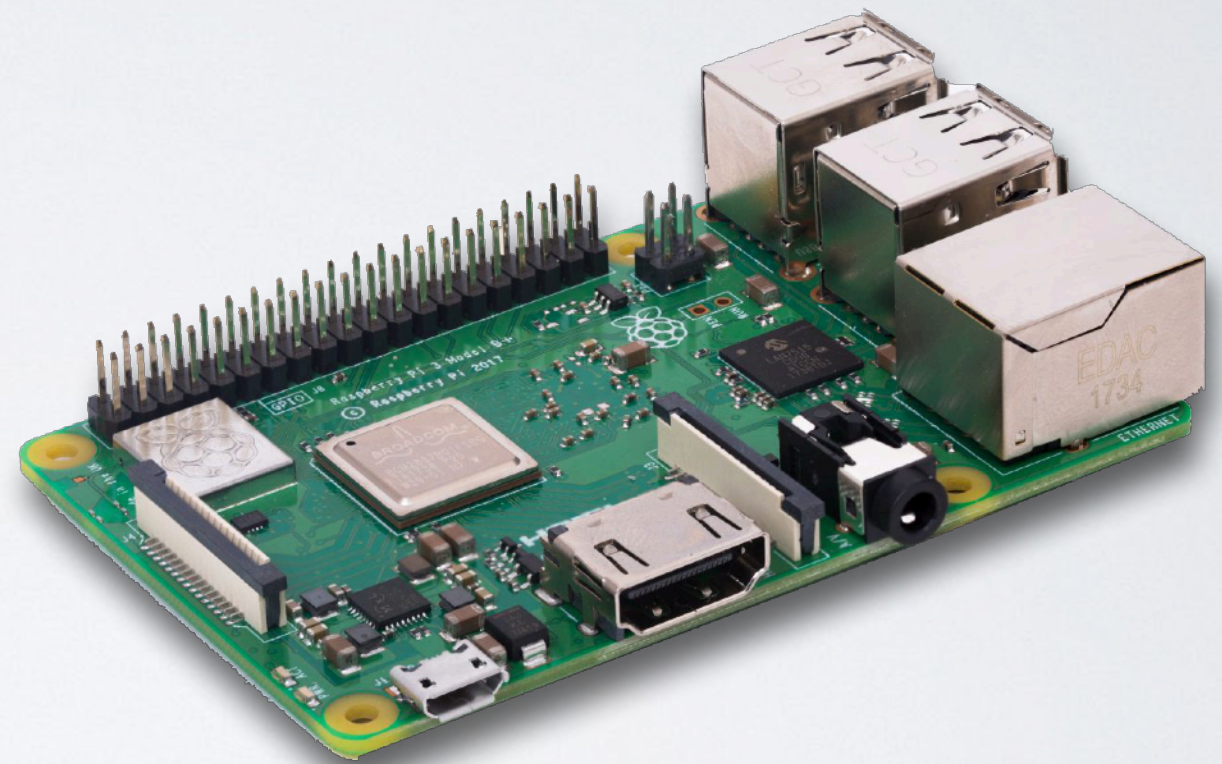
Overview & Examples

Board components

Basic setup

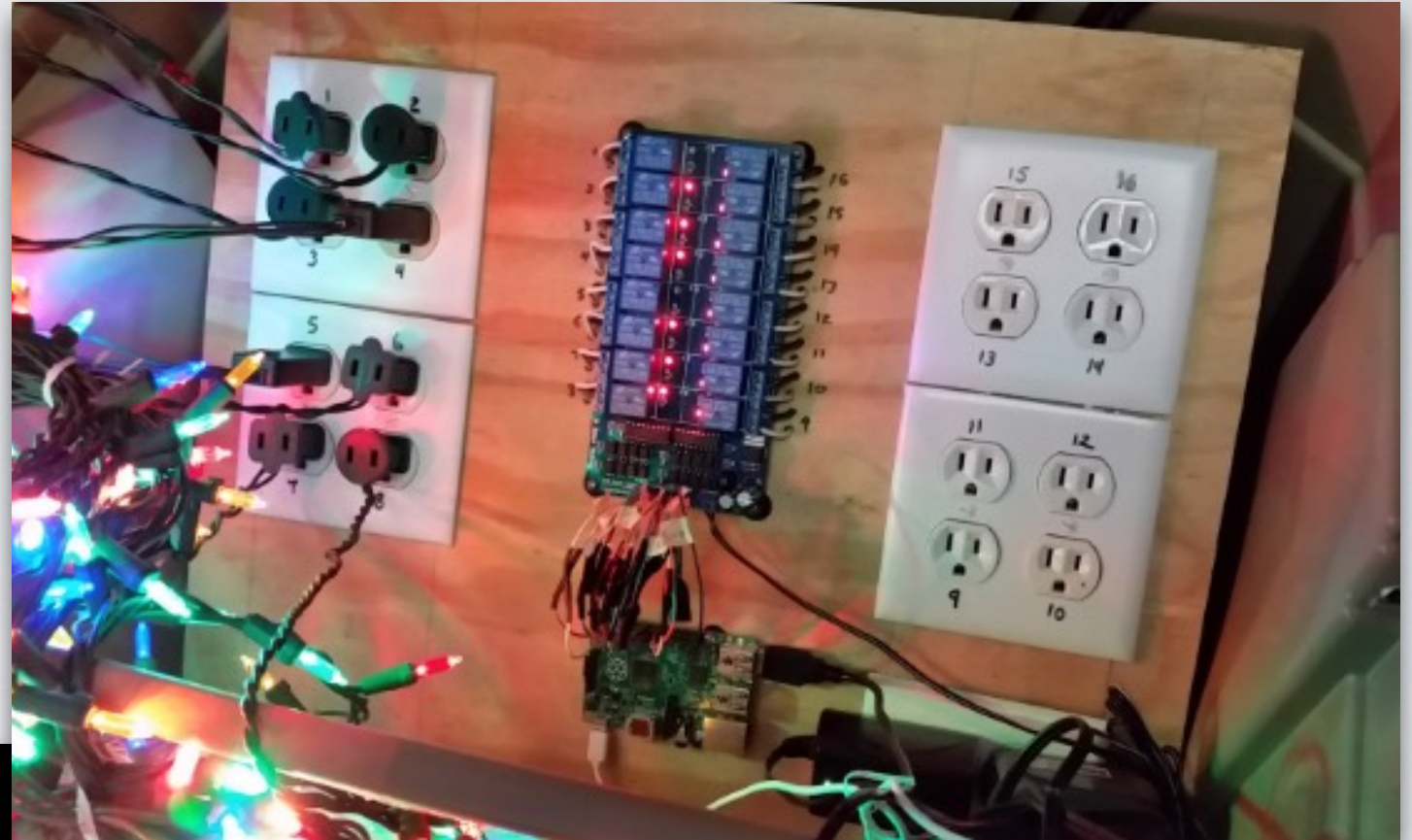
Software dev

Resources





<https://github.com/skypanther/PiLit>



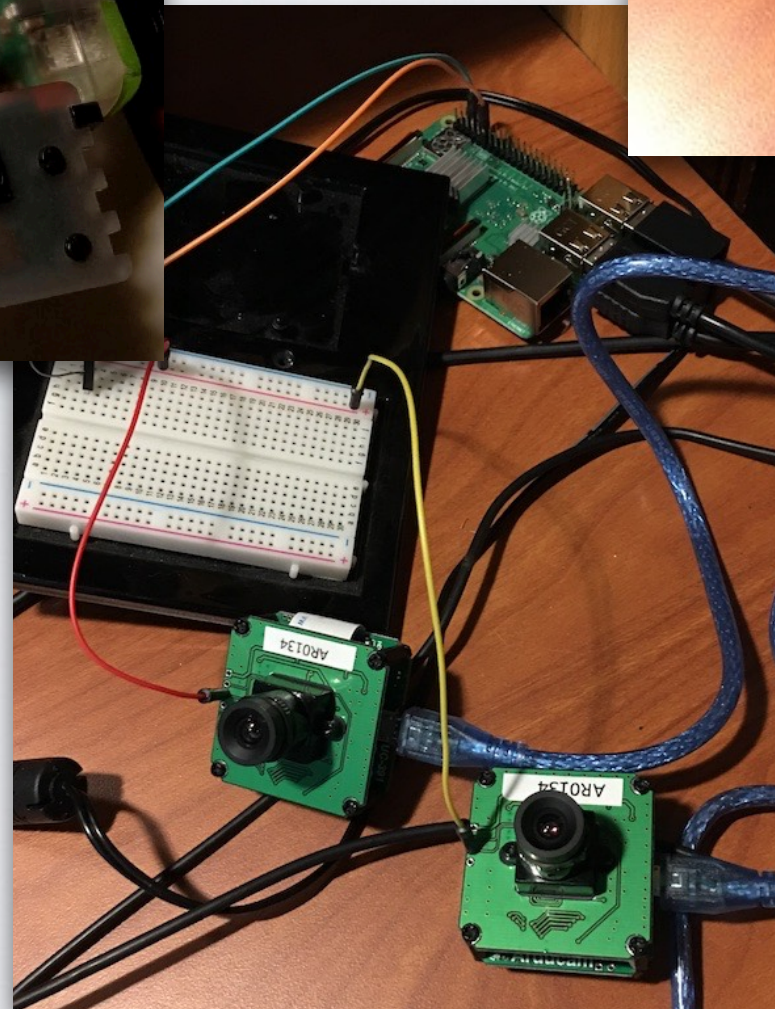
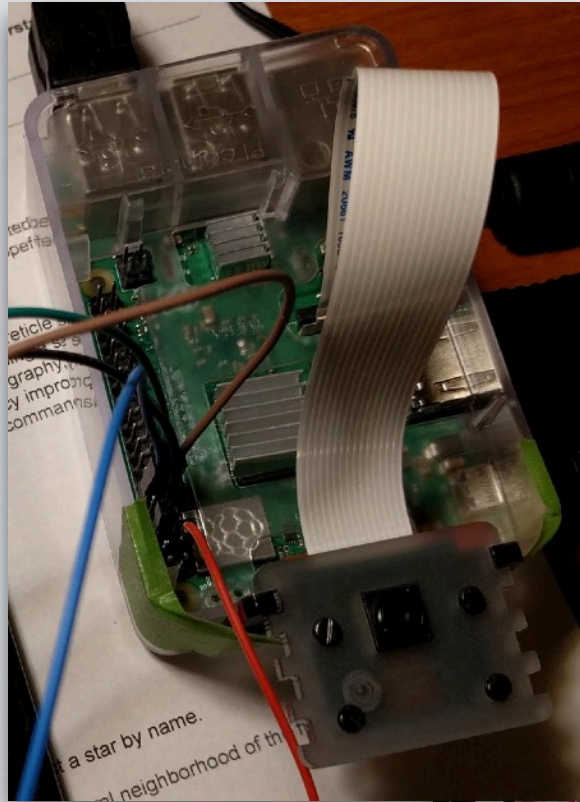


<https://www.timpoulsen.com/2018/pi-birdcam.html>





# Camera controller board





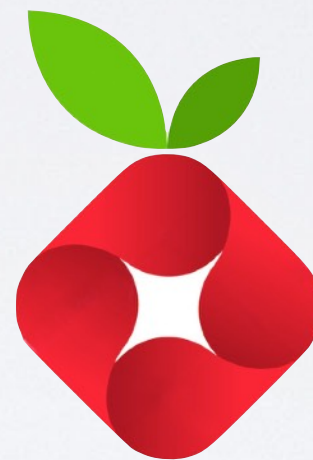
# POPULAR PROJECTS



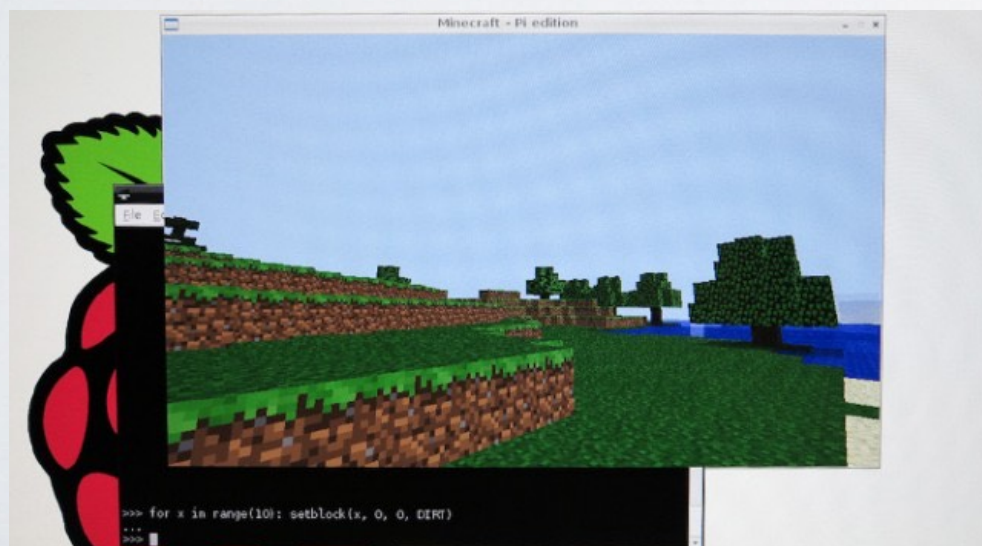
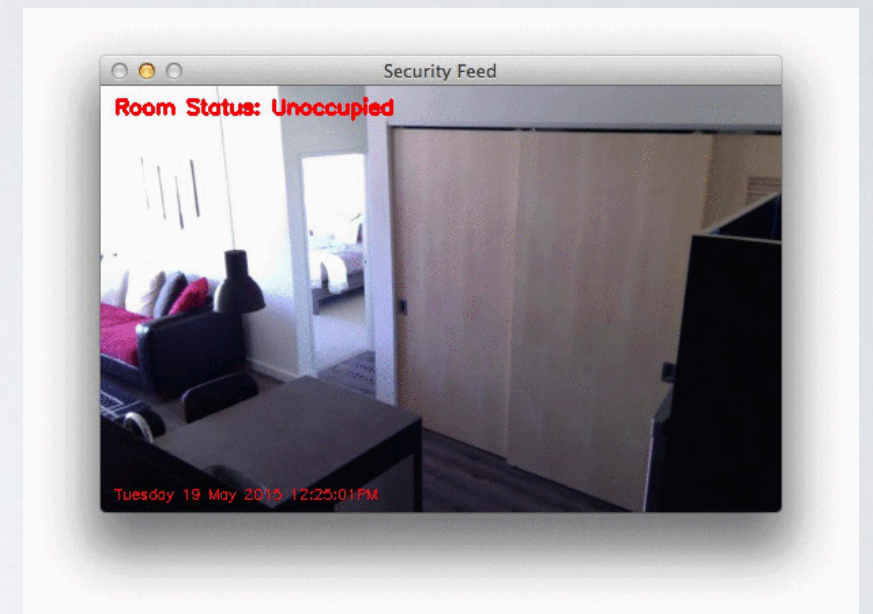
VPN



File server



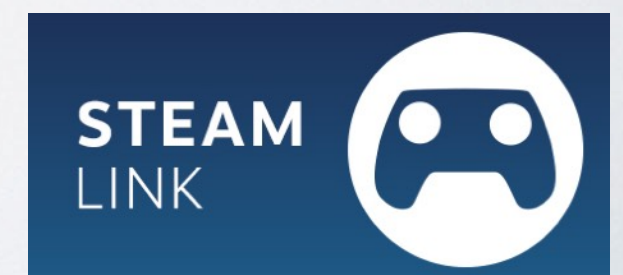
Pi-Hole



Web server

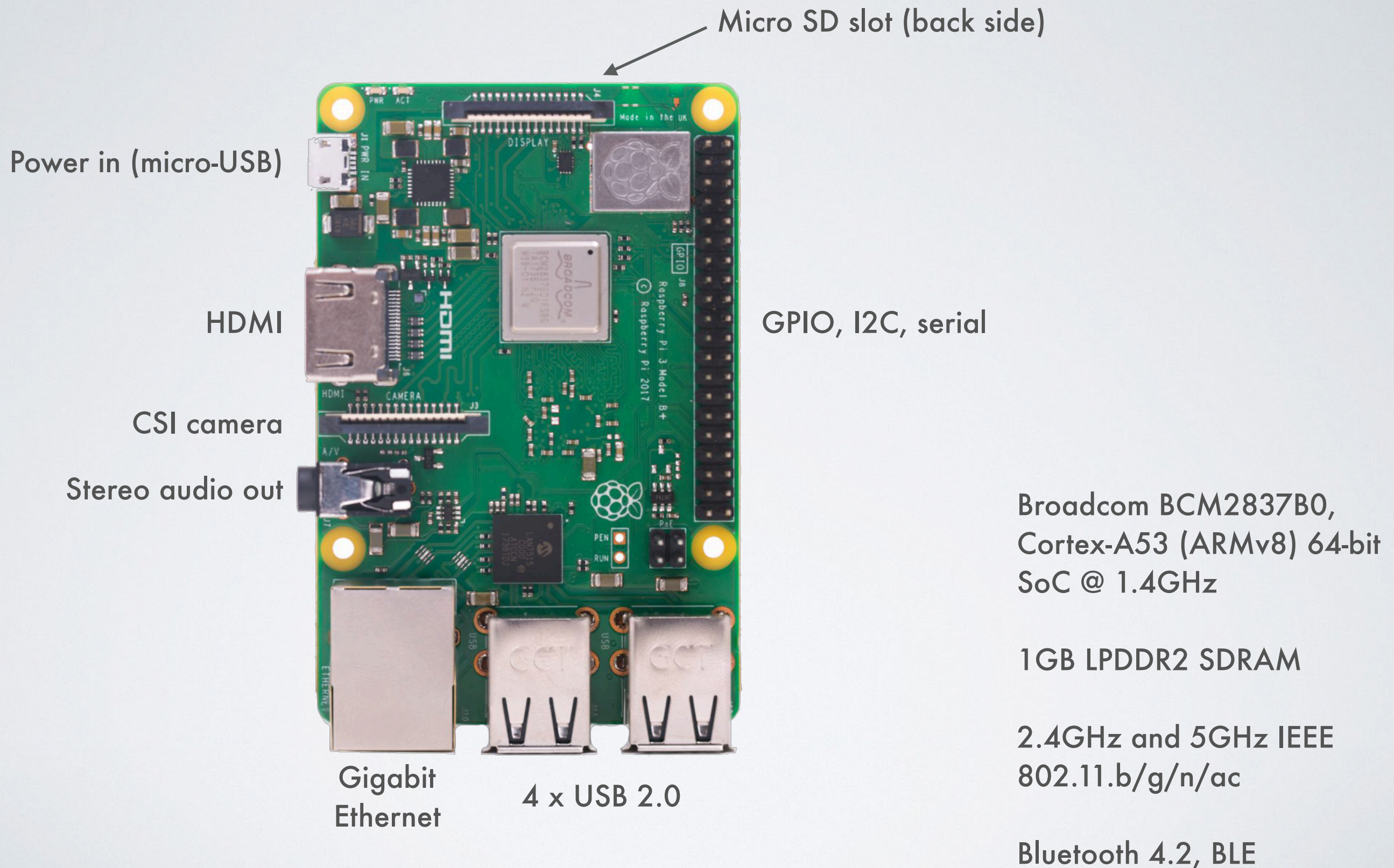
RaspbiCast  
(Chromecast)

AirPlay receiver

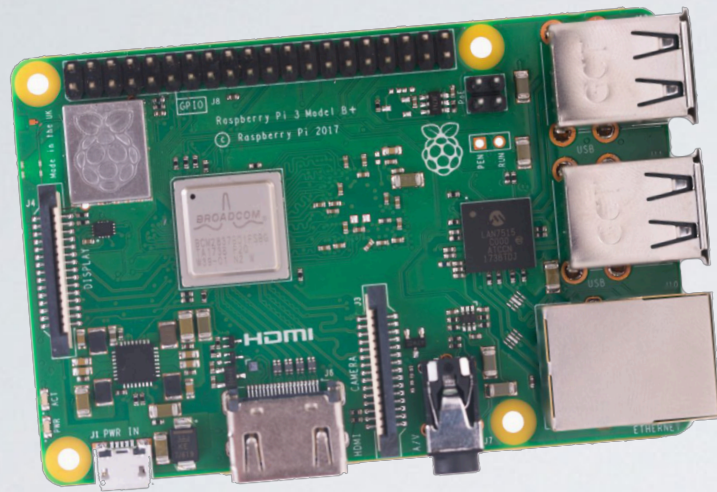




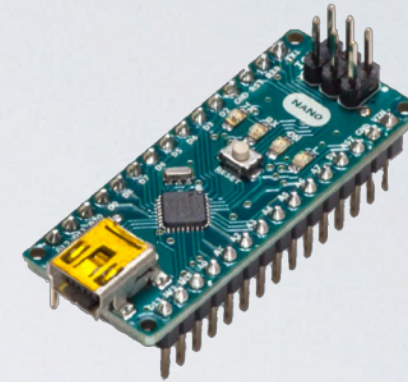
# PI 3B+







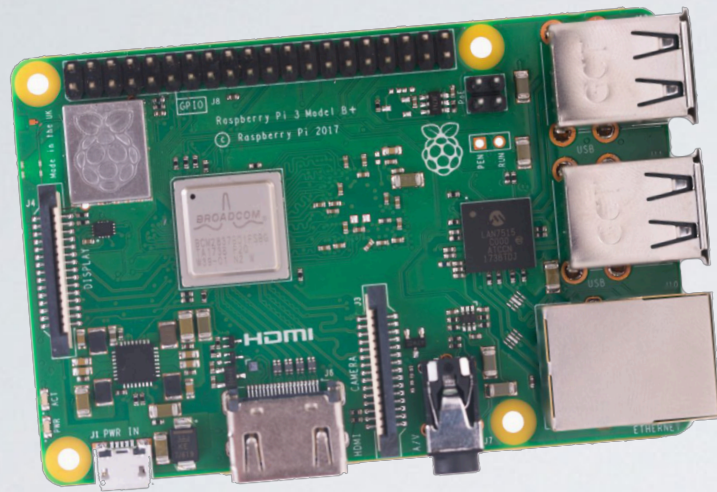
VS



- Full computer
- Full OS
- Peripheral ports, networking, HDMI, audio
- Best for:
  - Software-based projects
  - Complex projects

- CPU, RAM, GPIO
- No OS
- Requires separate computer for programming
- Best for:
  - Hardware-based projects
  - Simple / single-task projects
  - Real-time





## VS Alternatives

- Widest hardware / add-on options
  - Widest software / library options
  - Most tutorials, blogs, community support
  - The standard to which others are compared
  - Various sub-types:
    - Pi 3B+ (standard)
    - Pi Zero / Pi Zero W
- BeagleBone, OrangePi, BananaPi, Tinker, ODROID, etc.
  - Better hardware specs
  - Wider OS options such as Android, full Windows 10 (not IoT core), etc.
  - Fewer hardware / add-on options
  - Fewer software / library options, more compatibility problems
  - Limited community support and info



# OS OPTIONS

- Raspbian (customized Debian distro)
  - *NOOBs - an installer, not an OS*
- Third-party provided    Ubuntu, Windows 10 IoT Core, RISC OS, etc.
- Raspberry Pi Desktop
  - *Run Raspbian on a PC/Mac*
  - *VirtualBox or native install*



# RASPBIAN OPTIONS



## Raspbian Stretch with desktop and recommended software

Image with desktop and recommended software based on Debian Stretch

Version: November 2018  
Release date: 2018-11-13  
Kernel version: 4.14  
Release notes: [Link](#)

[Download Torrent](#) [Download ZIP](#)

SHA-256: 0ca644539fdaf4e19ec7cebf9e61c049b82ba45b1a21cdec91fa54bd59d660d2



## Raspbian Stretch Lite

Minimal image based on Debian Stretch

Version: November 2018  
Release date: 2018-11-13  
Kernel version: 4.14  
Release notes: [Link](#)

[Download Torrent](#) [Download ZIP](#)

SHA-256: 47ef1b2501d0e5002675a50b6868074e693f78829822eef64f3878487953234d



## Raspbian Stretch with desktop

Image with desktop based on Debian Stretch

Version: November 2018  
Release date: 2018-11-13  
Kernel version: 4.14  
Release notes: [Link](#)

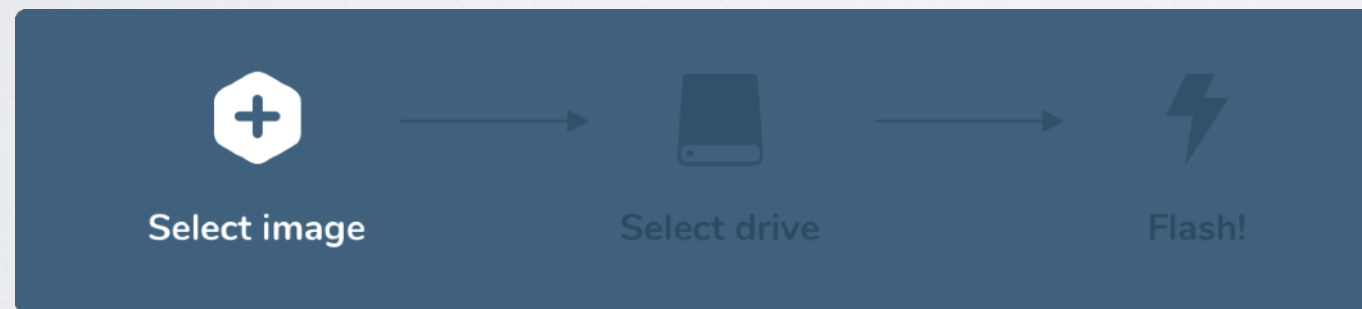
[Download Torrent](#) [Download ZIP](#)

SHA-256: a121652937ccde1c2583fe77d1caec407f2cd248327df2901e4716649ac9bc97



# OS SETUP

1. Download OS “image” .img file
2. Copy to SD card with `dd` or Etcher



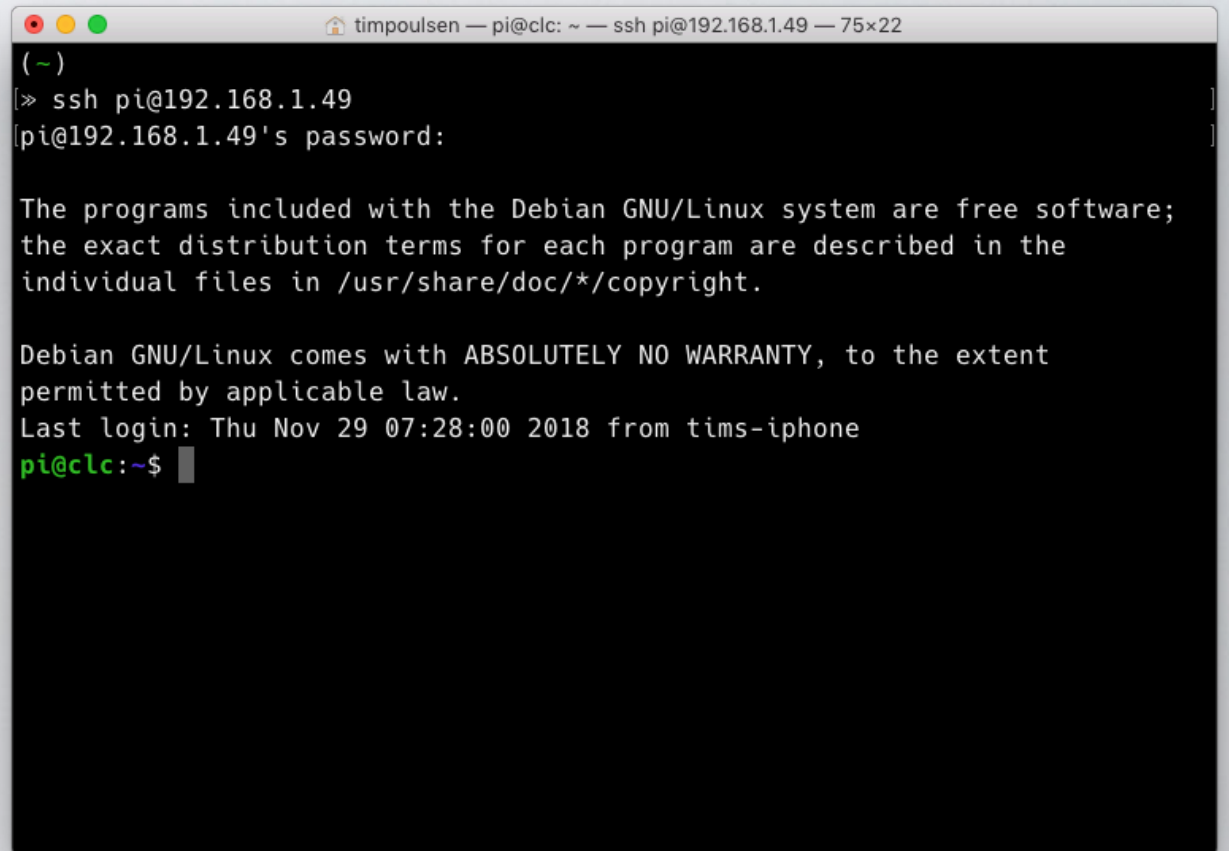
*Etcher app - <https://www.balena.io/etcher/>*

3. Insert in Pi and boot



# HARDWARE ACCESS

- Local
  - USB keyboard & mouse, HDMI monitor
- Remote / headless
  - ssh
    - Enabled by default in current Raspbian builds

A terminal window titled 'timpoulsen — pi@clc: ~ — ssh pi@192.168.1.49 — 75x22'. The window shows an SSH session. The user enters 'ssh pi@192.168.1.49' and is prompted for the password. After entering the password, the terminal displays the Debian GNU/Linux system's free software notice and the last login time. The prompt 'pi@clc:~\$' is visible at the bottom.

```
(~)
[> ssh pi@192.168.1.49
[pi@192.168.1.49's password:

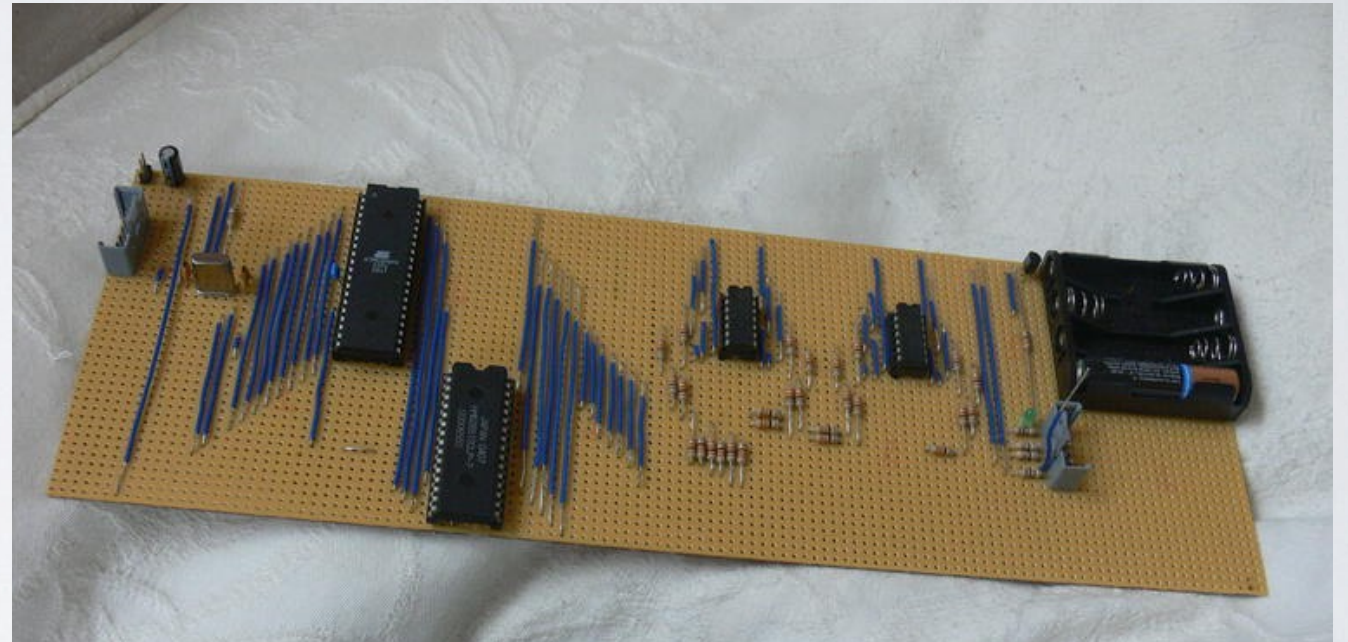
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 29 07:28:00 2018 from tims-iphone
pi@clc:~$
```



# EARLY VISION

- Computer on which kids could learn programming
- Inspired by Acorn's BBC Micro
- Meant to boot directly to a Python prompt
- The “Pi” actually comes from “Python”
- Thus, some bias towards Python in the ecosystem





# DEV ENVIRONMENT

(BUILT-IN)

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



# DEV ENVIRONMENT

(OTHERS)

- Node
- Go
- Kotlin
- Haskell
- Smalltalk  
(Squeak built-in, but needs some add-ons)
- others...

# 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

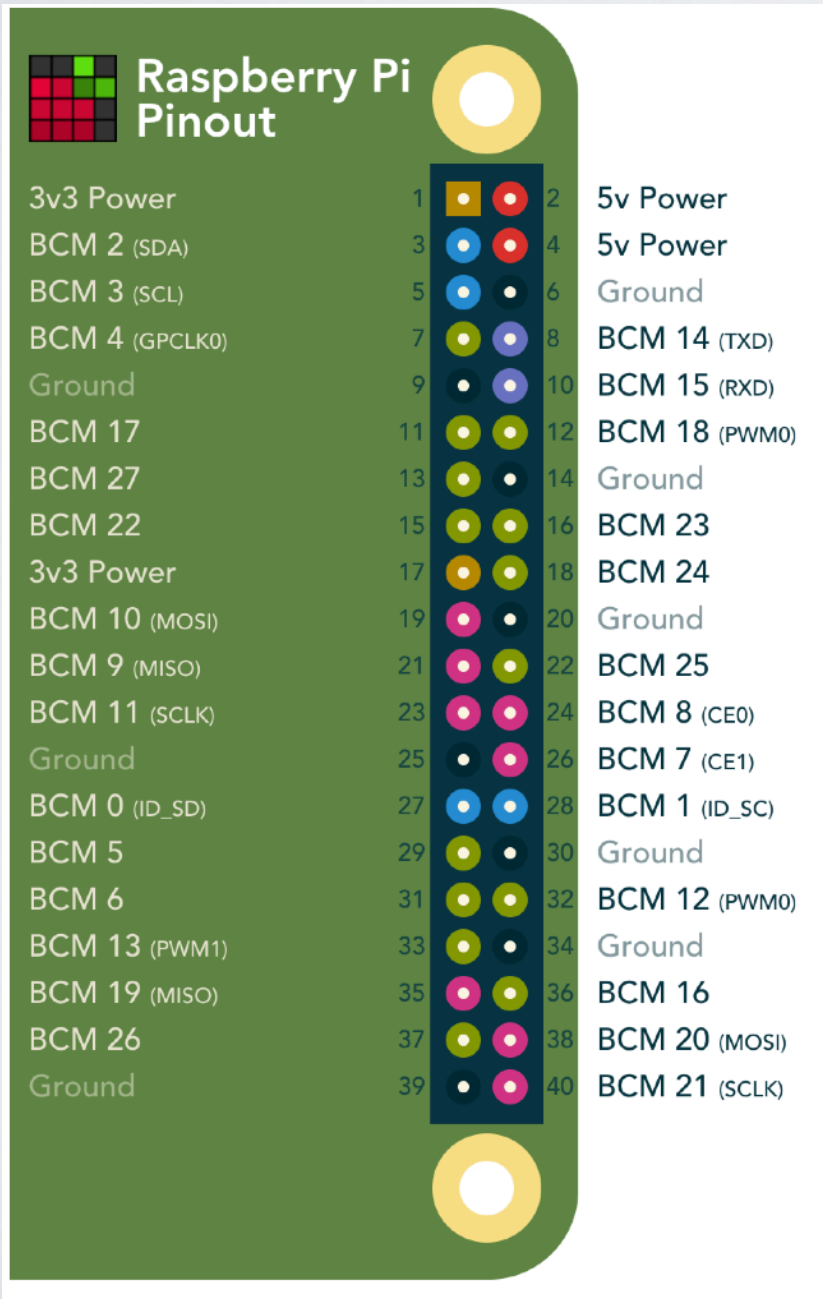
On Raspbian Stretch  
pip looks here before PyPI

## Installable:

- Any Python-only library
- Or, any that will compile for ARM
- [www.piwheels.org](http://www.piwheels.org) - precompiled packages
  - e.g. OpenCV, Tensorflow, WiringPi, etc.

# GPIO / I2C / SERIAL

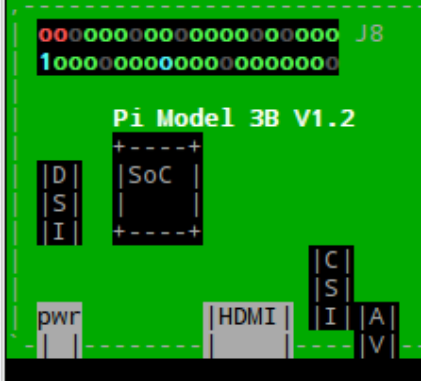
<https://pinout.xyz/>



```

pi@raspberrypi:~ $ pinout

```



```

Revision      : a02082
SoC           : BCM2837
RAM           : 1024Mb
Storage       : MicroSD
USB ports     : 4 (excluding power)
Ethernet ports : 1
Wi-fi        : True
Bluetooth     : True
Camera ports (CSI) : 1
Display ports (DSI): 1

J8:
  3V3 (1) (2) 5V
  GPIO2 (3) (4) 5V
  GPIO3 (5) (6) GND
  GPIO4 (7) (8) GPIO14
  GND (9) (10) GPIO15
  GPIO17 (11) (12) GPIO18
  GPIO27 (13) (14) GND
  GPIO22 (15) (16) GPIO23
  3V3 (17) (18) GPIO24
  GPIO10 (19) (20) GND
  GPIO9 (21) (22) GPIO25
  GPIO11 (23) (24) GPIO8
  GND (25) (26) GPIO7
  GPIO0 (27) (28) GPIO1
  GPIO5 (29) (30) GND
  GPIO6 (31) (32) GPIO12
  GPIO13 (33) (34) GND
  GPIO19 (35) (36) GPIO16
  GPIO26 (37) (38) GPIO20
  GND (39) (40) GPIO21

For further information, please refer to https://pinout.xyz/
pi@raspberrypi:~ $

```

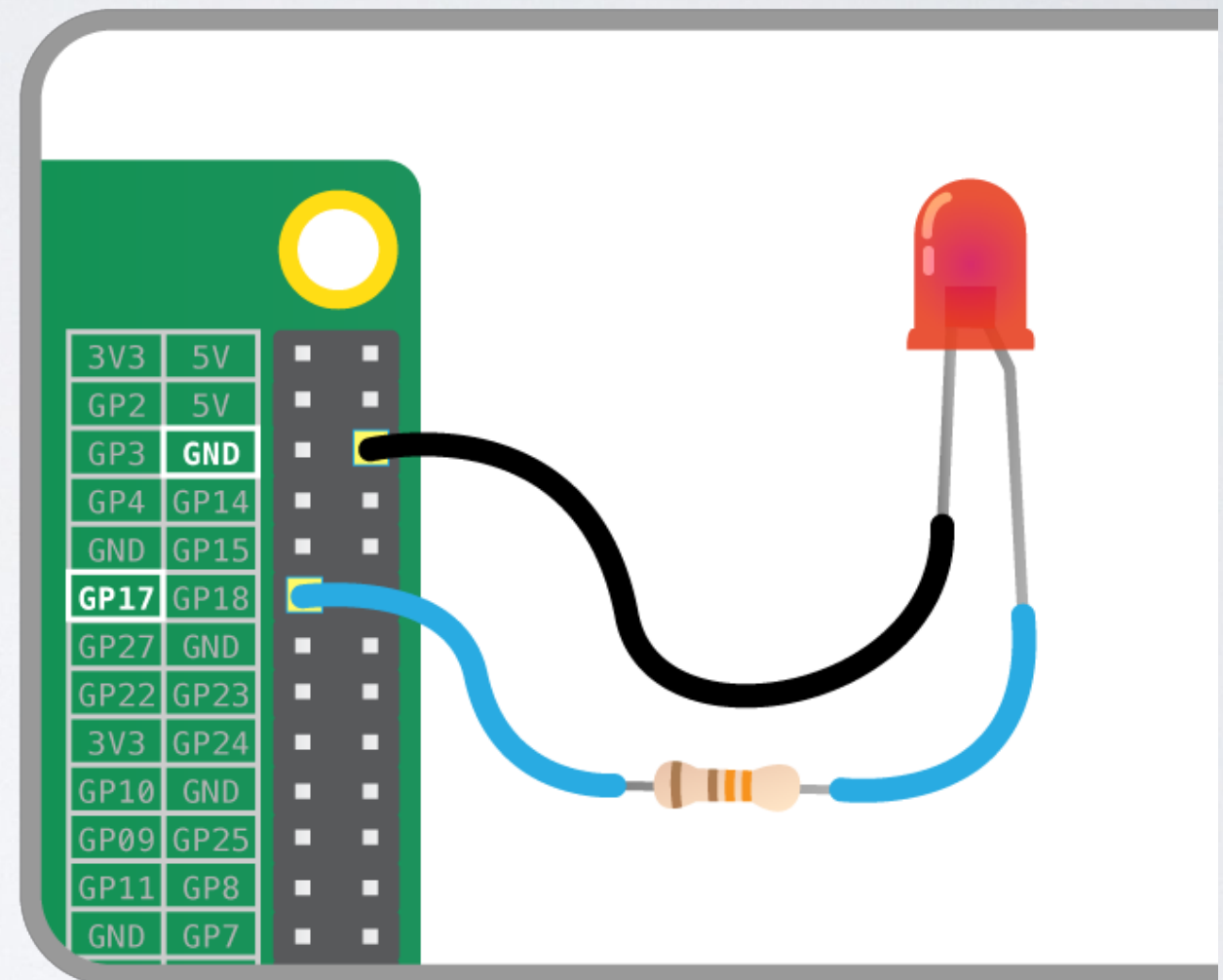


# GPIO BASICS

```
from gpiozero import LED  
from time import sleep
```

```
red = LED(17)
```

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





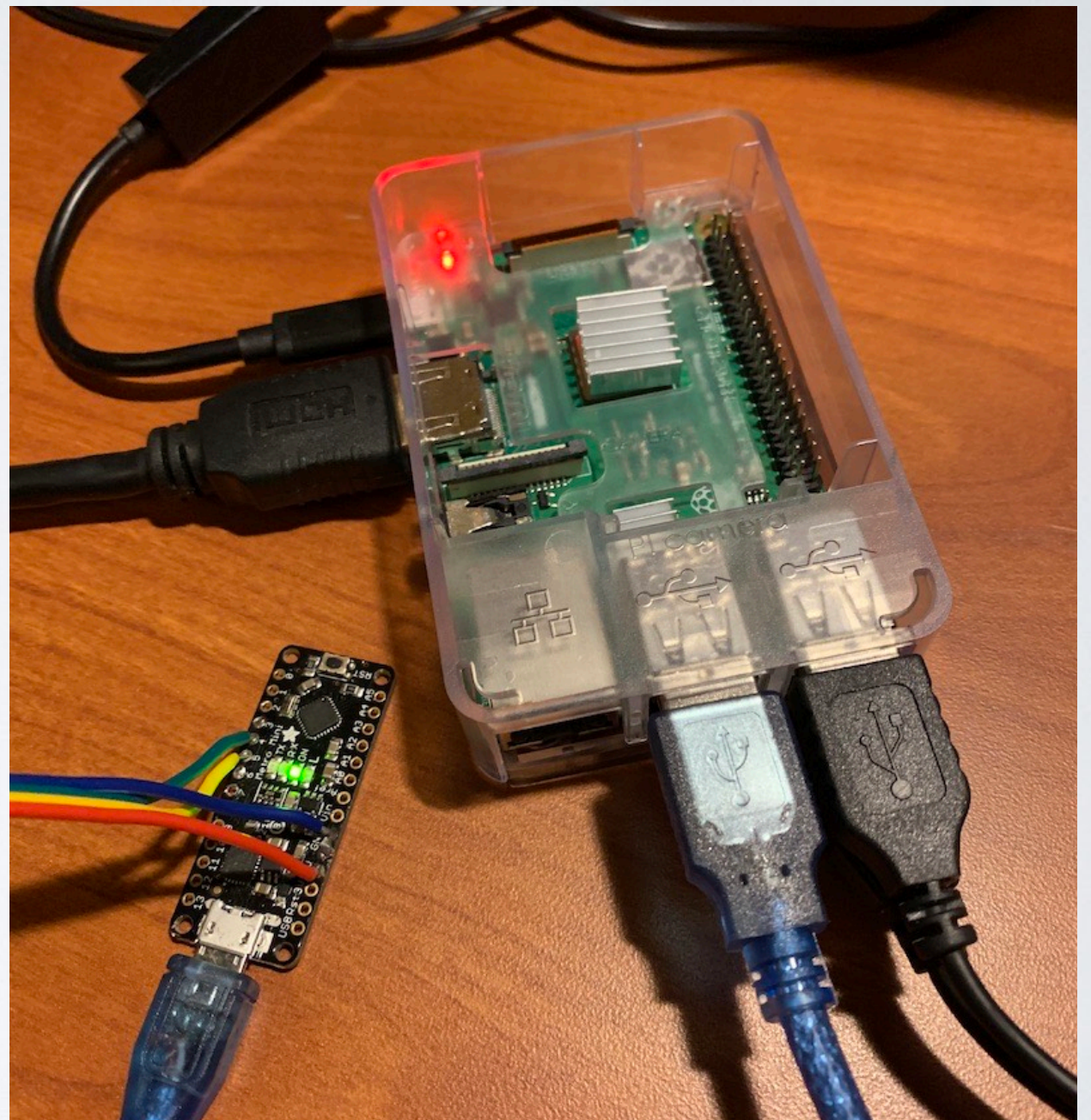
# SERIAL OVER USB

```
import serial

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

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

# close port
ser.close()
```



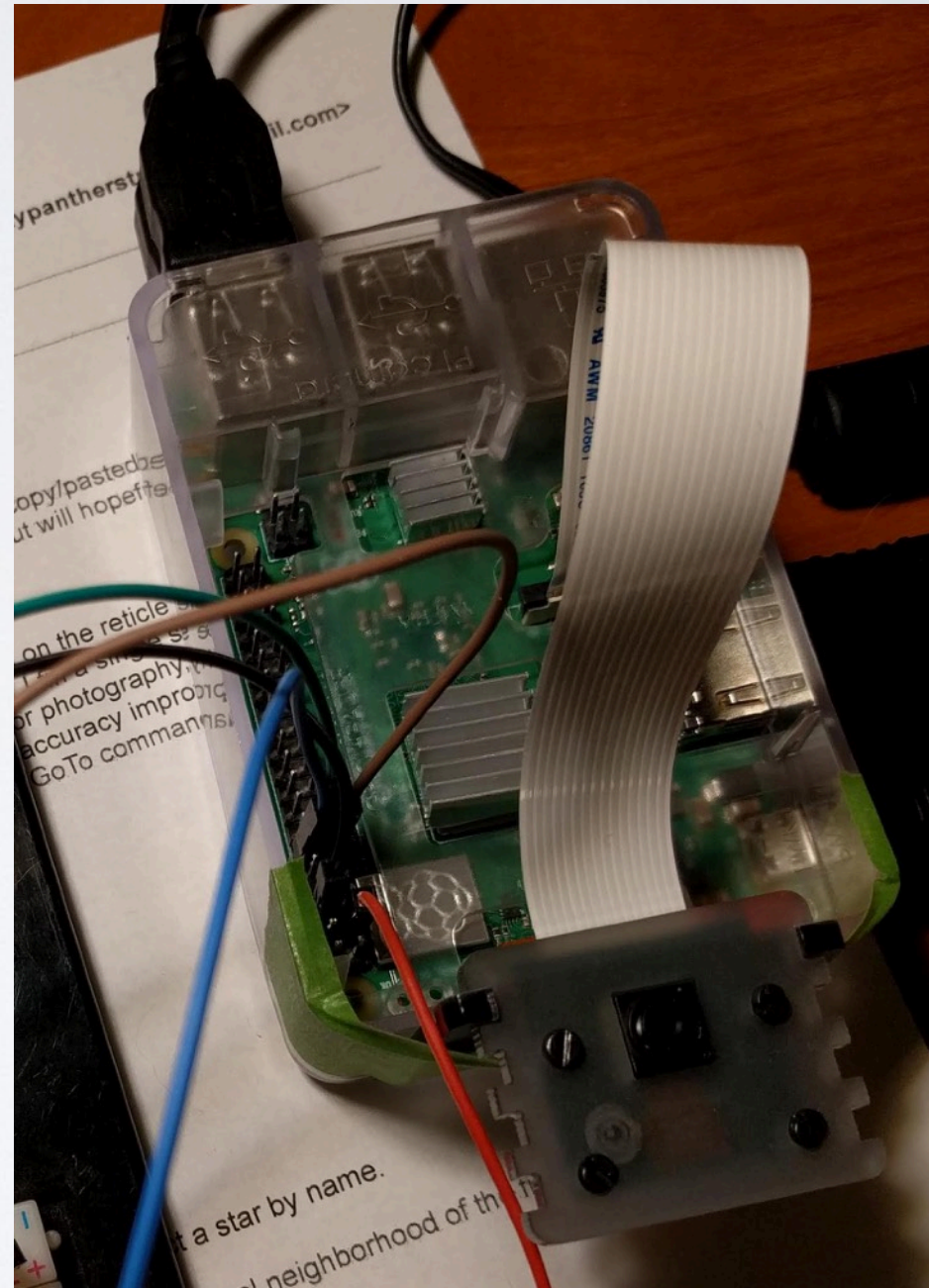


# PiCAMERA

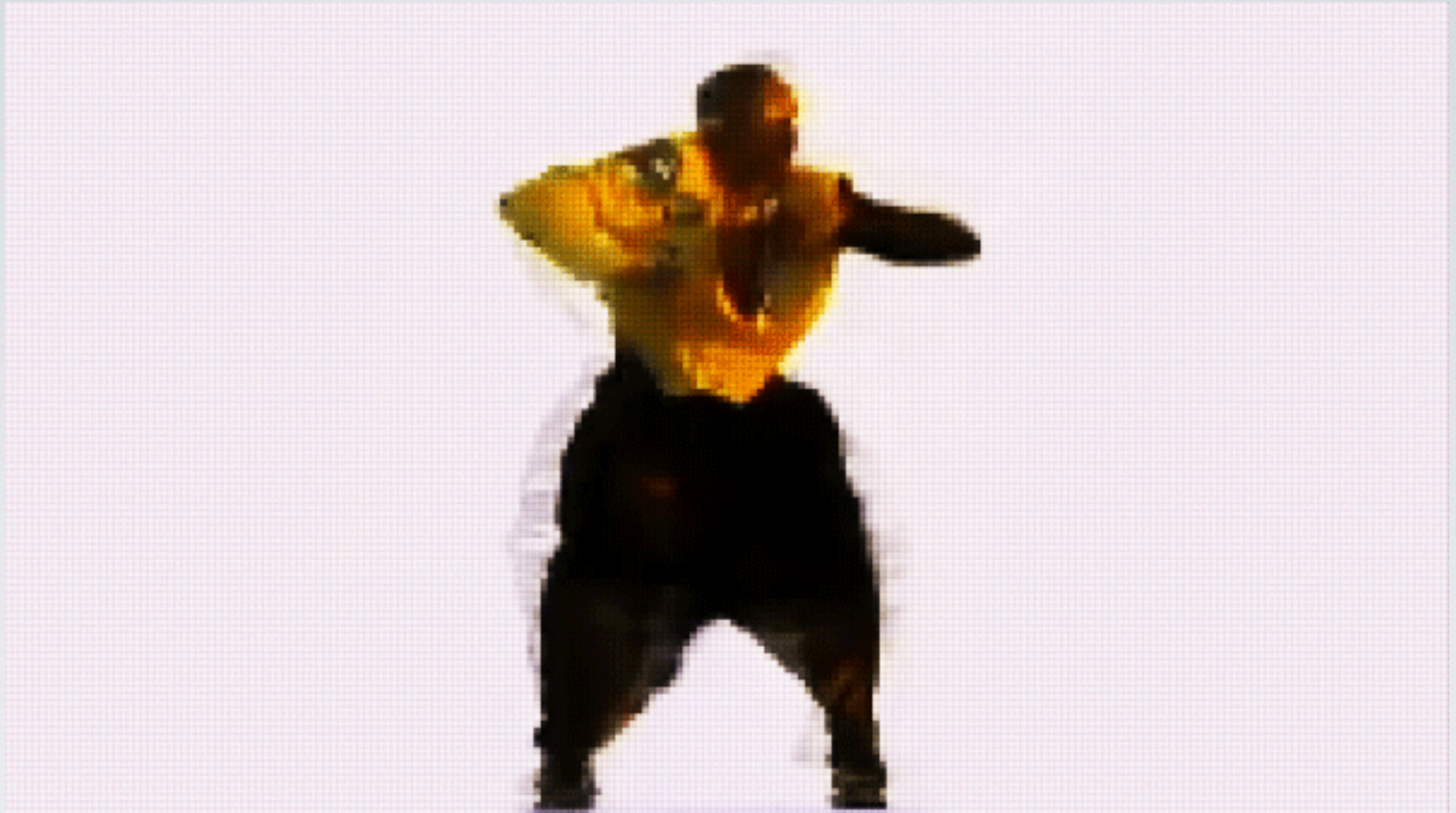
```
from picamera import PiCamera
from time import sleep

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

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







DEMO TIME



# THANKS!

- [timpoulsen.com](http://timpoulsen.com)
- [github.com/skypanther](https://github.com/skypanther)
- [@skypanther](#)
- [www.linkedin.com/in/timpoulsen](https://www.linkedin.com/in/timpoulsen)

