

Coding/programming is fun with tiny computers (credit-card size) - the “Raspberry Pi” computers

CS150 & CS250 Problem Solving I & II
Computer Science Dept
Old Dominion University

by

Dr. Ayman Elmesalami

Dr. Soad Ibrahim

Hussam Hallah, PhD student

What is Raspberry Pi?

- A low cost, \$35, credit-card sized computer
- Originally designed for educational purposes
- Early concept started in 2006. Officially launched in 2012
- Used in IoT, Internet of Things, a network of **Internet** connected objects (things) able to collect and exchange data.
- Used in embedded systems, computer system with a dedicated function within a larger mechanical or electrical system.
- Embedded systems Interface to surrounding environment through sensors and actuators.

Input (Sensor) → **Raspberry Pi** → Output (indicator or actuator)

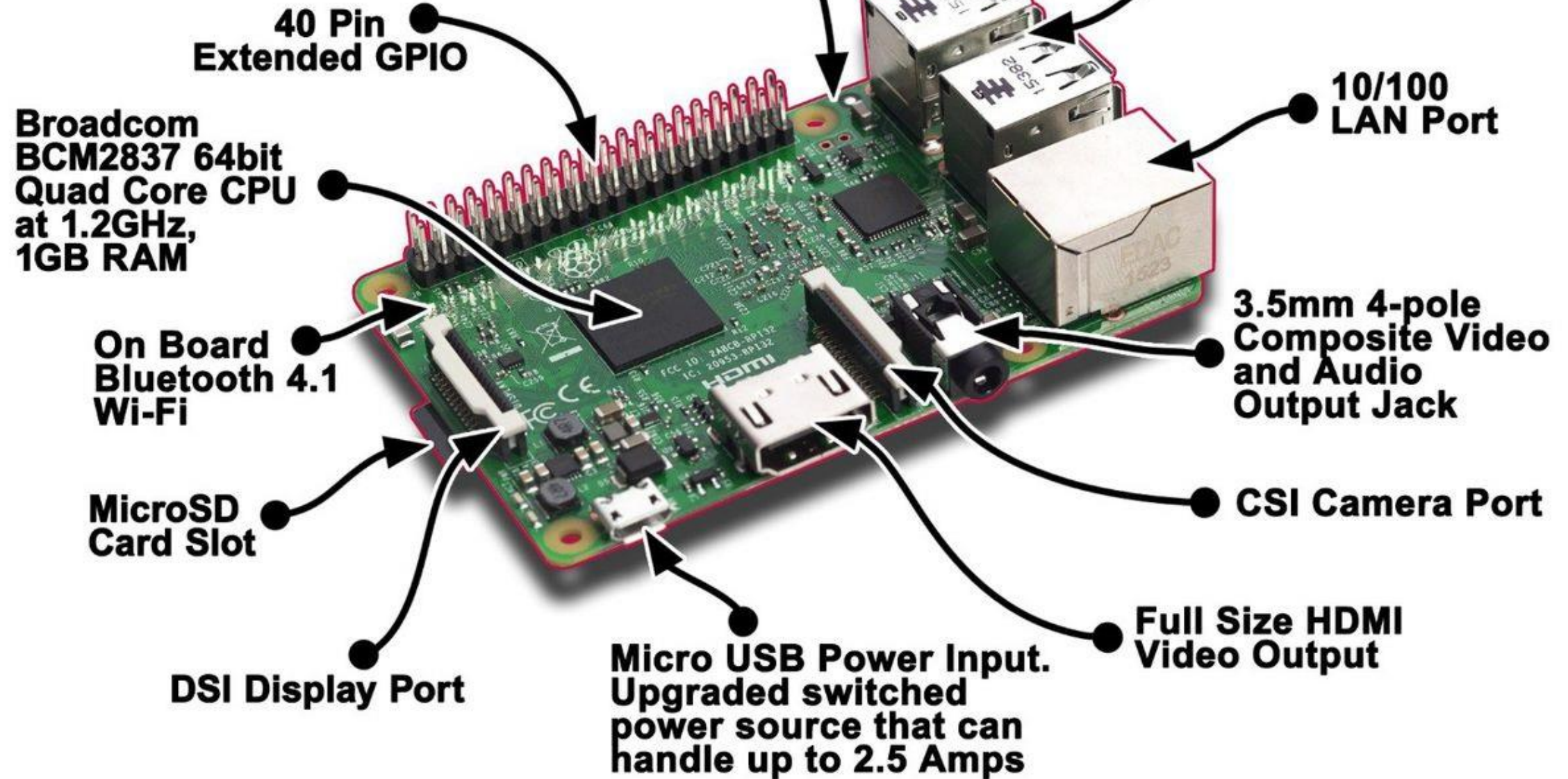
- Examples: Motion detectors, home automation, security systems, smart cities, leak detectors, etc.

Raspberry Pi 3 Model B: Hardware Specifications

- SoC: Broadcom BCM2837
- CPU: 1.2 GHz quad-core ARM Cortex A53
- GPU: Broadcom VideoCore IV @ 400 MHz
- Memory: 1 GB LPDDR2-900 SDRAM
- Storage: MicroSD (not included with Raspberry Pi)
- Ethernet: 10/100 MBPS
- Wireless LAN: 2.4 GHz 802.11 N
- Bluetooth: 4.1
- Ports: HDMI, 4x USB 2.0, 3.5mm analogue audio-video jack, micro-USB power input, Camera Serial Interface (CSI), Display Serial Interface (DSI).
- **GPIO: 40 pins** (General Purpose Input/Output)

Raspberry Pi 3 Model B

Dimensions
85.6mm x 56mm x 21mm



What you need

- Raspberry Pi: \$35
- Power adapter: \$8
- MicroSD card: 8GB or above recommended. \$20
- Case: \$8 (optional)
- HDMI cable: \$5
- USB keyboard: \$15
- USB mouse: \$12
- Monitor: \$110
- Total: About \$200

Raspberry Pi 3 Model B kit

Raspberry Pi 3 Model B



Case



Power Adapter



MicroSD Card



HDMI Cable



Connection Checklist

- Insert SD Card
- Connect HDMI cable to the monitor
- Connect Keyboard & Mouse
- Connect Network
- Connect Headphones
- Connect Power...wait

Operating Systems

- Multiple options with pros and cons each

- **Most popular operating systems:**

1. Raspbian OS (Official):

Debian based Linux distribution, free & open-source, lightweight, designed specifically for Raspberry Pi, and comes with pre-installed software like Chromium web browser, LibreOffice, ...etc.

2. Windows 10 IoT Core:

Free for hobbyist developers

- It is possible to buy an microSD card with a pre-installed operating system.

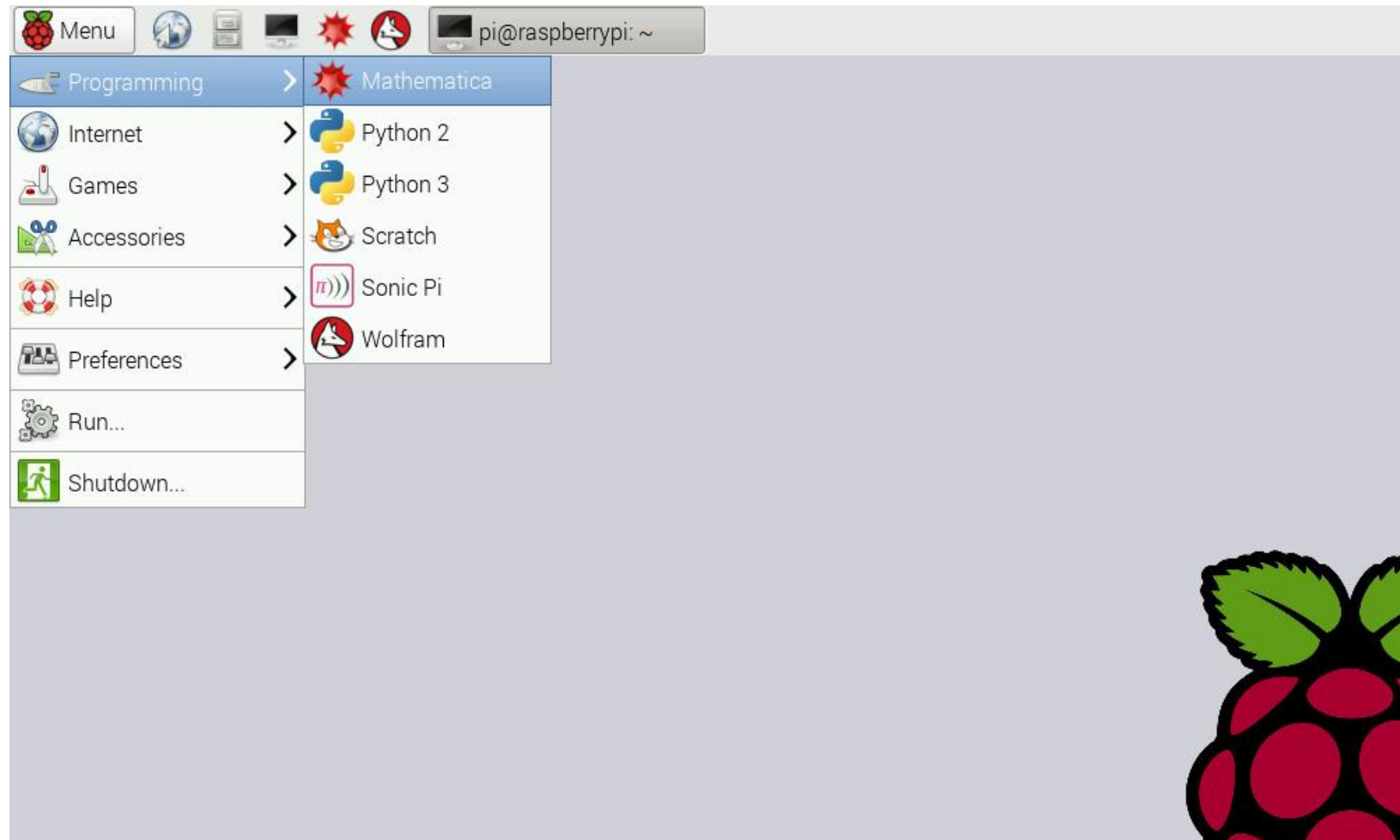
Boot up & Login

- If SD card is loaded with an operating system, Raspbian, the system will load upon booting up.

- Default login:

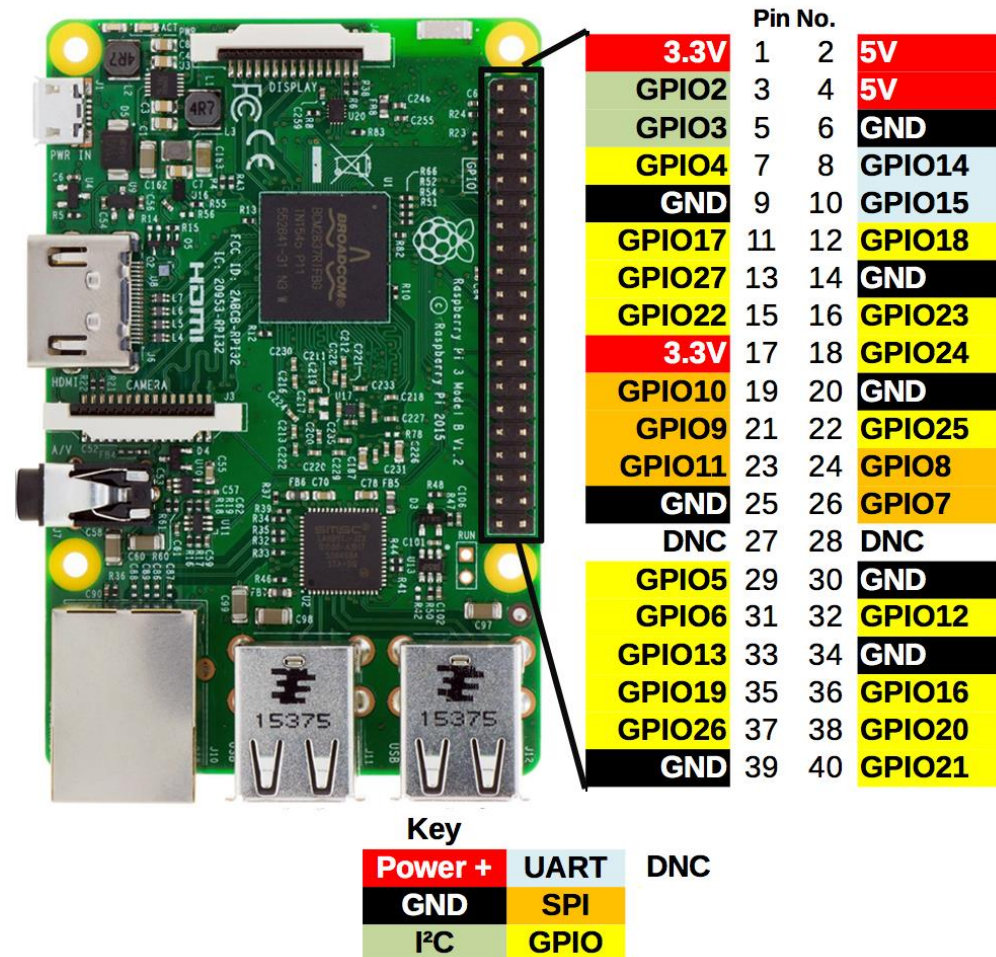
Username: pi

Password: raspberry



GPIO (General Purpose Input/Output)

- The most powerful feature of the Raspberry Pi
- 40 pins can be designated (in software) as input or output for a wide range of purposes
- The numbering of the GPIO pins is **NOT** in numerical order.
- It is possible to control GPIO pins using a number of programming languages like Python and C++
- Outputs: A GPIO pin used as an output pin can be set to high (3.3 volts) or low (0 volts)
- Inputs: A GPIO pin used as an input pin can be read as high (3.3 volts) or low (0 volts)



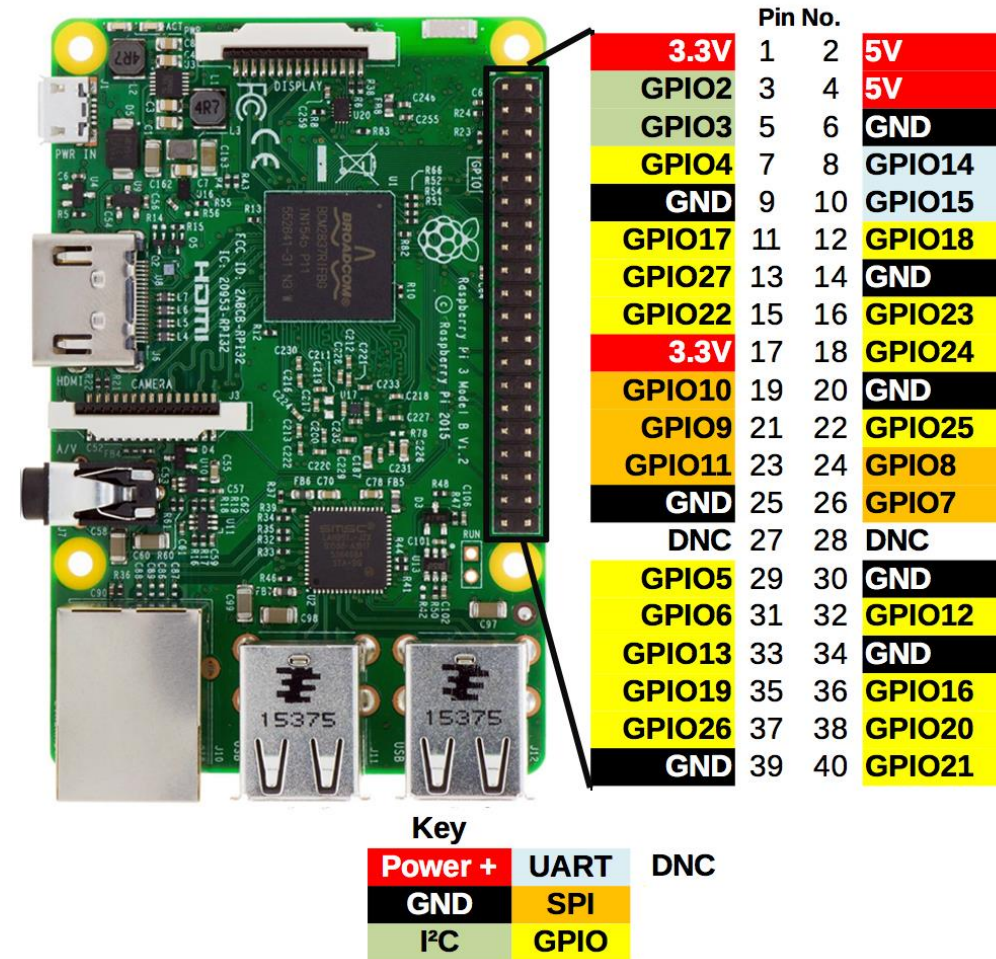
\$ pinout

- Use the **pinout** command-line tool to output a board diagram and details of the GPIO pins available on the Pi.

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ pinout  
-----  
00000000000000000000 J8  
10000000000000000000  
-----  
Pi Model 3B V1.2  
-----  
[D] [SoC] [C] [S] [I] [A] [V]  
[S] [I] [S] [I] [S] [I]  
[I] [I] [I] [I] [I] [I]  
-----  
pwr HDMI Net  
-----  
Revision : a02082  
SoC : BCM2837  
RAM : 1024Mb  
Storage : MicroSD  
USB ports : 4 (excluding power)  
Ethernet ports : 1  
Wi-fi : True  
Bluetooth : True  
Camera ports (CSI) : 1  
Display ports (DSI): 1  
  
J8:  
3V3 (1) (2) 5V  
GPIO2 (3) (4) 5V  
GPIO3 (5) (6) GND  
GPIO4 (7) (8) GPIO14  
GND (9) (10) GPIO15  
GPIO17 (11) (12) GPIO18  
GPIO27 (13) (14) GND  
GPIO22 (15) (16) GPIO23  
3V3 (17) (18) GPIO24  
GPIO10 (19) (20) GND  
GPIO9 (21) (22) GPIO25  
GPIO11 (23) (24) GPIO8  
GND (25) (26) GPIO7  
GPIO0 (27) (28) GPIO1  
GPIO5 (29) (30) GND  
GPIO6 (31) (32) GPIO12  
GPIO13 (33) (34) GND  
GPIO19 (35) (36) GPIO16  
GPIO26 (37) (38) GPIO20  
GND (39) (40) GPIO21  
  
For further information, please refer to https://pinout.xyz/  
pi@raspberrypi:~ $
```

GPIO pins simplified

- **GPIO:** Standard pins that can be used to turn devices on and off. For example, an LED.
- **I2C** ([Inter-Integrated Circuit](#)): Pins can be used to connect and communicate with hardware modules that support the I2C protocol.
- **SPI** ([Serial Peripheral Interface Bus](#)) pins can be used to connect and communicate to SPI devices. Same as I2C but makes use of a different protocol.
- **UART** ([Universal asynchronous receiver/transmitter](#)) is the serial pins used to communicate with other devices.
- **DNC** stands for do not connect, self-explanatory.
- The **power pins** pull power directly from the Raspberry Pi. These are the only 5 volts pins on the Pi.
- **GND:** Pins used to ground devices. It doesn't matter which pin is used as they are all connected to the same line.



Breadboards are NOT made for cutting bread

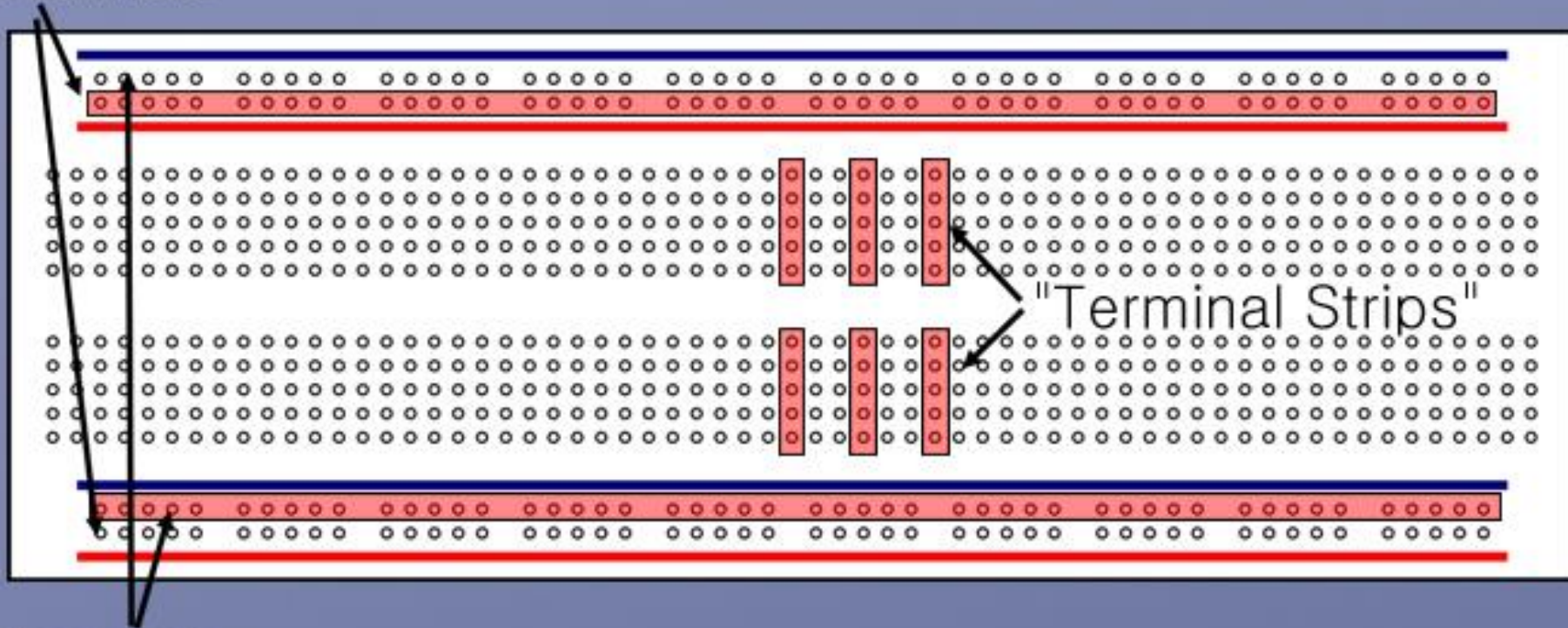
- Let's build circuits!



Breadboards are made for prototyping

- Building circuits with breadboards

Voltage Channels



Power Channels

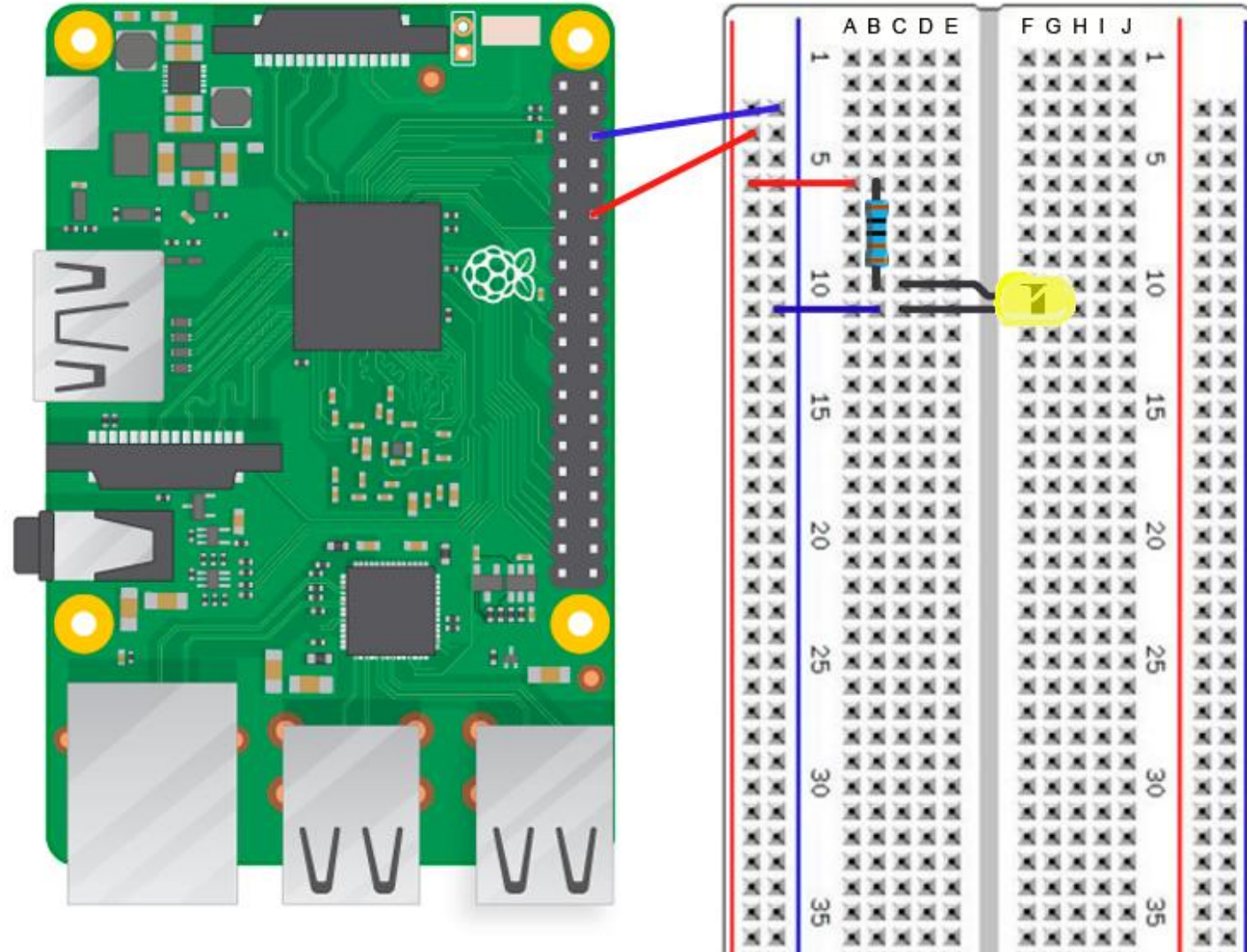
Ground Channels

Before connecting Anything!

- LEDs should have resistors, in series, to limit the current passing through them.
- Do **NOT** use 5 volts for 3.3 volts components.
- Do **NOT** connect motors directly to the GPIO pins (motors must be connected using an H-bridge circuit)

Connecting LED to Raspberry Pi GPIO

- Using numerical order, physical numbering, for GPIO pins.
- Pin 6 (GND) is connected to ground rail.
- Pin 12 (GPIO18) is connected to the voltage rail.
- A resistor is connecting the voltage channel to the positive end on the LED (long leg).
- The negative end (short leg) of the LED is connected to the ground rail.



Choosing the right resistor

- The formula to calculate the correct resistor to use is given by Ohms law:

$$R = \frac{V_s - V_f}{I}$$

Where:

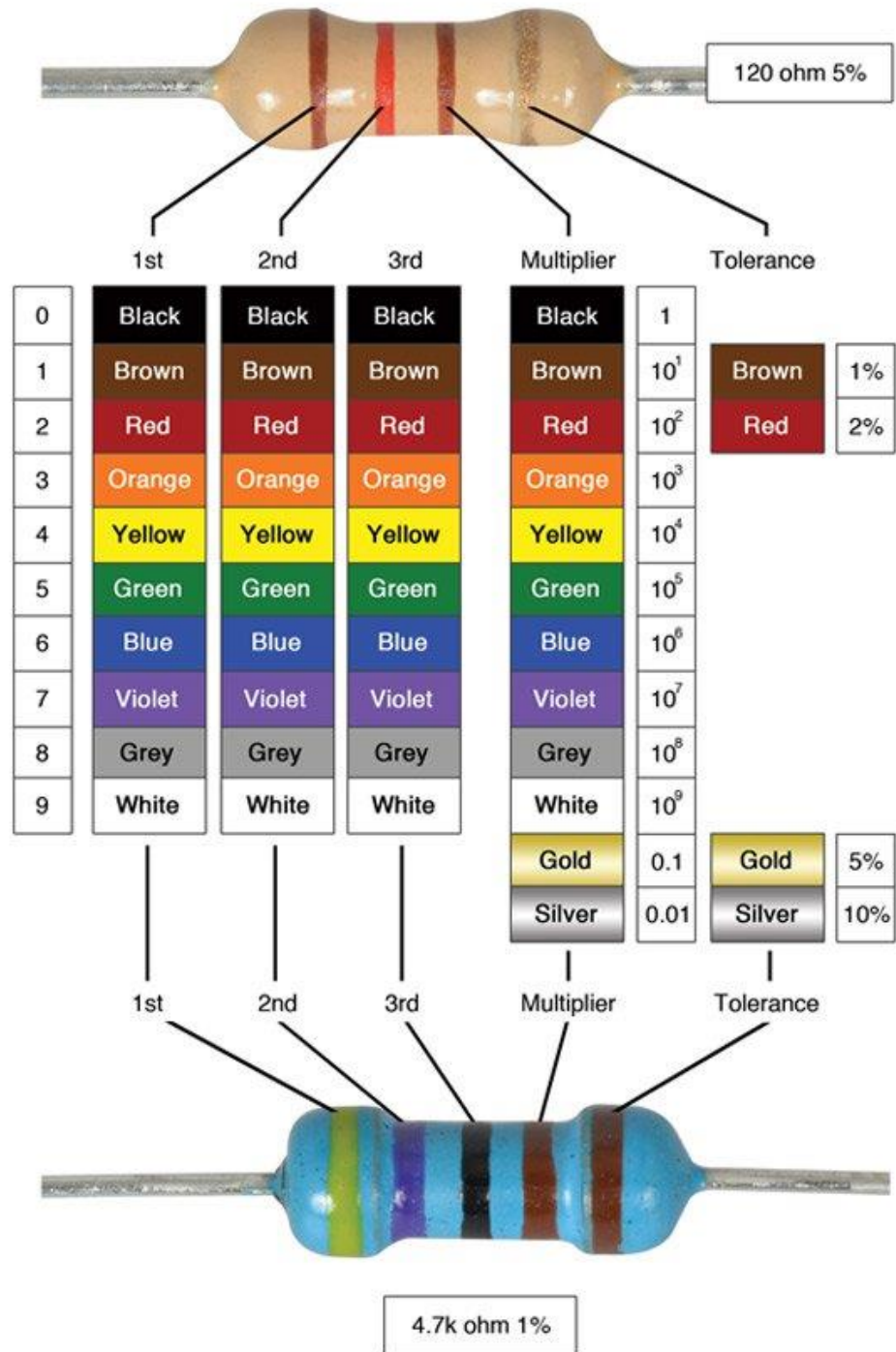
V_s is supply voltage (3.3 Volts for Raspberry Pi)

V_f is the LED forward voltage across the LED

I is the desired current in Amps

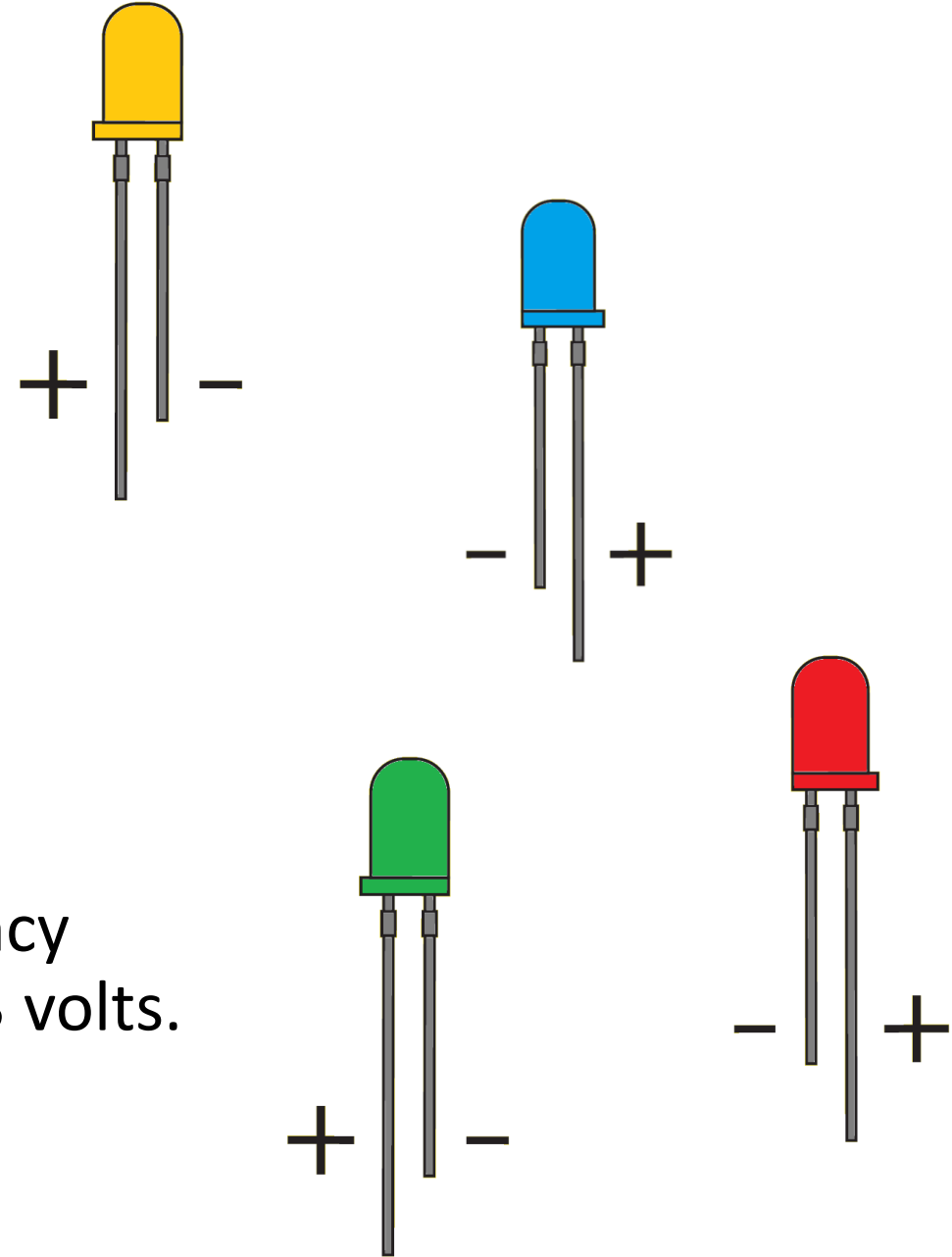
R is the resistance in Ohms

Note: Resistors are color coded



LEDs forward voltage

- Typically between 1.8 and 3.3 volts.
- It varies by the color of the LED.
- A red LED typically drops 1.8 volts
- Voltage drop normally rises as the light frequency increases, so a blue LED may drop from 3 to 3.3 volts.



Maximum current draw for GPIO pins

- The GPIO pins can draw **50mA** safely, that is 50mA distributed across all pins.
- An individual GPIO pin can only draw **16mA** safely

- Applying Ohms law for LED with a voltage drop of 1.8 Volts

$$R = (3.3 - 1.8)/0.016 = \mathbf{93.75 \text{ Ohms}}$$

- Choosing a resistor slightly higher than 100 Ohms will still work but will cause the LED to dim.
- Much higher resistors will prevent the LED from working.
- Choosing a lower resistor or connecting LEDs without resistors may permanently damage the pins or the Pi.

C++ Programming for Raspberry Pi GPIO

- Two ways to communicate with GPIO pins using code:

1. Write your own i/o driver for the Raspberry Pi, but you cannot just write to i/o memory (security issues).

<https://www.codeproject.com/Articles/1032794/Simple-I-O-device-driver-for-RaspberryPi>



2. Use a library to handle i/o to GPIO pins.



Wiring Pi GPIO Interface library for the Raspberry Pi

- **PIN** based GPIO access library written in C for the BCM2835, BCM2836 and BCM2837 SoC devices used in all Raspberry Pi.
- Includes a command-line utility **gpio** which can be used to program and setup the GPIO pins.
- Official Web site <http://wiringpi.com/>

Wiring Pi download and installation

- First, check that wiringPi is not already installed. In the terminal, run:
\$ gpio -v
- If installed as a package, remove the package:
\$ sudo apt-get purge wiringpi
\$ hash -r
- Install GIT if not installed:
\$ sudo apt-get install git-core
- If you get any errors, make sure your Pi has the latest version of Raspbian
\$ sudo apt-get update
\$ sudo apt-get upgrade
- Use GIT to obtain WiringPi
\$ cd
\$ git clone git://git.drogon.net/wiringPi
- If you have already used the clone operation for the first time, fetch an updated version
\$ cd ~/wiringPi
\$ git pull origin
- To build/install, run:
\$ cd ~/wiringPi
\$./build

Implementation

```
// Include wiringPi library
```

```
#include <wiringPi.h>
```

```
int main() {
```

```
// Define a constant and assign 7 to it. This represents the physical pin's number
```

```
const int LED_GRN_7 = 7;
```

```
// Use physical pins numbering order to communicate with pins
```

```
wiringPiSetupPhys();
```

```
// Set pin #7 mode to output
```

```
pinMode (LED_GRN_7, OUTPUT);
```

```
// Send high voltage (3.3 Volts) to pin #7. You could use "1" instead of "HIGH" for the 2nd argument
```

```
digitalWrite(LED_GRN_7, HIGH);
```

```
// Send low voltage (0 Volts) to pin #7. You could use "1" instead of "LOW" for the 2nd argument
```

```
digitalWrite(LED_GRN_7, LOW);
```

```
return 0;
```

```
}
```

Compile & Run

- The g++ compiler comes with Raspbian OS
- To compile the program:
\$ g++ main.cpp -o main -l wiringPi
- To run the program:
\$./main

In-class project: Prime numbers guessing game

- Connect two LEDs, **green** and **red**, to the Raspberry Pi GPIO pins.
- Write a C++ program that generates a random number between 1 and 500, and prints it on the screen.
- The program asks the user to guess if the printed number is a prime number.
- If the answer is correct, turn on the green LED and print “You win!” on the screen.
- If the answer is not correct, turn on the red LED, and print “You lose!” on the screen.
- The program then asks the user if he/she wants to continue playing or exit

Assignment: Credit Card Company Program

- Connect three LEDs, **green**, **red**, and **yellow** to the Raspberry Pi GPIO pins.
- Write a C++ program to manage a credit card company with at least one ADT (Account) with the following members:
card number, customer name, credit limit, and balance.
- The customer can pay the total amount of his/her balance or part of it.
- The customer can make a purchase using the credit card.
- The user can create, modify, and delete accounts.
- All new accounts are created with \$300 credit limit.
- Customers' data is stored in a binary file.

Assignment: Credit Card Company Program Cont'd

- The program's main menu has the following options:
 1. Create a new account
 2. Pay balance
 3. Make a purchase
 4. Check balance
 5. Edit an account
 6. Close an account
 7. View all accounts
 8. Exit

Assignment: Credit Card Company Program Cont'd

- Two types of output:

1. Text output printed on the screen.

Messages like “account closed!”.

(Part of your homework).

2. LED indicators' lights on a breadboard connected to Raspberry Pi GPIO port.

(Part of your graded work in the lab)

Assignment: Credit Card Company Program Cont'd

- Text Output (Part of your homework):
- Create a new account: Print “Account created” on success or “Creation Failed” on failure (duplicate credit card number)
- Pay balance: Print “Fully paid off” on full payment or “partially paid off” if partial payment is made.
- Make a purchase: Print “Purchase made” on success or “Purchase Failed” on failure (purchase amount exceeds the remaining credit limit)
- Check balance: Print “No balance to pay” if balance is 0, “Card maxed out” if balance = credit limit (credit limit is reached), “Your balance is XXX” otherwise.
- Edit an account: Print “Account modified” on success or “Modification Failed” on failure (duplicate credit card number)
- Close an account: Print “Account closed” on success or “Closing Failed” on failure (cannot close an account if balance is not paid off first).

Assignment: Credit Card Company Program Cont'd

- Raspberry Pi LED lights output (Not part of your homework, added during lab):
- Create a new account: Green LED lights up on success, Red on failure (duplicate credit card number)
- Pay balance: Green LED lights up if paid in full, Yellow if partial payment is made.
- Make a purchase: Green LED lights up on success, Red on failure (purchase amount exceeds remaining credit limit)
- Check balance: Green LED lights up if balance is 0 (all paid off), Red if credit limit is reached, Yellow otherwise.
- Edit an account: Green LED lights up on success, Red on failure (duplicate credit card number)
- Close an account: Green LED lights up on success, Red on failure (cannot close account if balance is not paid off first).