Intro to Linux and Hardware Using The Raspberry Pi

HackUSU 2019

Objectives

Discussion

- Introduce Linux OS
- Introduce the Raspberry Pi

Labs

- Lab 1: Connecting an LED
- Lab 2: Traffic Light
- Lab 3: Button Input
- Lab 4 : Button with LED

What is Linux? Why should I use it?

- Linux is an OS - manages communication between user input/software and hardware
- Most distributions are free
- Software is typically open-source
- Extremely reliable

From linux.com

The Linux Command Line

- A terminal emulator can be used for entering commands
- BASH: Bourne Again SHell
- BASH shell is most common command language and shell
- Used for direct access to system without writing code
- All programs can be launched from system calls accessible through the terminal

Common Linux Terminal Commands

```
$ ls : lists current directory contents
$ pwd: print working (current) directory
$ cd  <path>: change directory to <path>
$ mkdir <directory name>: mkdir directory named <directory name>
$ touch <filename> : creates a file named <filename>
$ cp  <source file> <destination file> : copies source to destination
$ mv <source file> <destination location> : moves\renames file to destination
$ rm <filename> : remove <filename>
$ make : build linux program using a makefile
$ ./<executable> : runs the program <executable>
$ date : prints out the current system date and time
$ chmod +x <filename> : make a file executable
$ leafpad <filename>: Opens the file in raspberry pi’s text editor
```

Other Terminal Helps and Tips

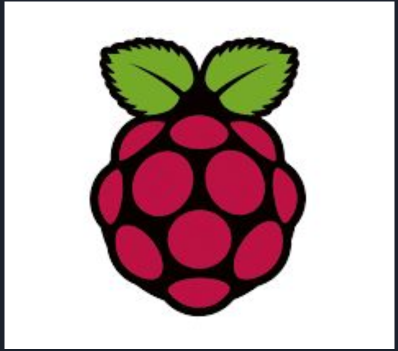
- . - (a single period) a symbol for your current directory. Helpful for copying/moving
- .. - (double periods) the directory up one level. Navigate up one directory with cd ..
- ~ - (tilde) symbol for the current user’s the home directory (/home/<current user>/)
- The up arrow lets you scroll through previous entered commands
- Pressing tab will autocomplete your current word in the command
    - If multiple possibilities, pressing tab twice will display all conflicting autocomplete possibilities
- Cut/Paste into the terminal with Ctrl+Shift+C / Ctrl+Shift+V
- Ctrl+C will force kill your current command/program

What is a Raspberry Pi?

The Raspberry Pi is a single-board mini computer

Can be used to interface with external devices

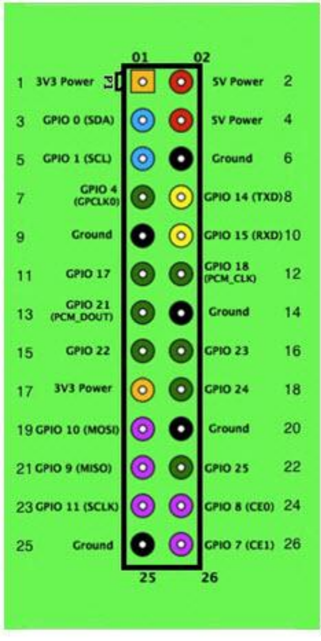
Used by hobbyists for projects and in professional environments

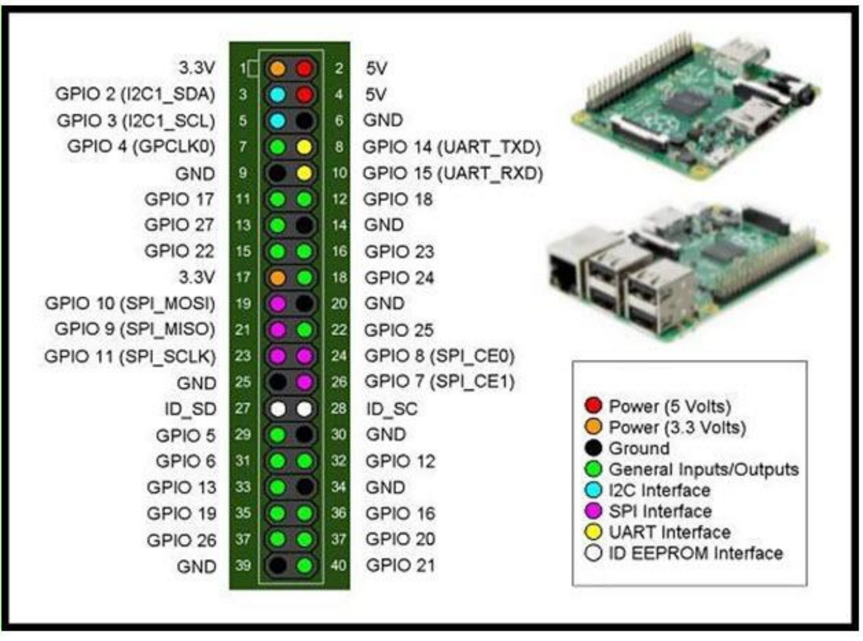


Raspberry Pi Talks to Hardware Easily

Digital communication ports mounted to the filesystem

Easy GPIO access





Lab 1 : Connecting an LED

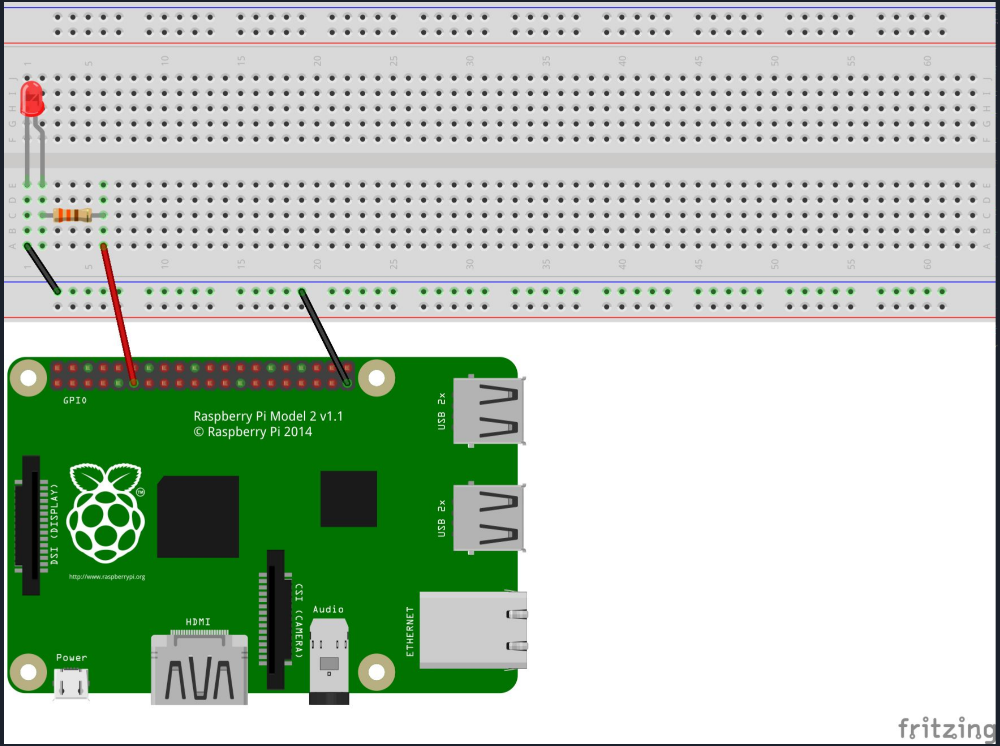
```
import RPi.GPIO as GPIO #gives us access to the GPIO pins
GPIO.setmode(GPIO.BCM)  #sets the pin numbering system we want
to use

LED = 17                #pin number driving the LED
GPIO.setup(LED,GPIO.OUT)

   GPIO.output(LED, True)  #turn LED on

   GPIO.output(LED, False) #turn LED off
```

LED Wiring Circuit



Lab 1 : Connecting an LED

```
from time import sleep

delay = 1     #change this number to adjust the frequency at 
which the light blinks

while True:
   GPIO.output(LED, True)  #turn LED on
   sleep(delay)

   GPIO.output(LED, False) #turn LED off
   sleep(delay)
```

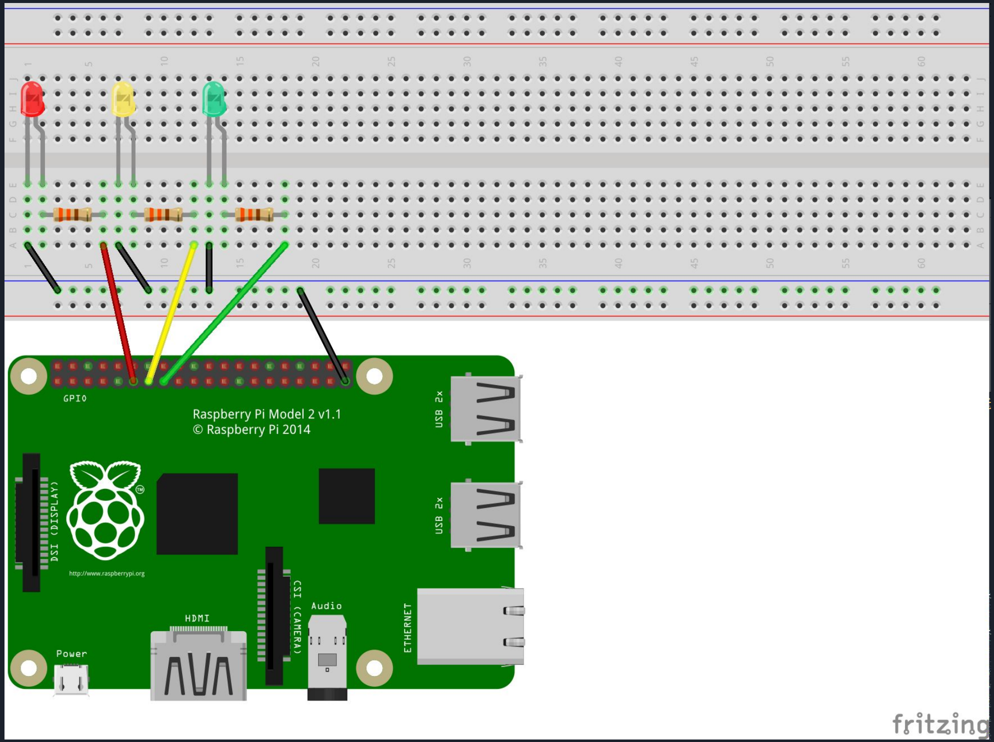
Lab 2: Traffic Light

```
green = 27  #set variables for the pin numbers driving each LED
yellow = 22
red = 17
while True:
   GPIO.output(red, False)  #turn on green light for 2 seconds
   GPIO.output(green, True)
   sleep(2)

   GPIO.output(green, False) #turn on yellow light for 1 second
   GPIO.output(yellow, True)
   sleep(1)

   GPIO.output(yellow, False) #turn on red light for 3 seconds
   GPIO.output(red, True)
   sleep(3)
```

Lab 2: Traffic Light



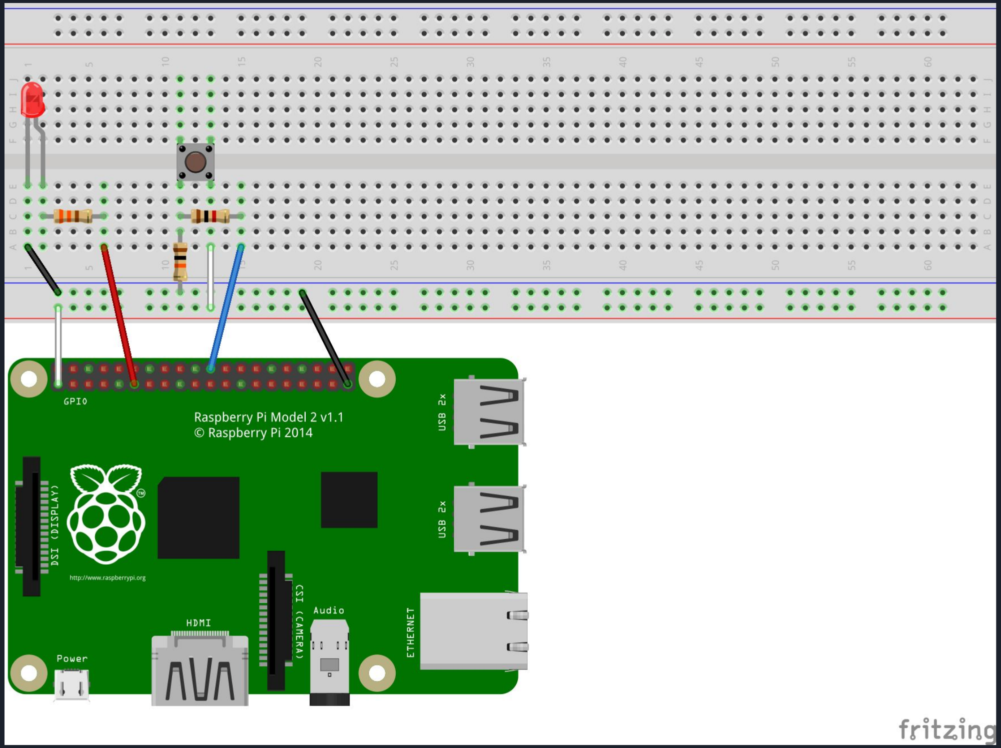
Lab 3: Button

```
button = 25  #set variable for the pin number used
```

```
GPIO.setup(button, GPIO.IN)   #configure GPIO pin to input mode
```

```
while True:
    if GPIO.input(button):
        print("Button pressed")
    else:
        print("Button not pressed")
    sleep(0.1)
```

Lab 3: Button


GPIO
Raspberry Pi Model 2 v1.1
© Raspberry Pi 2014
USB 2x
USB 2x
DSI (DISPLAY)
http://www.raspberrypi.org
CSI (CAMERA)
Audio
HDMI
Power
ETHERNET
fritzing

Lab 4 : Button with LED

```
toggle = False
while True:
    GPIO.output(LED, toggle)

    if GPIO.input(button):
        toggle = not toggle
        print("Button pressed")
        while GPIO.input(button):
            pass
```