Practical no: 3

Title: Study of Connectivity and configuration of Raspberry-Pi /Beagle board circuit with basic peripherals, LEDS. Understanding GPIO and its use in program

Name: Aditi Dinesh Mulay

Class: T.E. Computer

Subject: ES&IOT

Div: A

Roll no: 02

PRN No. 71918146B

ESIOT

Aditi Dinesh Mulay
T.E. Comp Div: A
Roll no: 02

Practical 3

Aim:

Study of connectivity & configuration of Raspberry-Pi/ Beagle Board circuit with basic peripherals, LEDs. Understanding GPIO and its use in program.

Theory

1. raspi-config

The Raspberry-Pi configuration tool in Raspbian, allowing you to easily enable features such as the camera, and to change your specific setting such as keyboard layout.

2. config.txt. -The Raspberry-Pi configuration file.

3. Wireless - Configuring your-Pi to connect to a wireless network using the Raspberry Pi3 Pi zero W's inbuilt wireless connectivity, or a USB wireless dongle.

4. Wireless Access Points - Configuring your Pi ~~to~~ as a ~~connect to~~ wireless access point using Raspberry Pi3 and Pi zero W's, In built wireless connectivity, or a USB wireless dongle.

5. Audio Config - Switch your audio o/p between HDMI and 3.5mm jack.

6. Camera Config - Installing & setting up the Raspberry-Pi camera board.

7. External storage config - Mounting & setting up external storage on a Raspberry-Pi.

8. Localisation - Setting up your Pi to work in your local language / time / zone.

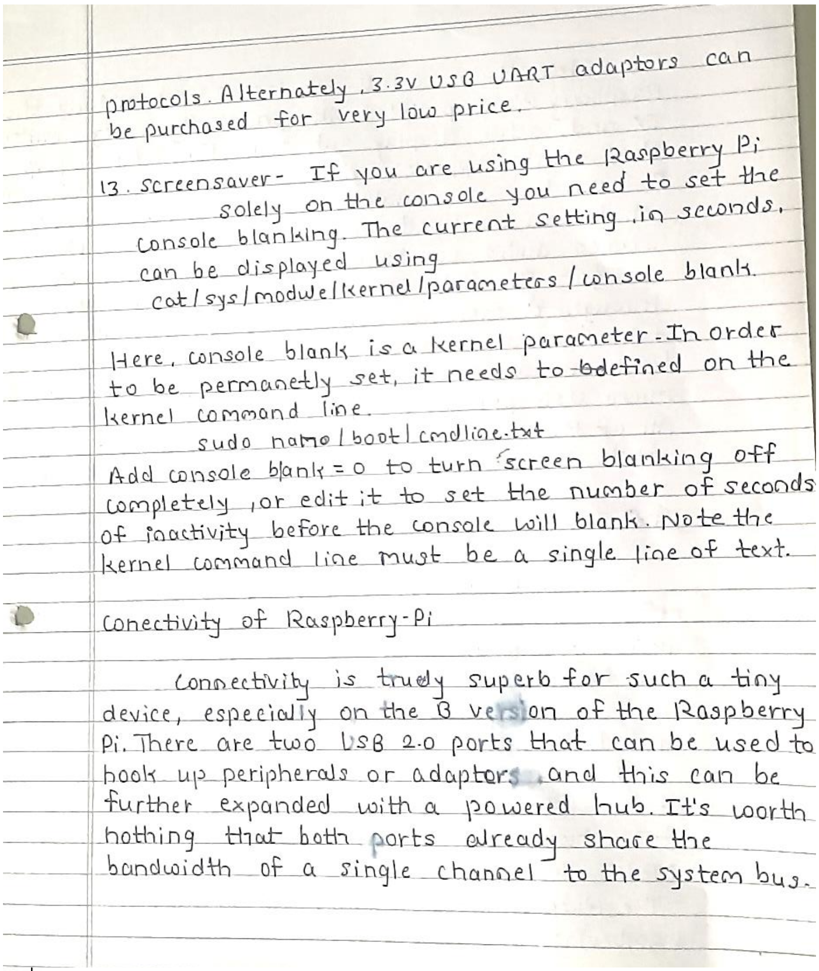
9. Default pin Config - Changing the default pin states.

10. Device Trees Config - Device Trees, overlays &parameters.

11. Kernel Command Line - Linux kernel accepts a command line of parameters during boot. On the Raspberry Pi, this command line is defined in a file in the boot partition, called cmdline.txt. This is a simple text file that can be edited using any text editor, eg. Nano.

```
sudo nano /boot/cmdline.txt.
```

12. UART Configuration - The SOCs used on the Raspberry Pis have two built in UARTs, a PL011 and a mini UART. They are implented using different hardware blocks, so they have slightly different characteristics. However, both are 3.3V devices which means extra care must be taken when connecting up to an RS232 or other system that utilizes different voltage levels. An adaptor must be used to convert the voltage levels between the two.



protocols. Alternately, 3.3V USB UART adaptors can be purchased for very low price.

13. Screensaver- If you are using the Raspberry Pi solely on the console you need to set the console blanking. The current setting, in seconds, can be displayed using

```
cat/sys/module/kernel/parameters/console blank.
```

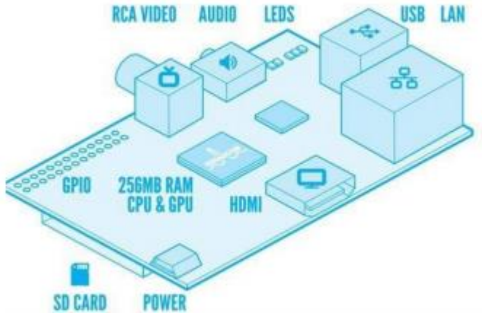
Here, console blank is a kernel parameter. In order to be permanetly set, it needs to ~~b~~defined on the kernel command line.

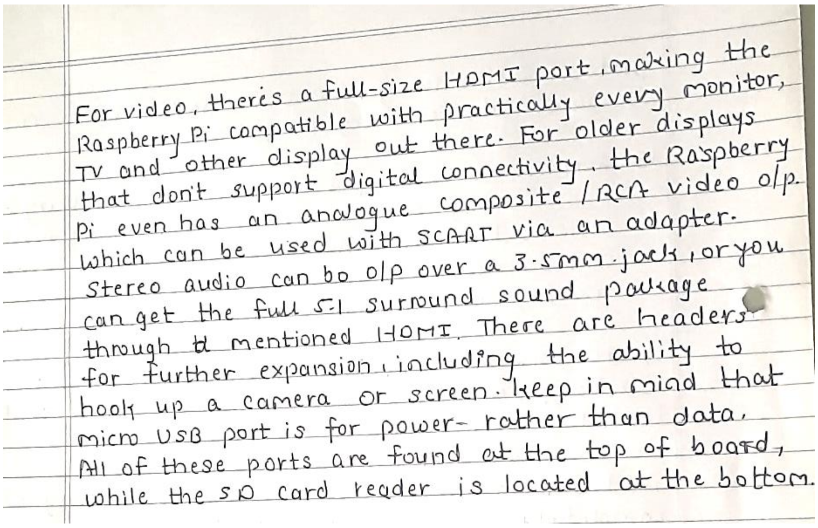
```
sudo nano/boot/cmdline.txt
```

Add console blank = 0 to turn screen blanking off completely, or edit it to set the number of seconds of inactivity before the console will blank. Note the kernel command line must be a single line of text.

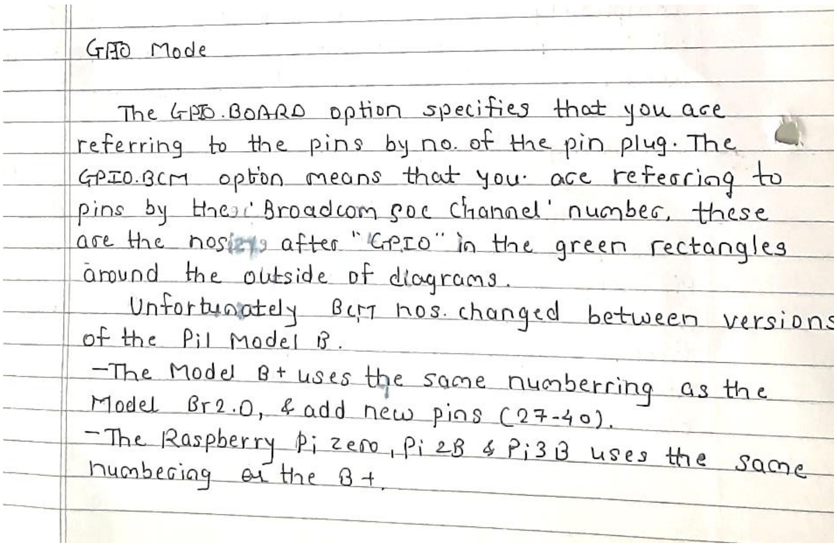
Conectivity of Raspberry-Pi

Connectivity is truly superb for such a tiny device, especially on the B version of the Raspberry Pi. There are two USB 2.0 ports that can be used to hook up peripherals or adaptors, and this can be further expanded with a powered hub. It's worth hothing that both ports already share the bandwidth of a single channel to the system bus.





For video, there's a full-size HDMI port, making the Raspberry Pi compatible with practically every monitor, TV and other display out there. For older displays that don't support digital connectivity, the Raspberry Pi even has an analogue composite / RCA video o/p. which can be used with SCART via an adapter. Stereo audio can bo o/p over a 3.5mm jack, or you can get the full 5.1 surround sound package through ~~tt~~ mentioned HDMI. There are headers for further expansion, including the ability to hook up a camera or screen. Keep in mind that micro USB port is for power- rather than data. All of these ports are found at the top of board, while the SD card reader is located at the bottom.

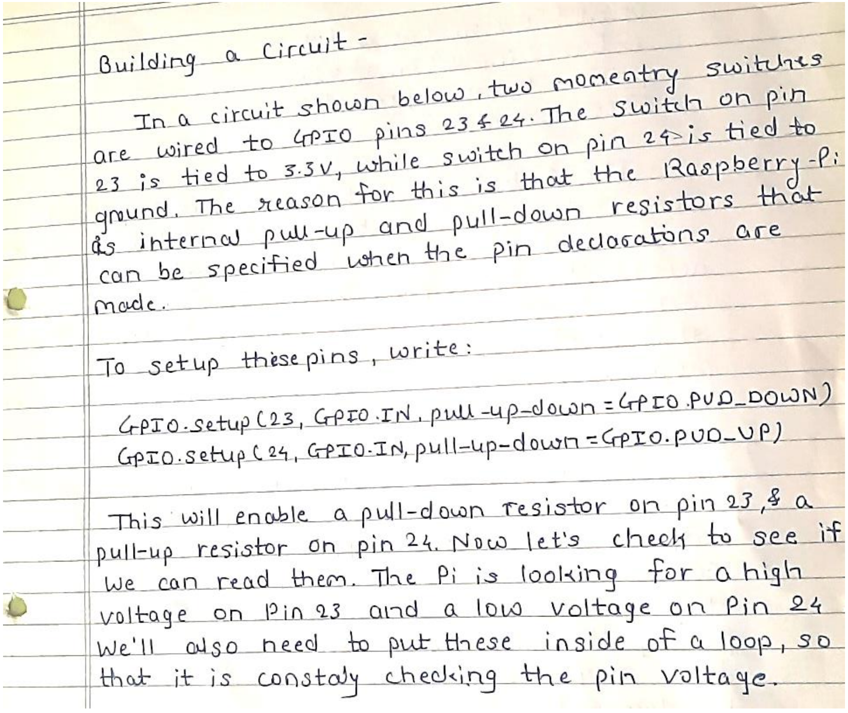


GPIO Mode

The GPIO.BOARD option specifies that you are referring to the pins by no. of the pin plug. The GPIO.BCM option means that you are referring to pins by the 'Broadcom SOC Channel' number, these are the nos after "GPIO" in the green rectangles around the outside of diagrams.

Unfortunately BCM nos. changed between versions of the Pi Model B.

- The Model B+ uses the same numberring as the Model Br2.0, & add new pins (27-40).
- The Raspberry Pi zero, Pi 2B & Pi3B uses the same numbering as the B+.



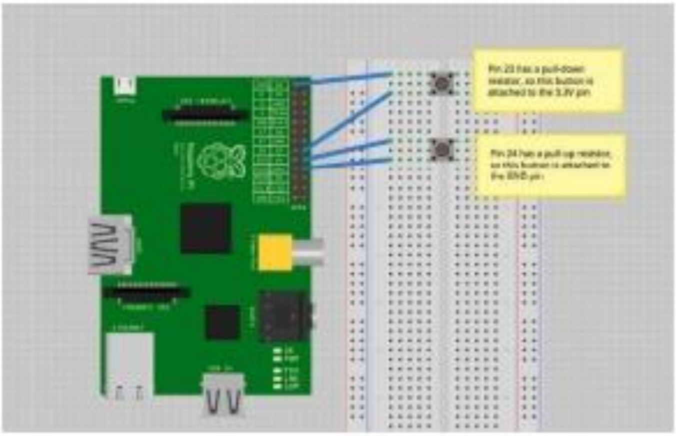
Building a Circuit -

In a circuit shown below, two momeatry switches are wired to GPIO pins 23 & 24. The switch on pin 23 is tied to 3.3V, while switch on pin 24 is tied to ground. The reason for this is that the Raspberry-Pi is internal pull-up and pull-down resistors that can be specified when the pin declarations are made.

To setup these pins, write:

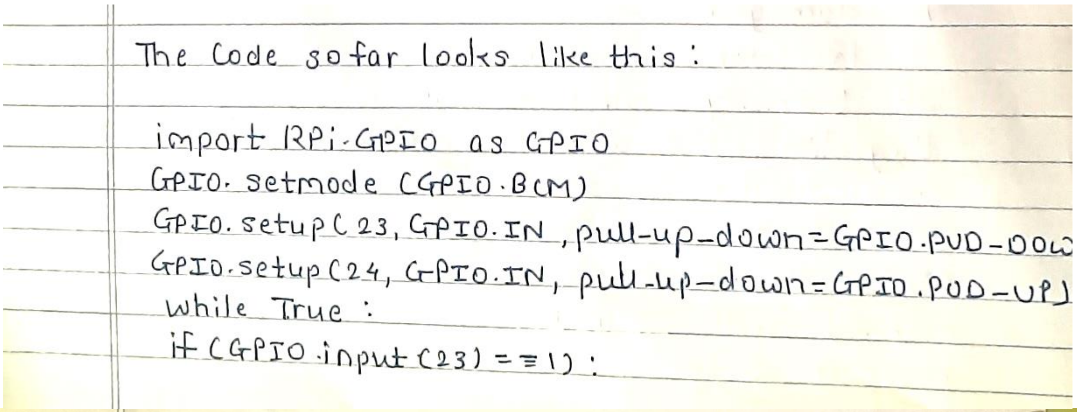
```
GPIO.setup(23, GPIO.IN, pull-up-down=GPIO.PUD_DOWN)
GPIO.setup(24, GPIO.IN, pull-up-down=GPIO.PUD_UP)
```

This will enable a pull-down resistor on pin 23, & a pull-up resistor on pin 24. Now let's check to see if we can read them. The Pi is looking for a high voltage on Pin 23 and a low voltage on Pin 24. We'll also need to put these inside of a loop, so that it is constaly checking the pin voltage.



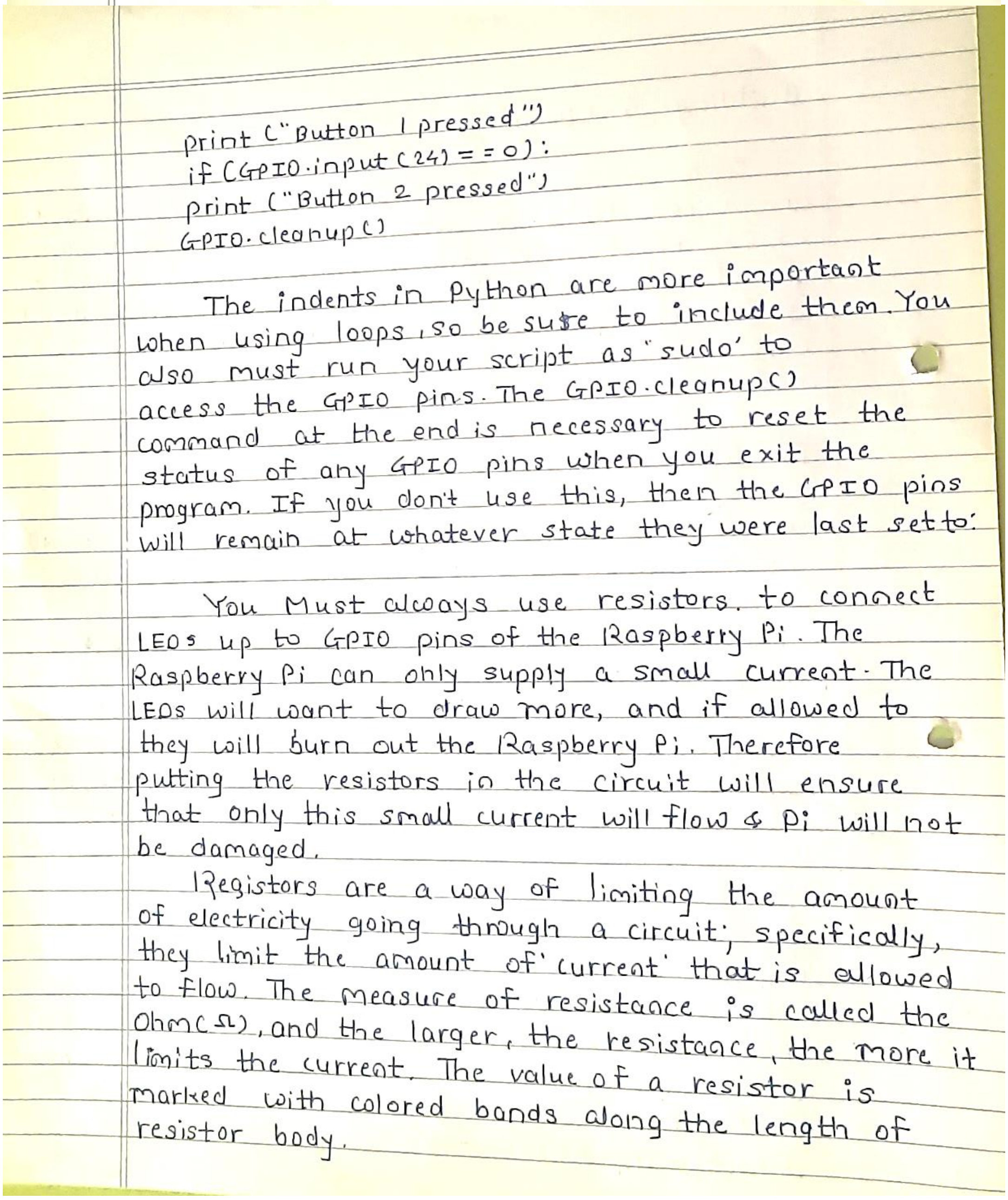
Pin 23 has a pull-down resistor, so this button is attached to the 3.3V pin

Pin 24 has a pull-up resistor, so this button is attached to the GND pin.



The Code sofar looks like this:

```
import RPi.GPIO as GPIO
GPIO.setmode (GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull-up-down=GPIO.PUD-DOW
GPIO.setup(24, GPIO.IN, pull-up-down=GPIO.POD-UP)
 while True :
 if (GPIO.input(23) ==1):
```

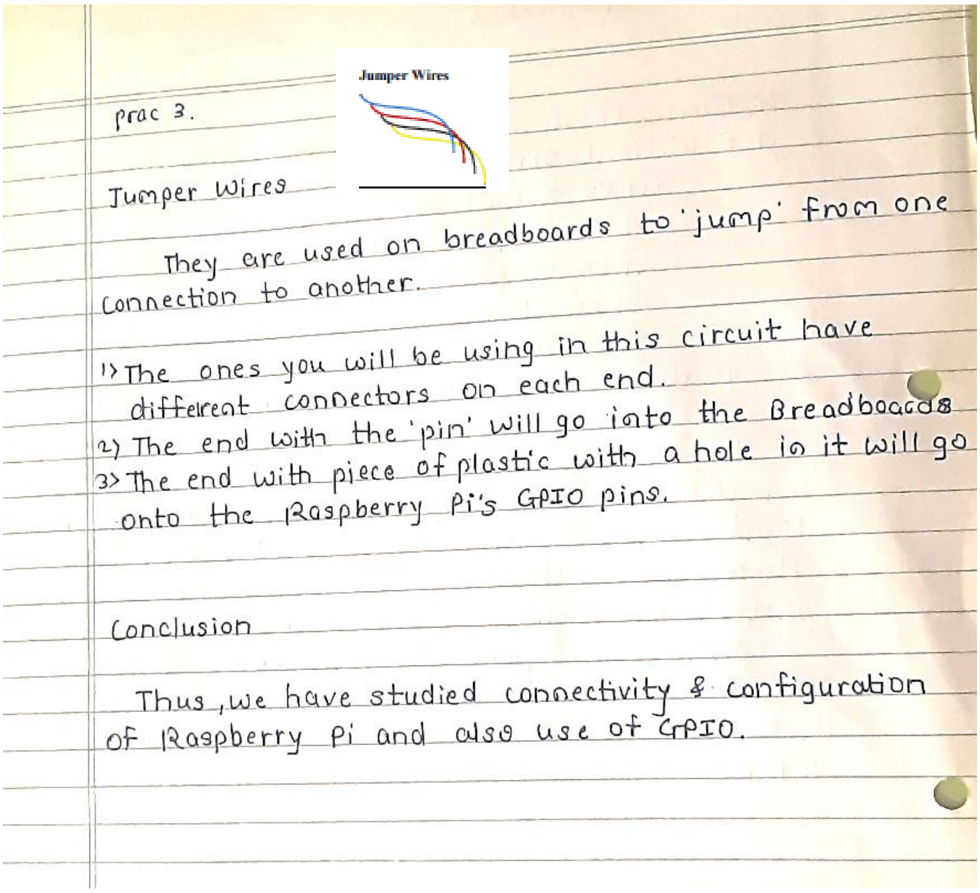


```
  print ("Button 1 pressed")
  if (GPIO.input(24)==0):
  print ("Button 2 pressed")
  GPIO.cleanup()
```

The indents in Python are more important when using loops, so be sure to include them. You also must run your script as 'sudo' to access the GPIO pins. The GPIO.cleanup() command at the end is necessary to reset the status of any GPIO pins when you exit the program. If you don't use this, then the GPIO pins will remain at whatever state they were last set to.

You Must always use resistors to connect LEDs up to GPIO pins of the Raspberry Pi. The Raspberry Pi can only supply a small current. The LEDs will want to draw more, and if allowed to they will burn out the Raspberry Pi. Therefore putting the resistors in the circuit will ensure that only this small current will flow & Pi will not be damaged.

Registors are a way of limiting the amount of electricity going through a circuit; specifically, they limit the amount of 'current' that is allowed to flow. The measure of resistance is called the Ohm(Ω), and the larger, the resistance, the more it limits the current. The value of a resistor is marked with colored bands along the length of resistor body.



Jumper Wires

prac 3.

Jumper Wires

They are used on breadboards to 'jump' from one connection to another.

1> The ones you will be using in this circuit have different connectors on each end.
2) The end with the 'pin' will go into the Breadboards
3> The end with piece of plastic with a hole in it will go onto the Raspberry Pi's GPIO pins.

Conclusion

Thus, we have studied connectivity & configuration of Raspberry Pi and also use of GPIO.