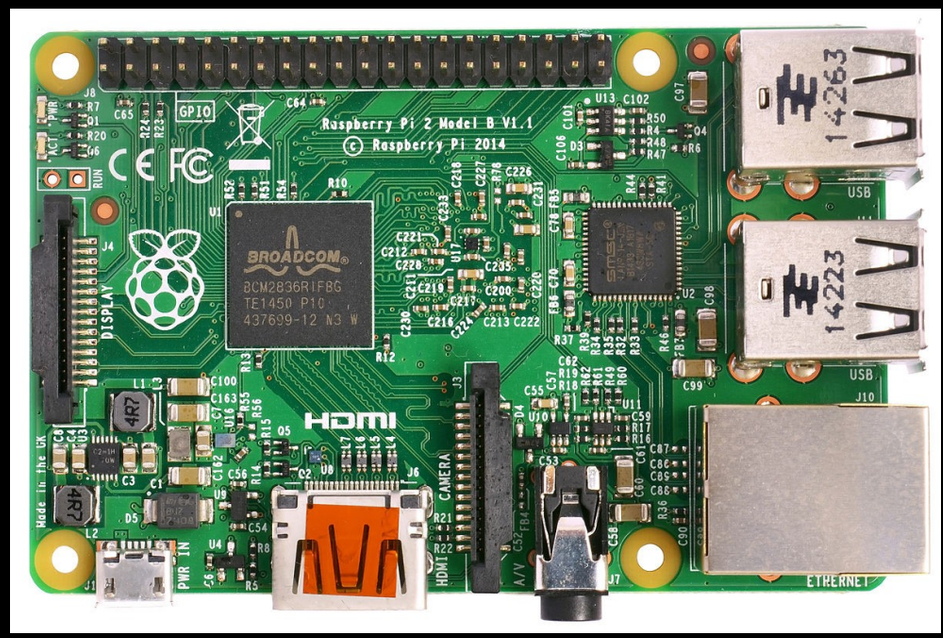
Raspberry Pi
Python Users Group Nepal



Keya Lea Horiuchi

Project website: Soldering and Electronics - Fosslounge.org

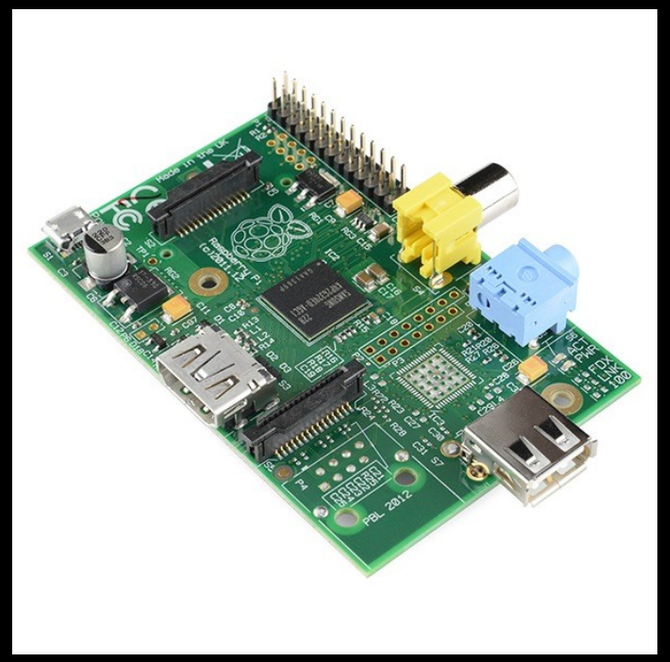
keyalea@gmail.com

What we'll cover

- What possible, different projects
- What it is
- The basics: RPi, Electricity, GPIO pins, Python, LEDs, jumper wires & the breadboard for prototyping

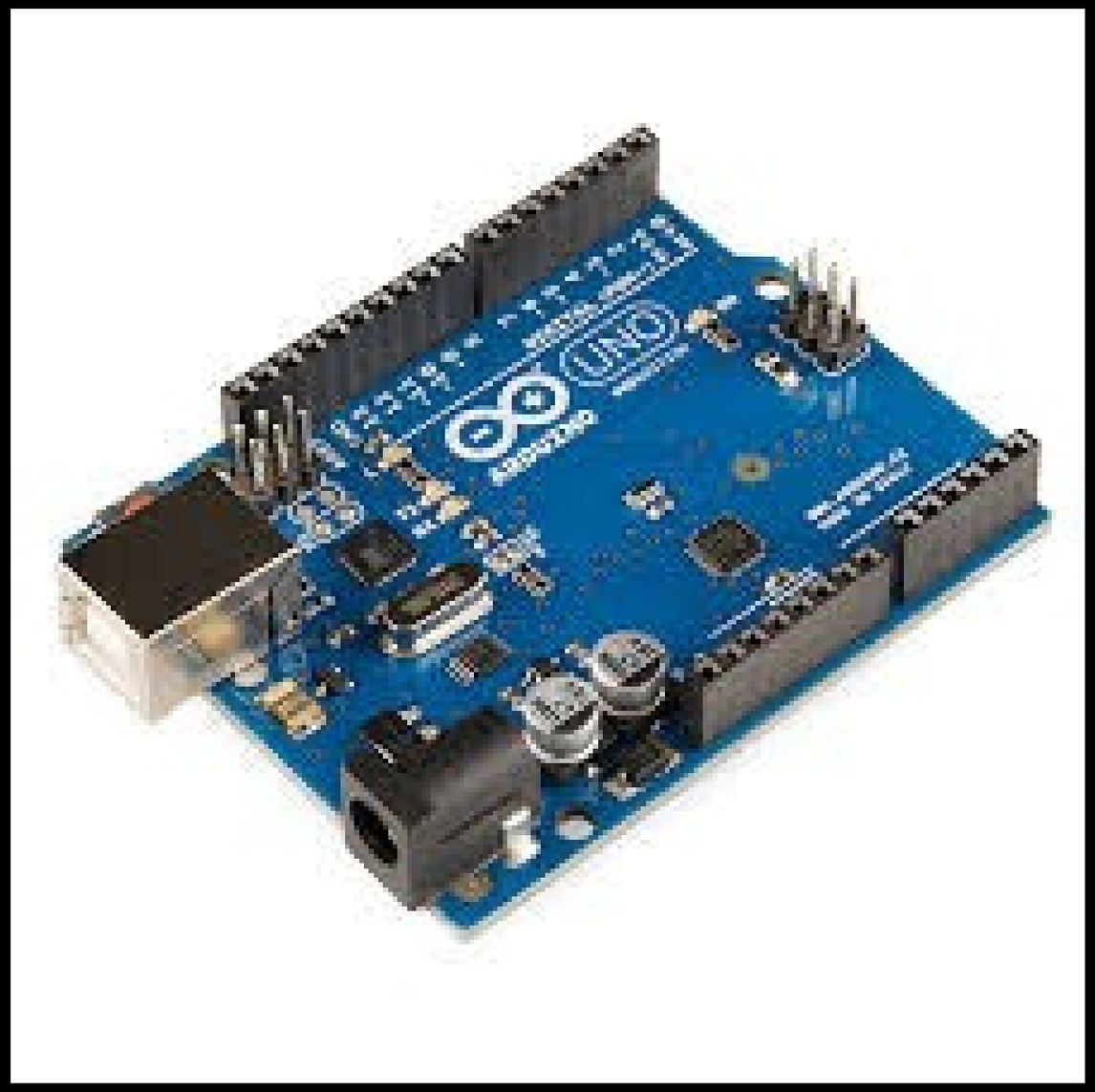
What is the RPi?

- It’s minimalist.
- Name came from the tradition of fruit computers like Apricot, Tangerine & Acorn
- Pi stands for Python
- Base for a complete computer, capable of running a LAMP stack.



This is Eben Upton.

- Create an inexpensive, build-able, intellectually-free device that people could use and learn from.
- Started selling Model A in Feb 2012.
- As of February 2016, 8 million RPis have been sold.
- Made in the UK



An Arduino.
Also cool, but used for different purposes.

Possibilities

Yes, it can be a computer, but it can be a lot more.

DIY!

- Use it to prototype and solve things
- Python modules
- Attach different sensors
- Build solutions

DIY!

- Use it to prototype and solve things
- Python modules
- Attach different sensors
- Build solutions

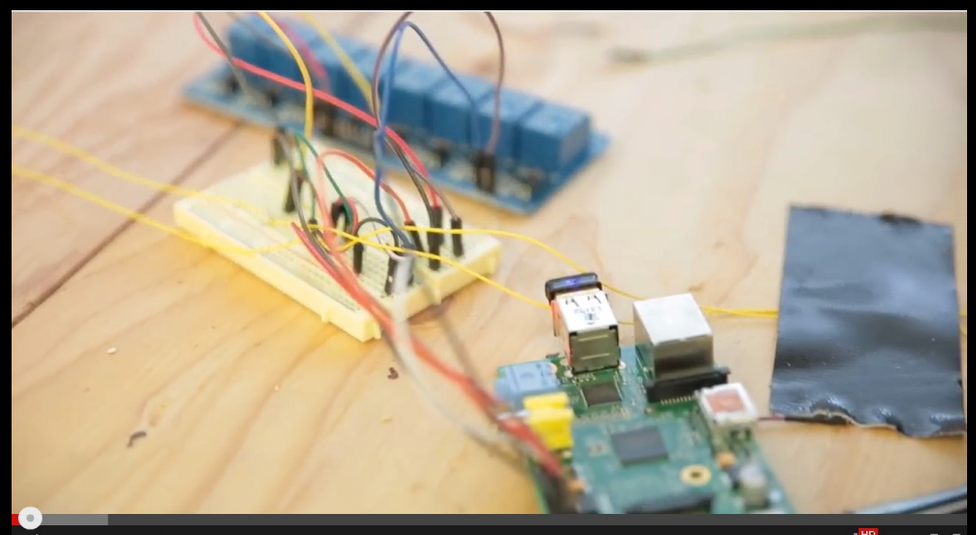
Use a RaspPi to Fix Everyday Problems

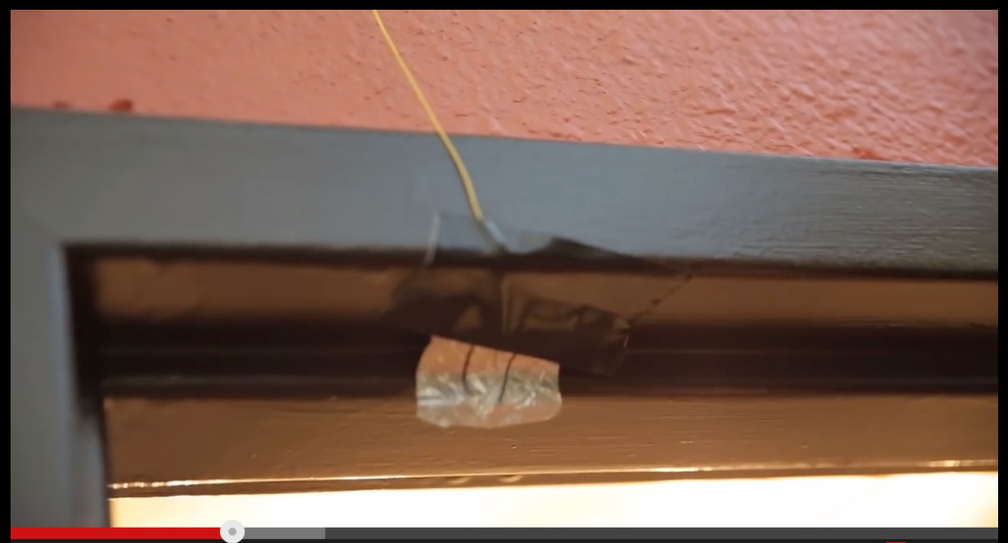
Blender Defender: motion detection, lights, blender

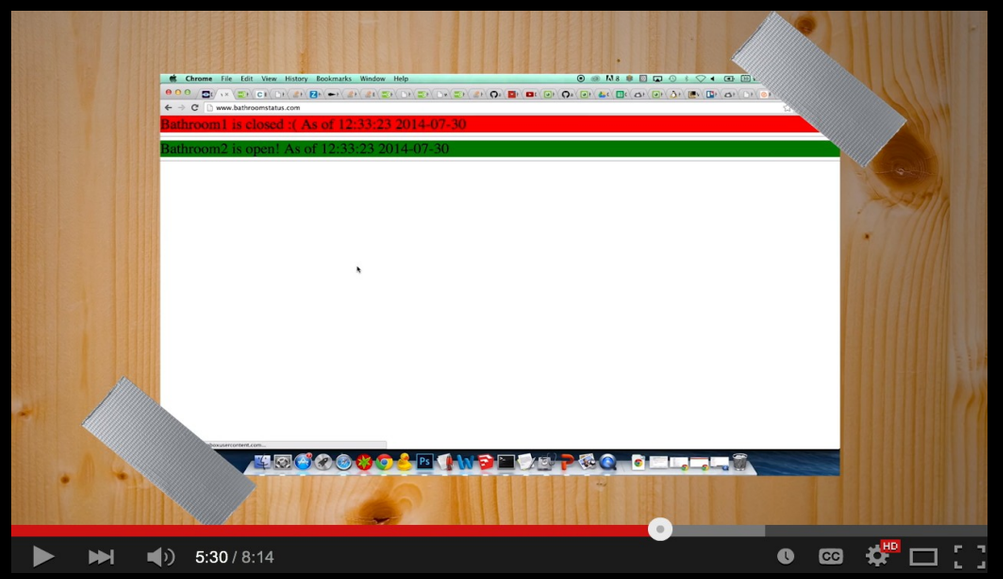
https://www.youtube.com/watch?v=-b9m8BpmD0o

http://www.plasma2002.com/blenderdefender/

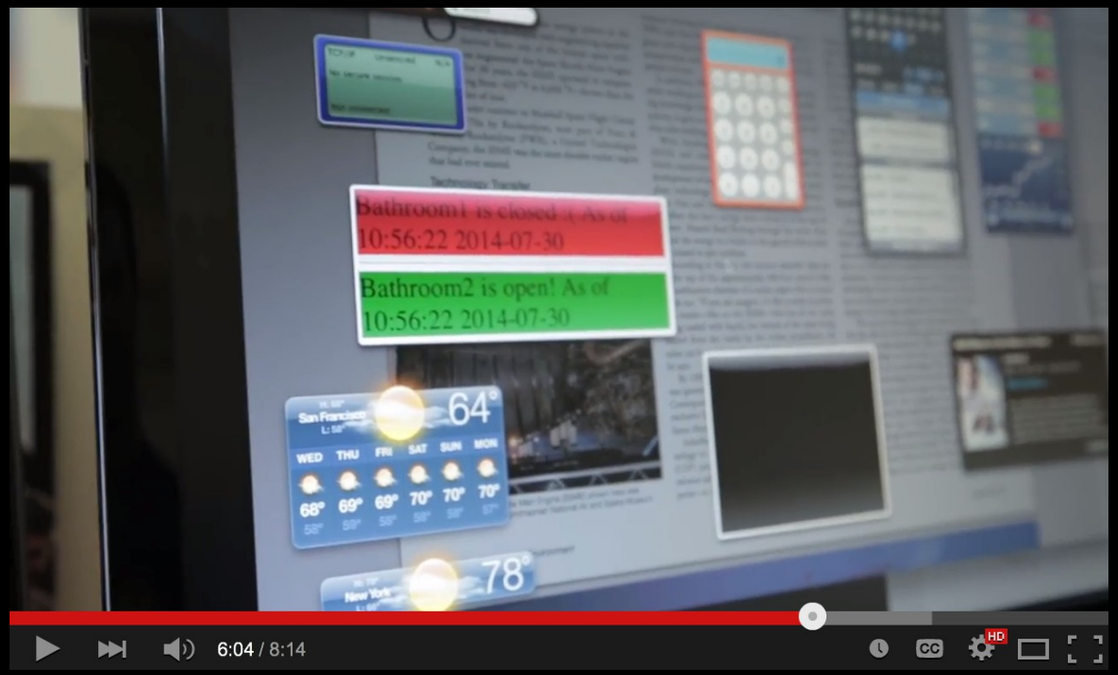
Web-based Open Bathroom



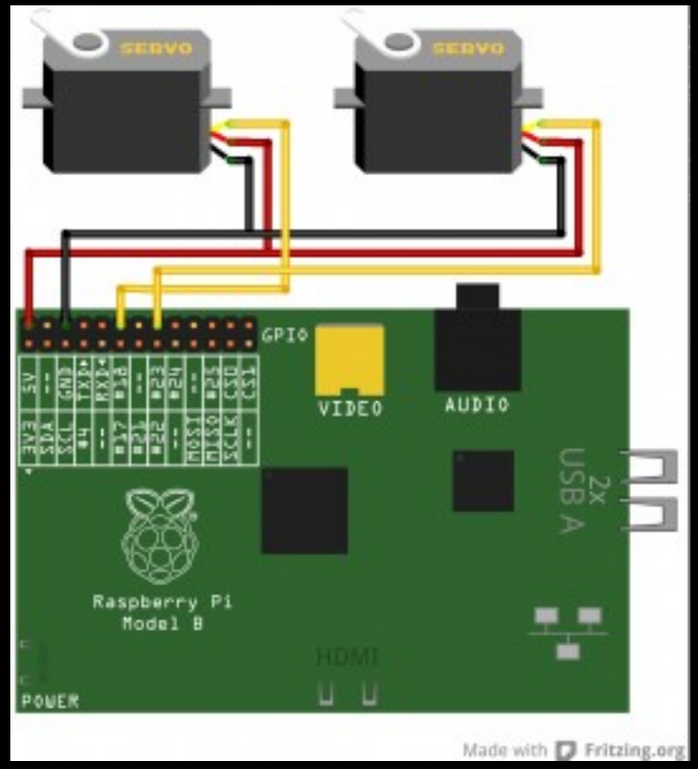




Sensors + Website + Rasp Pi = Office Productivity

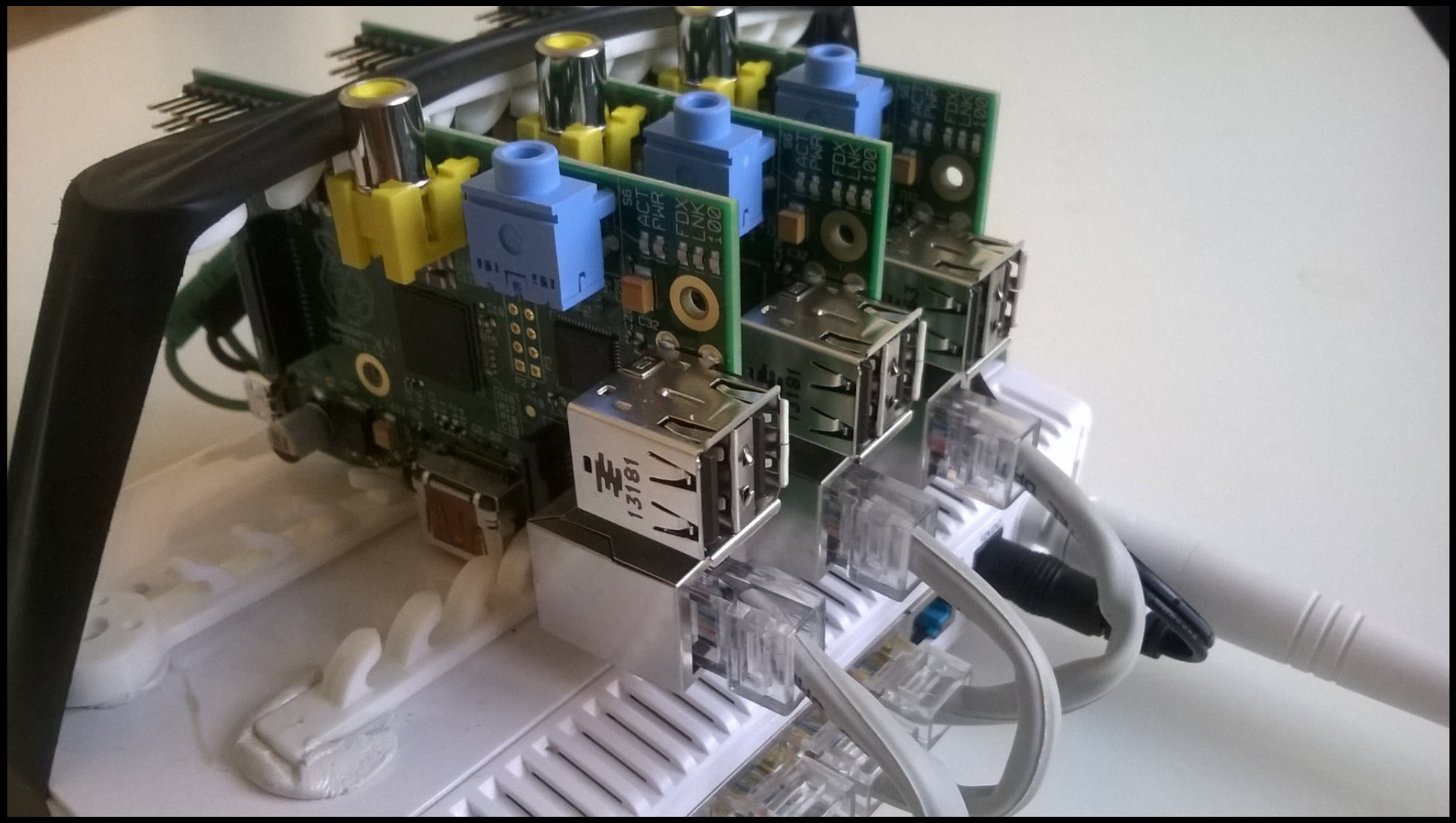


https://www.youtube.com/watch?v=ZszlVVY1LXo 8:14



http://mitchtech.net/raspberry-pi-servo-face-tracker/

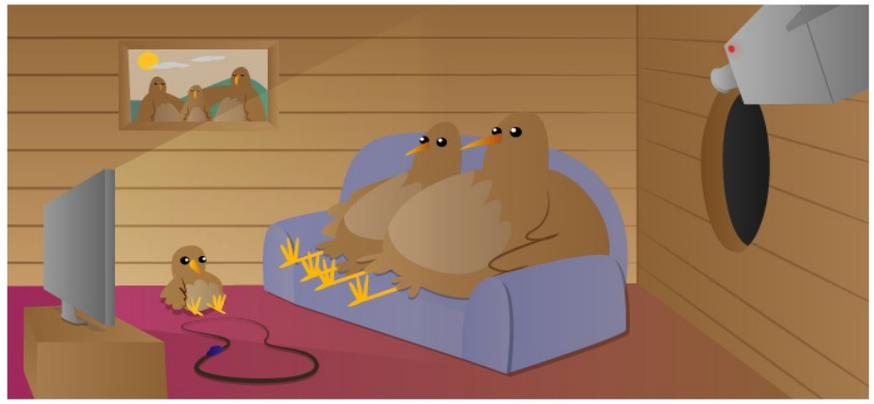
Build Hadoop cluster



http://www.widriksson.com/raspberry-pi-hadoop-cluster/

BirdCamera Box

INFRARED BIRD BOX



WHAT YOU WILL MAKE

In this resource you will make an infra-red bird box. This project will not only teach you about electronics and programming, but can help support the bird population in your area. Having a camera inside a nesting box can be tremendously rewarding; however, with the Raspberry Pi you can also share the nesting box with the world by streaming the video content to the Internet. Watch as your birds gain their own Internet following!

WHAT YOU WILL LEARN

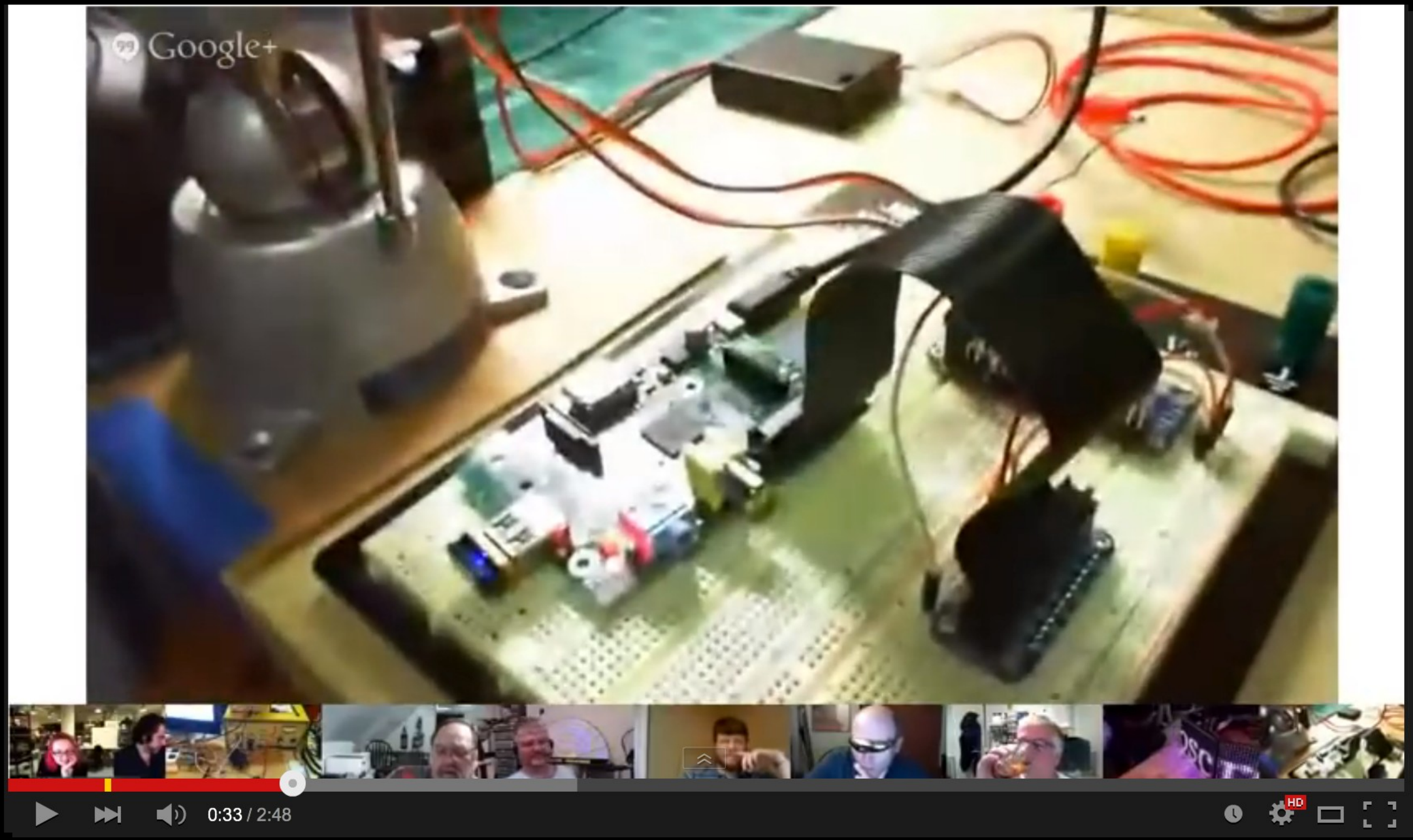
By creating an infra-red bird box you will learn:

- How to setup up a Raspberry Pi with an infra-red camera module
- How to connect an infra-red LED to a Raspberry Pi
- How to adjust the camera focus on an Pi camera module
- How to control the camera and LED to see what is happening inside the bird box

WHAT YOU WILL NEED

GET STARTED

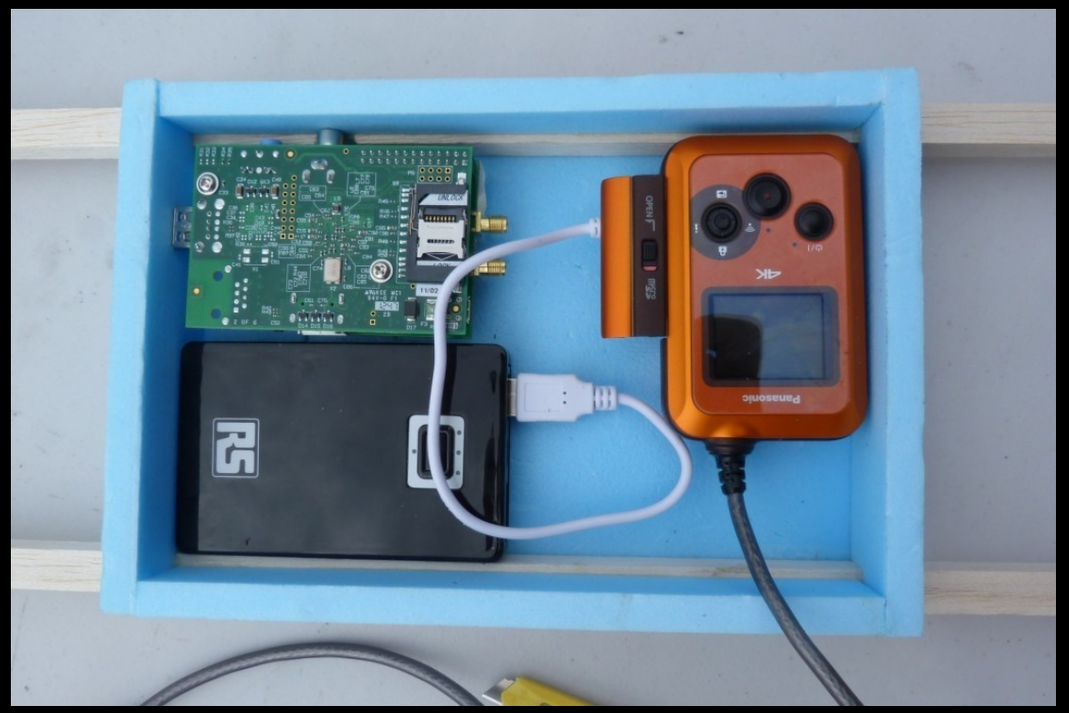
https://www.raspberrypi.org/learning/infrared-bird-box/



https://www.youtube.com/watch?v=RVYByGU6uYI

Screenshot of AdaFruit Forum

Sent the Rpi into Space



Map | SUPER | Settings

Spacenear.us Sat 12th July

GILWELL24 0900, Epping, *434.613 MHz* 300 baud TurboHAB & 50 baud 7n2 RTTY; *434.200 MHz* 600 baud SSDV
HADFIELD 1000, Walsall, *434.350* MHz, 50 baud 7n2
SUPER 1030, Ross-on-Wye, *434.450* SSDV 300 baud; *434.480* MHz DomEX22; *434.475* 50 baud 7n1
XABEN75 1100, Cambridge, *434.250* & *434.300* MHz, 50 baud 7n1

Come chat to us: **#highaltitude** on irc.freenode.net
Current time is : 15:31 UTC

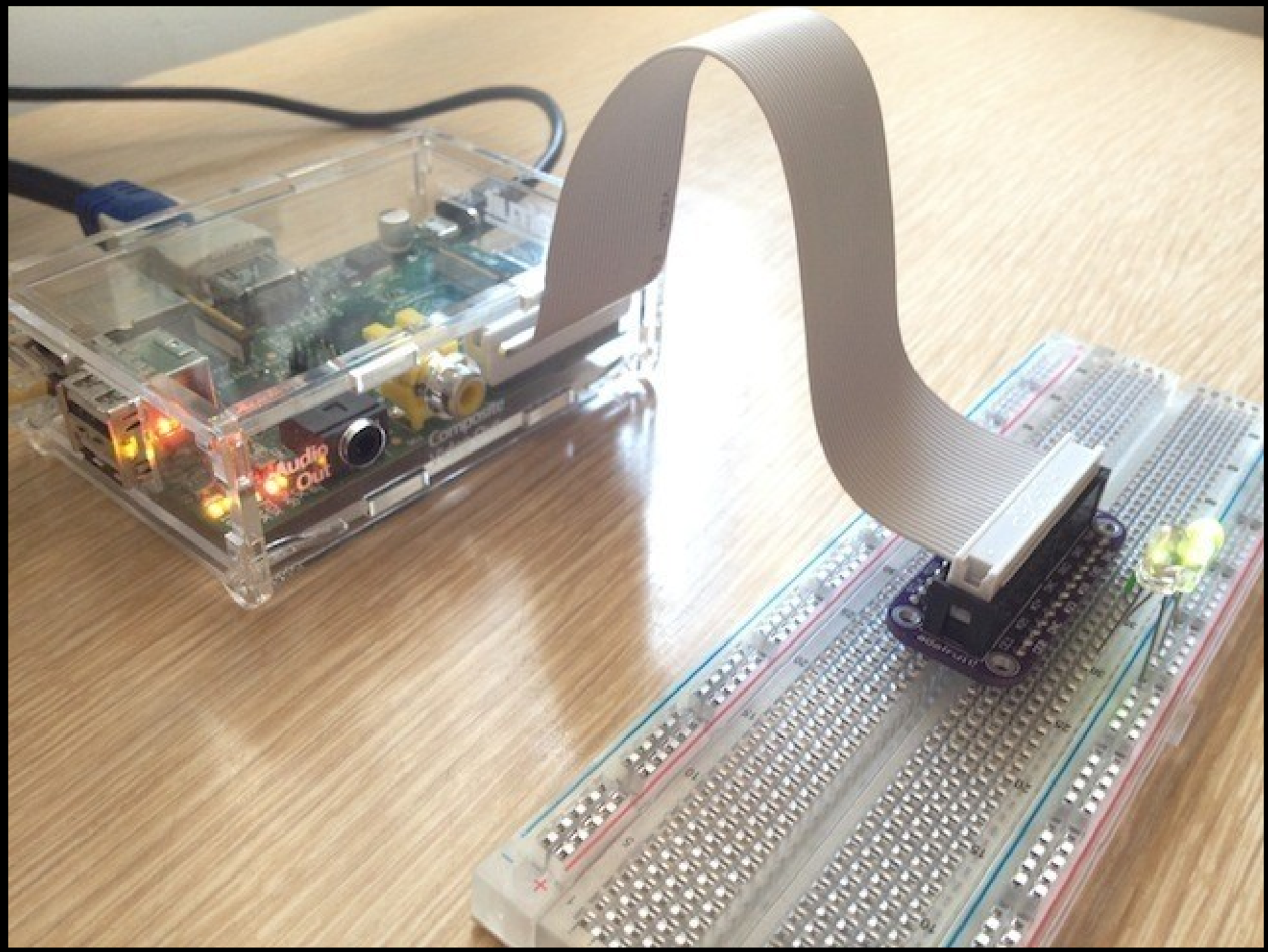
If this page does not load correctly please try the **Mobile Tracker**.

Map | Satellite

SUPER
Time: 2014-07-12 12:31:44
Position: 51.87255,-2.49006
Altitude: 160 m **Rate:** 0.0 m/s
Max. Altitude: 35569 m
Heading: 0°
Speed: 0 km/h
Temperature: 22.5C
Battery: 5.9 V
Satellites: 7
Receivers: MORPI_chase
Pan To | Path | Follow

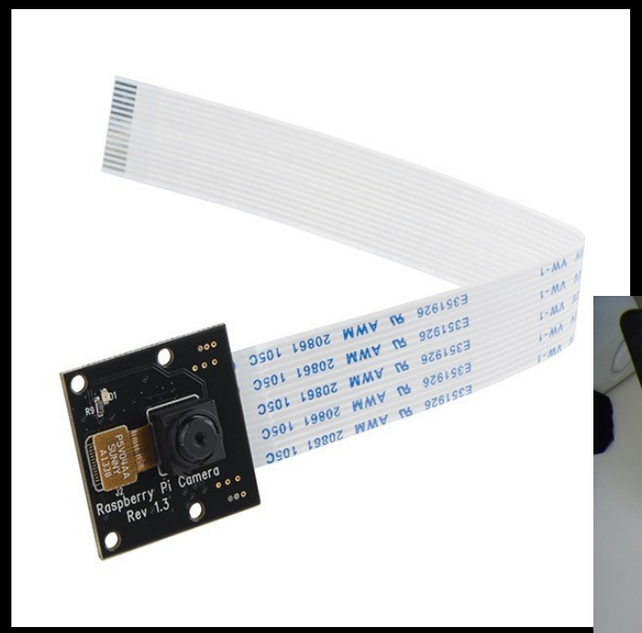
MORPI_chase
Time: 2014-07-12 12:29:30
Position: 51.87328,-2.486979
Altitude: 157 m **Rate:** 0.1 m/s
Heading: 0°
Speed: 0 km/h
Pan To | Path | Follow

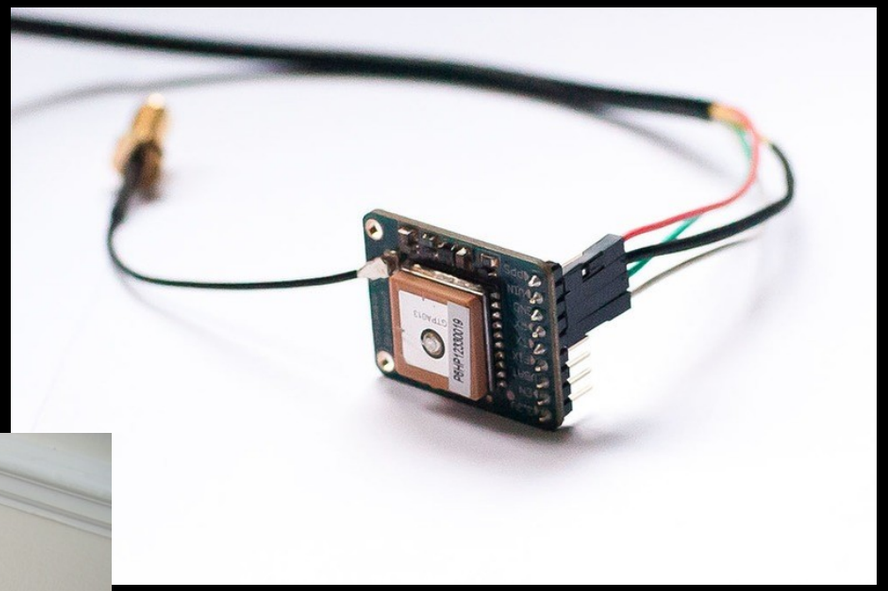
powered by spacenear.us

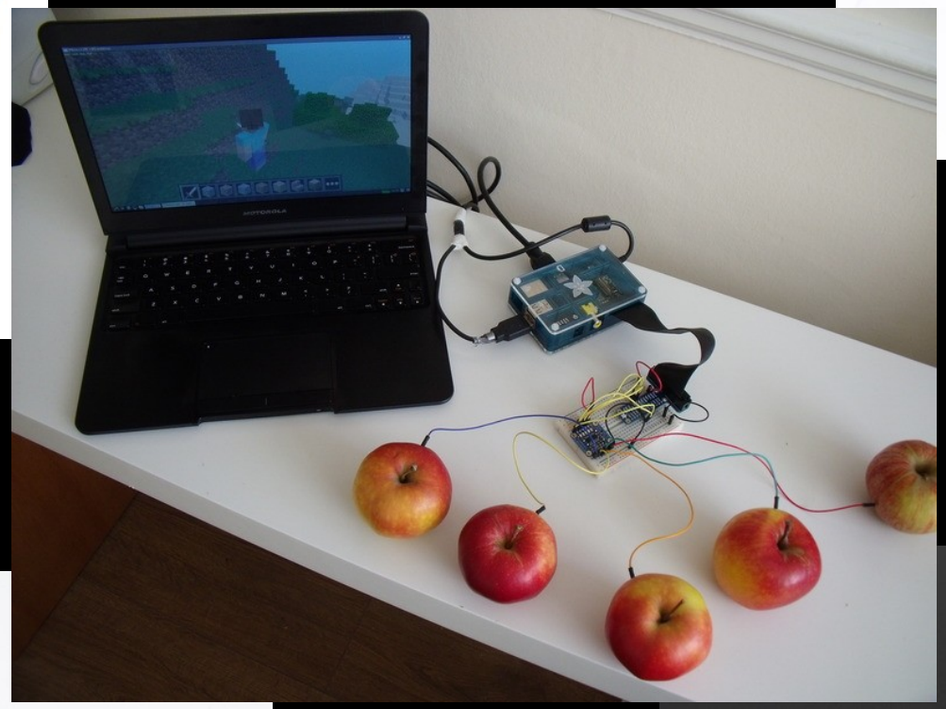


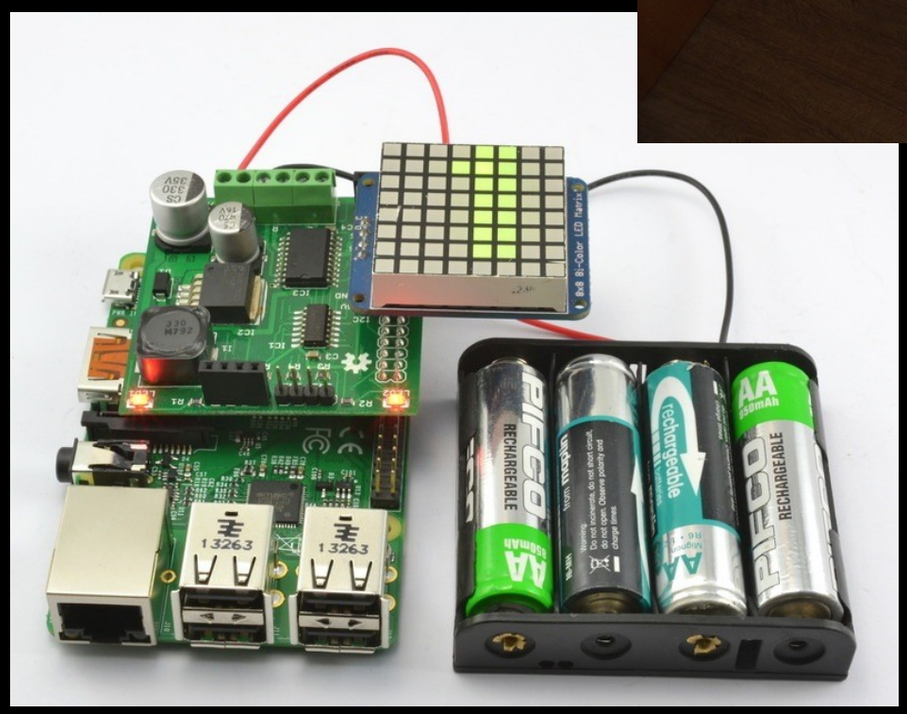
https://learn.adafruit.com/raspberry-pi-e-mail-notifier-using-leds/pr

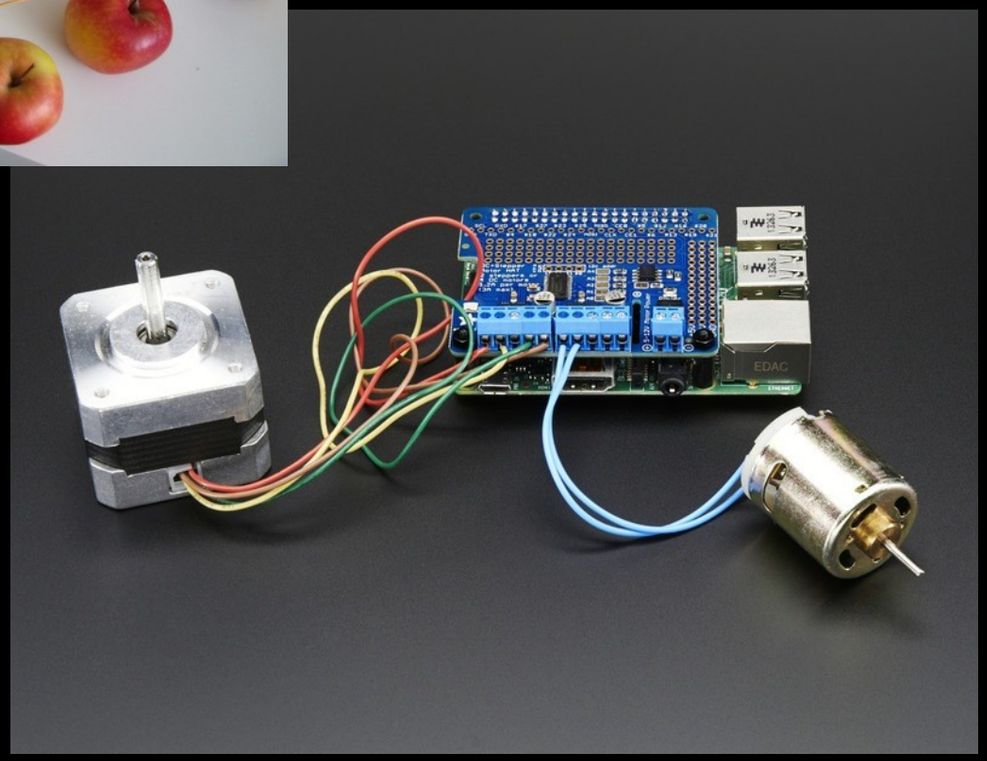
IoT


Raspberry Pi Camera
Rev 1.3






8x8 Bi-Color LED Matrix
AA
2450mAh
RECHARGEABLE
rechargeable

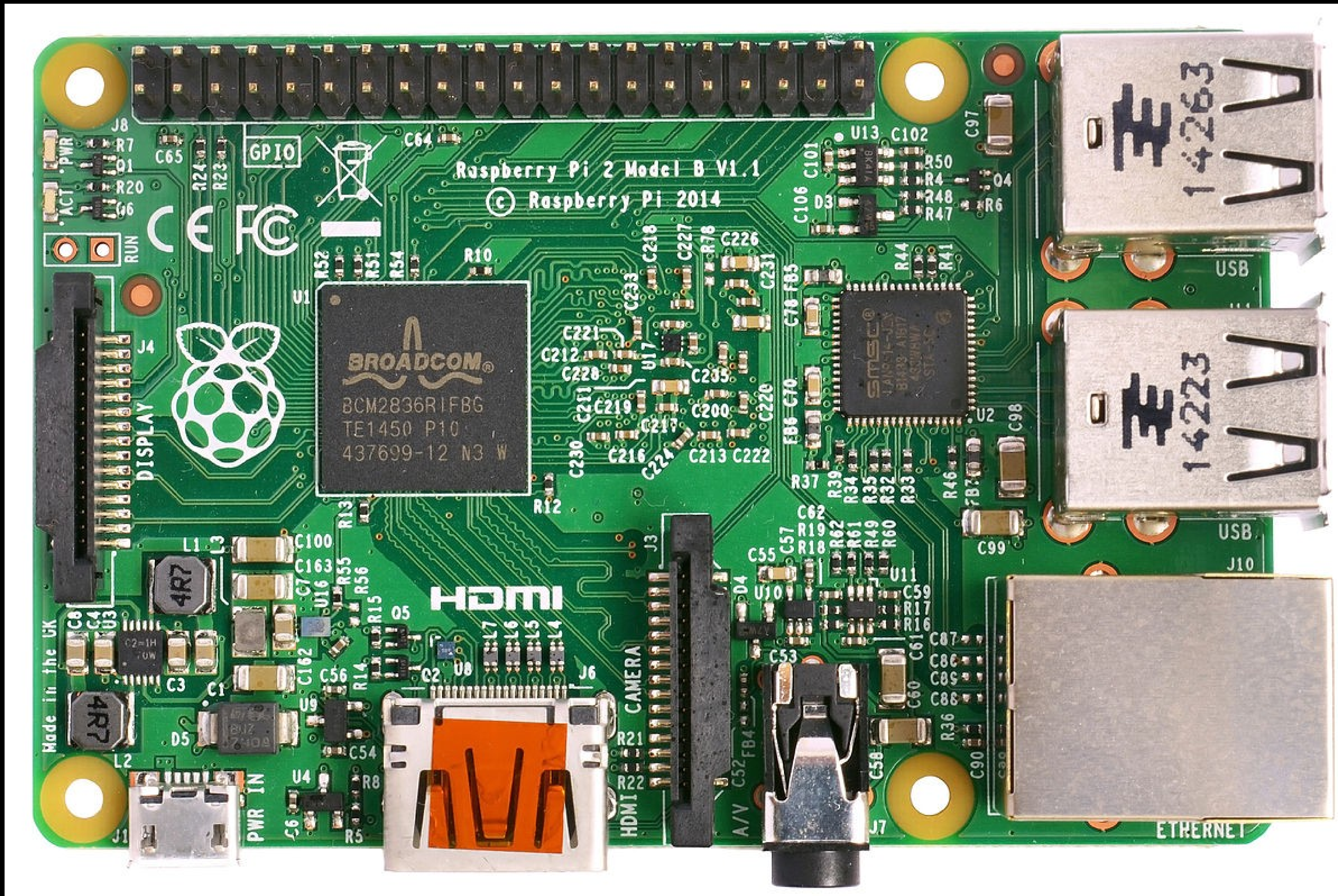

EDAG

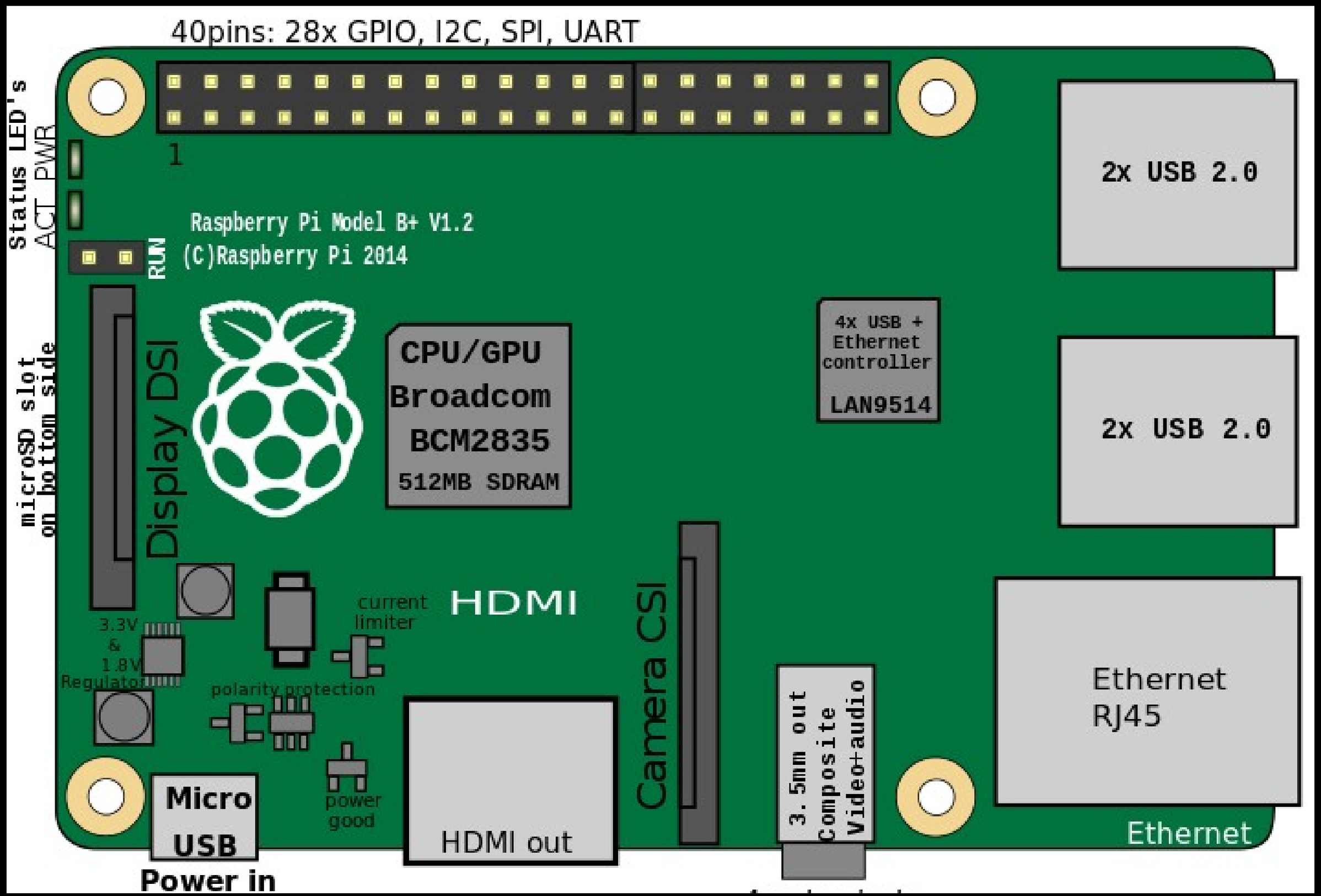
32+ different OS (Wikipedia)

- First need to install an operating system on the MiniSD card
  - ArchLinux (minimalist)
  - Raspbmc (digital media center)
  - Raspian (default)
  - Pidora (Fedora remix)
  - FreeBSD
  - Ark OS (website and email hosting)
  - Minepion (mine cryptocurrency)
  - IPFire (firewall/router distro, only runs on RPi1)
  - Windows 10 IoT Core

Specs

- $35
- Model, B Generation 3
- SoC - Broadcom BCM2837
- **CPU 1.2GHz quad-core ARM, Cortex-A53**
  (A-B+ 700 MHz single-core ARM, B 900MHz)
- GPU Broadcom VideoCore IV @ 250MHz, MPEG-2 and VC-1 1080p30 H.264/MPEG-4 AVC
- Memory: 1GB RAM (shared with GPU)
- Power ratings: 800 mA / 4.0 W (B+ 600mA / 3.0 W)
- **Power source 5 V via MicroUSB or GPIO**

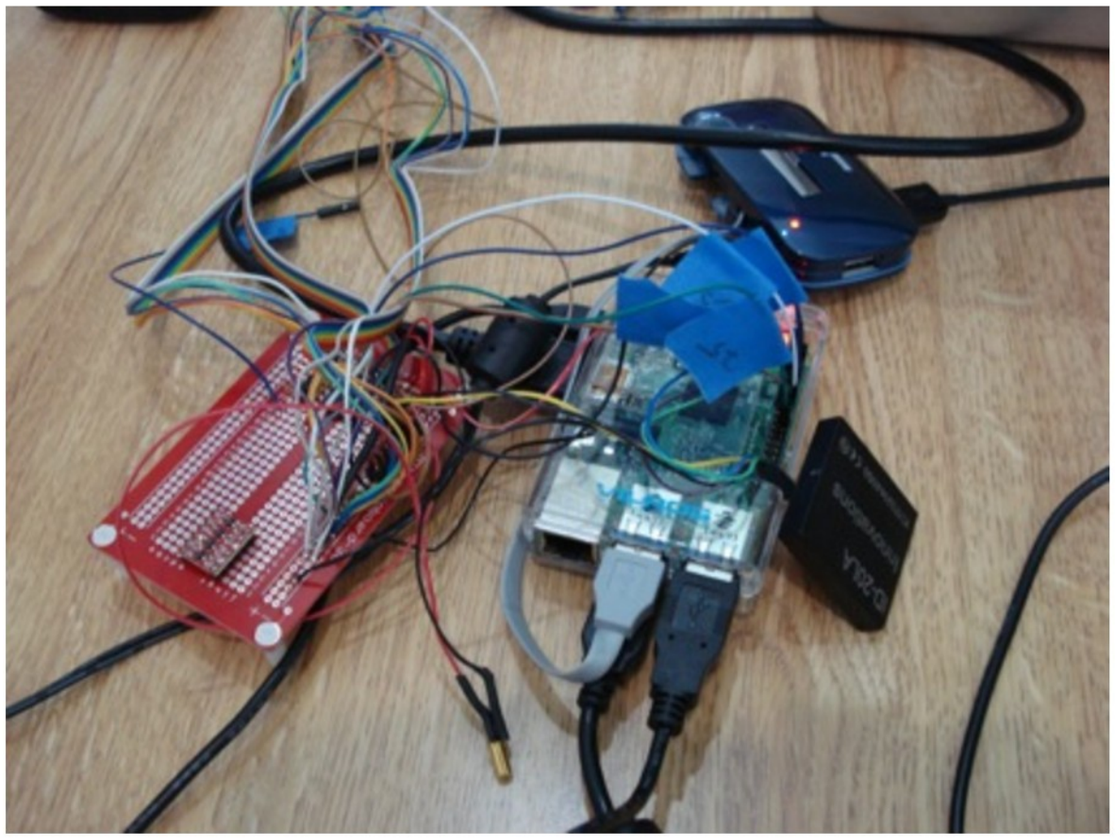

Raspberry Pi 2 Model B V1.1
© Raspberry Pi 2014
GPIO
BROADCOM
BCM2836RIFBG
TE1450 P10
437699-12 N3 W
DISPLAY
HDMI
CAMERA
A/V
PWR IN
USB
USB
ETHERNET
Made in the UK


40pins: 28x GPIO, I2C, SPI, UART
1
Status LED's
ACT PWR
RUN
Raspberry Pi Model B+ V1.2
(C)Raspberry Pi 2014
microSD slot
on bottom side
Display DSI
CPU/GPU
Broadcom
BCM2835
512MB SDRAM
4x USB +
Ethernet
controller
LAN9514
2x USB 2.0
2x USB 2.0
HDMI
current
limiter
3.3V
&
1.8V
Regulator
polarity protection
power
good
Camera CSI
3.5mm out
Composite
Video+audio
Ethernet
RJ45
Micro
USB
Power in
HDMI out
Ethernet

Specs (cont.)

- USB 2.0 ports: 4
- Video input: 15 pin MIPI camera interface connector
- Video output: HDMI
- Audio input - I2S (2 is squared)
- On-board storage: MicroSD 4 – 64+ G
- On-board network: 10/100 Mbit/s Ethernet (8P8C)
- Low-level peripherals: 17 GPIO pins (LOOK :-))

Mobile Medical Disaster Relief Dispensation Unit



PCM Portable Computerized Module

The Problem

Topic: Infrastructure Security

- There are no systemic methods of tracking medicine in disaster relief scenarios.
- The current method used is usually paper-based, allowing for error and data leaks.
- The paper-based system forces medical personnel to spend unnecessary energy tracking and logging medicine when they should be concentrating on saving lives.

Solution

A portable medical dispensation system called the PCM. Each module allows the medical personnel to quickly access the needed medicine without having to worry about tracking it.

- Authenticates medical personnel
- Allows rapid access to medicine
- Tracks actions in a database
  - Personnel who accessed medicine
  - Medicine accessed

- GPS awareness
- Securely stores and transmits data

Prototype 1

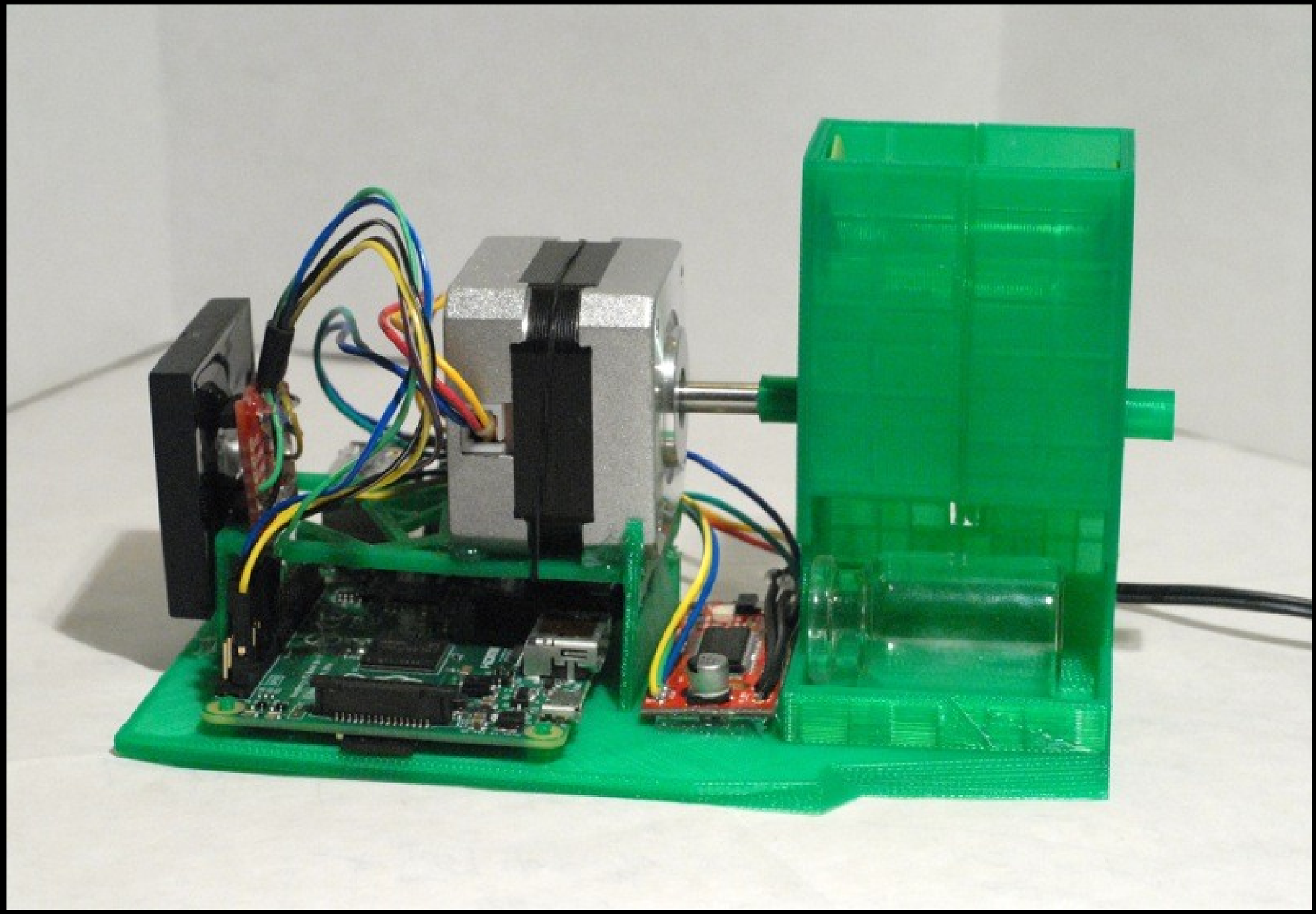
- The first prototype authenticated on **RFID** (Radio Frequency Identification).
  - A small RFID button was placed in a bracelet.

- If the medical personnel is recognized, it engages a **stepper motor** to dispense medicine.

- A **Raspberry Pi** (small computer) was programmed with **Python** to run these basic functions.
  - Used GPIO (General Purpose In/Out) pins

Components

- Raspberry Pi B+, Raspbian, 8GB SD card
- Stepper Motor 12V, Mercury Motor, bipolar, external power
- EasyDriver, SchmalzHous.com controller
- RFID Reader ID-20LA (125 kHz)
- SparkFun RFID Reader Breakout
- RFID Tags 125kHz



```python
import serial, time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)
GPIO.setup(16, GPIO.OUT) #16 goes to stepmotor
GPIO.setup(18, GPIO.OUT) #18 for direction

#set up pulse rate modulation object for pulses for 16
p = GPIO.PWM(16,500) #500Hz
def SpinMotor(direction, number_steps):
        GPIO.output(18, direction)
        while number_steps > 0:
                p.start(1)
                time.sleep(0.01) # change impacts rotation .02 175 full rotation
# or if value .01, then 350 steps for full rotation
                number_steps -= 1
        p.stop()
        return True

# I realize that adding cards in elif statements not the best way to code
# cards = ['6A004A16C0', '6A0049F913']
# rfid reader
ser = serial.Serial('/dev/ttyAMA0', 9600, timeout=0.5)
worked = False
try:
        while True:
                string = ser.read(12)
                if len(string) == 0:
                        continue
                else:
                        string = string[1:11]
                        if string == '6A0049F913':
                                worked = SpinMotor(True, 170)
                        elif string == '6A004A16C0':
                                worked = SpinMotor(True, 350)
                        elif string == '770096EE82':
                                worked = SpinMotor(True, 350)
                        elif string == '77009700B7':
                                worked = SpinMotor(True, 350)
                        elif string == '770096CD0B':
                                worked = SpinMotor(True, 350)
                #       print(worked)
except KeyboardInterrupt:
        pass
finally:
        GPIO.cleanup()
```

Not many lines of code were needed.

Most lines are comments.

This is neither efficient or elegant code, but it worked!

Prototype 2

The second prototype was designed to be more practical and closer to a model that can be used in the field.

- A **touchscreen user interface** was added
- A **larger carrying case**
  - Stores **bins** that hold **different types of medicine** and supplies.
- It authenticates using **RFID** as the previous version.
- If the medical personnel is recognized, depending on the **level of access**, it begins **tracking and logging** actions in a **database**.
- **Potentiometers** used for automated tracking of medicine.

rfidmed / RFIDMed2 / source / — Bitbucket - Mozilla Firefox

FOSS Lounge ‹ Log In | NSF Project Update … | rfidmed / RFIDMed2… | Python Meetup # 10… | Bitbucket — The Git …

Atlassian, Inc. (US) https://bitbucket.org/rfidmed/rfidmed2/sr

Search

Atlassian Bitbucket Features Pricing Find a repository… English Sign up Log in

RFID Med / Untitled project / RFIDMed2

Source

master | RFIDMed2 /

README.md	1.1 KB	2015-06-01	Updated the README file to V2
dbReset.py	26.5 KB	2015-06-05	Fix updateQty
db_connect.py	52.9 KB	2015-06-11	resolve conflicts
encrypt.py	2.1 KB	2015-05-26	changes to connectDB.py and medbox2.sql
gpsCalc.py	7.1 KB	2015-06-08	gpscalc now has serial back
gui.py	34.0 KB	2015-06-16	changed gui to register all 8 bins
pots.py	3.0 KB	2015-06-16	changed pots to register all 8 bins

README

This is a work in progress to prototype a medical dispensation device that authenticates on RFID.

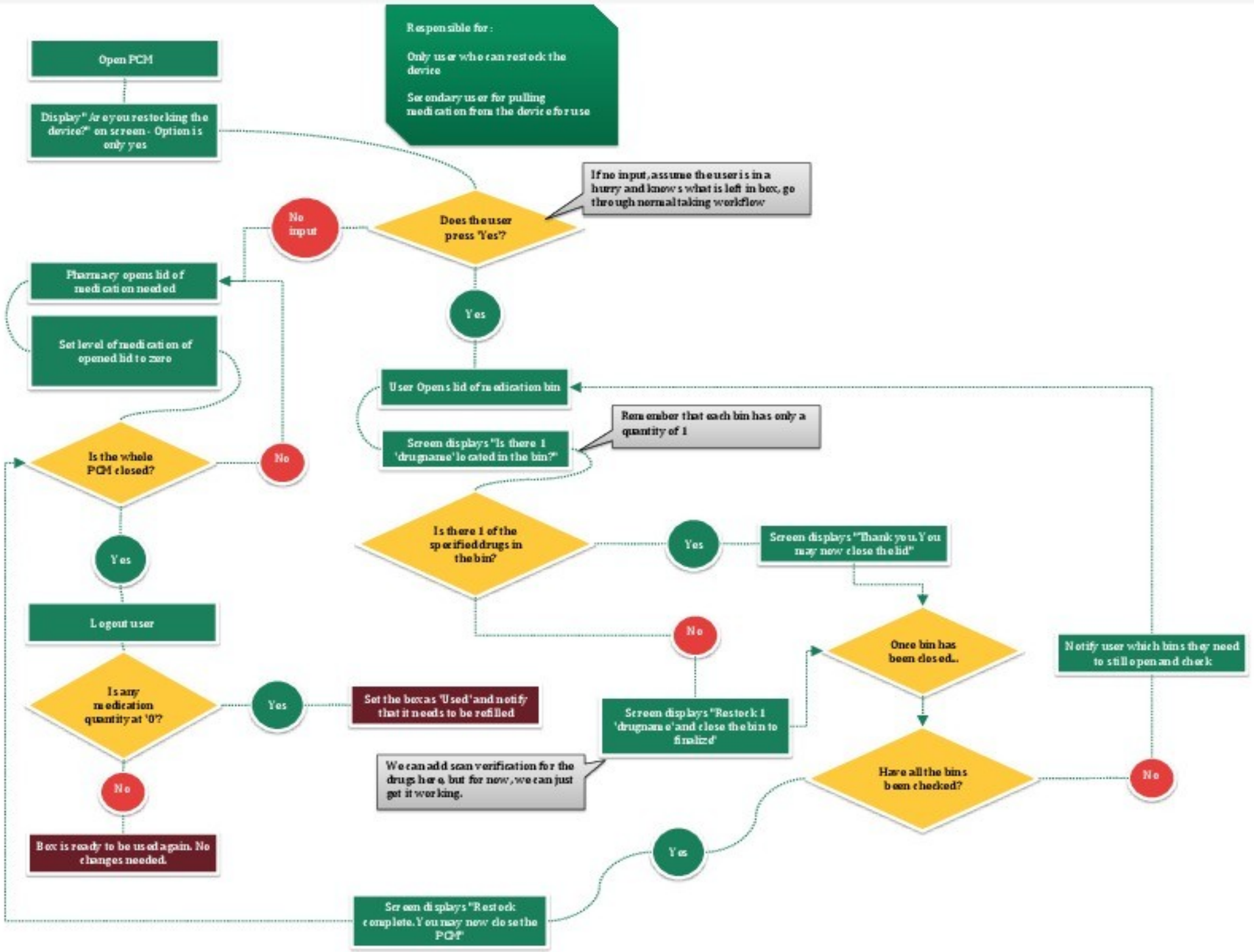
RFID Med

- Version 0.02

Equipment we're using

- Raspberry Pi 2, Raspbian, 8GB

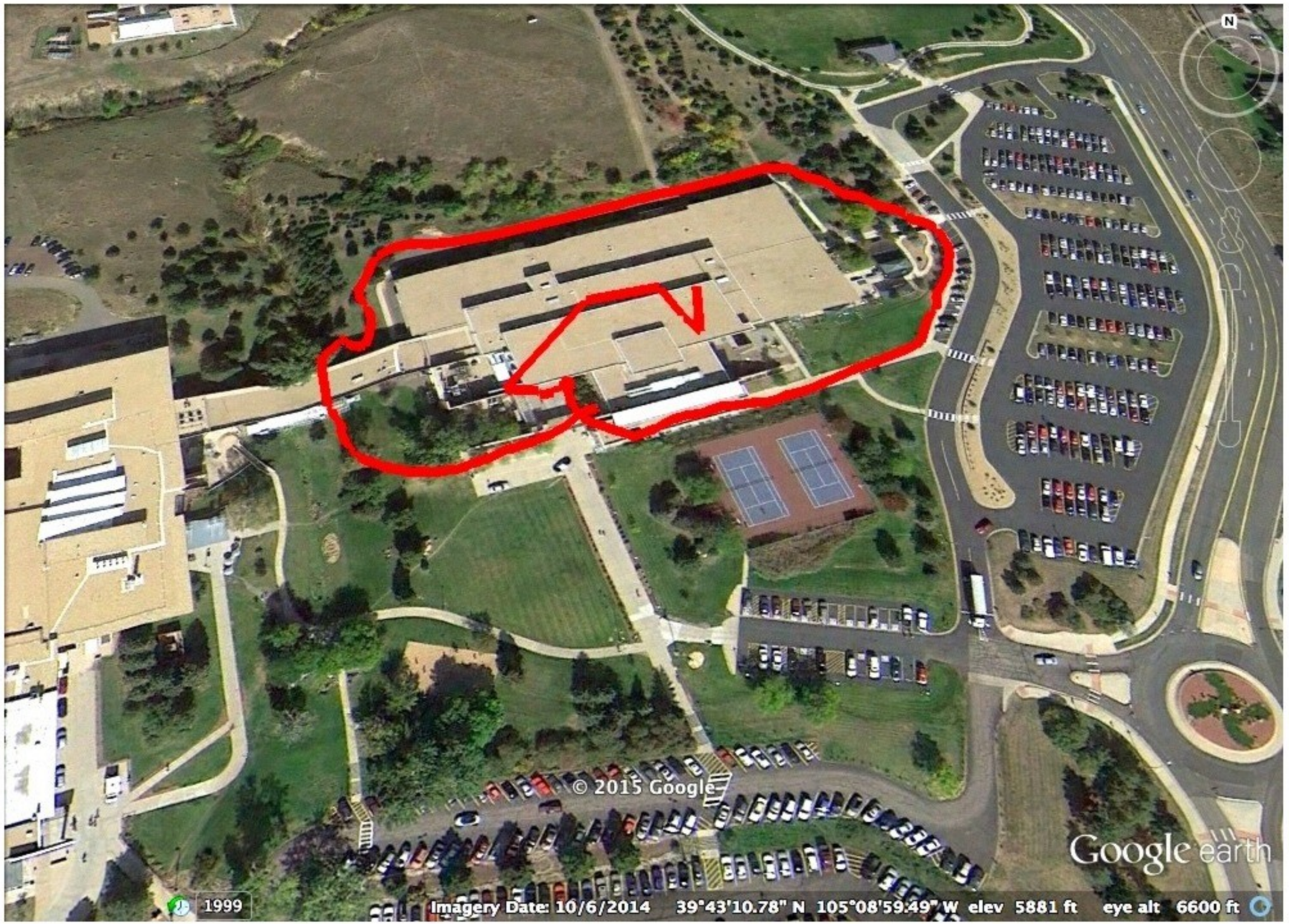
Pharmacy


Responsible for :
Only user who can restock the device
Secondary user for pulling medication from the device for use
Open PCM
Display "Are you restocking the device?" on screen - Option is only yes
If no input, assume the user is in a hurry and knows what is left in box, go through normal taking workflow
Does the user press 'Yes'?
No input
Yes
Pharmacy opens lid of medication needed
Set level of medication of opened lid to zero
User Opens lid of medication bin
Remember that each bin has only a quantity of 1
Screen displays "Is there 1 'drugname' located in the bin?"
Is there 1 of the specified drugs in the bin?
Yes
Screen displays "Thank you. You may now close the lid"
No
Once bin has been closed..
Notify user which bins they need to still open and check
Screen displays "Restock 1 'drugname' and close the bin to finalize"
We can add scan verification for the drugs here, but for now, we can just get it working.
Have all the bins been checked?
No
Yes
Is the whole PCM closed?
No
Yes
Logout user
Is any medication quantity at '0'?
Yes
Set the boxes 'Used' and notify that it needs to be refilled
No
Box is ready to be used again. No changes needed.
Screen displays "Restock complete. You may now close the PCM"


ANAPHYLAXIS
PCM
Portable Computerized Module

Components

- Raspberry Pi 2, Raspbian, 8GB
- RFID Reader ID-20LA (125 kHz)
- SparkFun RFID Reader Breakout
- RFID Tags 125kHz
- Potentiometers - to sense if a bin is open or closed
- A tilt sensor - to see if the box is open or closed
- RGB and Gesture Sensor
- 5" Touchscreen monitor for the user interface.
- GlobalStat BU-353 USB GPS Receiver


N
© 2015 Google
Google earth
1999
Imagery Date: 10/6/2014 39°43'10.78" N 105°08'59.49" W elev 5881 ft eye alt 6600 ft

Life Saving Technology

That Fits in a Backpack

Connecting and starting the RPi

1. Connect the keyboard and mouse into the USB ports.

2. Make sure the monitor is OFF. Connect the HDMI cable and push the button to turn the monitor on.

3. Plug the RPi into the power-strip and plug the miniUSB cable in to power it on.

Demo!

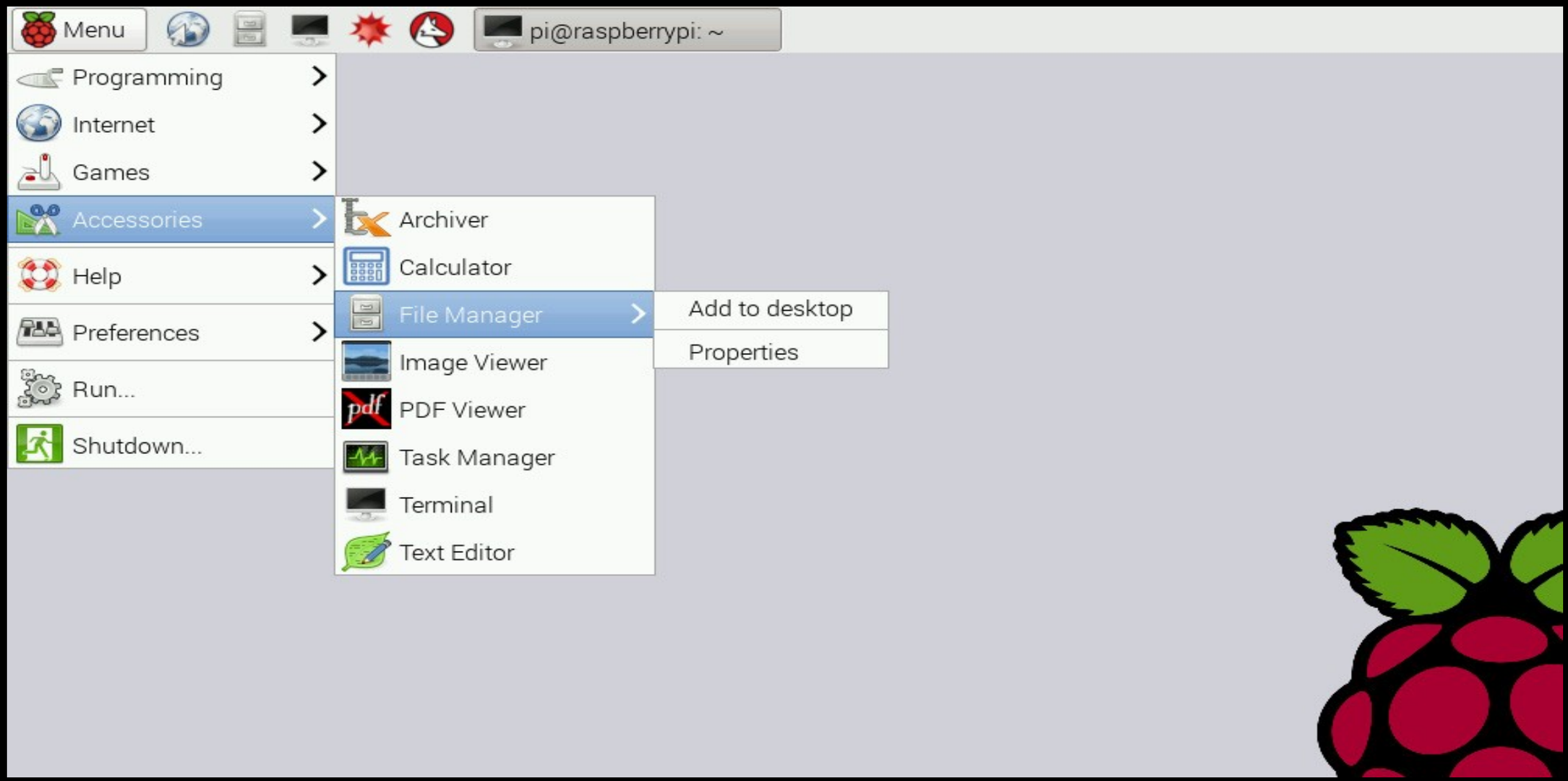
After the initial setup, (connect to a keyboard, screen, and mouse) need a way to access the device.

- SSH – if configured

- USB Console cable – USB to TTL Serial Cable

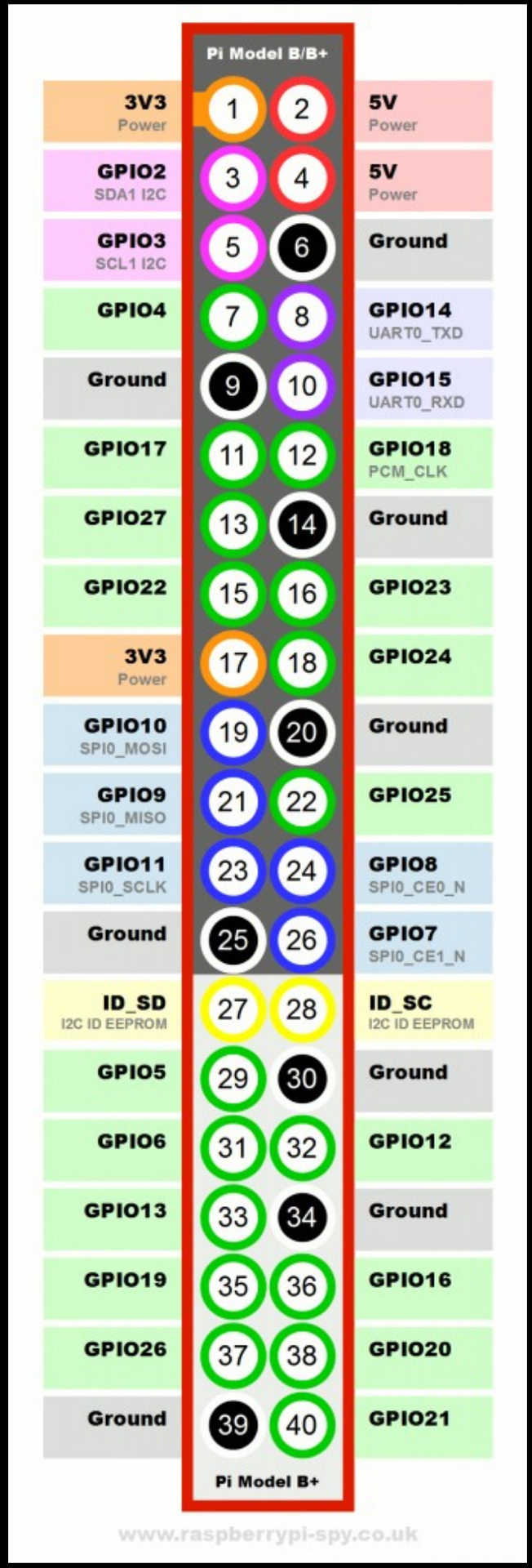
(Powered by 5V, but logic RX and TX is 3.3V)

There's also the GUI

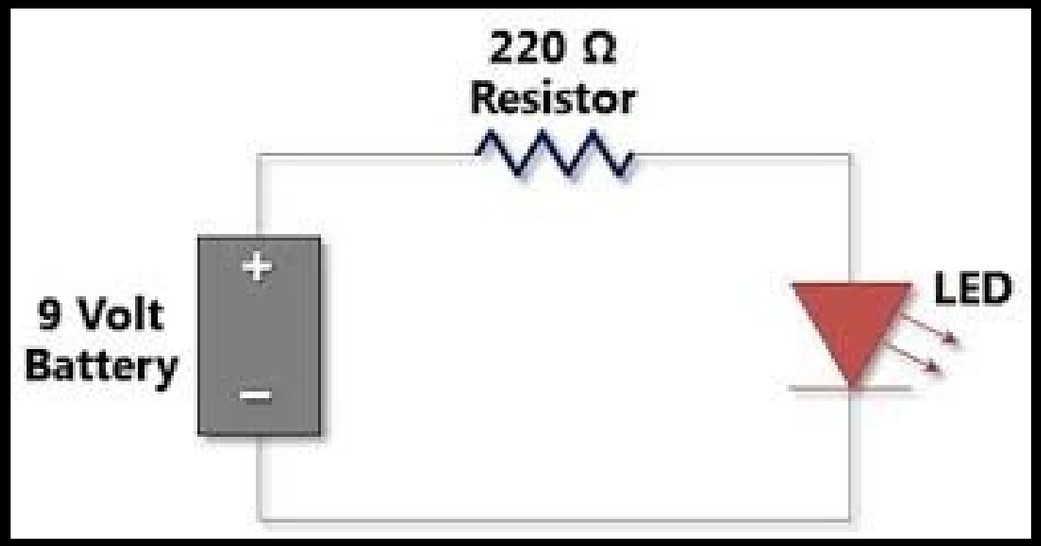


GPIO Pins

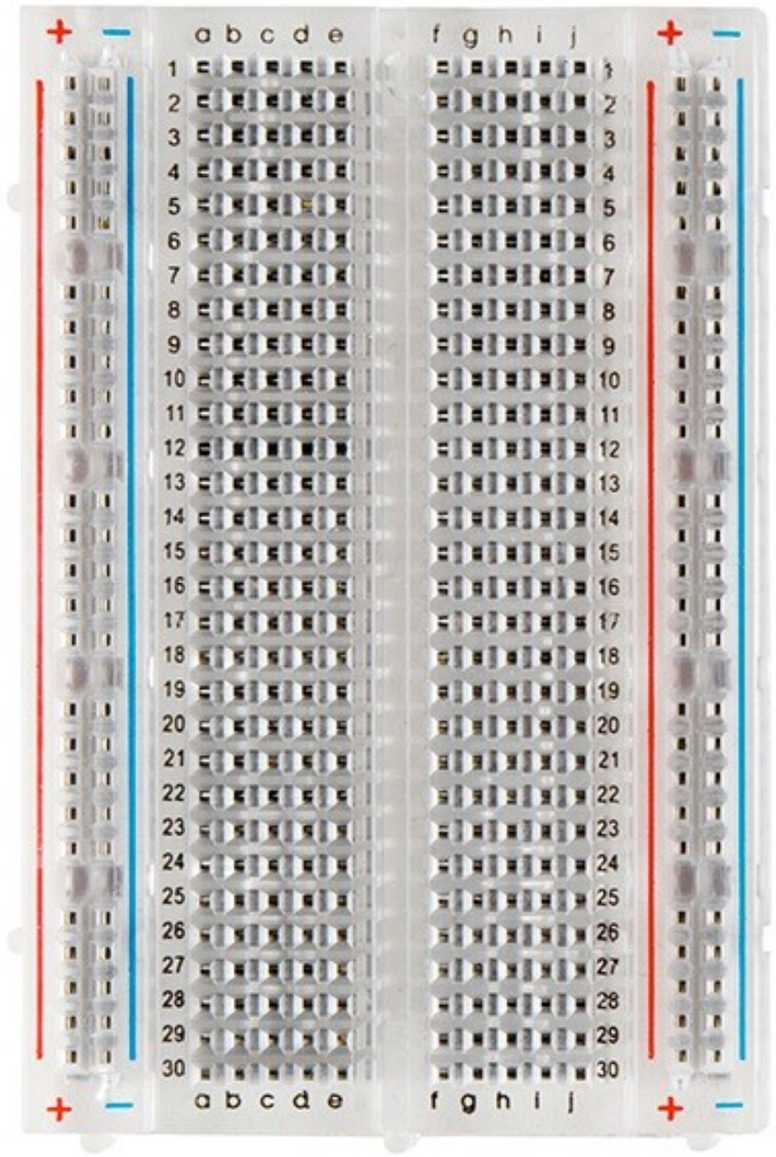
- Logic is 3.3 volts
- Power is 3.3v on left, 5v on right.
- Applying 5v into RPi will damage over time.
- GPIO pin numbers don’t match with pin numbers.
- If don’t have handy chart, flip over. Square pin is #1.

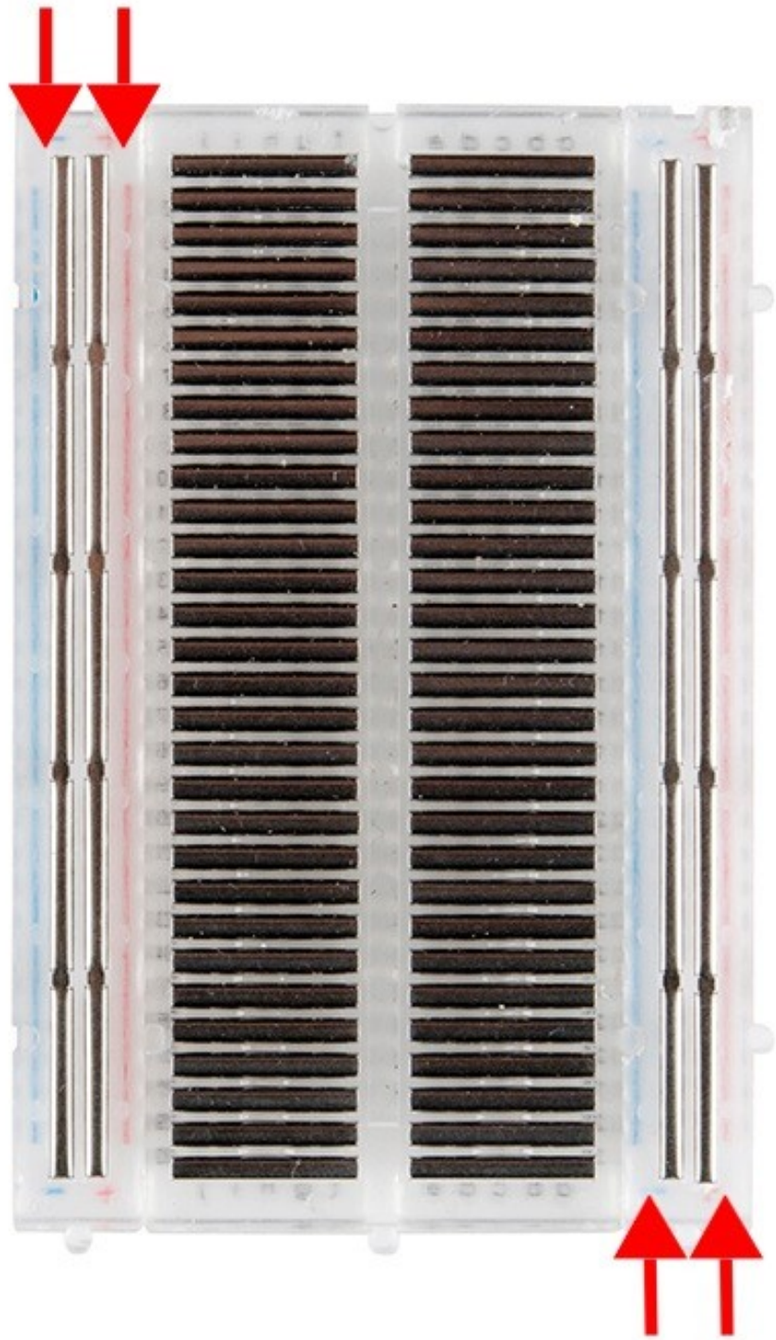


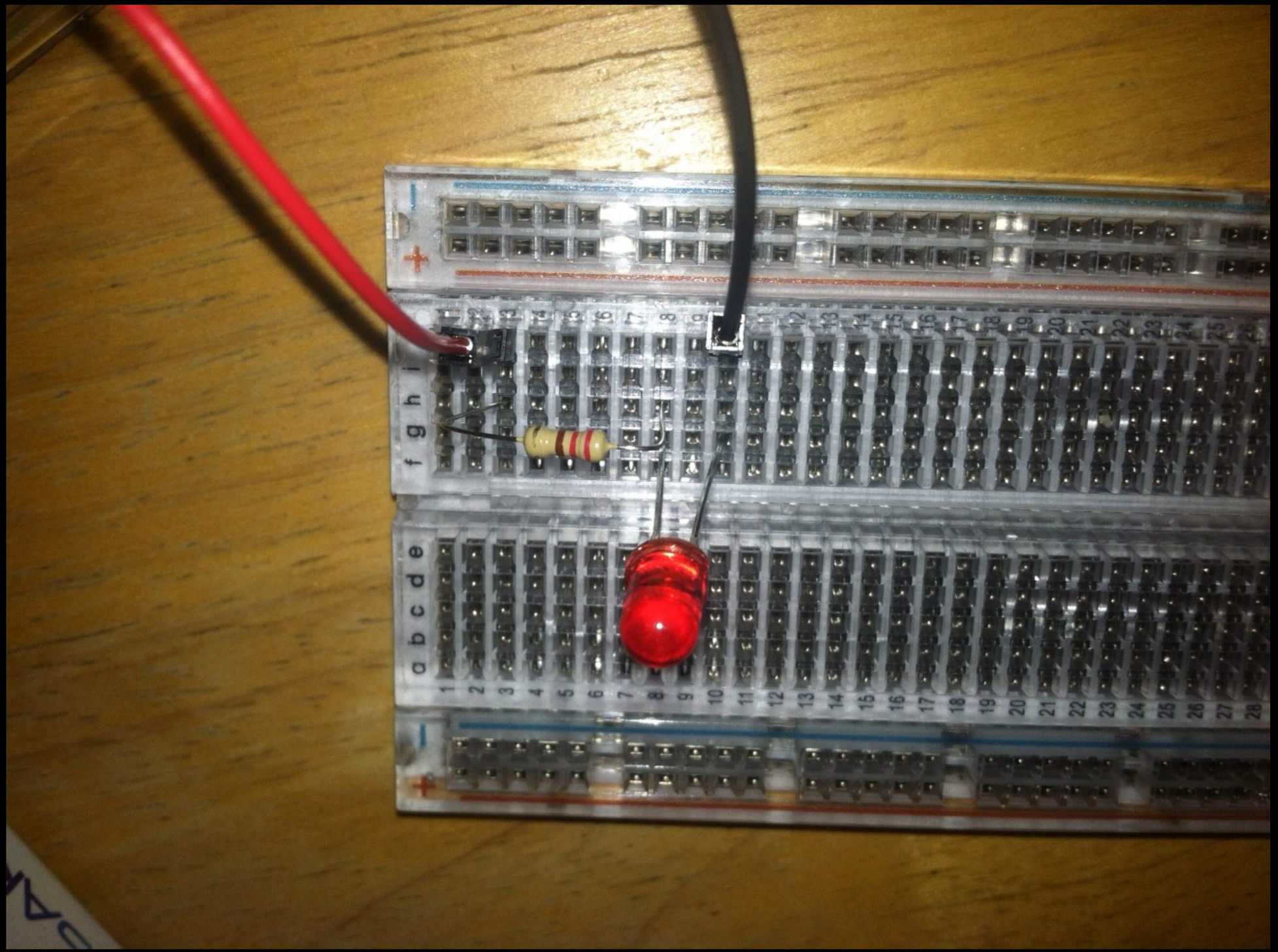
Simple Circuit



- Breadboard
- 330 Ω (Ohm) resistor
- 2 jumper wires (male-to-female)
- LED


+ −
a b c d e
f g h i j
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30




f g h i
a b c d e

```
$ screen /dev/cu.usbserial-A403CSYM
115200

$ minicom -b 115200 -o D /dev/ttyUSB0
```

-b is the baud rate,
-o – don't initialize,
-D specify the device

python

```
import time
import RPi.GPIO as GPIO

LED = 11
GPIO.setmode(GPIO.BOARD)
GPIO.setup(LED, GPIO.OUT)
GPIO.output(LED, GPIO.LOW)

try:
  while True:
    GPIO.output(LED, GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output(LED, GPIO.LOW)
    time.sleep(0.5)
except  KeyboardInterrupt:
  GPIO.cleanup()
```

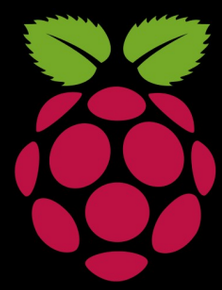
- $ sudo python blinky.py # to run
- $ rm blinky.py # to delete file
- $ sudo shutdown -h now # to shutdown
- (minicom, control A, q for quit)
- Wait for green light to stop blinking, then unplug power (miniUSB).

Resources

SparkFun Electronics
https://www.sparkfun.com/

AdaFruit
https://www.adafruit.com/

Raspberry Pi Foundation
https://www.raspberrypi.org



What are your ideas?

- Go to **FossLounge.org** and place them into the “Interested?” form

- Soldering classes – Teachers / students

Questions?

Thank you!

FossLounge.org