IoT Physical Devices and Endpoints - RaspberryPi

- **Introduction to RaspberryPi**
- **About the RaspberryPi Board:**

  Hardware Layout

  Operating Systems on RaspberryPi

  Configuring RaspberryPi

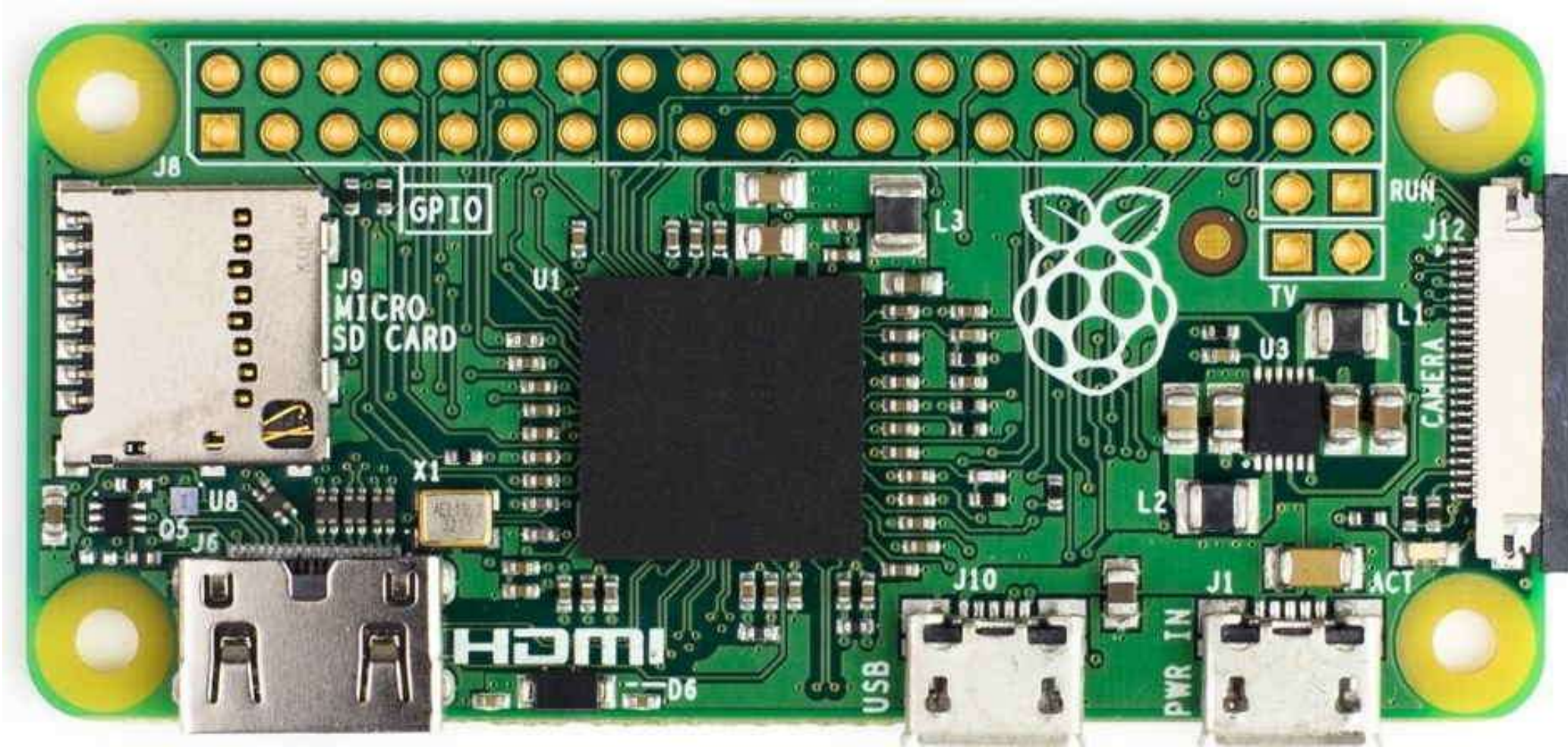
- **Programming RaspberryPi with Python.**

	Arduino	Raspberry Pi
Architecture	A **microcontroller** platform. It's designed to run simple programs and perform tasks like reading sensors, controlling LEDs, motors, or other actuators. Common models like the **Arduino Uno** use microcontrollers such as the ATmega328.	A **single-board computer**. It runs a full operating system (often **Raspberry Pi OS**, based on Linux) and can handle complex tasks like running multiple programs, networking, or even acting as a lightweight server.
Operating System	**Doesn't have an operating system**. It runs a single program at a time, loaded directly onto its memory	Has a **fully functional OS**, allowing for multitasking, web browsing, file management, and more.
Power and Performance	Limited in terms of processing power (16-32 MHz) and memory (usually 2-8 KB RAM). It's designed for low-power, real-time tasks.	Far more powerful, with multi-core processors, higher RAM (512MB to 8GB), USB ports, HDMI output, and the ability to run a desktop environment.
Connectivity and Interfaces	Typically has fewer interfaces. It comes with digital and analog input/output pins to interface with sensors, motors, or other electronics. It lacks native support for peripherals like monitors or keyboards.	Has a wide range of connectivity options like USB ports, HDMI, Ethernet, Wi-Fi, and Bluetooth. It also supports GPIO pins for hardware interfacing like Arduino, but with more capabilities.

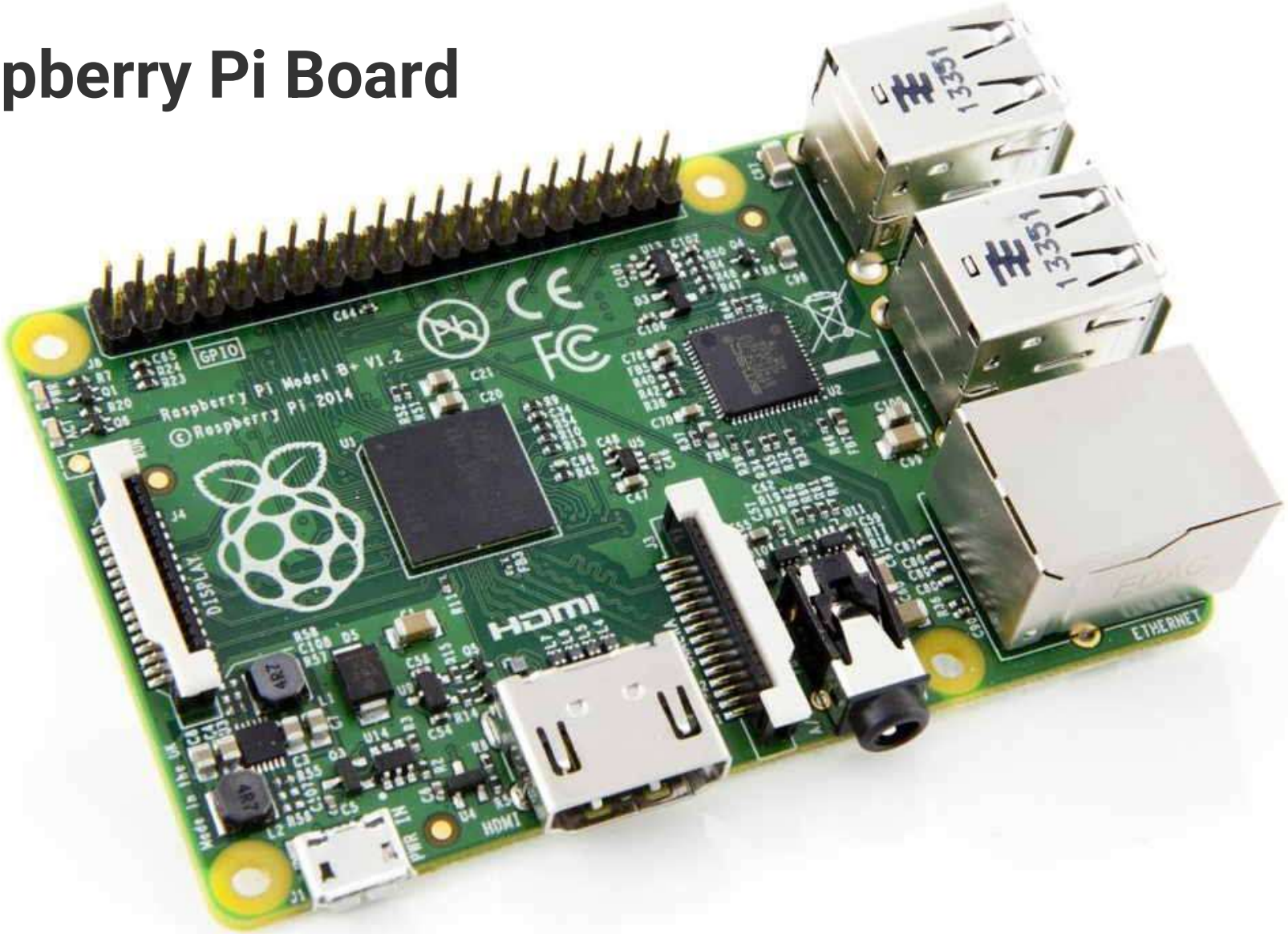
Programming	Primarily programmed using the **Arduino IDE** with **C/C++**. It's simpler and more focused on interacting with hardware.	You can program it using many languages like **Python, C, Java**, etc. It supports a wide variety of software and development environments since it's a full computer.
Use Case	Ideal for simpler projects like robotics, sensor monitoring, motor control, or other embedded systems where real-time control and low power are essential.	Suited for more complex applications such as programming, running servers, media centers, or even AI applications. It's often used for educational purposes, prototyping, or projects that require significant processing power.

- **Arduino** is for simple, low-power hardware control tasks.
- **Raspberry Pi** is for more complex, high-power computing tasks.

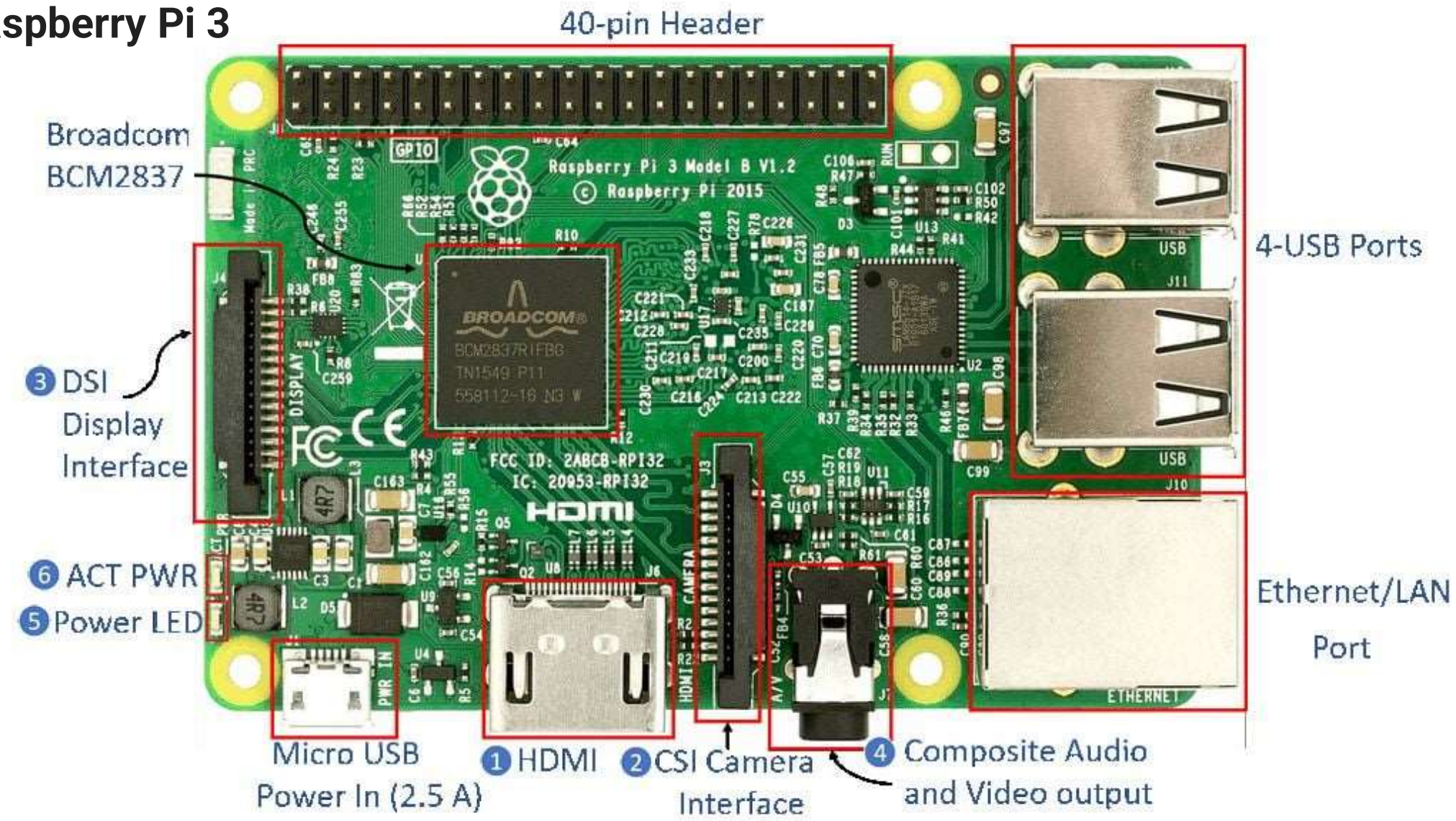
Raspberry Pi zero Board

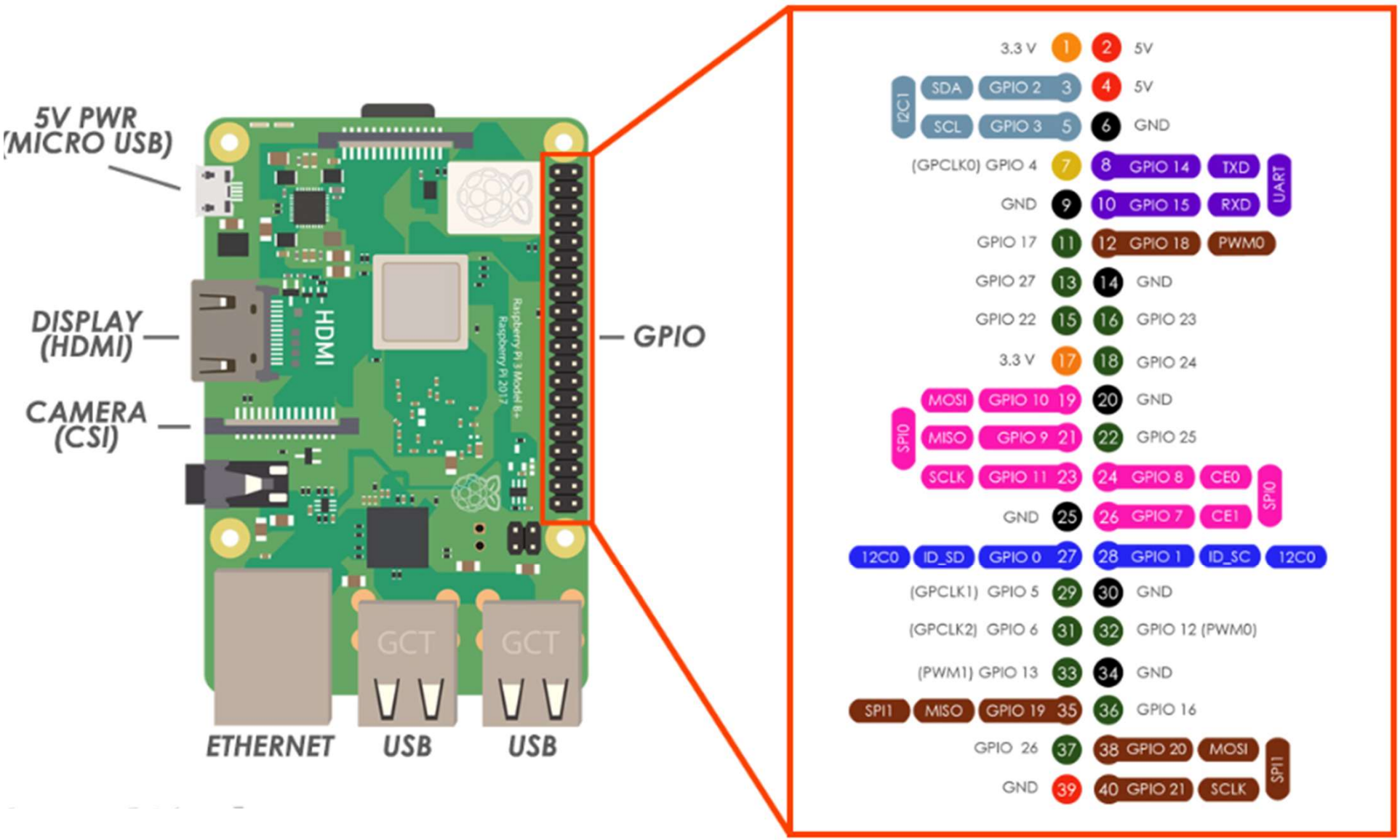


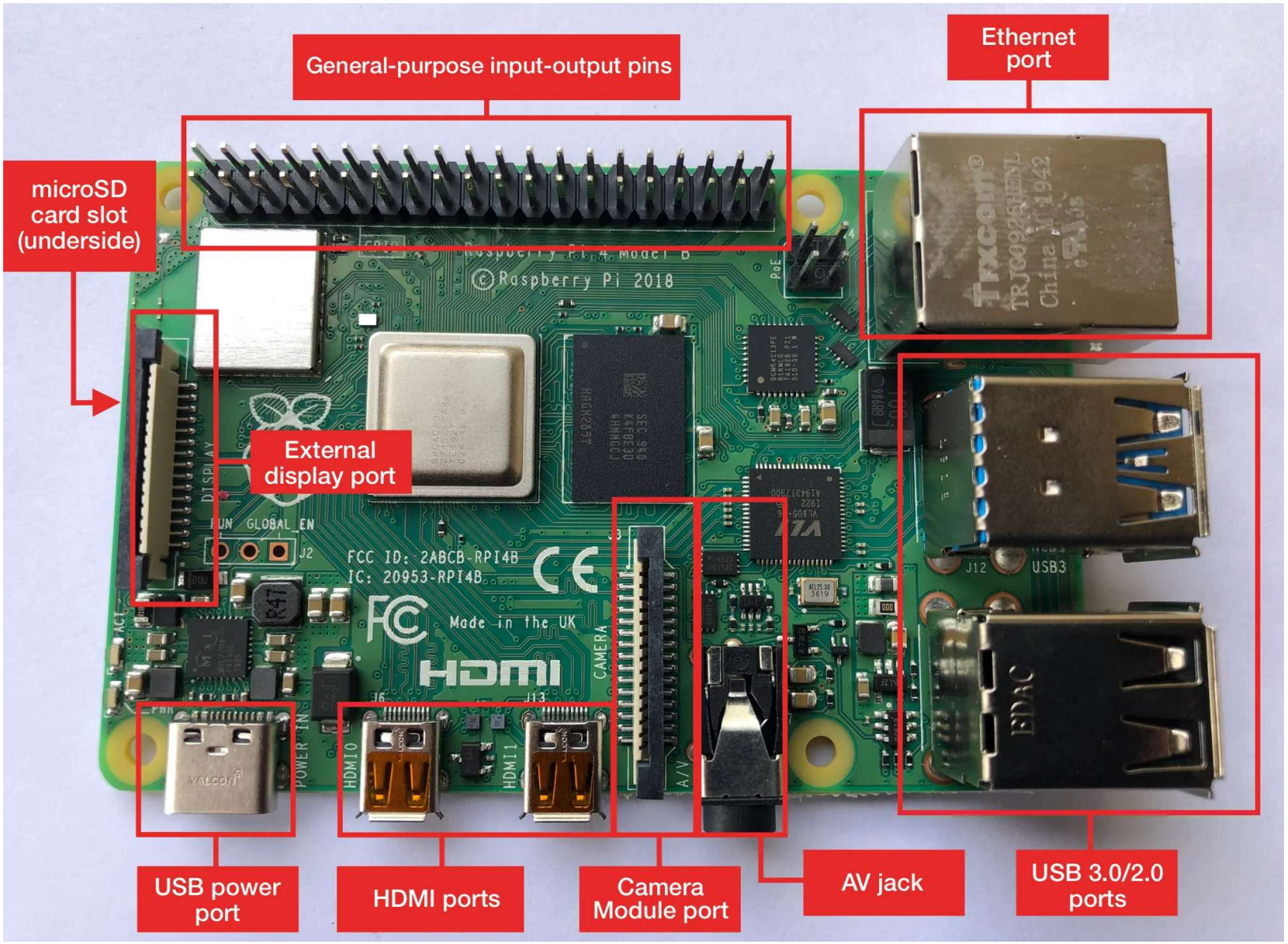
Raspberry Pi Board



Raspberry Pi 3




5V PWR (MICRO USB)
DISPLAY (HDMI)
CAMERA (CSI)
GPIO
ETHERNET
USB
USB
HDMI
GCT
GCT
Raspberry Pi 3 Model B+
Raspberry Pi 2017
3.3 V 1 2 5V
I2C1 SDA GPIO 2 3 4 5V
I2C1 SCL GPIO 3 5 6 GND
(GPCLK0) GPIO 4 7 8 GPIO 14 TXD UART
GND 9 10 GPIO 15 RXD UART
GPIO 17 11 12 GPIO 18 PWM0
GPIO 27 13 14 GND
GPIO 22 15 16 GPIO 23
3.3 V 17 18 GPIO 24
SPI0 MOSI GPIO 10 19 20 GND
SPI0 MISO GPIO 9 21 22 GPIO 25
SPI0 SCLK GPIO 11 23 24 GPIO 8 CE0 SPI0
GND 25 26 GPIO 7 CE1 SPI0
I2C0 ID_SD GPIO 0 27 28 GPIO 1 ID_SC I2C0
(GPCLK1) GPIO 5 29 30 GND
(GPCLK2) GPIO 6 31 32 GPIO 12 (PWM0)
(PWM1) GPIO 13 33 34 GND
SPI1 MISO GPIO 19 35 36 GPIO 16
GPIO 26 37 38 GPIO 20 MOSI SPI1
GND 39 40 GPIO 21 SCLK SPI1



3V3	5V
GPIO2	5V
GPIO3	GND
GPIO4	GPIO14
GND	GPIO15
GPIO17	GPIO18
GPIO27	GND
GPIO22	GPIO23
3V3	GPIO24
GPIO10	GND
GPIO9	GPIO25
GPIO11	GPIO8
GND	GPIO7
ADV	ADV
GPIO5	GND
GPIO6	GPIO12
GPIO13	GND
GPIO19	GPIO16
GPIO26	GPIO20
GND	GPIO21

- A Raspberry Pi 3 board has **40** pins on it.
- Among these pins, we have **4** power pins on the Raspberry Pi, **2** of which are **5v** pins and another **2** are **3.3v** pins.
- The 5v power pins are connected directly to the Raspberry Pi's power input and we can use these pins to run low power applications.
- There are 8 ground pins and all of these are connected to each other
- **28** GPIO pins, labeled starting from GPIO 0 and going up to GPIO 27.
- The GPIO (General Purpose Output and Input) pins, as indicated by their full form, can be programmed to be output pins or input pins. So we can set values of output pins and we can even read values of input pins.
- The GPIO pins can be digitally programmed so that they can be turned ON or OFF.
- The output of any GPIO pin is 3.3v and can be used to control output components like an LED or a motor. These ON/OFF conditions can also be interpreted as a Boolean True/False, 1/0 or HIGH/LOW.
- These are the common types of pins on a Raspberry Pi 3 board. Some of these pins also have a dual function. For example, pin 3 or GPIO 2 also acts like an I2C pin.

About the RaspberryPi Board: Hardware Layout

1. : Processor & RAM
2. : USB Ports
3. : Ethernet Port
4. : HDMI Output
5. : Composite Video Output
6. : Audio Output
7. : GPIO Pins
8. : Display Serial Interface (DSI)
9. : Camera Serial Interface (CSI)
10. : Status LEDs
11. : SD Card Slot
12. : Power Input

Processor & RAM

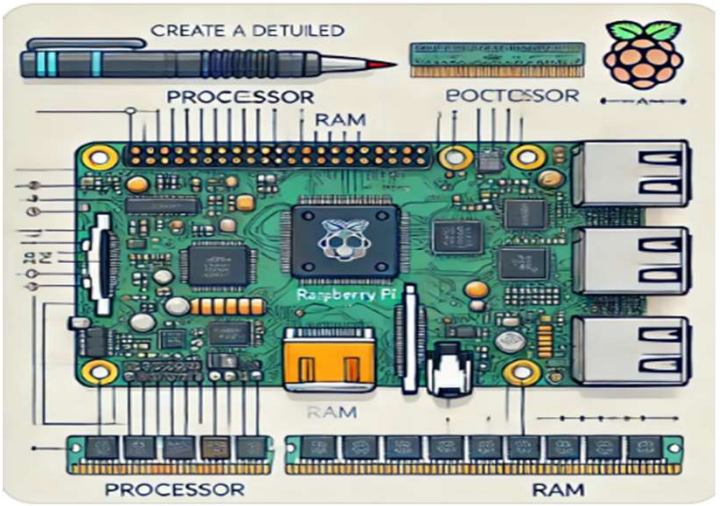
Processor:

ARM-based,

700 MHz Low Power **ARM1176JZ-F**

RAM: 512 MB SDRAM

Found in **Raspberry Pi Model B, Revision 2**



About the RaspberryPi Board: Hardware Layout

1. : Processor & RAM
2. : USB Ports
3. : Ethernet Port
4. : HDMI Output
5. : Composite Video Output
6. : Audio Output
7. : GPIO Pins
8. : Display Serial Interface (DSI)
9. : Camera Serial Interface (CSI)
10. : Status LEDs
11. : SD Card Slot
12. : Power Input

USB Ports

4 USB ports

Provides current up to **100mA**

External powered hub needed for devices that require more current

About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

Ethernet Port

RJ45 Ethernet port for wired internet connection

Supports **USB Wi-Fi adapters** for wireless internet

About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

HDMI Output

HDMI port for video and audio output

Can connect to monitors via **HDMI cable**

HDMI to DVI adapter can be used for DVI-only monitors

About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

Composite Video Output

RCA jack supports **PAL and NTSC video** output

Used to connect to old TVs with RCA input

About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

Audio Output

3.5mm audio jack for old TVs

Lower quality compared to HDMI audio output

About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

GPIO Pins

General-purpose input/output pins (GPIO)

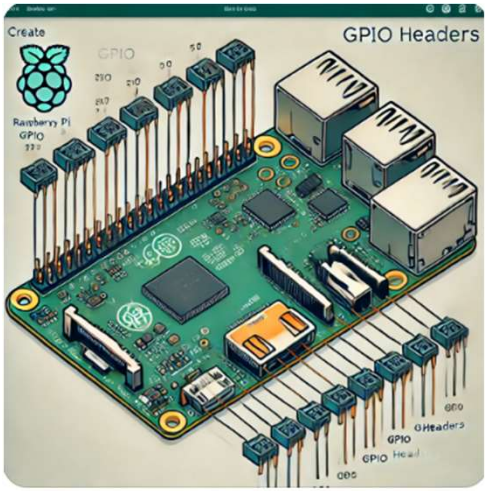
Types of Pins:

True GPIO

I2C interface

SPI interface

Serial Rx and Tx



About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

Display Serial Interface (DSI)

Connects to an **LCD panel**

About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

Camera Serial Interface (CSI)

Connects to a **camera module**

About the RaspberryPi Board: Hardware Layout

1. : Processor & RAM
2. : USB Ports
3. : Ethernet Port
4. : HDMI Output
5. : Composite Video Output
6. : Audio Output
7. : GPIO Pins
8. : Display Serial Interface (DSI)
9. : Camera Serial Interface (CSI)
10. : Status LEDs
11. : SD Card Slot
12. : Power Input

Status LEDs

5 Status LEDs

Indicate different system statuses

About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

SD Card Slot

No built-in OS or storage

Requires an **SD card** (at least 8GB) with a Linux image

Appendix A explains setting up **NOOBS**

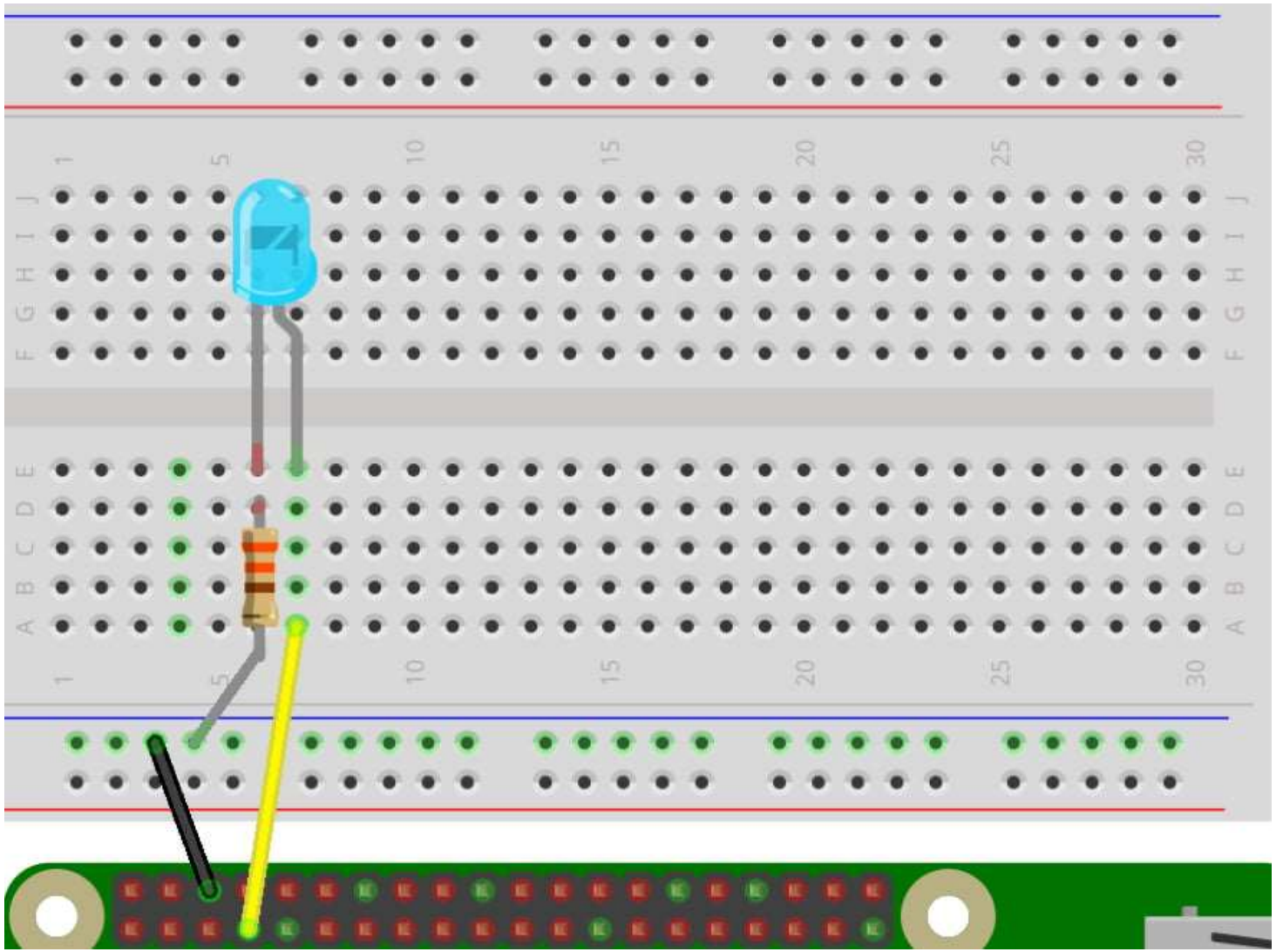
About the RaspberryPi Board: Hardware Layout

1. **: Processor & RAM**
2. **: USB Ports**
3. **: Ethernet Port**
4. **: HDMI Output**
5. **: Composite Video Output**
6. **: Audio Output**
7. **: GPIO Pins**
8. **: Display Serial Interface (DSI)**
9. **: Camera Serial Interface (CSI)**
10. **: Status LEDs**
11. **: SD Card Slot**
12. **: Power Input**

Power Input

Uses a **micro-USB connector** for power input

LED blink using the Raspberry Pi and Python



LED blink using the Raspberry Pi and Python

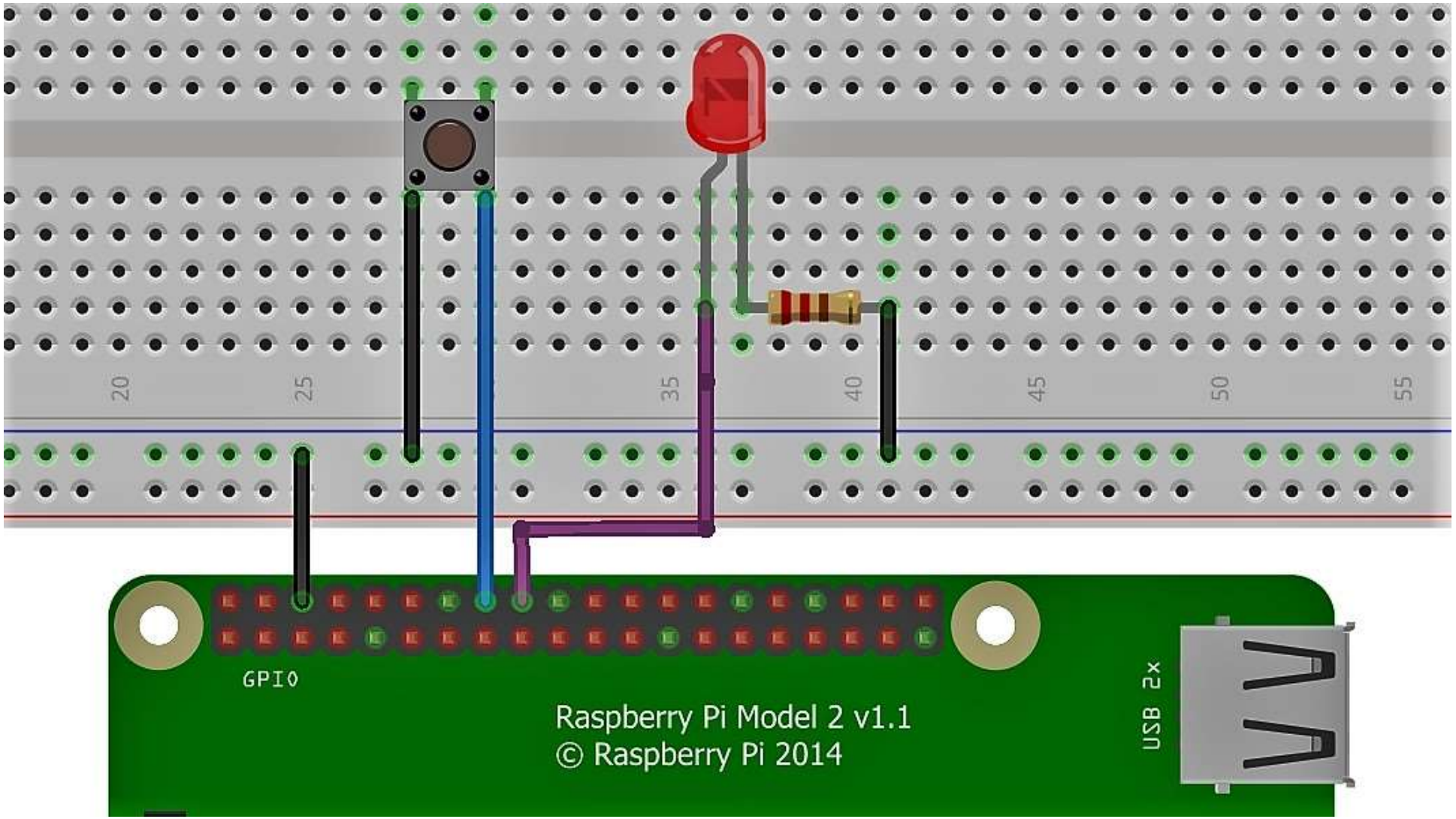
1. Initialize the GPIO ports
2. Turn the LED on and off in 1 second intervals

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

GPIO.setmode(GPIO.BOARD)                         # Use physical pin numbering
GPIO.setup(18, GPIO.OUT, initial=GPIO.LOW)       # Set pin 18 to be an output pin and set initial value to low (off)

while True:                                      # Run forever
        GPIO.output(18, GPIO.HIGH)               # Turn on
        sleep(1)                                 # Sleep for 1 second
        GPIO.output(18, GPIO.LOW)                # Turn off
        sleep(1)                                 # Sleep for 1 second
```

LED blink with Push Button using the Raspberry Pi and Python



```
1   import RPi.GPIO as GPIO
2   import time
3
4   GPIO.setmode(GPIO.BCM)
5
6   GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)#Button to GPIO23
7   GPIO.setup(24, GPIO.OUT)  #LED to GPIO24
8
9   try:
10      while True:
11          button_state = GPIO.input(23)
12          if button_state == False:
13              GPIO.output(24, True)
14              print('Button Pressed...')
15              time.sleep(0.2)
16          else:
17              GPIO.output(24, False)
18  except:
19      GPIO.cleanup()
```