RaRa Academy: Raspberry Pi

Karl Heinz Kremer - K5KHK

Why Are We Here?

- I cannot convert you into a Raspberry Pi (or Linux) expert in two hours

- I cannot teach you everything there is to know about using a Raspberry Pi

- I hope that I can give you some guidance about how to get started and create some interest in using a Pi in your own projects

What is a Raspberry Pi?

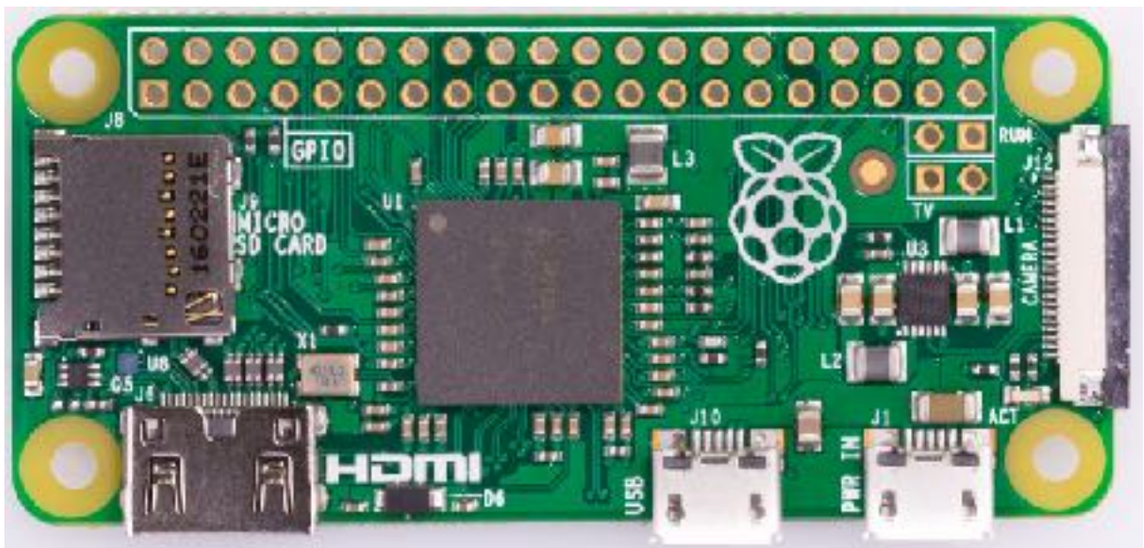
- It’s a REAL computer for $5 to $35 - BYOKMDP
- Created in 2012 by the Raspberry Pi Foundation in the UK to teach basic computer science in schools and developing countries
- Why Raspberry and why Pi?

The Pi Family

- Raspberry Pi 3 B+
    - 1.4GHz 64-bit quad-core CPU with 1GB of RAM
- Raspberry Pi 3 A+
    - same as above
- Raspberry Pi Zero and ZeroW
    - 1GHz single-core CPU with 512MB of RAM


PWR ACT
Made in the UK
DISPLAY
J1 PWR IN
GPIO J8
Raspberry Pi 3 Model B+
© Raspberry Pi 2017
HDMI
HDMI CAMERA J3
A/V
PEN
RUN
USB
USB
ETHERNET


PWR ACT
Made in the UK
DISPLAY
J1 PWR IN
GPIO J8
Raspberry Pi 3 Model A+
© Raspberry Pi 2018
HDMI
HDMI CAMERA J3
A/V
PEN
RUN
USB


J8
GPIO
MICRO SD CARD
U1
L3
RUN
TV
U3
L1
CAMERA
X1
U6
L2
J10
J1
ACT
HDMI
USB
PWR IN

More About the Family

- Raspberry Pi Zero/ZeroW is a cheaper model, with less memory and reduced input and output capabilities. The “W” stands for WiFi and Bluetooth

- Processor is based on the ARM architecture (it's not an Intel compatible processor as is used in Windows PCs and Mac computers)

So, it’s like an Arduino?

No, there is a big difference between an Arduino and a Raspberry Pi: The Arduino does not have an operating system, whereas the Pi is a “real” computer that can run applications. The Arduino is programmed with one “sketch”, and unless it’s re-programmed, that is all it will run. You may remember from the Arduino RaRa Academy presentation that the Arduino software runs in a loop.

The Pi can run many different programs at the same time.

Software

- The Pi can run a number of different operating systems, but it's "normal" OS is Linux in the form of something called "Raspbian" - this is a Linux distribution specifically created for the Raspberry Pi

- Raspbian is available in three different “sizes”:
  - Graphical user interface with applications
  - Graphical user interface
  - Command line interface (“Lite”)

Linux

- It’s a UNIX style operating system
- Created in 1991 by a kid named Linus Torvalds from Finland
- Remember DOS? The command line should be your friend again.
- When installed with a GUI (graphical user interface), it looks and behaves very similar to Windows or macOS.

SHUT A LINUX SYSTEM DOWN PROPERLY!

I Have A Pi, Is That It?

- BYOKMDP - Bring Your Own Keyboard, Mouse, Display, Power
- Can use a TV with HDMI input as display
- USB Mouse and Keyboard
- USB Power Supply that can deliver at least 2.5A

USB

- The Pi has two different types of USB ports: One micro USB connector that is used for it's power supply, and one or more "normal" USB ports that allow you to connect devices (mouse, keyboard, GPS, audio devices, disk drives …).

- When connecting too many power hungry USB devices, use a powered USB hub

One More Thing

- The Pi does not have a disk drive. It uses a SD memory card (these days, a micro SD card) for storage purposes.

- It should be fast!

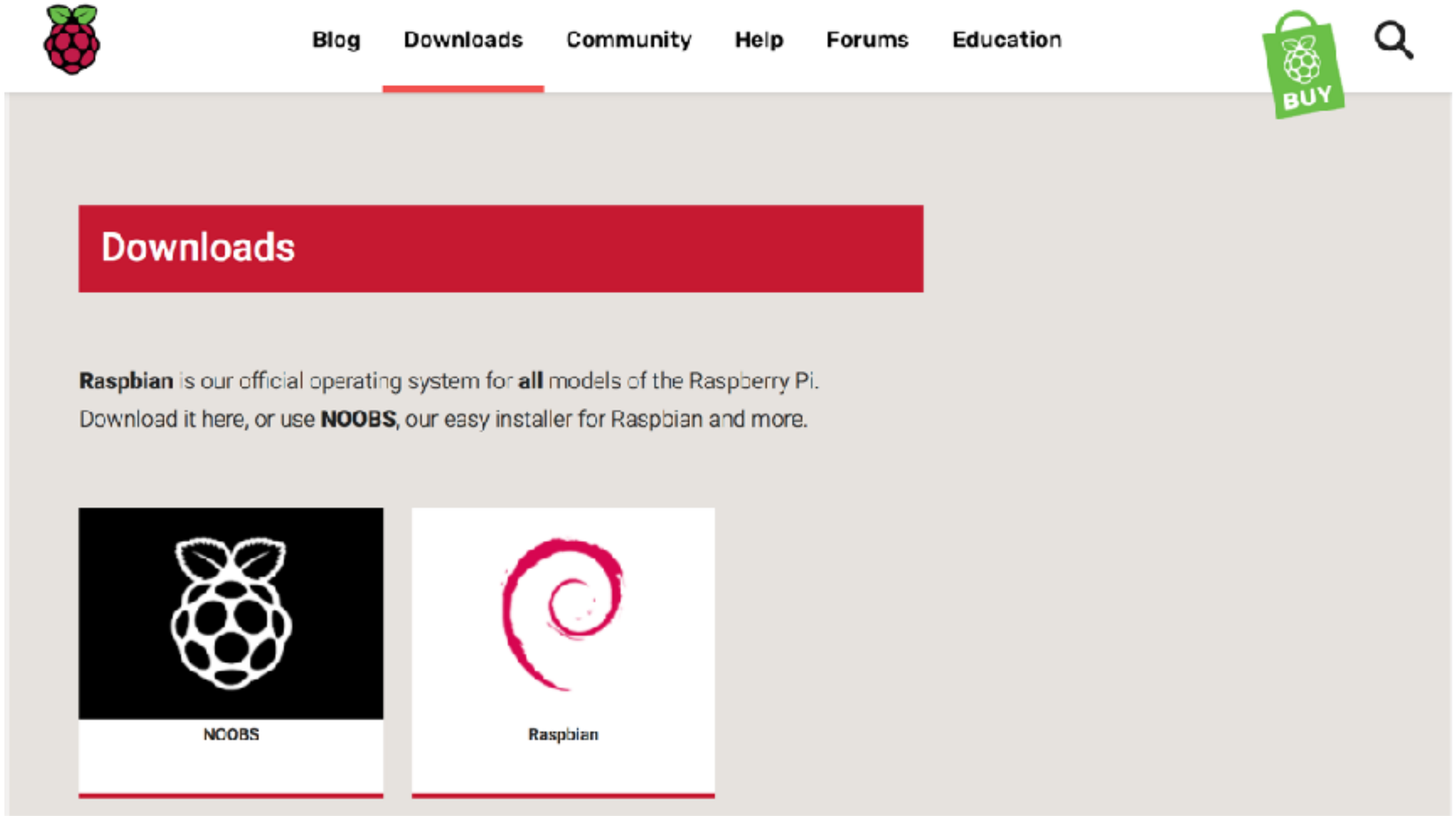
- SD cards were not designed for this, and may not last forever.

That's all you need as long as the memory card contains an operating system configured to "do something"

Setting up the SD Card

- How do we get the operating system on the SD card?
- Two methods:
  - Use the “NOOBS” approach (“**N**ew **O**ut **O**f the **B**ox **S**oftware”)
  - Write a card image to the card using a specific application

Downloading the OS



NOOBS

- Format the memory card with the FAT file system (chances are, if it's a new one, it is already formatted that way).

- Copy & Paste the files from the ZIP file to the memory card.

- Insert the card into the Pi and apply power.

- Done - OK, not quite yet, there are a few things that need to be set up once the system comes up

- It is possible to buy a SD card with pre-installed operating system, but that could potentially mean that you start out with an old version of the system.

- Installing from scratch is not complicated and pretty straight forward.

- The “Noobs” system makes the installation easier, creates a recovery partition, but uses up more space on your memory card. The Raspbian version is the same.

- If you want to hear the “bits crackle” use a direct Raspbian download - but be prepared for a more involved installation.

- You need to set the keyboard layout (and timezone) – if you don’t, your US keyboard will not work correctly.

Programming in Python

- Use the “Idle” development environment
- Follow a tutorial!
- Practice Practice Practice

Small Sample Program

```
for i in range(2):
    print(“A")
print("B")
```

The Same in C

```
for (int i=0; i<2; i++) {
    printf(“A”);
}
printf(“B”);
```

Or this:

```
for (int i=0; i<2; i++) { printf(“A”); }
printf(“B”);
```

Or this:

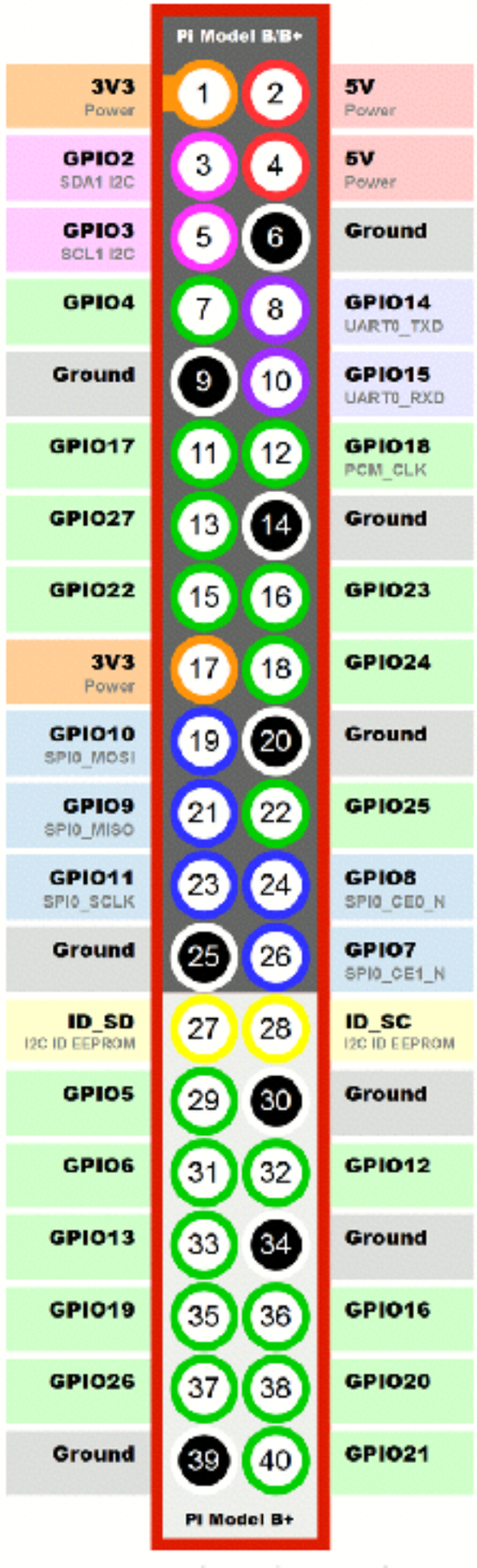
```
for (
   int i=0;
      i<2;
         i++)
            {
printf(“A”);
   }
      printf(
         “B”
               )
                  ;
```

- Find a good Python tutorial and start playing with this programming language.
- There is also “Scratch”, which is a graphical programming environment that might be a good introduction to thinking in “code” for people who want to ease into programming slowly.

GPIO

- GPIO stands for “General Purpose Input/Output” and refers the the header block that is on the Raspberry Pi. The user can read from the input pins, and write to the output pins.

- The Pi uses 3.3V logic – make sure you do not connect 5V logic level devices without a level shifter.

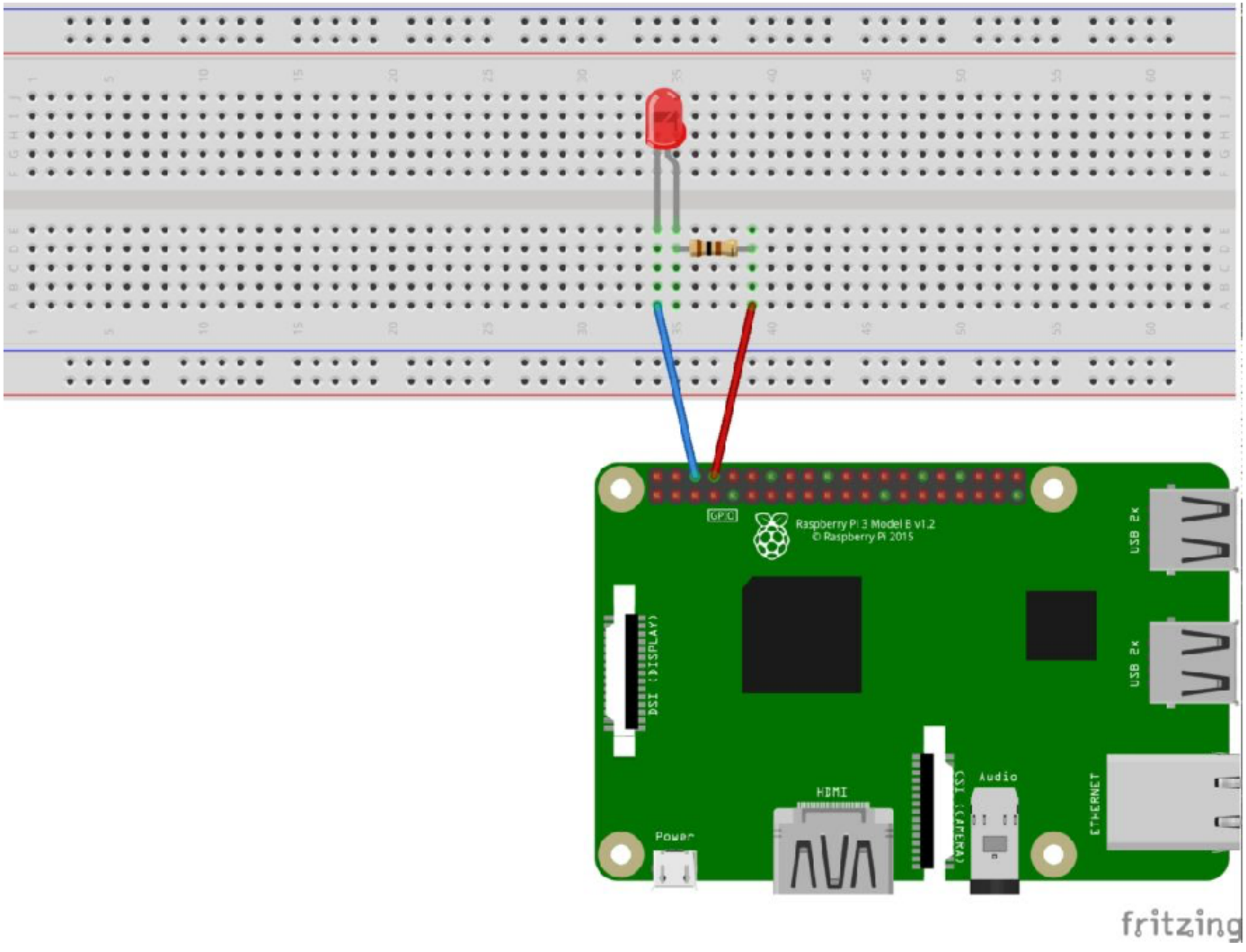


Blink LED, Blink!

```
import RPi.GPIO as GPIO     # Import Raspberry Pi GPIO library
from time import sleep      # Import the sleep function from the time module

GPIO.setwarnings(False)     # Ignore warning for now
GPIO.setmode(GPIO.BOARD)    # Use physical pin numbering
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW)   # Set pin 8 to be an output pin and
                                            # set initial value to low (off)

while True: # Run forever
    GPIO.output(8, GPIO.HIGH) # Turn on
    sleep(1)                  # Sleep for 1 second
    GPIO.output(8, GPIO.LOW)  # Turn off
    sleep(1)                  # Sleep for 1 second
```


GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

Keeping the Pi Up To Date

- Software “ages” very fast
- Need to keep the operating system up to date
- Use the following commands to update the operating system:

```
sudo apt-get update
sudo apt-get upgrade
```

Updating the Pi

- The two commands first UPDATE the list of available software, and then UPGRADE those software packages that are out of date.

- Some software packages will not get upgraded to the latest version, there is another command that will do that:

```
sudo apt-get dist-upgrade
```

- A normal “upgrade” will never remove any packages, a “dist-upgrade” will remove packages to resolve conflicts.

Installing Software

- There are different methods to install software. The most straight forward one uses the “apt-get” tool to manage packages. The graphical tool to manage software on a Pi also uses “apt-get” in the background.

- If you are using the GUI version of Raspian, it’s convenient to search.

- To use the command line version, use something like this:

```
sudo apt-get install some_package_name
```

- This works as long as there is a “package” available for the Raspbian distribution, and stored in a repository that apt-get can access.

- If that’s not the case, we have to resort to some lower level processes to get software installed.

- If a “.dpk” file is available (e.g. WSJT-X), the best way to install it is using the GUI file manager. Just open (double-click) the file in the file manager and let the operating system figure out what to do.

- If that’s not possible either, we may have to compile software.

- That’s when we can get into trouble, because nobody is protecting us from ourselves :)

- Let’s take a look at an example: FreqShow

FreqShow

- Shows a frequency spectrum using an RTL-SDR dongle
- LCD touch screen screen (Pi Hat)

- Requires that we install software using the package manager, but also compile and install software on our own.

APRS Tracker

- Uses a GPS to determine position, and then sends that information out using an HT.

- Need to interface the audio with the HT

- Raspberry Pi only has audio output, no input, so for anything that requires input, we need to use an external audio interface (“sound card”)

- Need to use GPIO to control PTT

- Used “Direwolf” as the main software to handle the tracking. Instructions for the setup are included in the Direwolf documentation.

APRS Receive Only Gateway

- Uses RTL-SDR dongle to monitor APRS frequency 144.39MHz
- Uses Direwolf to forward received messages to an online gateway

WSPR Transmitter

- A Raspberry Pi can transmit!
- It can produce pretty much any type of modulation by turning a GPIO pin on and off.
- The output is very dirty and REQUIRES a low pass filter before an antenna can be attached. A simple piece of wire will act as an antenna.
- See https://www.tapr.org/kits_20M-wspr-pi.html for a 20m LP filter that also adds an amplifier in the form of a “Pi Hat”.
- See the documentation for information about software that can produce SSB/FSQ/NFM/AM/WSPR/CW/SSTV and a frequency sweep.

Fldigi and WSJT-X

- To connect the Raspberry Pi, you need a USB audio interface and a way to control PTT.

- If CAT control is available, that can usually handle PTT as well, otherwise a USB serial port is required.

- Don’t need expensive SignaLink interface or similar, but may need some ferrite cores to block HF getting into the Pi. The Pi is relatively sensitive to HF.

Other “Stuff”

- Signal Generator
- Digital (audio card based) oscilloscope
- ADS-B Flight Tracking - PiAware
- RTL_433 (e.g. capture data from weather station)
- DMR hot spot
- Interface for digital modes (DRAWS, MFJ RigPi)

Non-Raspberry Pi

- There is are other single-board computers that are very similar to the Raspberry Pi
- Some of them use the same GPIO port
- Some are faster, cheaper, come with built-in memory (no SD card required)…
- The community around the Raspberry Pi is definitely the largest

Resources

- The official Raspberry Pi site is full of good information: https://www.raspberrypi.org

- The MagPi is a magazine that is published in a printed edition, but can be downloaded for free.
  In addition to the MagPi magazine, they also publish other documents: A “Beginners Guide” and a “Projects Book” – volume 3 of the Projects Book has a Python tutorial: https://www.raspberrypi.org/magpi/issues/

- AdaFruit has a lot of add-ons for the Pi and also tutorials and guides: https://learn.adafruit.com/category/learn-raspberry-pi

- Instructables has lots of projects and tutorials: https://www.instructables.com/howto/raspberry+pi/

- How to install NOOBS (old but has a video): https://www.raspberrypi.org/blog/introducing-noobs/ https://www.raspberrypi.org/documentation/installation/noobs.md

- Noobs documentation: https://github.com/raspberrypi/noobs/blob/master/README.md

- Raspberry Pi for Ham Radio Group: https://groups.io/g/RaspberryPi-4-HamRadio

- Tiny Python Panadapter: https://aa6e.net/wiki/Tiny_Python_Panadapter

- APRS RX Only GW: https://github.com/wb2osz/direwolf/blob/master/doc/Raspberry-Pi-SDR-IGate.pdf

- APRS GPS Tracker: https://github.com/wb2osz/direwolf/blob/master/doc/Raspberry-Pi-APRS-Tracker.pdf

- Blinking LED Example: https://raspberrypihq.com/making-a-led-blink-using-the-raspberry-pi-and-python/

- PiAware Installation: https://flightaware.com/adsb/piaware/install
- RTL_433 https://github.com/merbanan/rtl_433