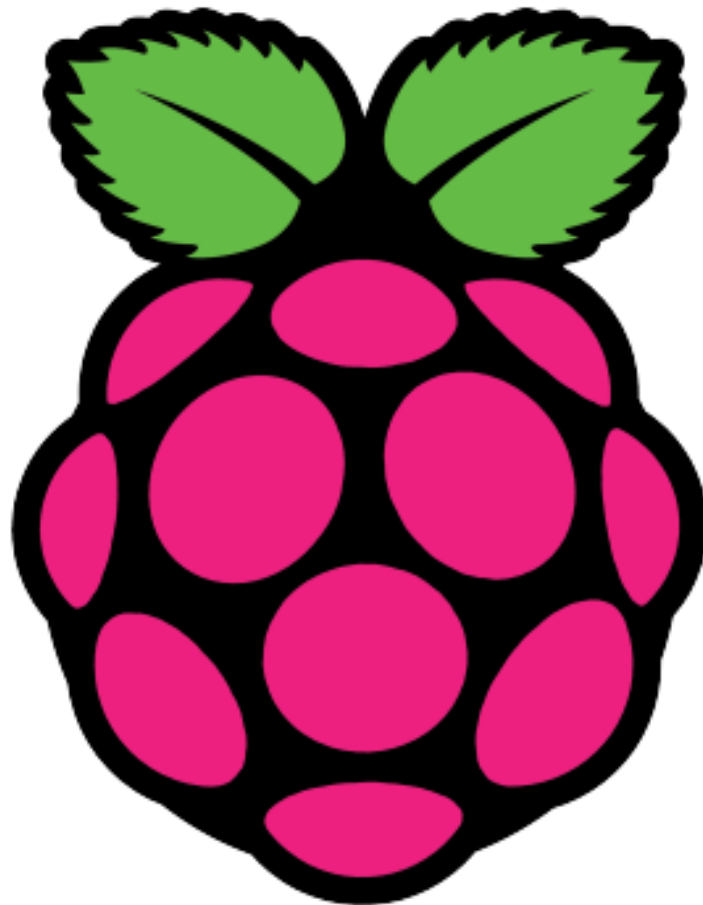


“Raspberry pi course”

ENG: AHMED MUBARAK

01020451375



SESSION NO.“5”

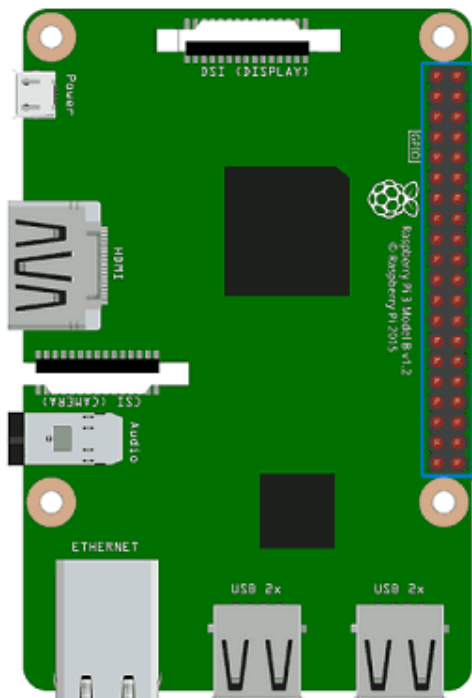
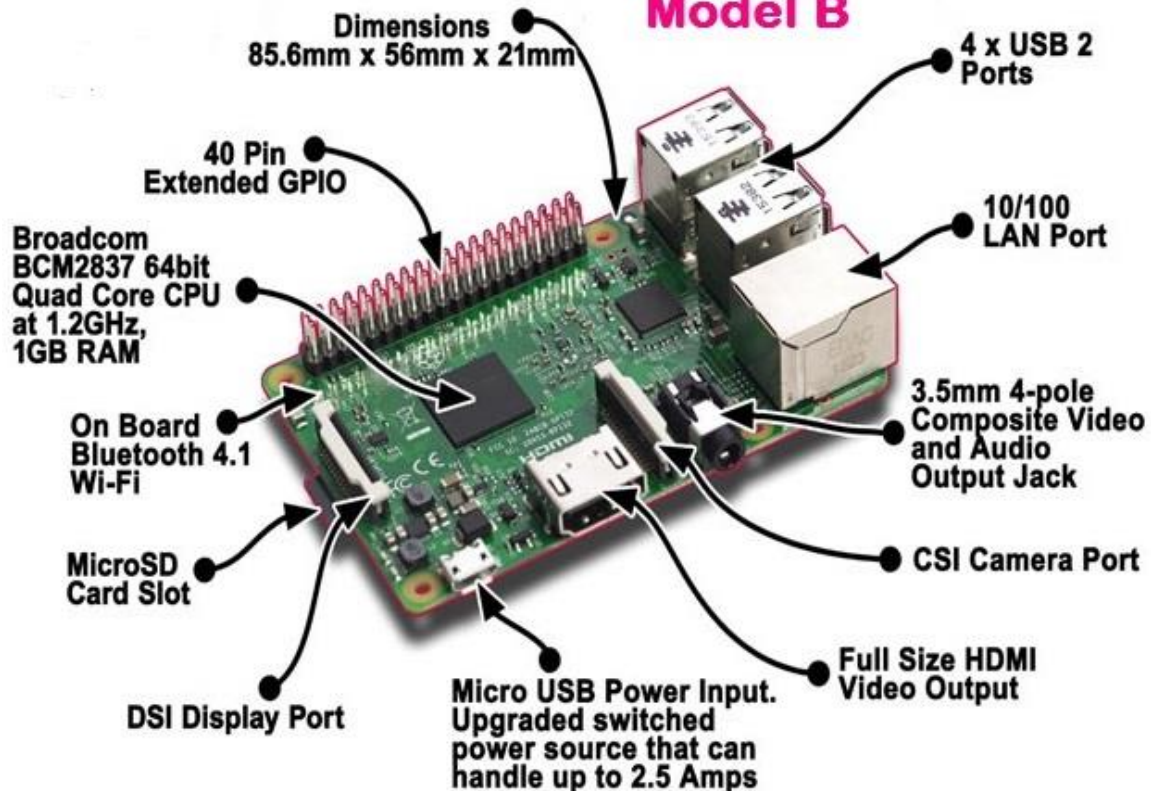
- INTRO TO RASPBERRY PI

ENG.AHMED MUBARAK

01020451375

BOARD EXPLANATION

Raspberry Pi 3 Model B



3.3V	1	2	5V
GPIO2 (SDA1)	3	4	5V
GPIO3 (SCL1)	5	6	GND
GPIO4 (GPIO_GCLK)	7	8	GPIO14 (UART_TXD0)
GND	9	10	GPIO15 (UART_RXD0)
GPIO17 (GPIO_GEN0)	11	12	GPIO18 (GPIO_GEN1)
GPIO27 (GPIO_GEN2)	13	14	GND
GPIO22 (GPIO_GEN3)	15	16	GPIO23 (GPIO_GEN4)
3.3V	17	18	GPIO24 (GPIO_GEN5)
GPIO10 (SPI0_MOSI)	19	20	GND
GPIO9 (SPI0_MISO)	21	22	GPIO25 (GPIO_GEN6)
GPIO11 (SPI0_CLK)	23	24	GPIO8 (SPI_CE0_N)
GND	25	26	GPIO7 (SPI_CE1_N)
ID_SD (I2C EEPROM)	27	28	ID_SC (I2C EEPROM)
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

COMPARE BETWEEN RASPBERRY PI AND ARDUINO

Spec	Arduino Uno	Raspberry Pi 3 B
CPU Type	8-bit Microcontroller	64-bit Microprocessor
Operating System	None	Some flavor of Linux
Storage	32 kB flash	Depends on size of SD card
Memory	2 kB	1 GB RAM
Speed	16 MHz	1.2 GHz
GPU	None	Built in
Networking	None	Ethernet, Wi-Fi, Bluetooth
Price	\$20-\$22	\$35
USB ports	1	4
Power consumption	Can be < 0.25 W	Several watts

ID_SD and ID_SC PINS:

These pins are reserved for ID EEPROM.

At boot time this I2C interface will be interrogated to look for an EEPROM that identifies the attached board and allows automatic setup of the GPIOs (and optionally, Linux drivers).

DO NOT USE these pins for anything other than attaching an I2C ID EEPROM. Leave unconnected if ID EEPROM not required.

INSTALL RASPBIAN ON RASPBERRY PI

- YOU MUST BUY A MEMORY SD CARD (IT MUST BE CLASS 10)
- CONNECT SD CARD TO YOUR PC USING CARD READER
- GO TO : <https://www.sdcard.org/downloads/formatter/>
- DOWNLOAD : **SD Memory Card Formatter**
- FORMAT YOUR SD CARD USING **SD Memory Card Formatter**
- GO TO : <https://www.raspberrypi.org/> TO DOWNLOAD RASPBIAN OS
- DOWNLOAD : **Raspberry Pi OS with desktop and recommended software**
- GO TO : <https://www.balena.io/etcher/>
- DOWNLOAD : [balenaEtcher](https://www.balena.io/etcher/) AND OPEN IT
- CHOOSE THE PACKAGE THAT YOU DOWNLOAD AND YOUR SD CARD AND CLICK FLASH .. WAIT TO FINISH FLASHING
- EJECT YOUR SD CARD AND PUT IT IN YOUR RASPBERRY PI
- CONNECT THE RASPBERRY TO DISPLAY USING HDMI
- USE USB PORTS TO CONNECT MOUSE AND KEYBOARD
- POWER YOUR RASPBERRY PI USING ADAPTOR : 5V .. 2.5A
- WAIT RASPBERRY TO BOOT ..
- AFTER FINISH BOOTING CHOOSE YOUR REGION AND LANGUAGE AND SET A PASSWORD TO IT AND FINISH CONFIGURATION
- CONGRATULATION YOU HAVE A RASPBIAN ON YOUR RASPBERRY ...

WHAT YOU MUST TO DO AFTER INSTALLING RASPBIAN ?

- CLICK ON RASPBERRY LOGO THEN CHOOSE PREFERENCES THEN CHOOSE RASPBERRY PI CONFIGURATION
- CHOOCE INTERFACES THEN ENABLE (SSH AND VNC)
- REBOOT YOUR RASPBERRY PY AND OPEN YOUR COMPUTER
- OPEN TERMINAL AND TYPE : `sudo nano /etc/dhcpd.conf`
- ADD THIS TEXT TO MAKE A STATIC IP:

```
interface eth0

static ip_address=192.168.0.10/24

static routers=192.168.0.1

static domain_name_servers=192.168.0.1
```

```
interface wlan0

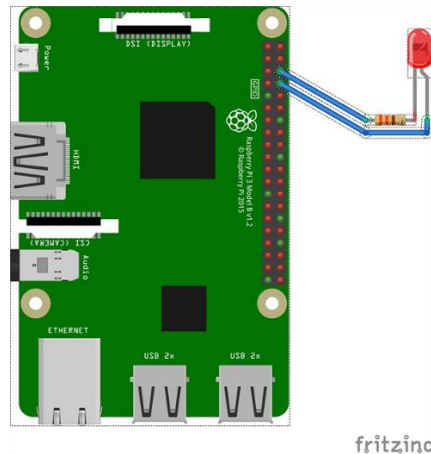
static ip_address=192.168.0.200/24

static routers=192.168.0.1

static domain_name_servers=192.168.0.1
```

- PRESS (CTRL + X) THEN PRESS (Y) THEN PRESS ENTER
- GO TO : <https://www.putty.org/> TO DOWNLOAD PUTTY AND INSTALL IT
- GO TO :
<https://www.realvnc.com/en/connect/download/viewer/>
TO DOWNLOAD VNC VIEWER
- NOW YOU CAN OPEN YOUR RASPBERRY PI ON YOUR COMPUTER BY IP ADDRESS ONLY FROM PUTTY OR VNC VIEWER .

CONTROL LED USING RASPBERRY PI



- OPEN TERMINAL
- TYPE : `nano led.py` TO CREATE A NEW FILE THAT YOU CAN TYPE YOUR PYTHON CODE ON IT .
- NOW IT'S PROGRAMMING TIME .. THIS IS AN EXAMPLE CODE TO CONTROL LED

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
print ("LED ON ")
GPIO.output(18,GPIO.HIGH)
time.sleep(1)
print("LED OFF")
GPIO.output(18,GPIO.LOW)
time.sleep(1)
```

- THE BACK CODE IS JUST BLINK LED ONE TIME

- IF YOU WANT TO LOOP BLINKING USE THE FOLLOWING CODE :

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW)
while True:
    print("HIGH")
    GPIO.output(8, GPIO.HIGH)
    sleep(1)
    print("LOW")
    GPIO.output(8, GPIO.LOW)
    sleep(1)
```

- TO RUN THIS CODE :
 1. SAVE THE FILE BY CLICKING (CTRL+X) THEN (Y) THEN (ENTER)
 2. USE THIS INSTRUCTION TO RUN IT :

```
sudo python led.py
```

With my best wishes:

ENG : AHMED MUBARAK
