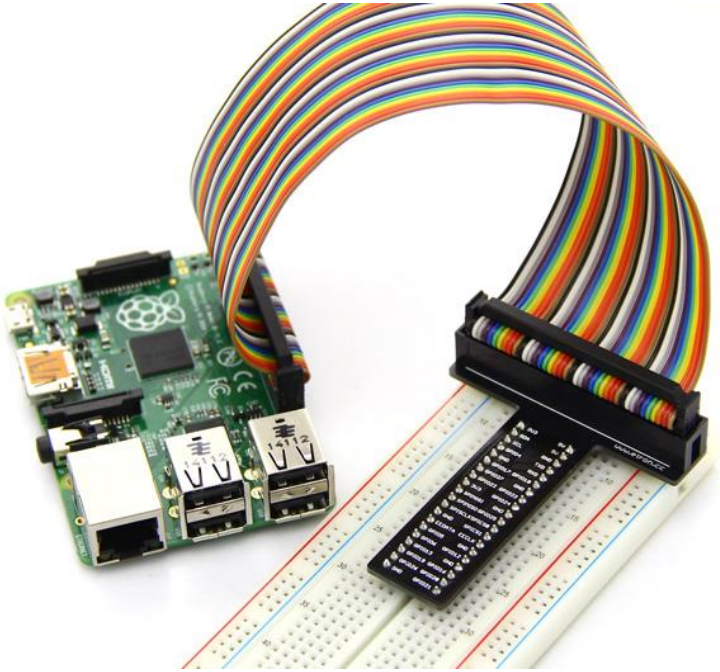
Linux For Embedded Systems

For Arabs

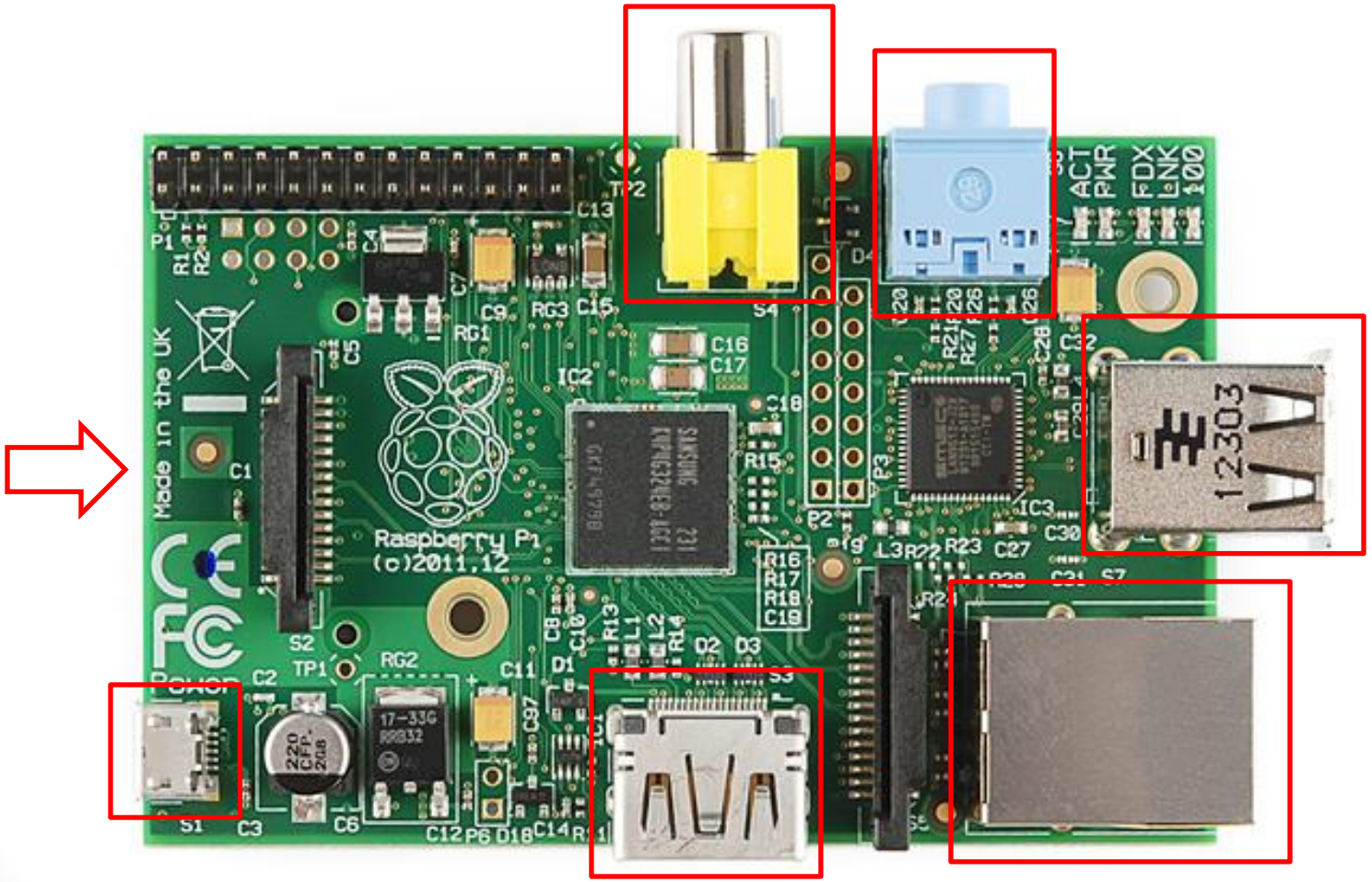
Cairo University
Computer Eng. Dept.
CMP445-Embedded Systems

Ahmed ElArabawy

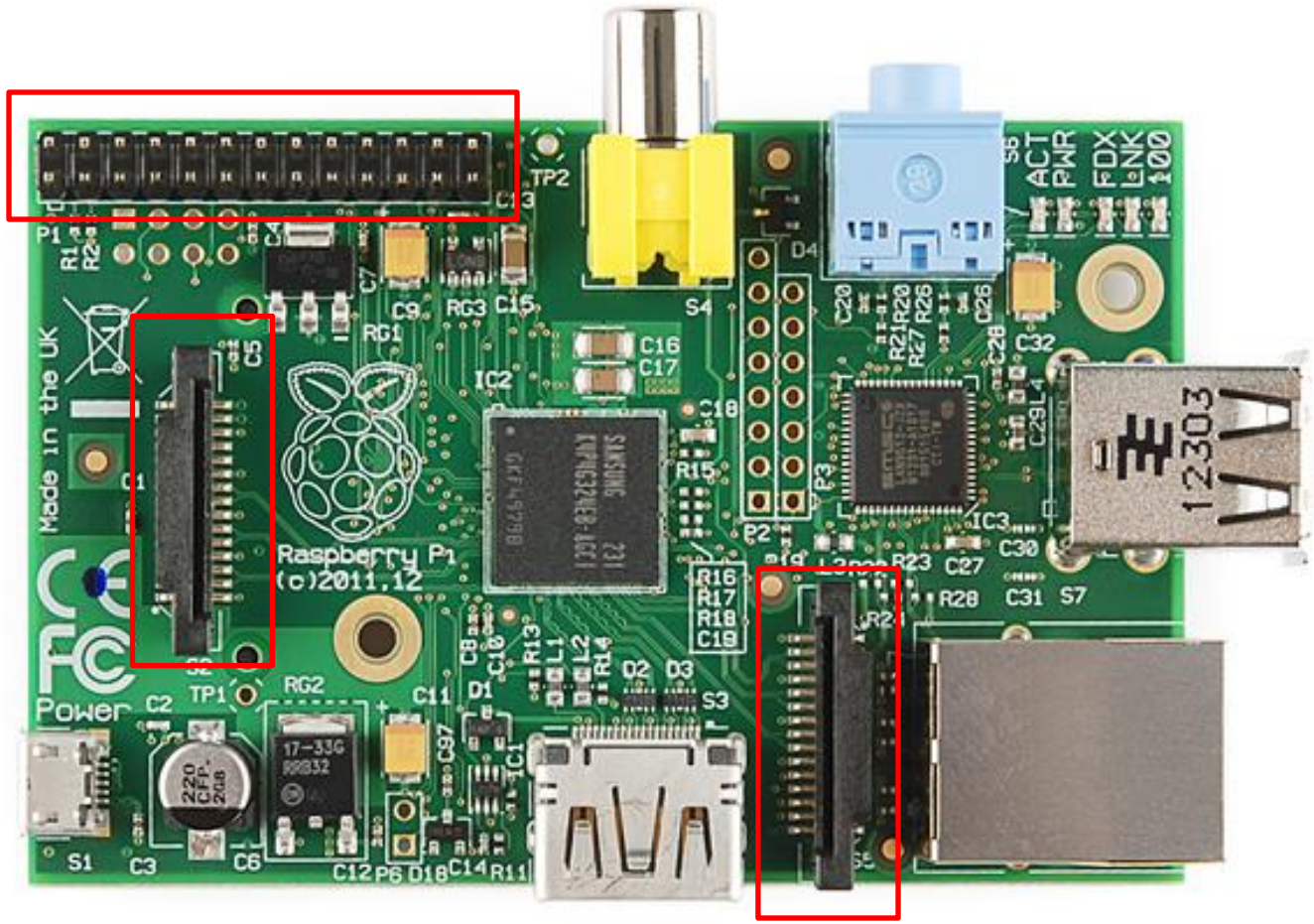


Lecture 9:
The Pi Control Arm

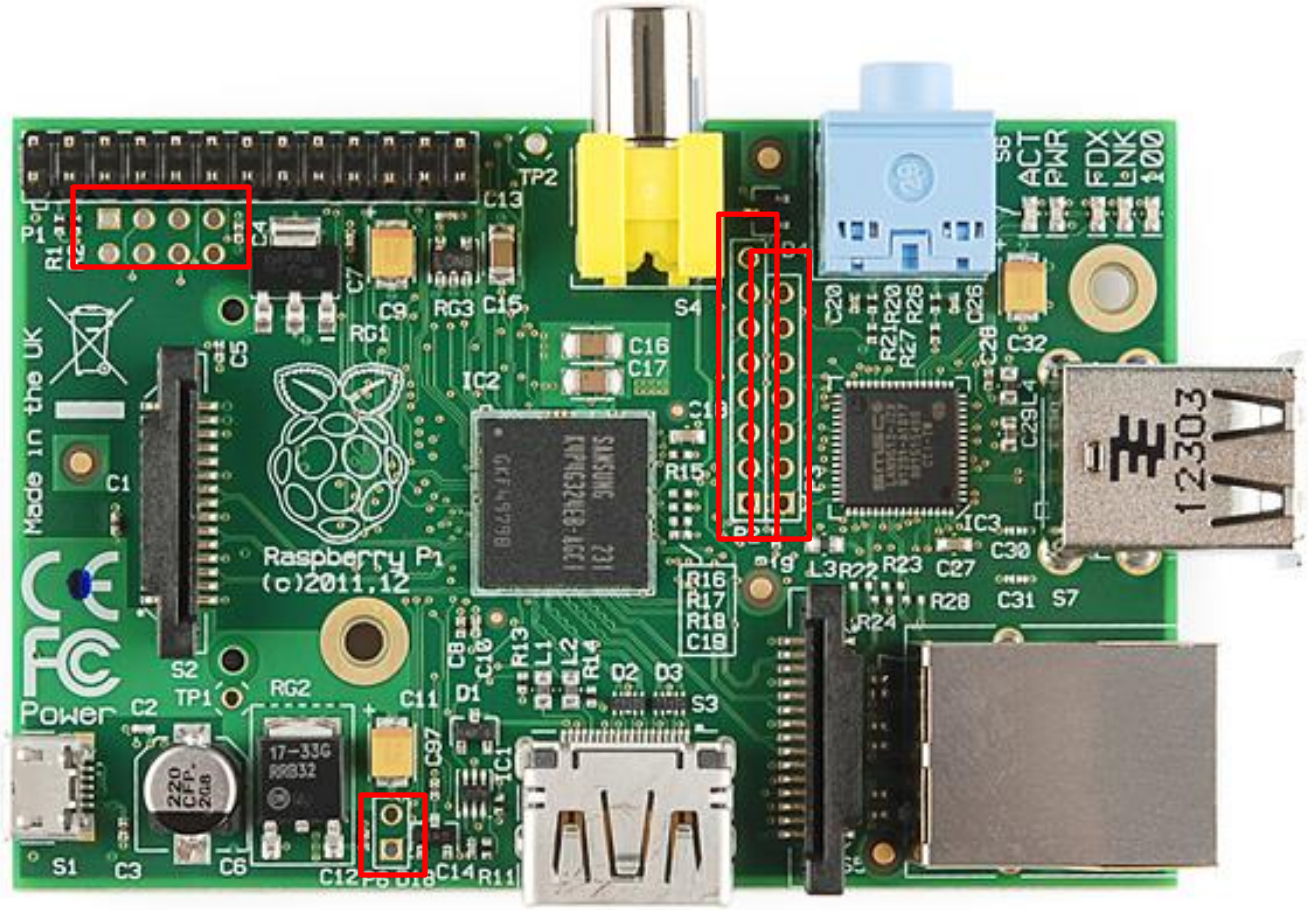
A Quick Look at Model B



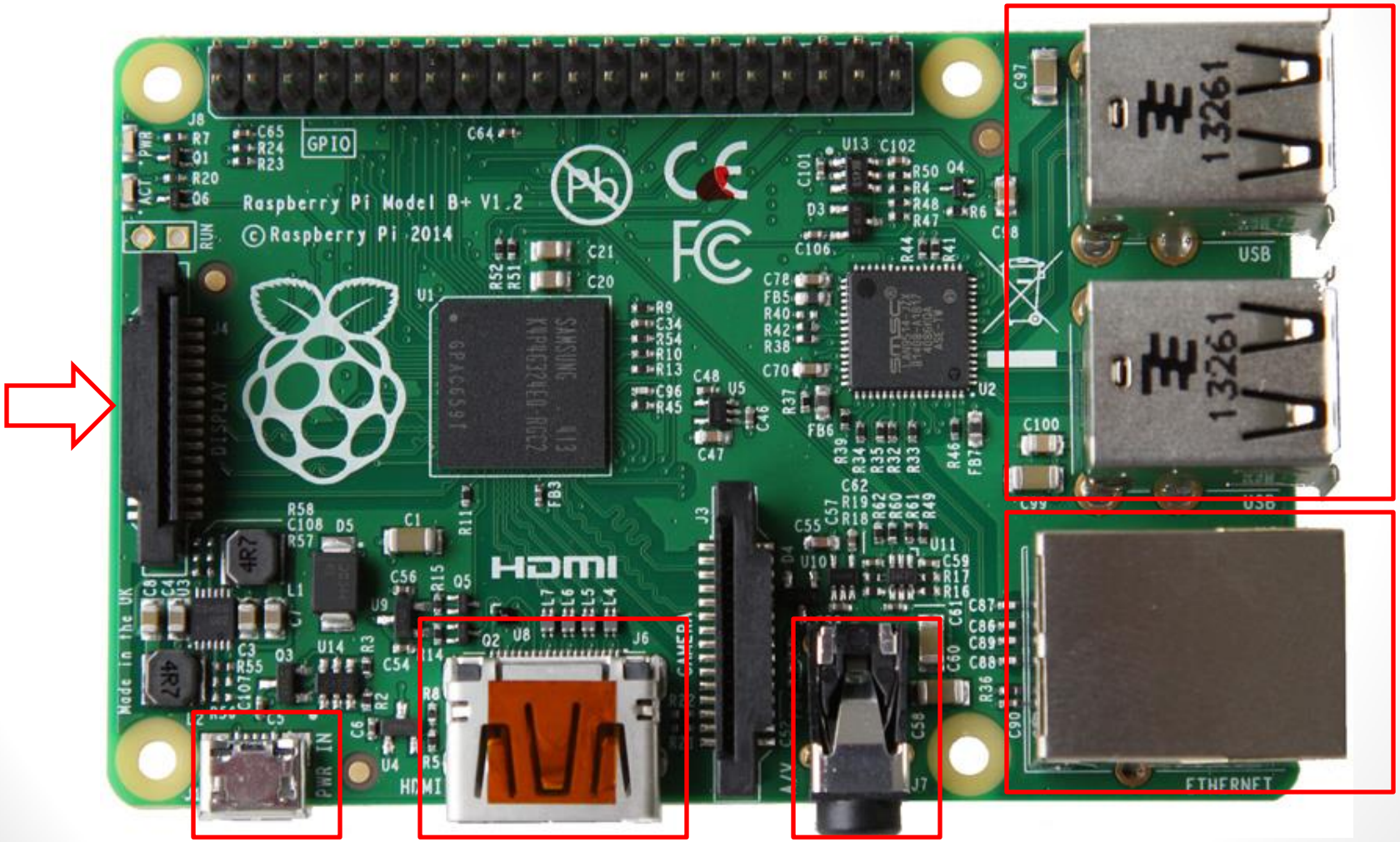
A Quick Look at Model B



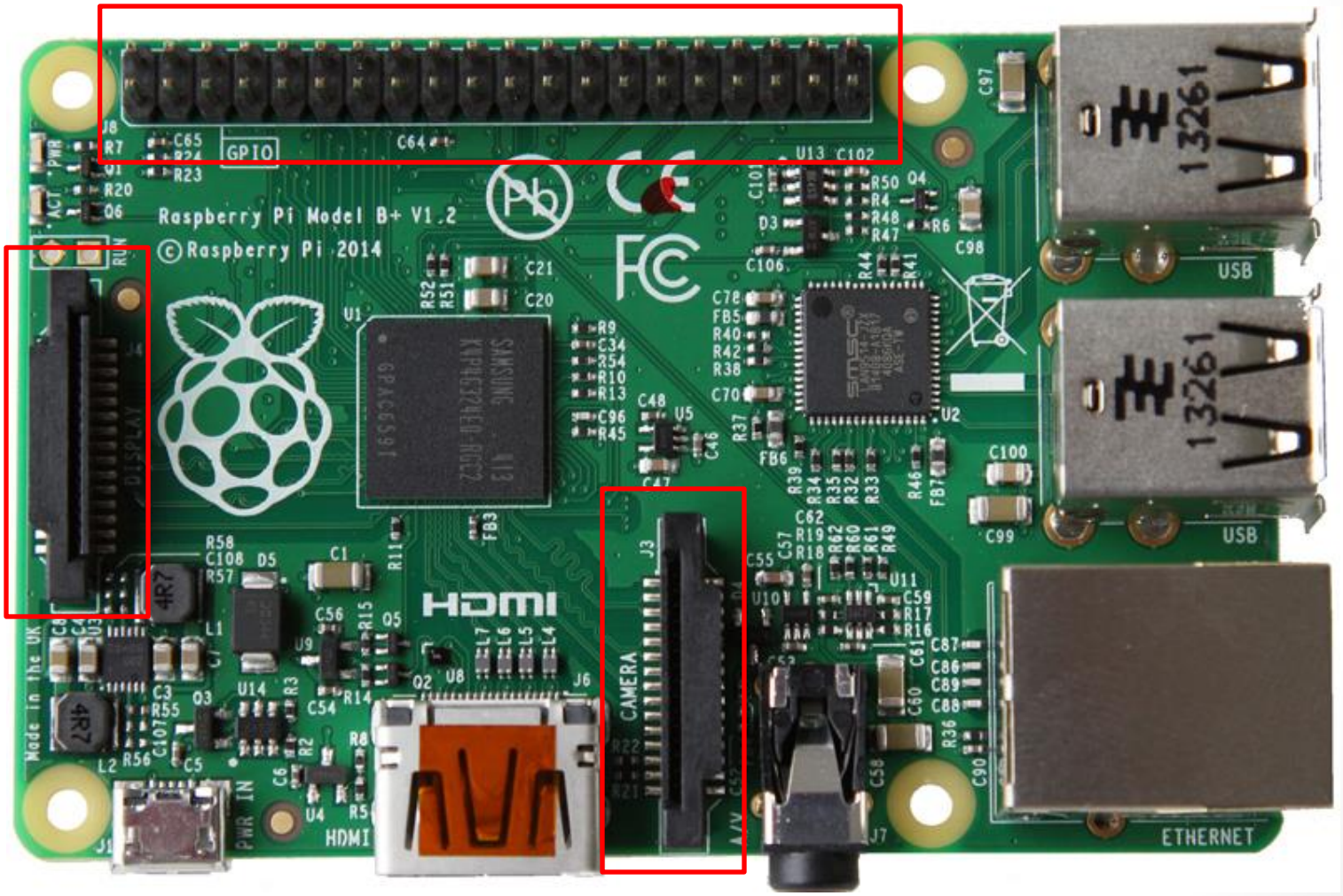
A Quick Look at Model B



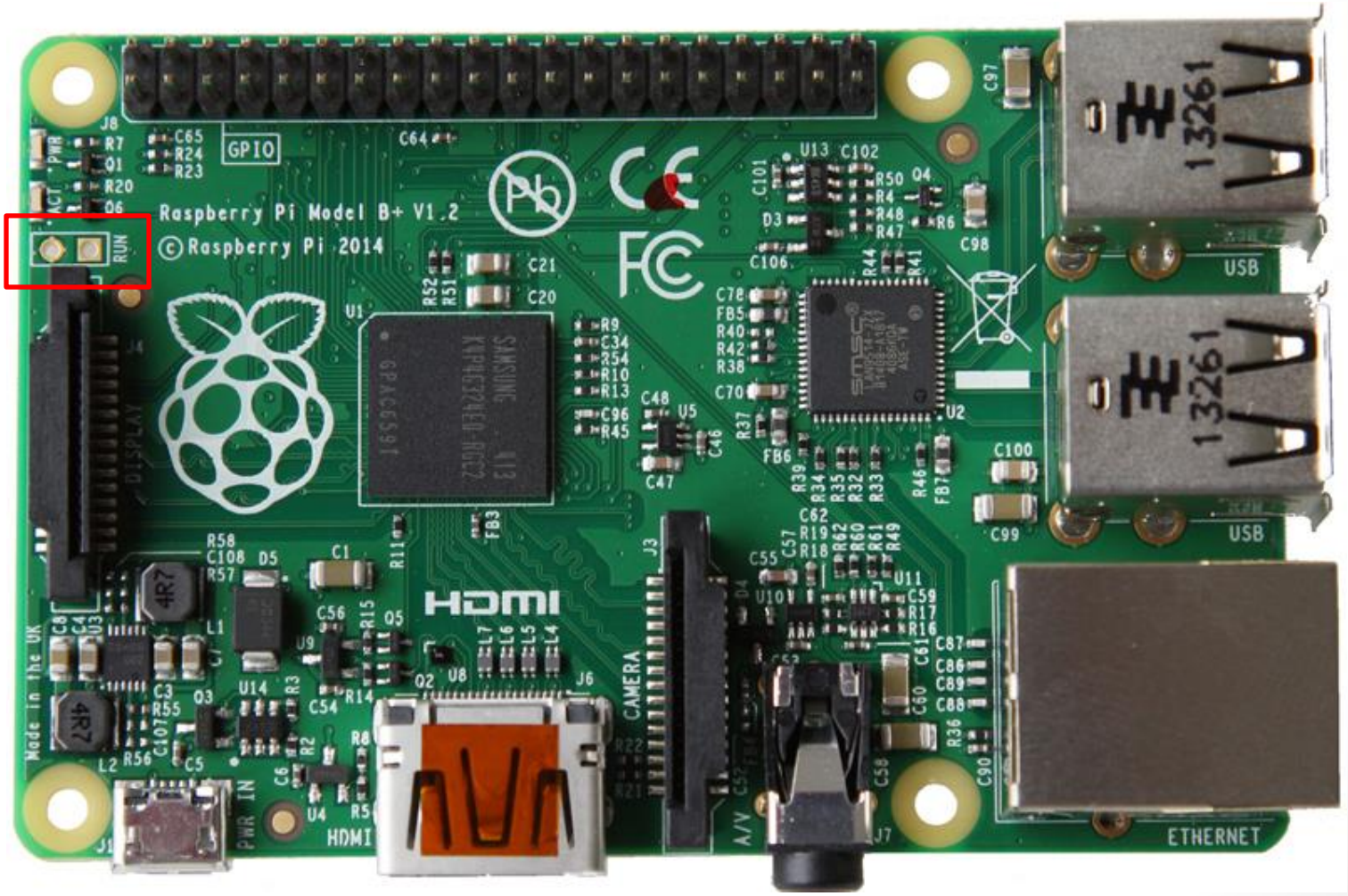
Moving to Model B+



Moving to Model B+



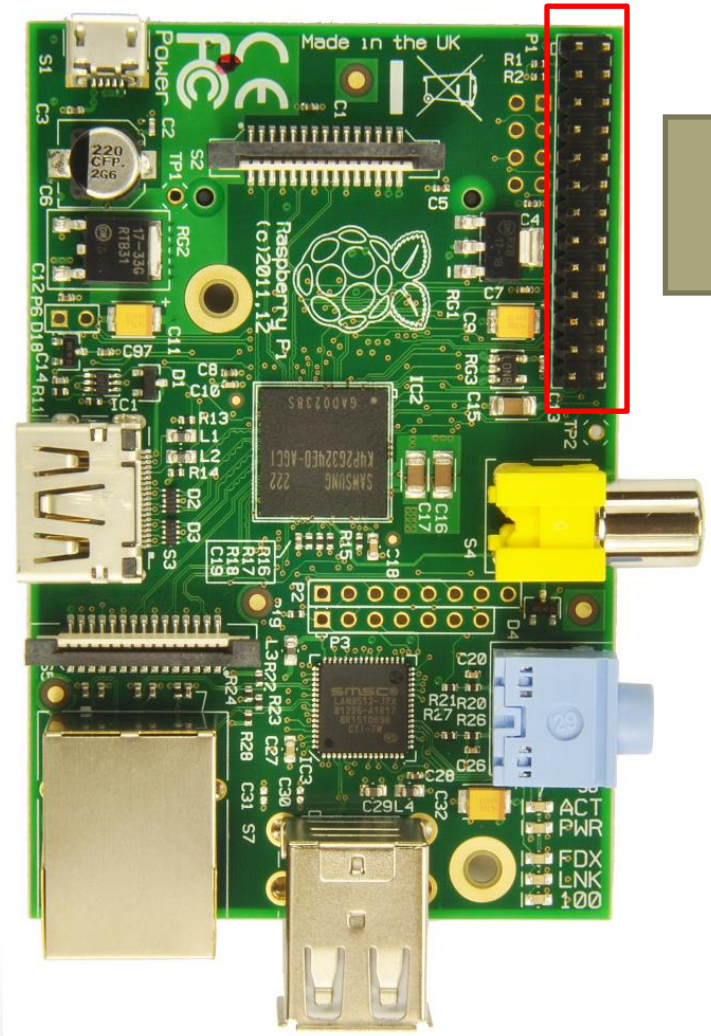
Moving to Model B+

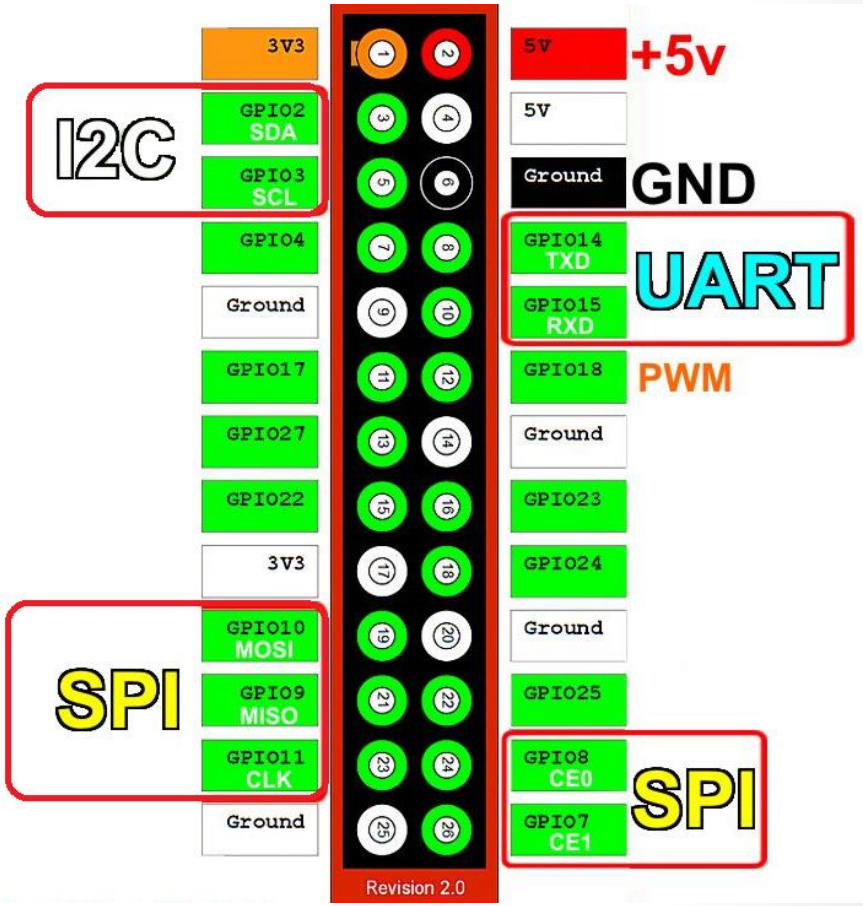


Interfacing the Pi

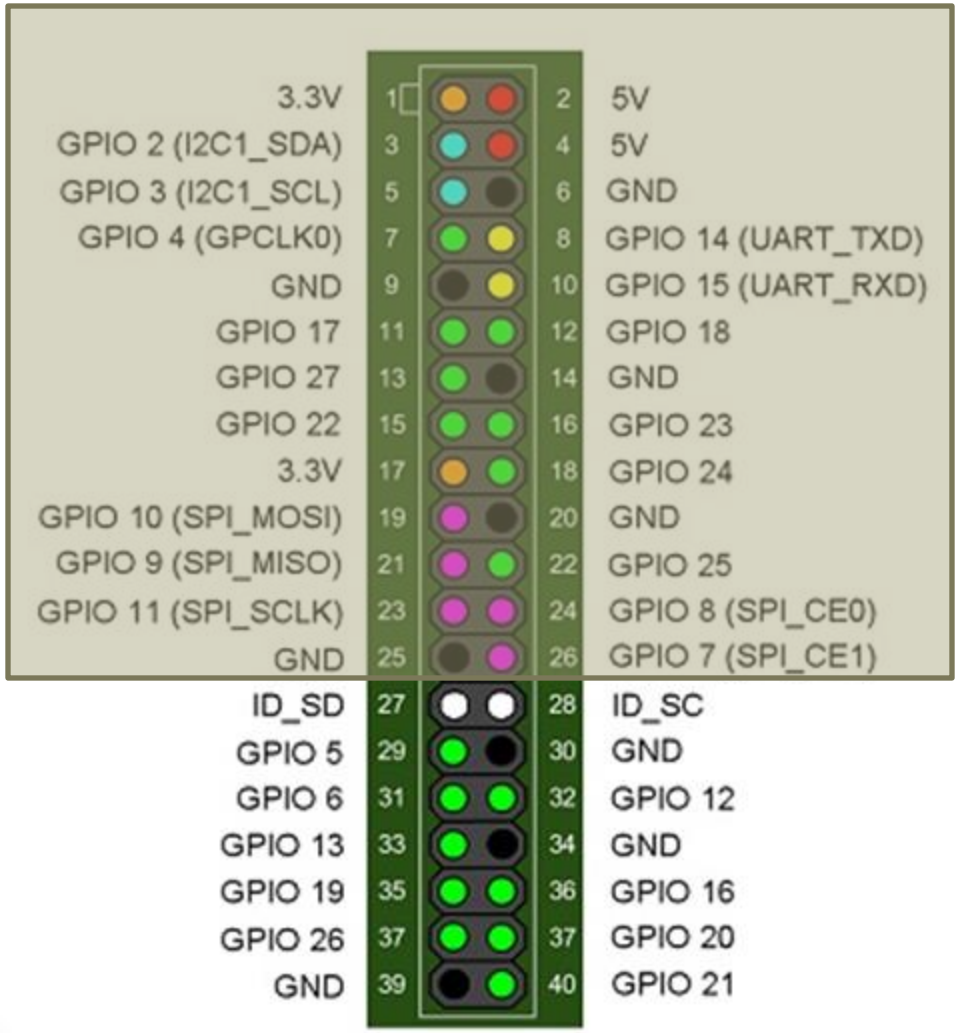
- The Raspberry Pi can be interfaced to external devices and peripherals via,
  - Existing Connectors
    - Ethernet, USB, A/V, HDMI, Power, SDIO
  - Signal Header Connectors
    - GPIO Header (26 pin in B, 40 Pin in B+)
    - CSI (For Camera Interface)
    - DSI (For Display Interface)
  - Unconnected Signal Headers
    - These headers need some soldering to use it
    - Mainly the chip RESET signal

GPIO Header in Model B



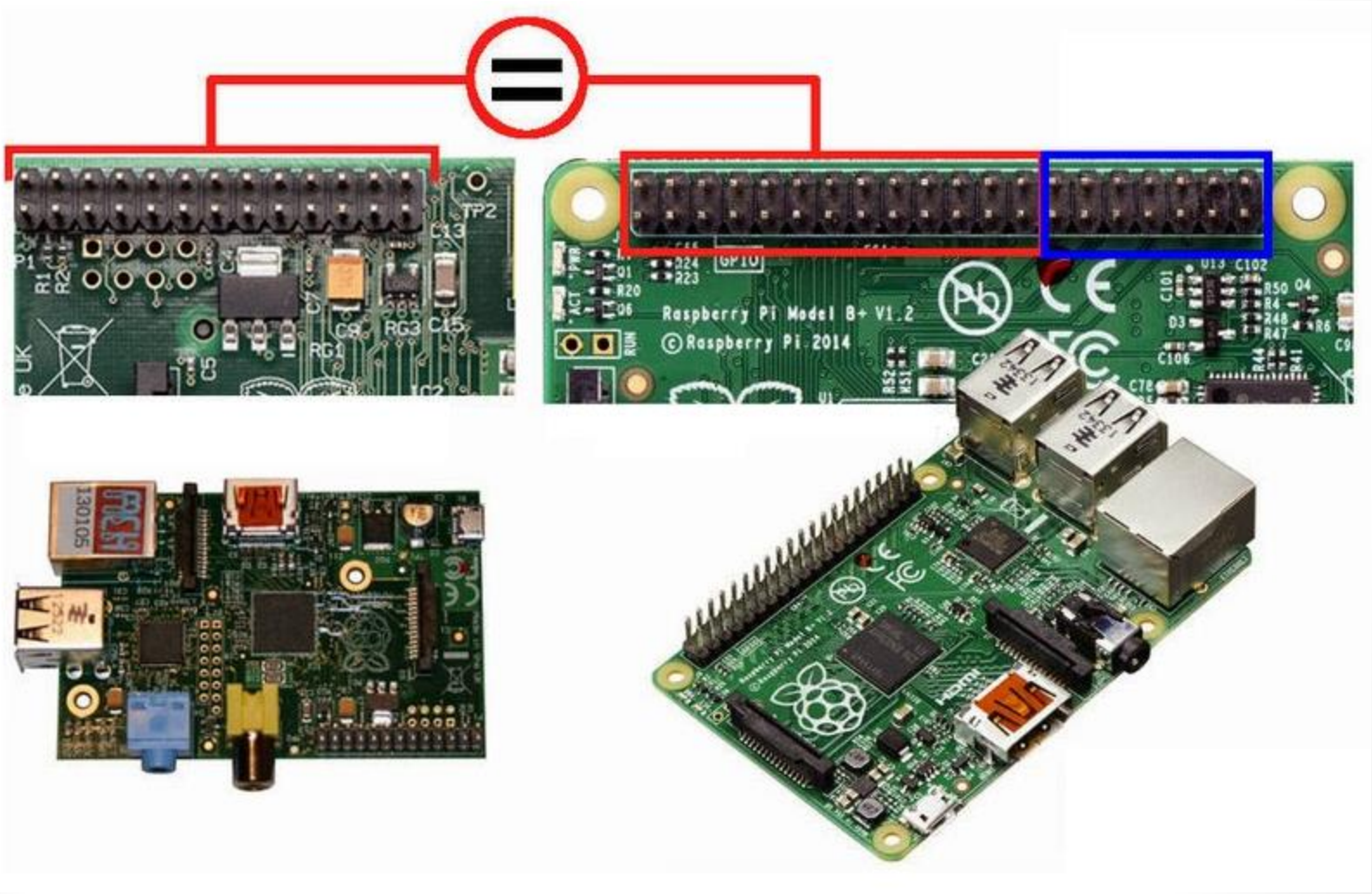


GPIO Header in Model B+



3.3V	1	2	5V
I2C1 SDA	3	4	5V
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
GPIO 10 MOSI	19	20	GROUND
GPIO 9 MISO	21	22	GPIO 25
GPIO 11 SCLK	23	24	GPIO 8
GROUND	25	26	GPIO 7

B Versus B+ Models

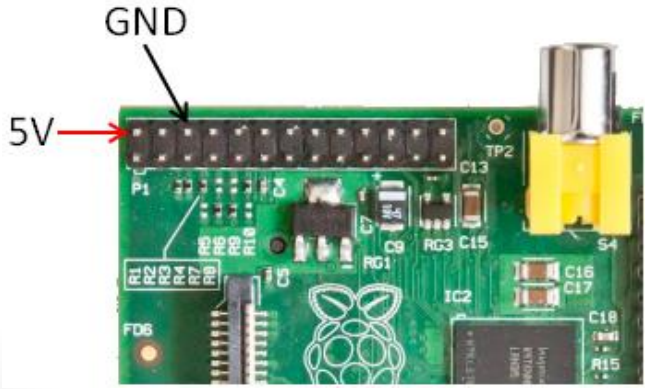


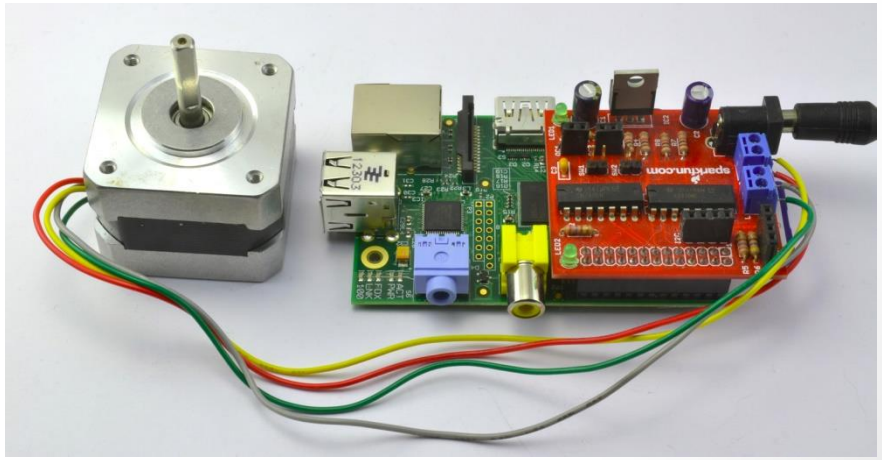
GPIO Header

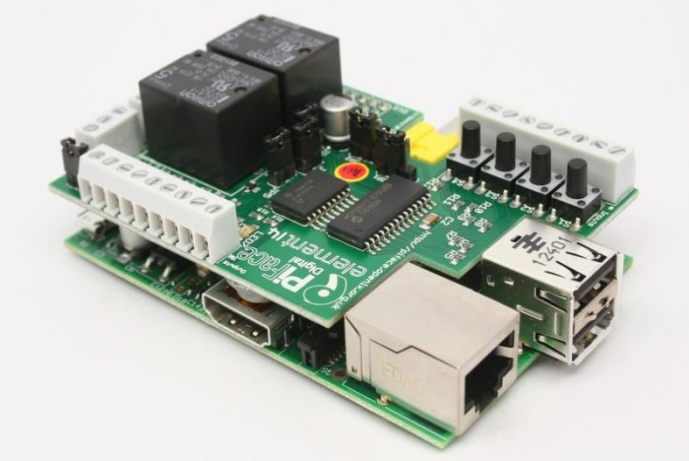
- The GPIO header is the main method for hardware interfacing
- In Model B, it contains 26 pins, and in Model B+ it was upgraded to 40 Pin
- The GPIO header in Model B+ is backward compatible with the older model, this means that pins 1-26 in model B+ are identical to those in model B
- This way, any hardware designed to interface with the Model B can interface with model B+ with no need for any change

GPIO Header Signals Power/GND

- The Raspberry Pi requires a 5V power line
- This is normally provided using the Micro-USB Connector
- You can also power the Raspberry Pi through the GPIO Header
- This is used when the Pi is powered from another board that is connected to it
- You can also use the power signal in the GPIO header to power other boards (as long as they are a light load)





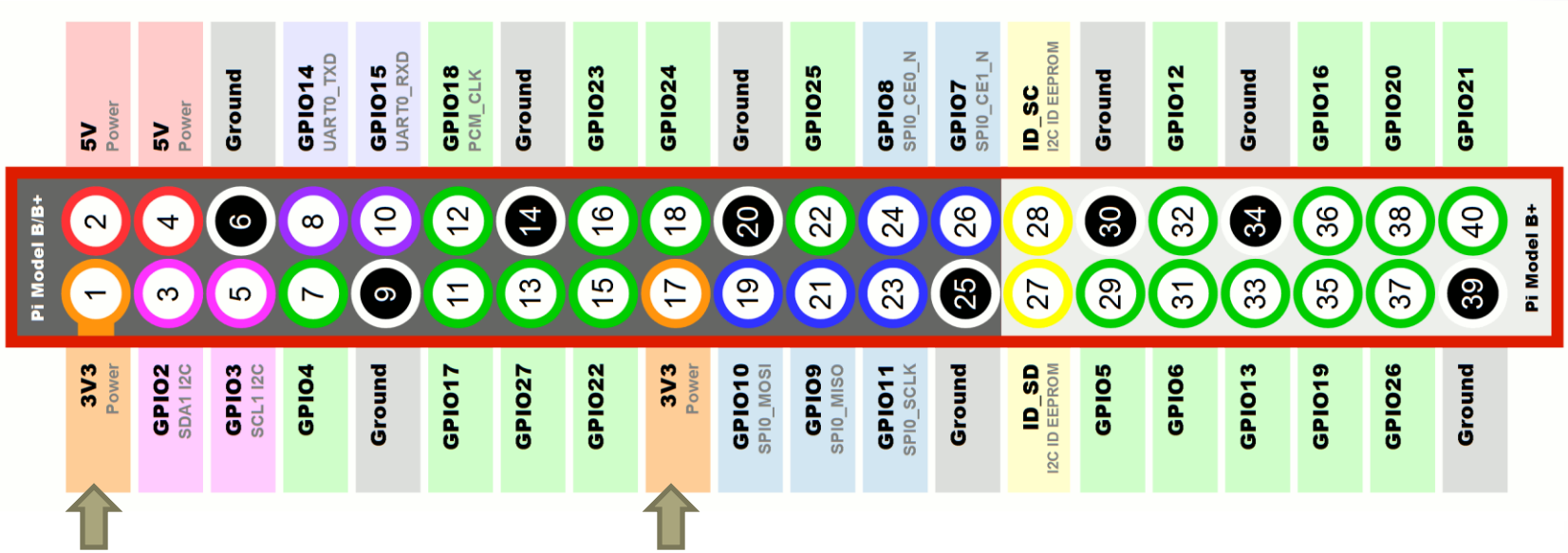


- The Pi can be used to feed power to other boards connected to it
- The maximum current you can take from the 5V rail is based on the used power supply
- The board takes around 700 mA from the 5V power supply, any extra current the power supply can provide can be used to external circuits

5V vs. 3.3 V

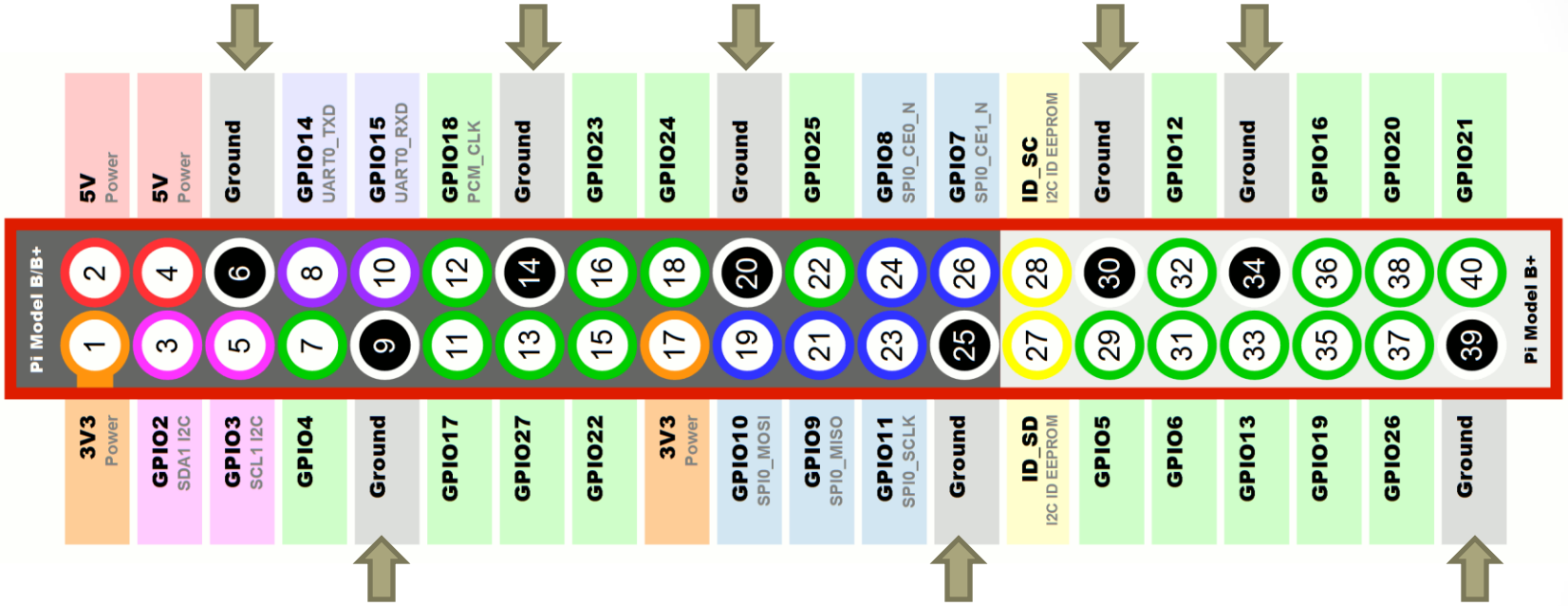
- Note that although that the Pi is powered using 5V, all of its signaling is done using 3.3V
- This is a **VERY IMPORTANT** thing to watch for, **NEVER** connect the Pi directly to any circuits using 5V signaling
- Note that Arduino uses 5V signaling, so all Arduino circuits can not be connected to the Pi
- The Pi does not have a over voltage protection for its pins, hence you can easily destroy the board by connecting it to 5V circuits
- If you need to use chips that run with 5V logic such as those running with Arduino boards,
  - If the chip only takes output from the Pi, sometimes, the 3.3V of the Pi is good enough for the external chip to detect logic 1
  - If the 3.3V of the Pi is not enough to drive the chip, then you need a 3.3V to 5V Level Shifter
  - If the chip provides input to the Pi, then you will always need to use a 5V to 3.3V Level Shifter. **Not doing that will damage your Pi Board**

GPIO Header Signals
3.3V Line



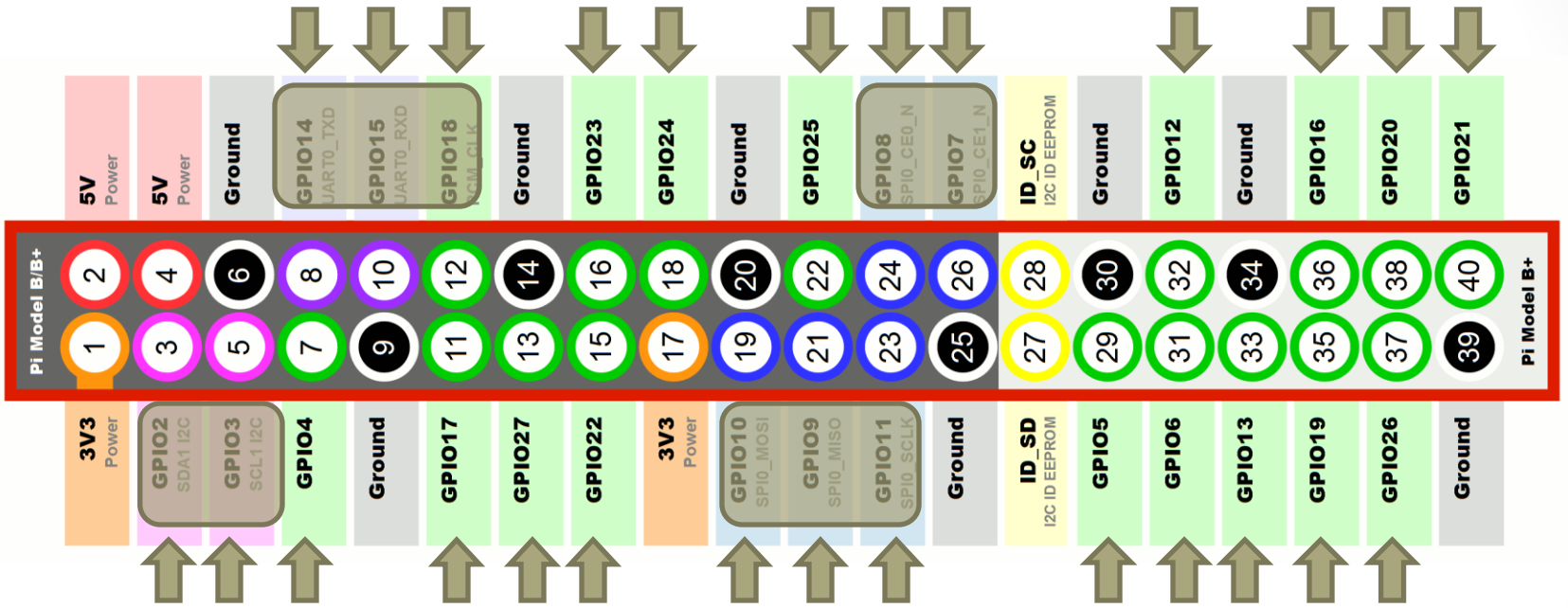
- These pins can be used to provide power to 3.3V circuits (If not going to overload the Pi)
- This is **NOT** an input signal, the Pi only takes 5V line, this is an output line

GPIO Header Signals
GND Line



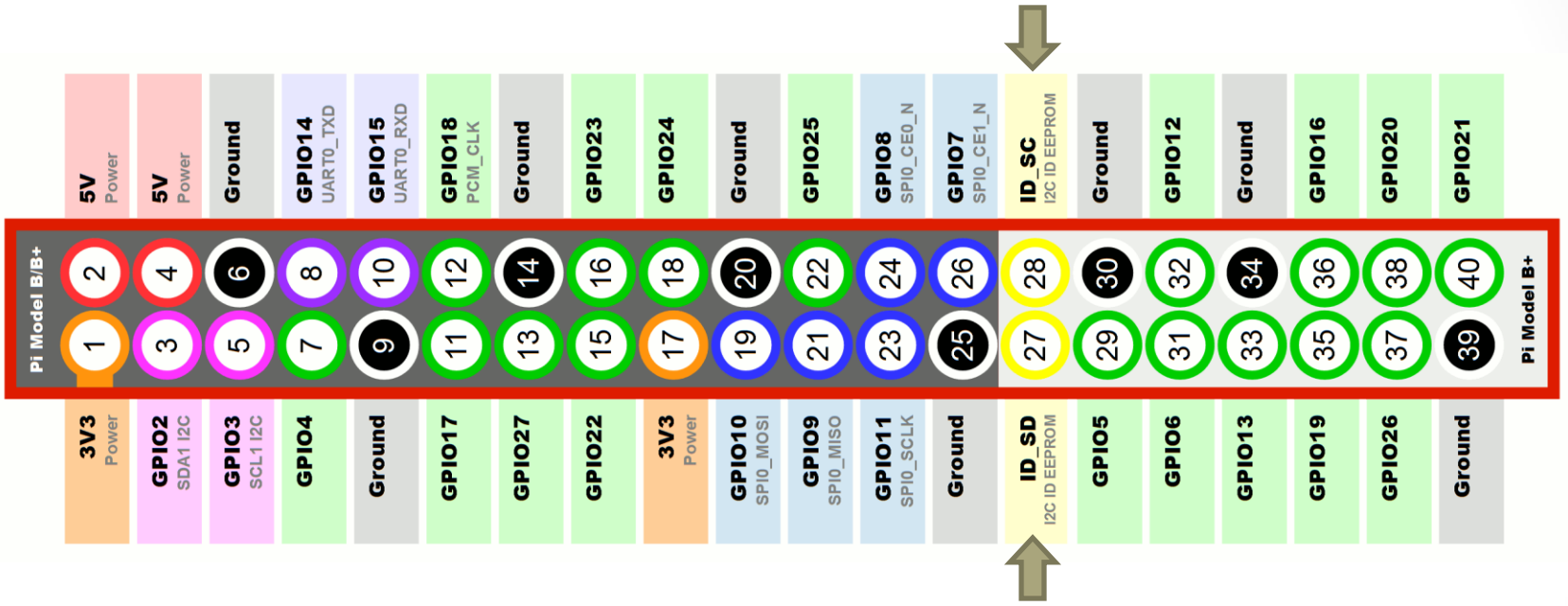
- These pins represent common signals between the Pi and external circuits connected to it

GPIO Header Signals
GPIO Signals



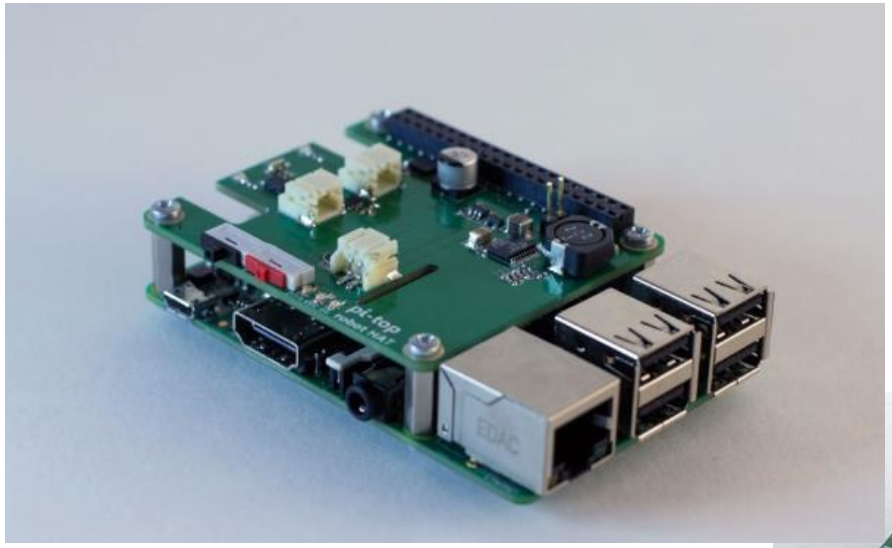
- 26 Pin for GPIO (General Purpose Input Output)
- These pins can be programmed to be either input or output signals
- These signals use **3.3V** logic
- Some of the GPIO pins have a dual role (either GPIOs or part of another interface)
- The default for these pins is to be a GPIO, to switch to the other role, you need to load the driver for the needed interface

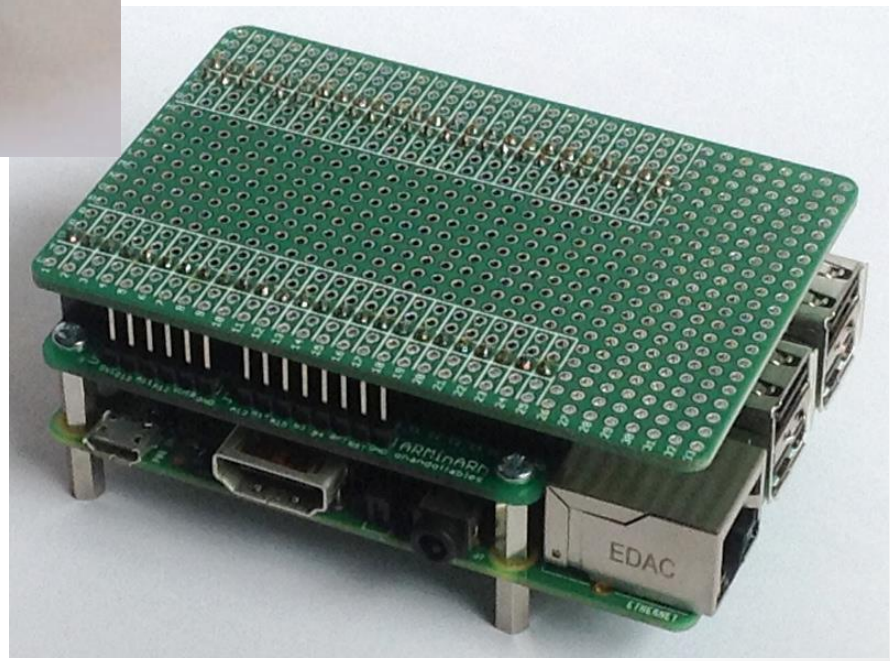
GPIO Header Signals
ID EEPROM Signals



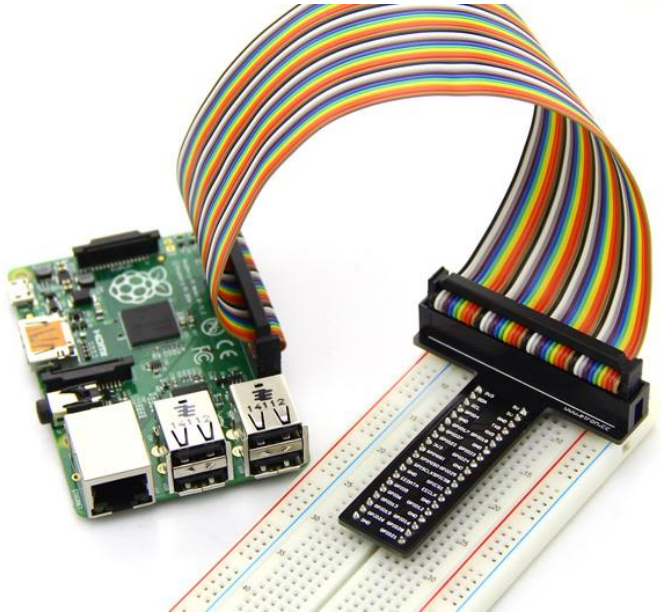
- These two pins are introduced in model B+ to enable a new concept called Raspberry Pi HATs
- A Pi HAT (Hardware Attached on Top), is a daughter board that can be connected on top of the Pi
- The Pi uses those two pins to read an ID EEPROM that describes the attached hardware and the required configuration of the GPIO pins

Raspberry Pi HATs





Interfacing to the GPIO Header



- You can connect wires directly to the Pi Header
- However, it is safer to use a breakout connector (also called Cobbler)
- Before you do any hardware connection, make sure that the cable is connected in the right direction (test that the signals on the breadboard maps correctly to the pins on the Pi)
- Also make sure that no connection or disconnection is done while the Pi is powered on (power on the Pi after all connections are made, and power it off before any modification in the connections)
- Review your connections thoroughly before power up of the board. **Any mistake can destroy the board**

How to Damage a Pi

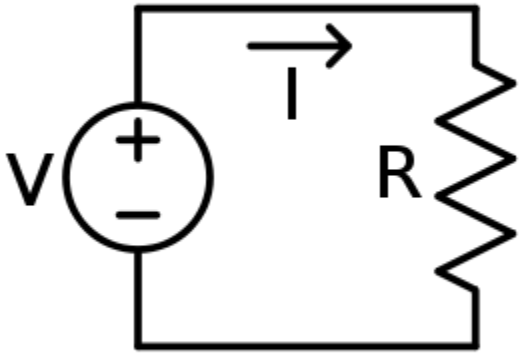
- The following actions may destroy the Pi,
  - Touching the chips of the Pi (Statics in your body may damage the chips)
  - Connecting the Pi to the wrong Power supply (It is not protected against over voltage)
  - Making new connections in the circuit while the Pi is connected and powered up (transient currents may result in damage to the board)
  - Disconnecting a USB while the Pi is powered
  - Connecting the Pi GPIO to 5V signal (the Pi uses 3.3V signaling)
  - Connecting the output of a device to a Pi GPIO while it is configured as output
  - Connecting an output GPIO directly to GND or Vcc

Basic Concepts for Electronic Circuits

Current Limiting Resistors

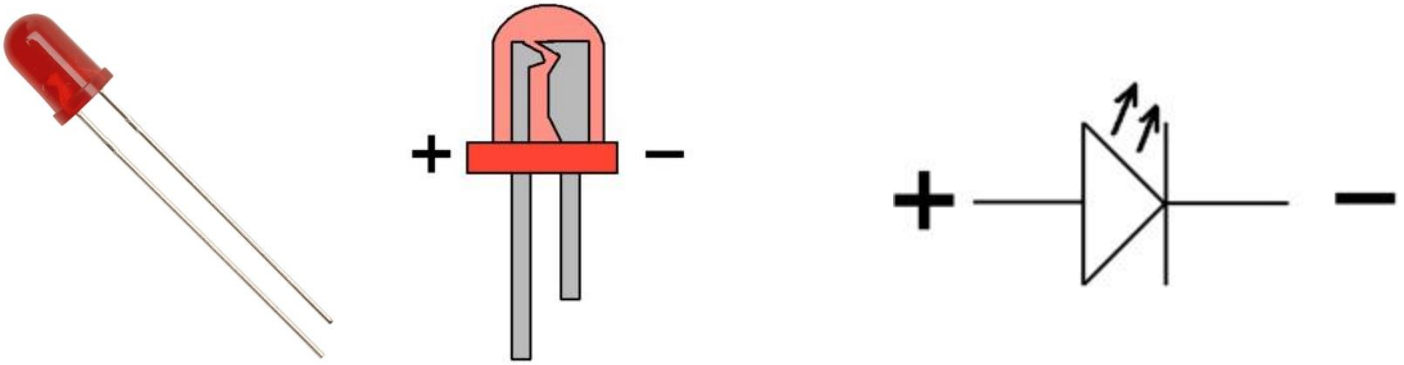
$$I = \frac{V}{R}$$

Electric current = Voltage / Resistance



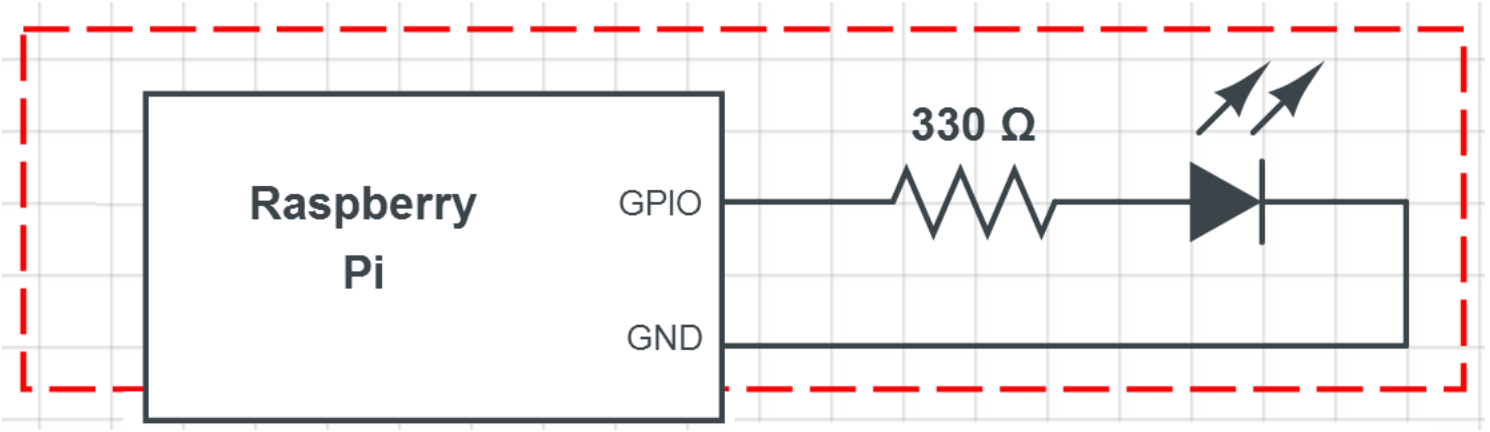
- We use resistance in our electronic circuits to limit the current running in the circuit
- In our circuits, we need to always calculate the expected current to make sure we are not overloading the Raspberry board
- For the GPIO Pins (3.3V signaling), each pin can provide a maximum of **50 mA**. Beyond that you can damage the board

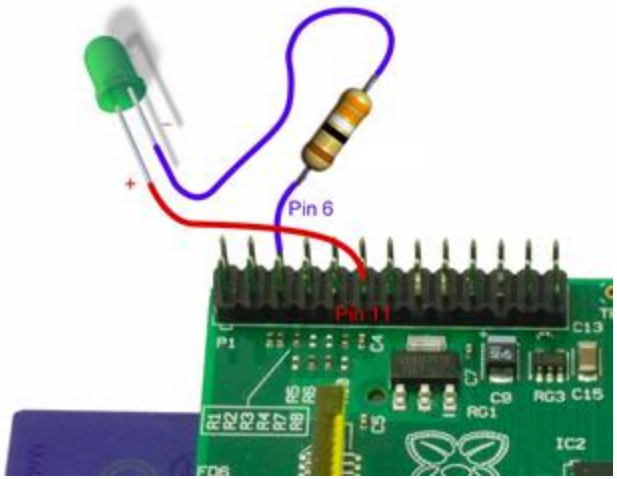
Using LEDs



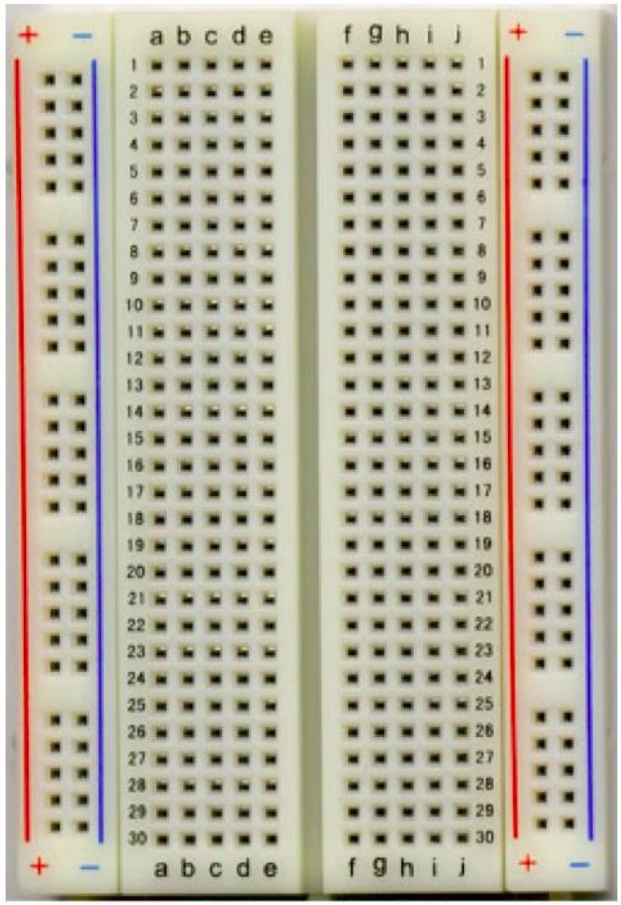
- LED (Light Emitting Diode) is used to show the status of a signal
- When it is high, the LED emits light, otherwise it remains off
- A LED is some sort of a diode, which means current only goes in one direction, from anode to Cathode

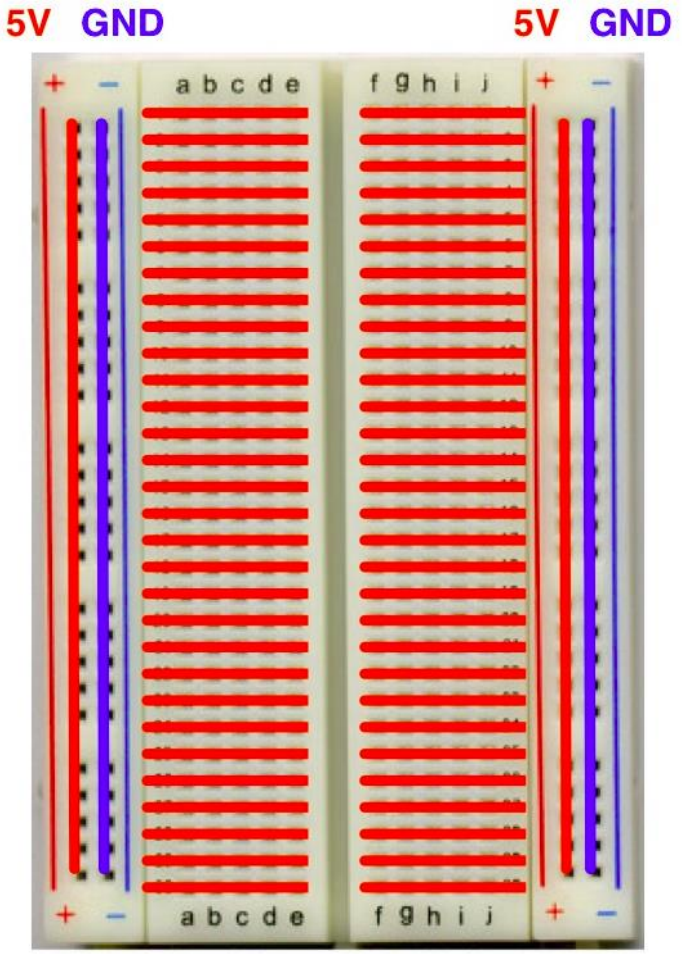
Interfacing to O/P GPIO



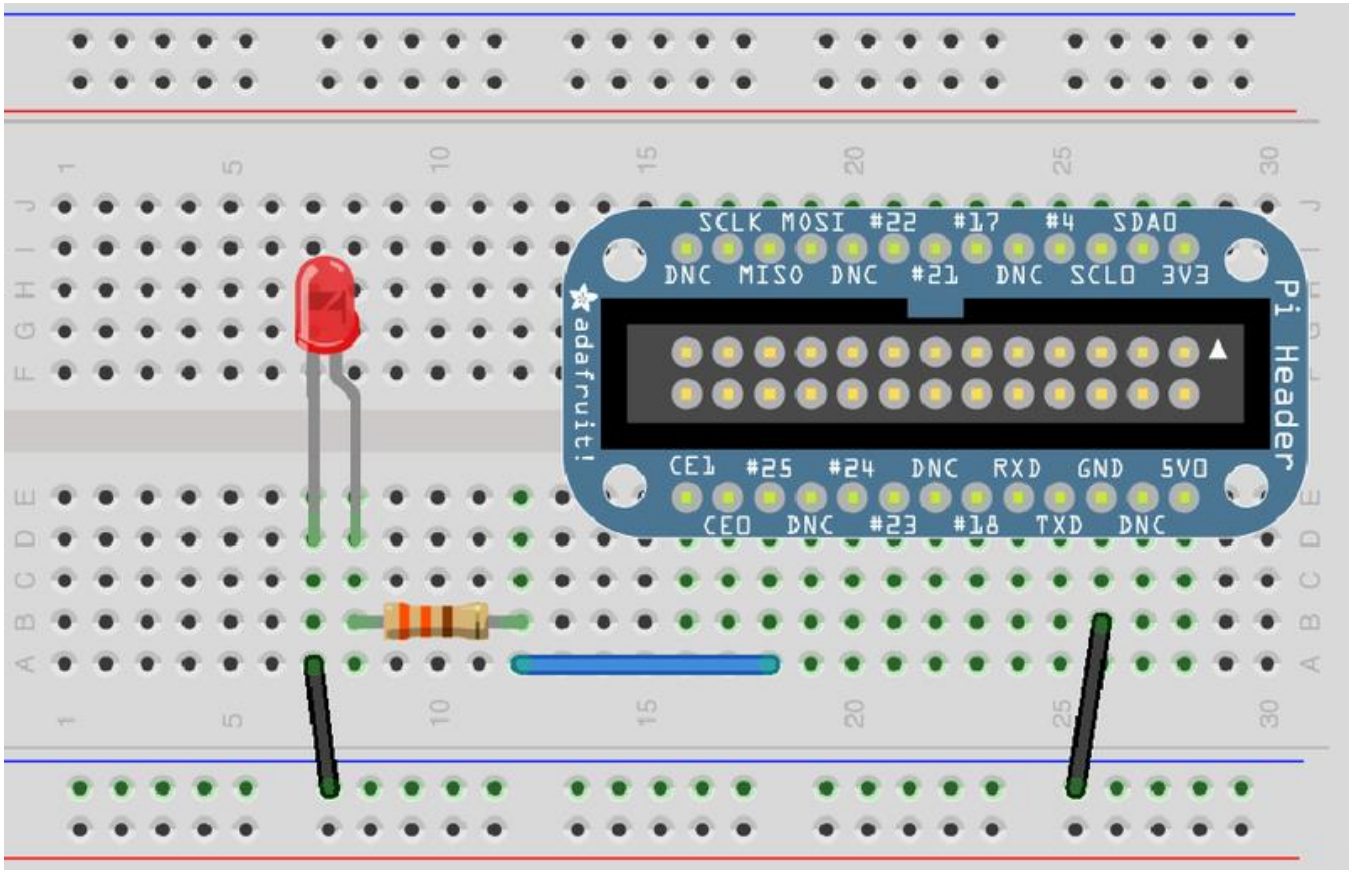


Using a BreadBoard

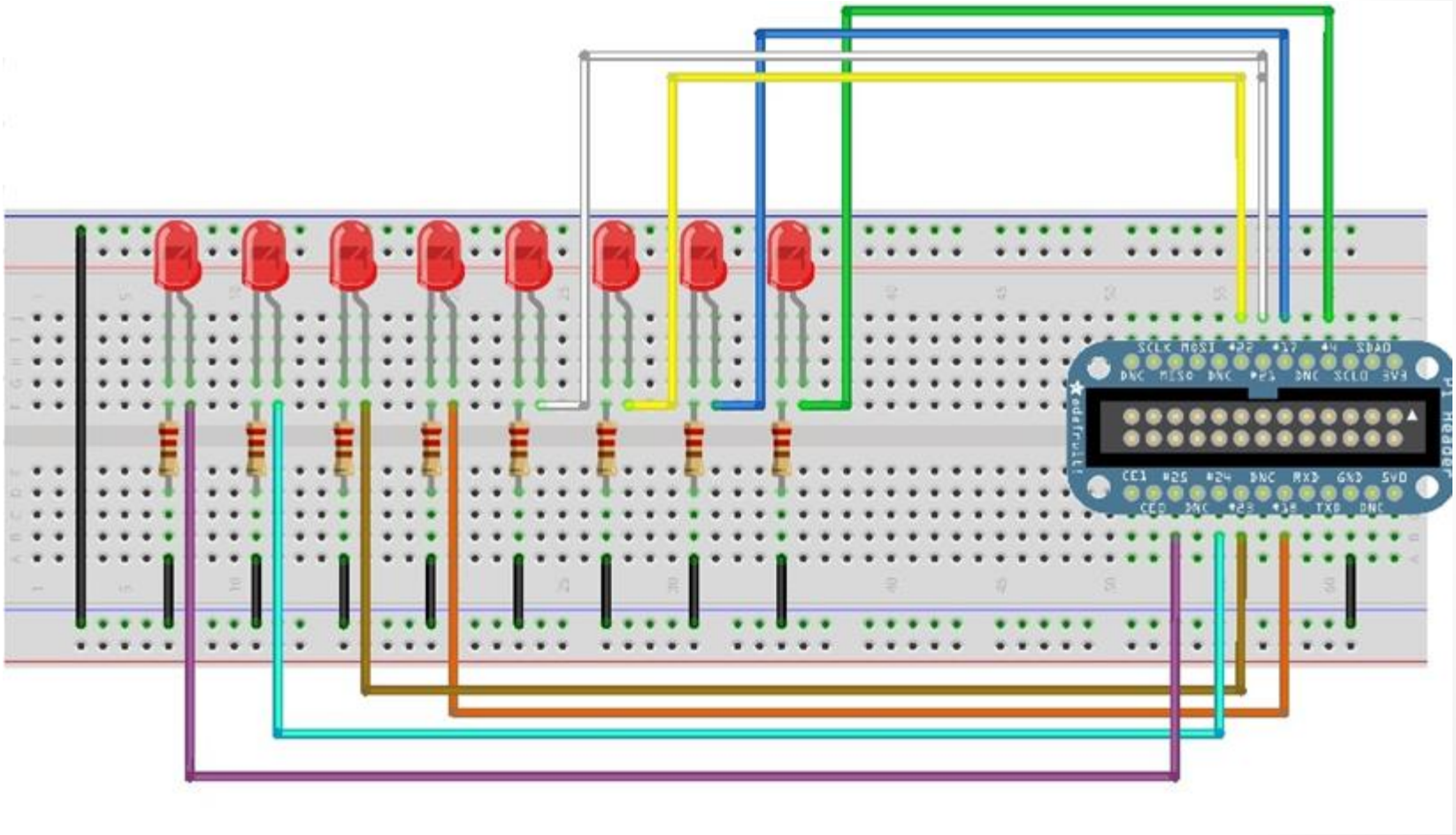




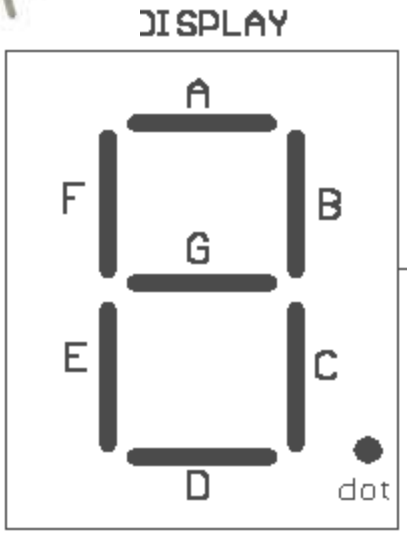
Making the Connection

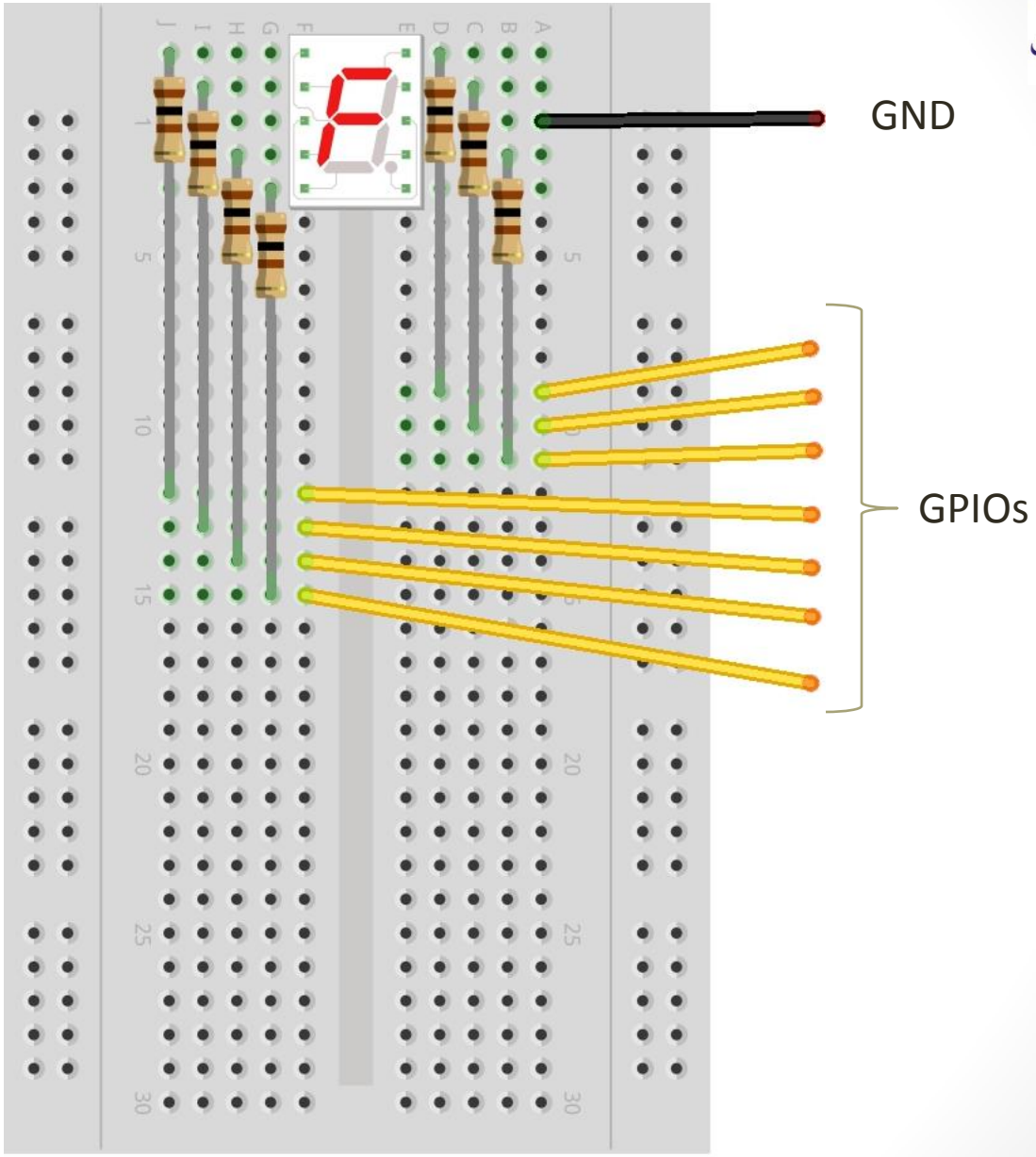


Using Multiple LEDs

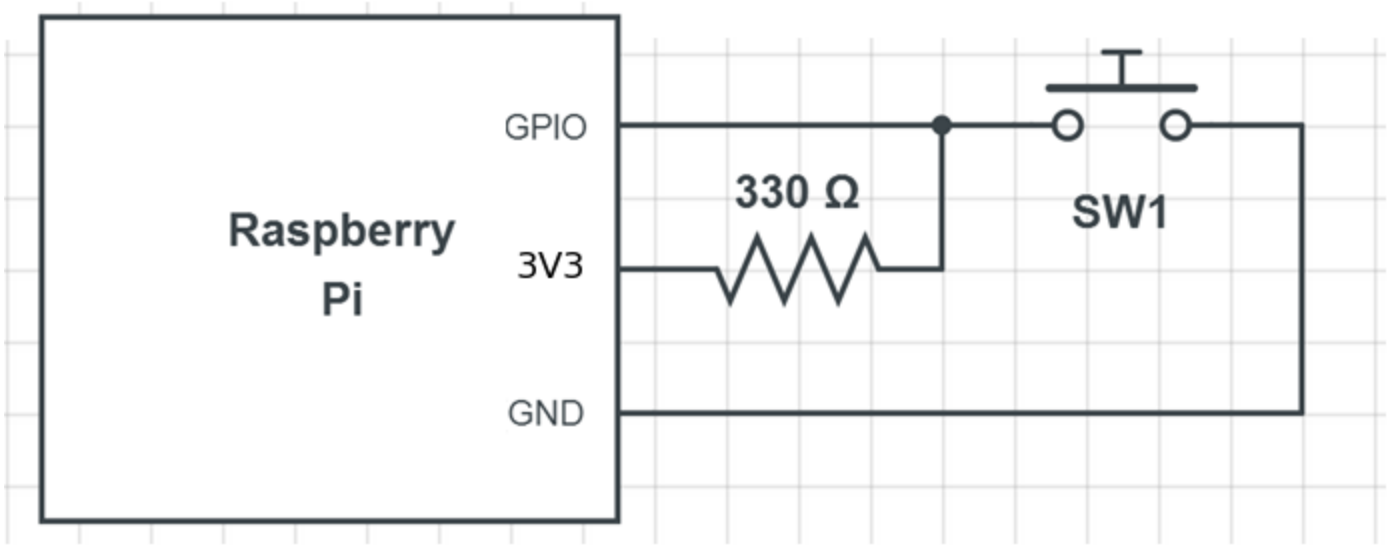


Using 7 Segment Display



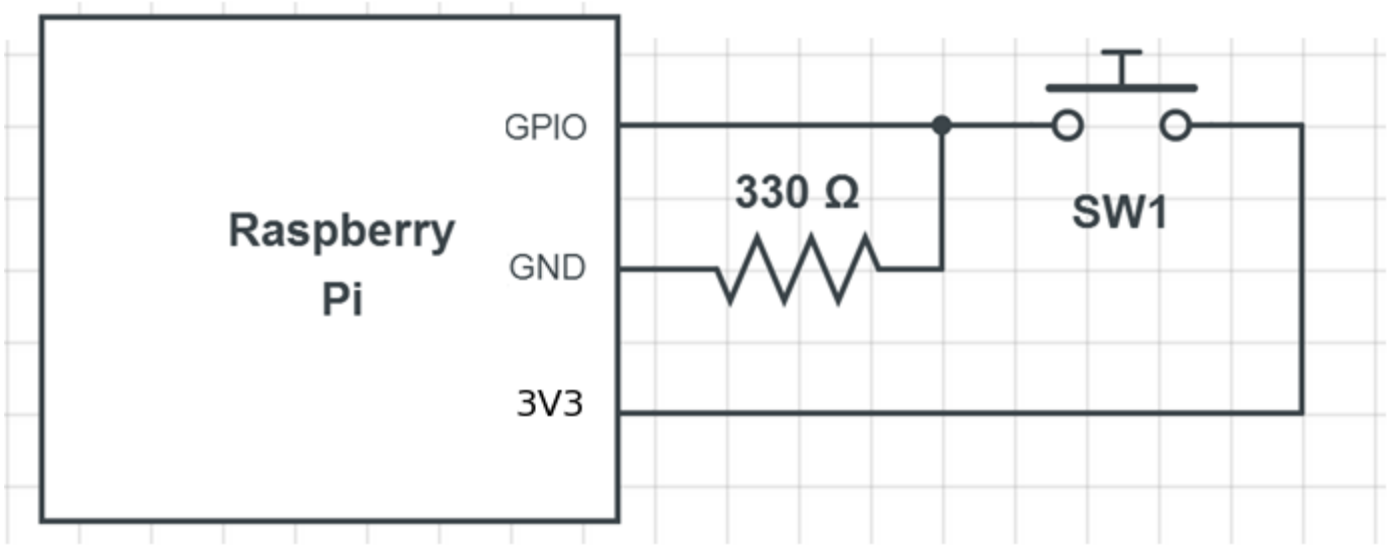

GND
GPIOs

Interfacing to I/P GPIO



- Now we program the GPIO as an input pin
- If the switch is pressed, a logic 0 is sent to the GPIO
- But if the switch is left open, what is the input to the GPIO pin?
- We need to use a ***pull up resistor***

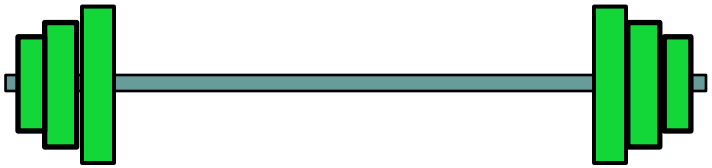
Interfacing to I/P GPIO



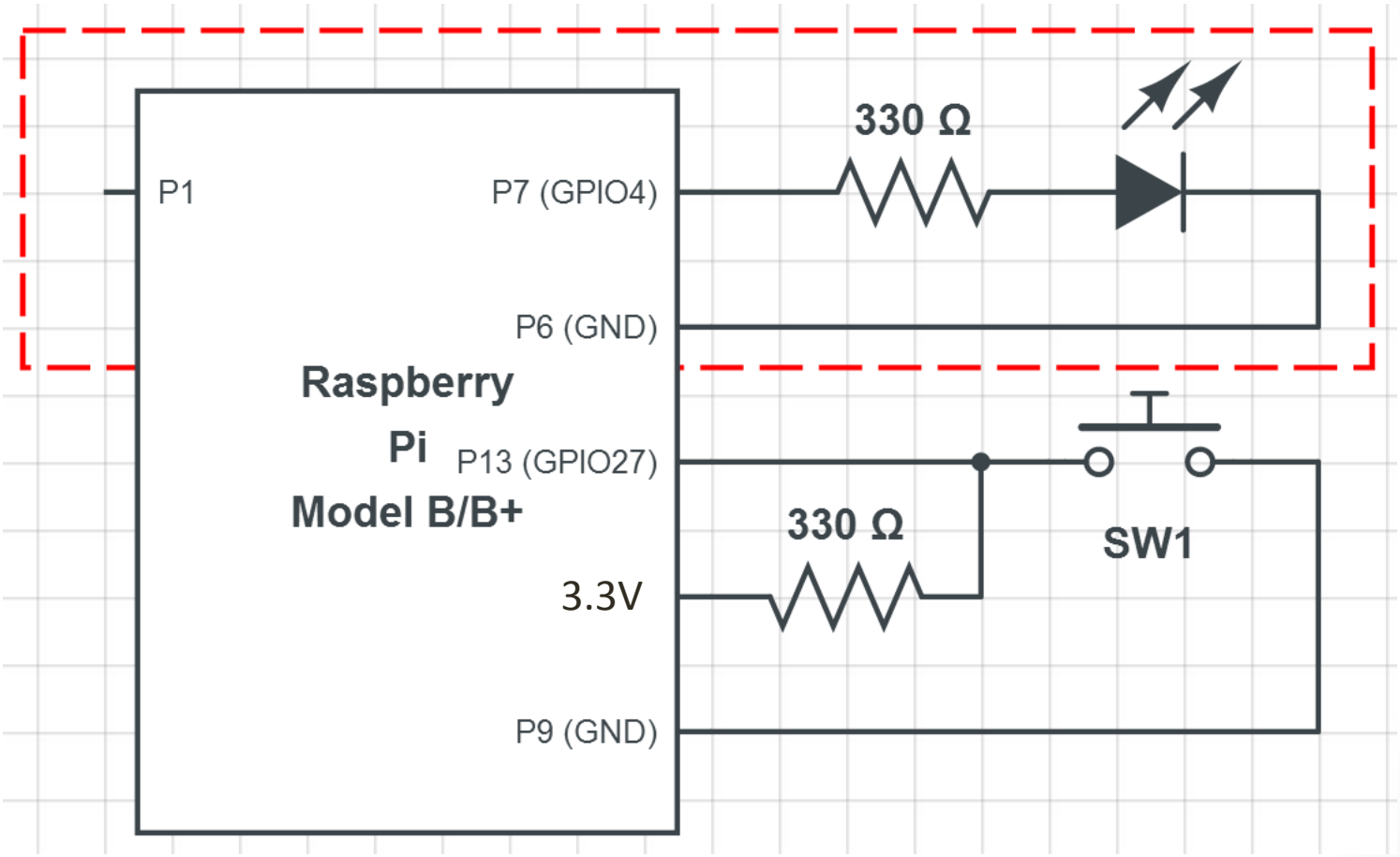
- Now we program the GPIO as an input pin
- If the switch is pressed, a logic 0 is sent to the GPIO
- But if the switch is left open, what is the input to the GPIO pin?
- We need to use a ***pull up resistor***
- We can reverse the role of the switch (input signal is high when pressed, and low when not pressed)
- In this case the resistor is called ***pull down resistor***
- *Note that the raspberry pi has internal pull up and pull down resistors for all input GPIOs, and they can be enabled/disabled through programming*

Using Pull Up Resistor

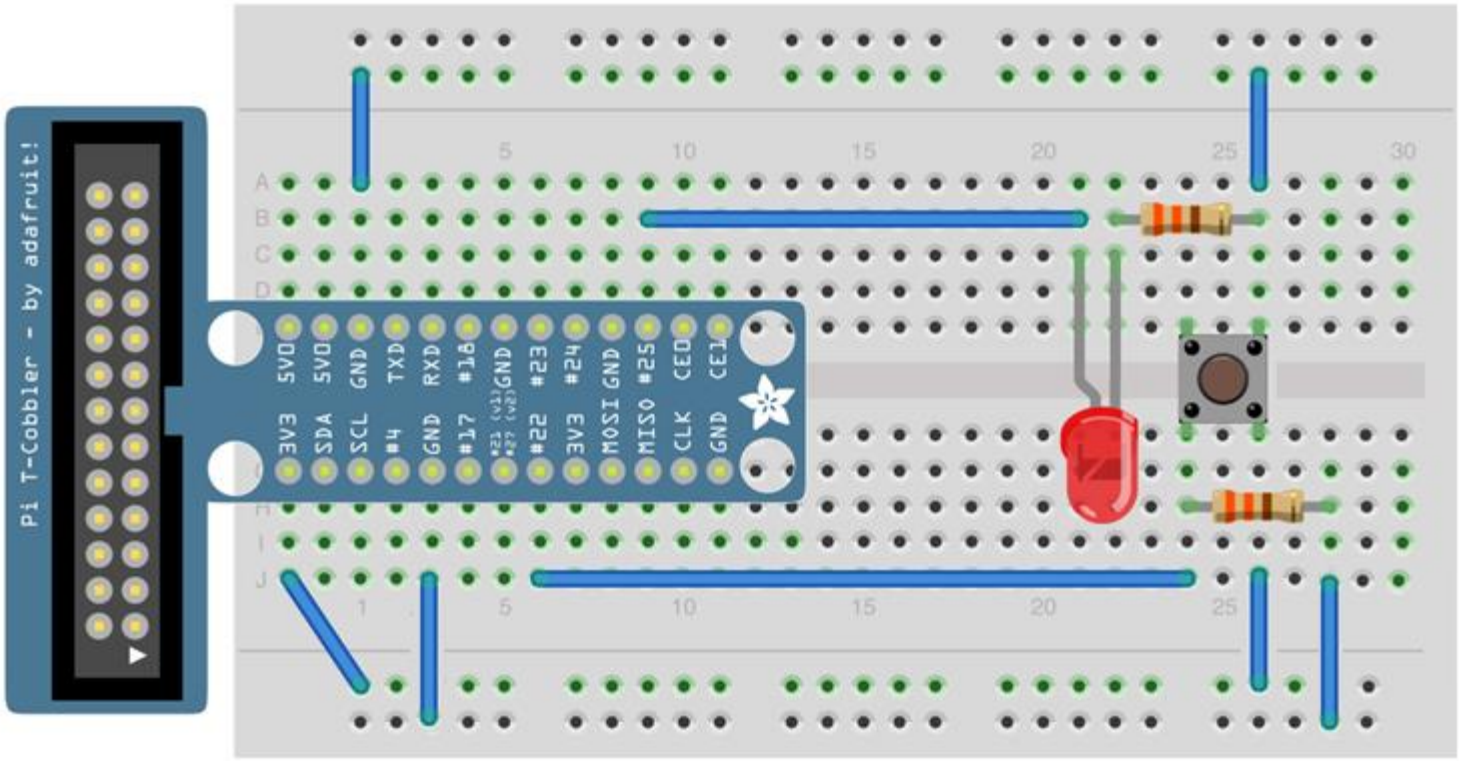
Using Pull down Resistor



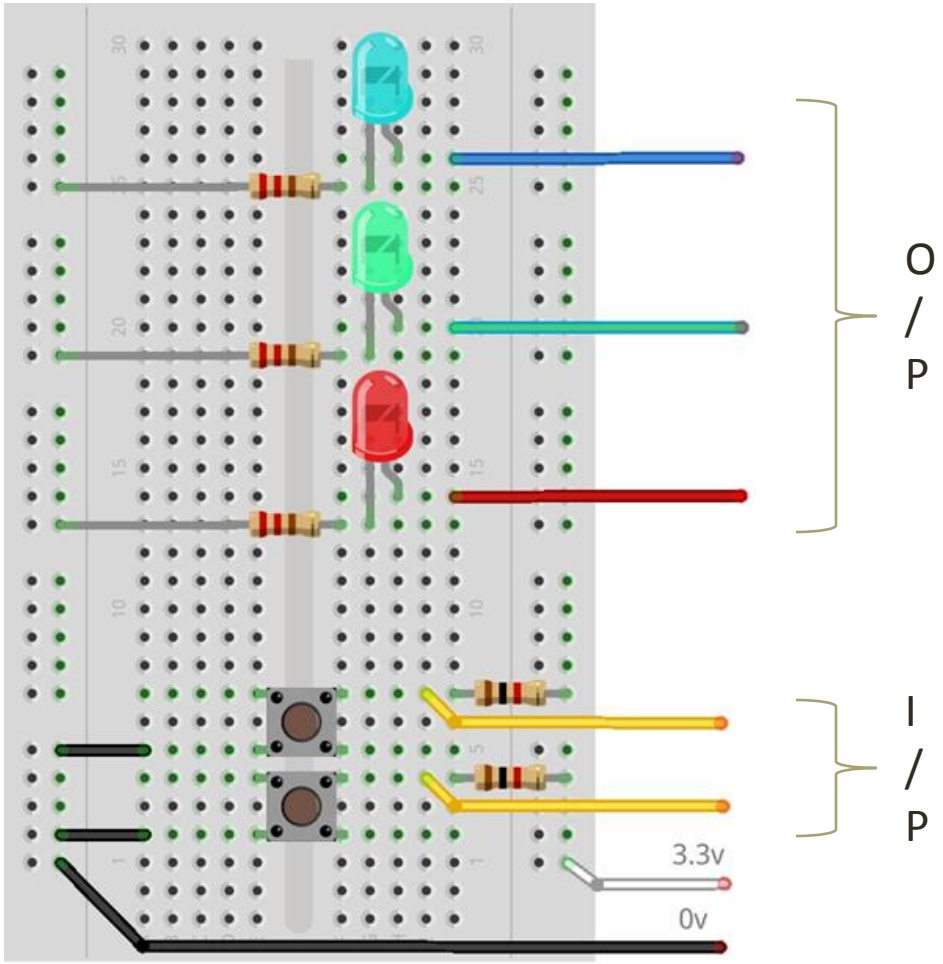
Both I/P & O/P GPIOs



Both I/P & O/P GPIOs



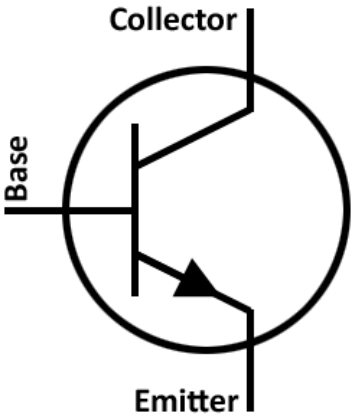
Both I/P & O/P GPIOs

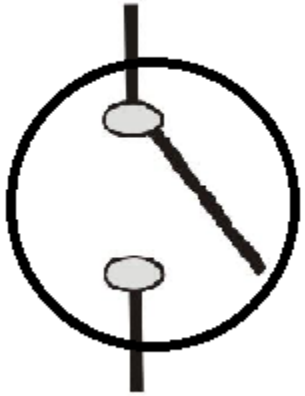


Handling Bigger loads

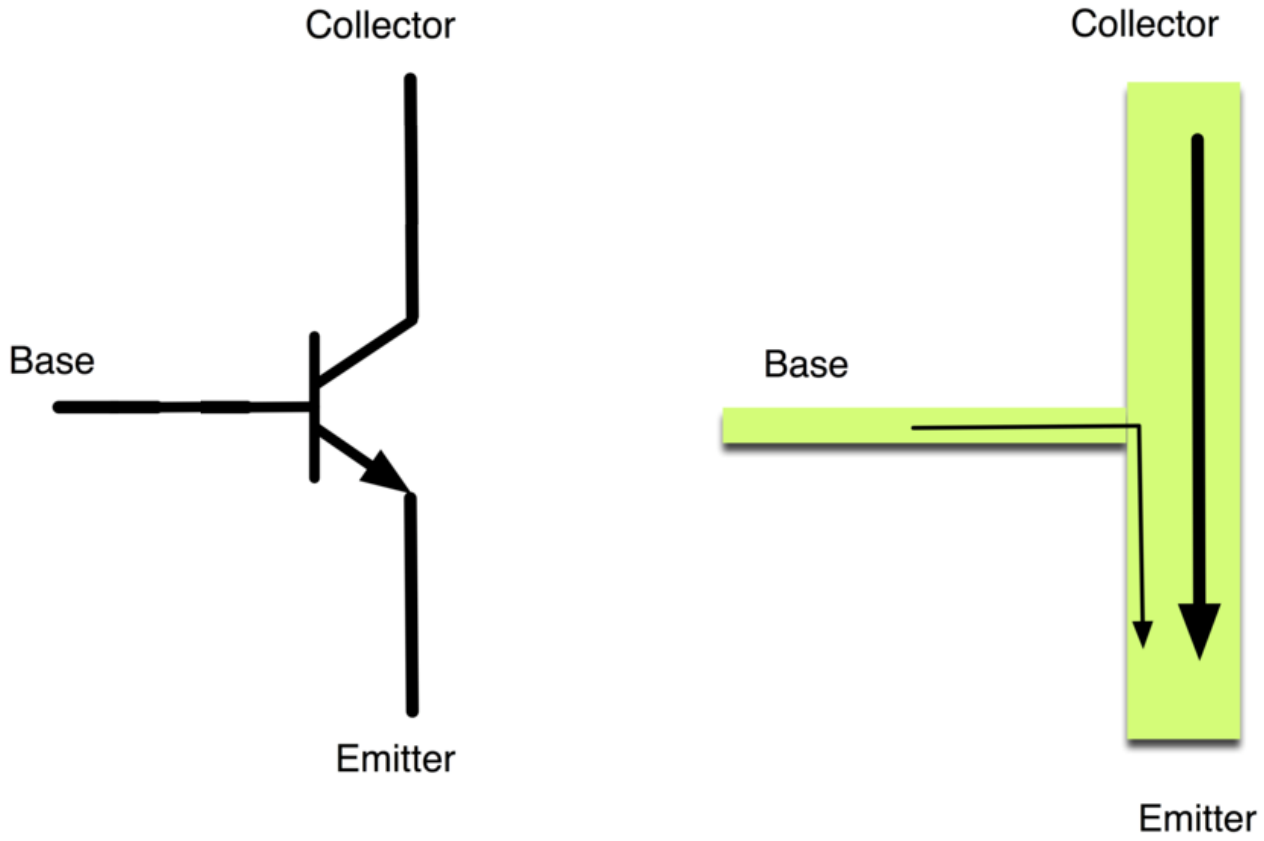
- The Pi GPIO can drive a max of **50 mA**
- This is suitable for simple logic circuits or a LED
- However this is not enough to drive bigger loads like a motor
- Also, some loads needs higher voltage than 3.3V
- For all of these cases, connecting the Pi GPIO pin directly to the load will not work, and may damage the Pi
- The solution for these situations is to use a ***transistor***
- The transistor can be used in different modes, but in this case it is used as a switch controller by the GPIO pin
- The GPIO pin acts a controller for that switch while the transistor provides the load with the higher current/voltage

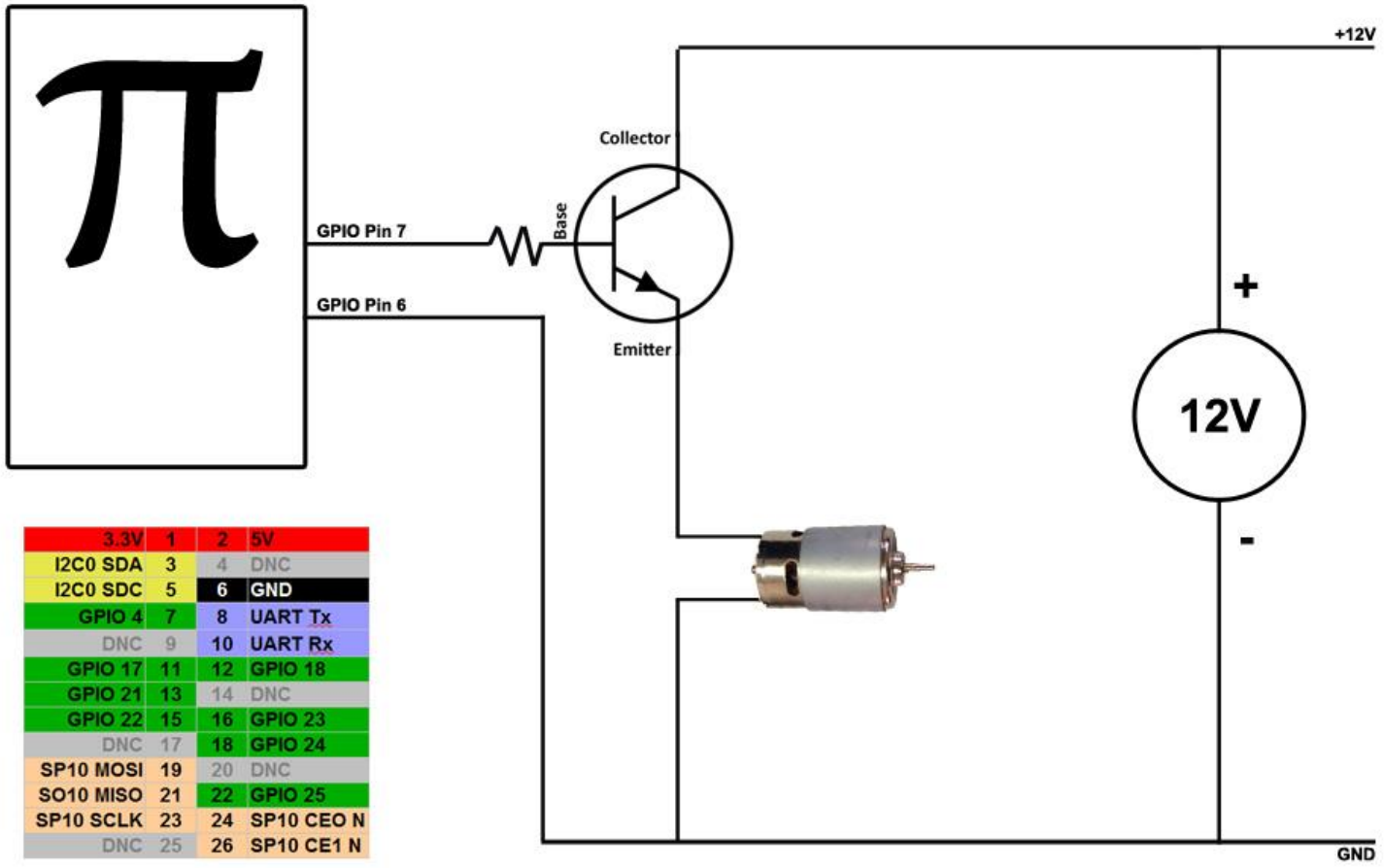
Using a Transistor





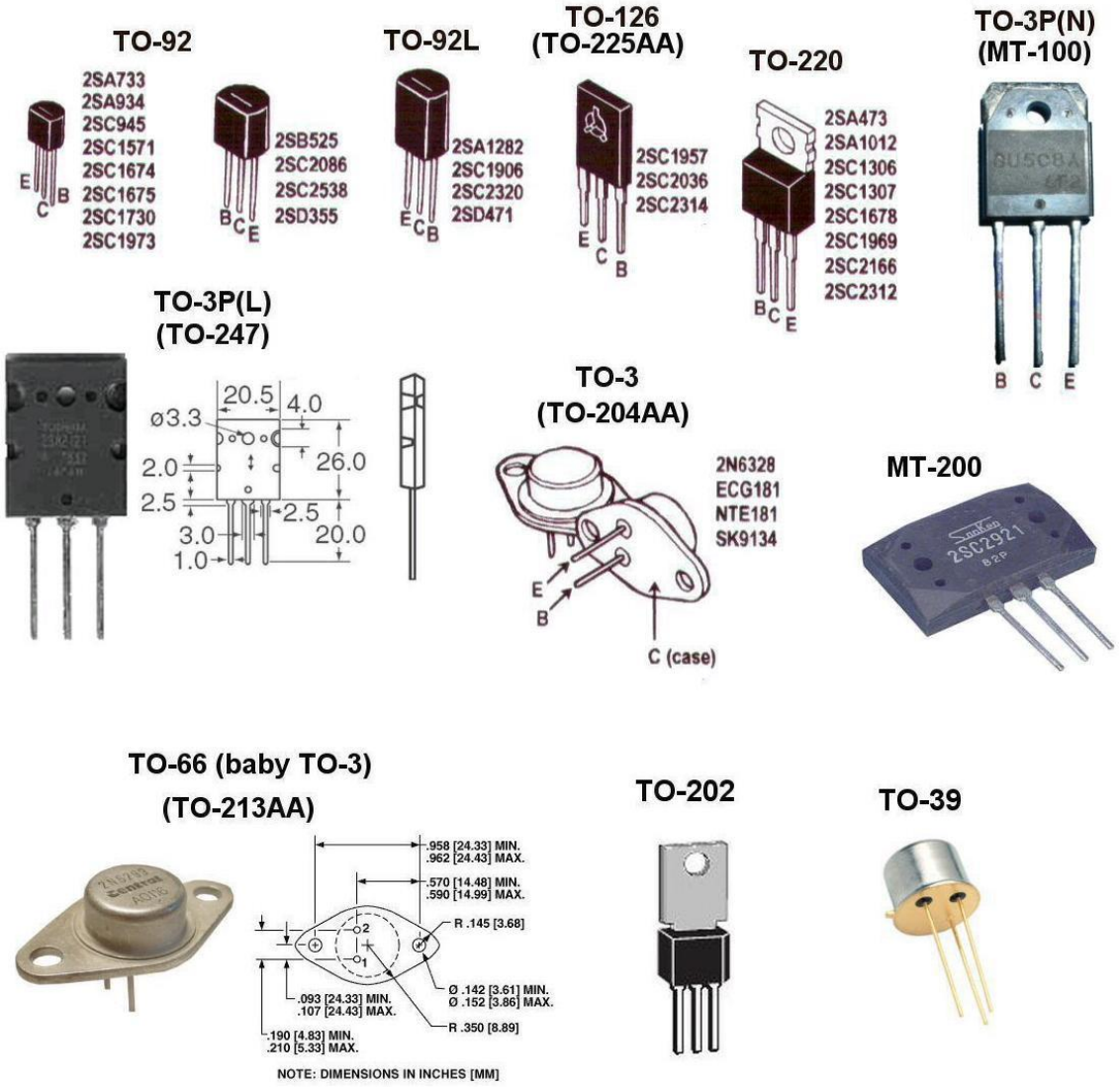
Using a Transistor



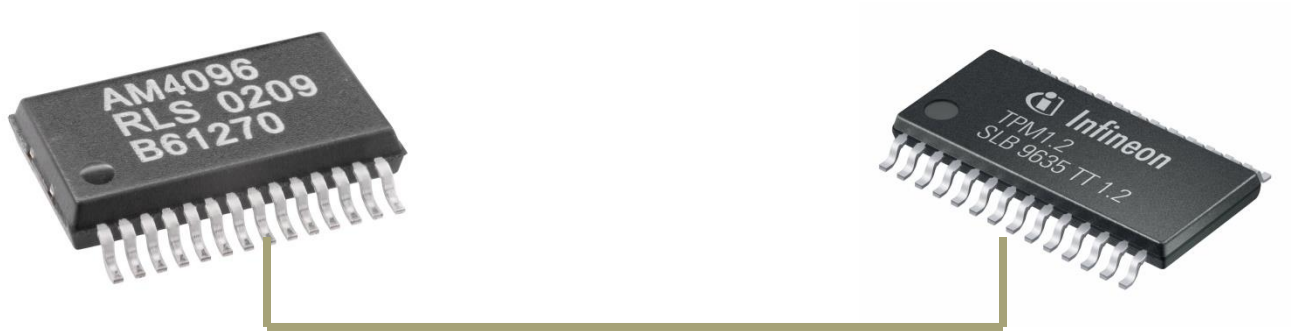


3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SDC	5	6	GND
GPIO 4	7	8	UART Tx
DNC	9	10	UART Rx
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SO10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CEO N
DNC	25	26	SP10 CE1 N

Transistor Form Factor

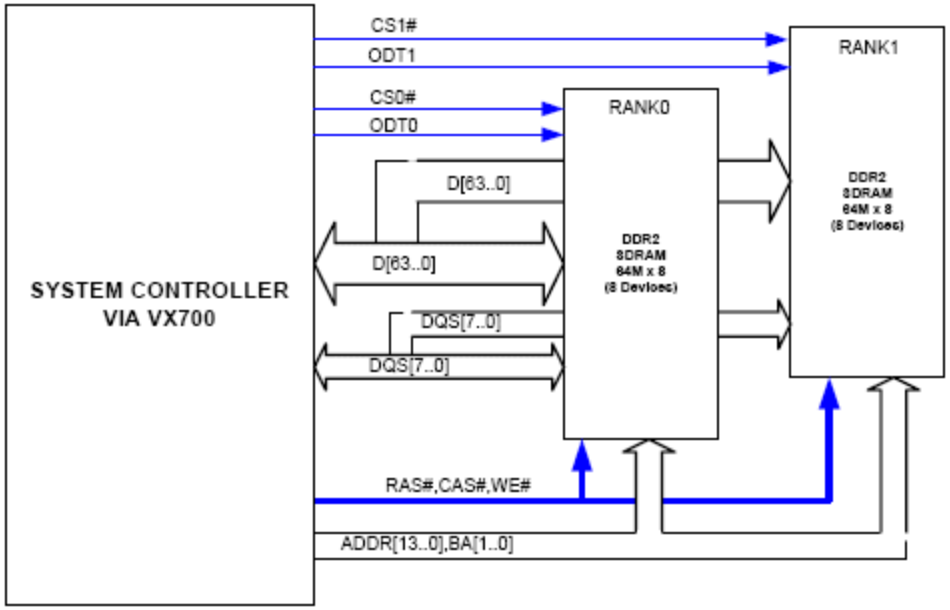


Connection Types
Point-to-Point



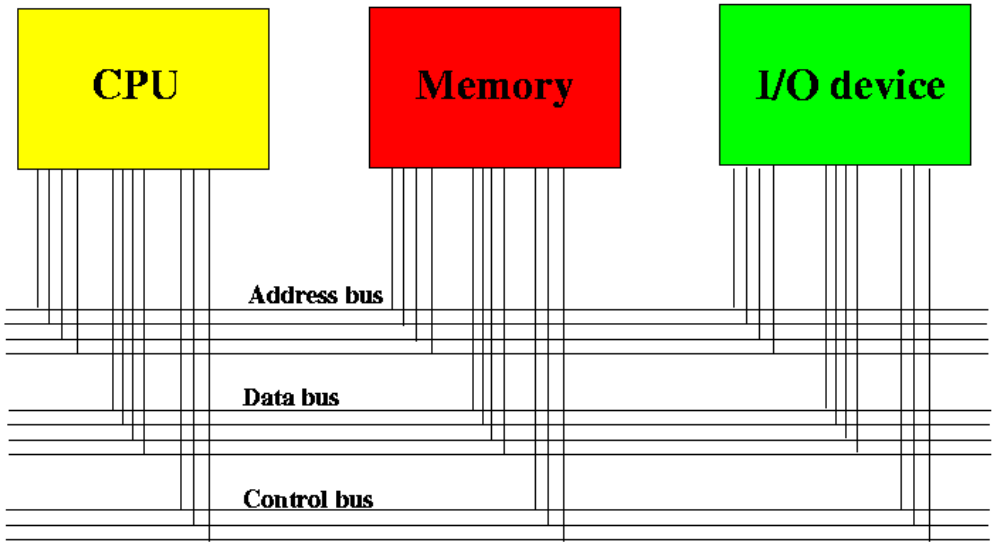
- In point to point connections,
  - One side is the **output** and the other side is the **input**
  - In some cases, the two sides may change roles
  - But we always have, one input and one output

Connection Types
Bus Connection



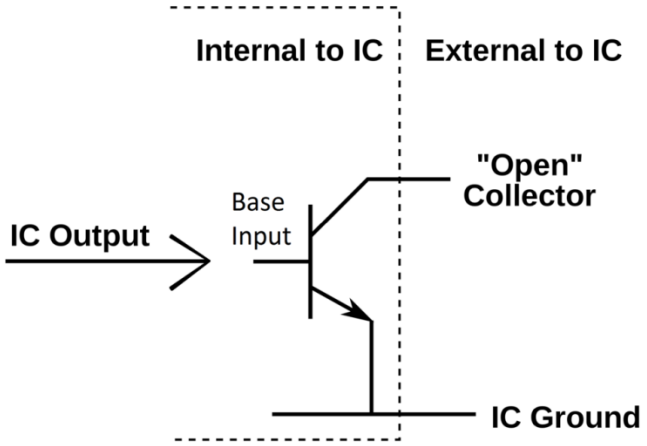
- In Bus connection,
  - Multiple entities connected to the same line
  - At any point in time, only one chip is enabled as output, and at least one entity enabled as input
  - We can not have multiple output signals connected together
  - We need chip select signals to tell which chip should enable its output, and which one should disable it

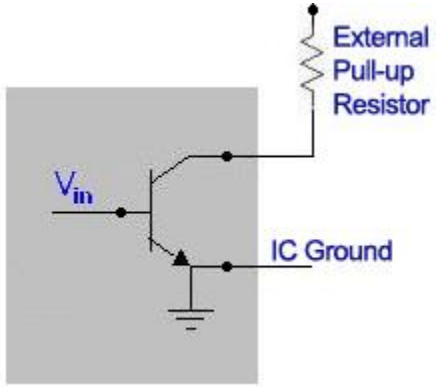
Bus Connection



- This means, that for an output signal to work on a bus it should be able to have 3 states
  - High/Low
  - Not Connected (High Impedance)
- An output is put in High Impedance, when the chip is not selected, and the output is driven by a different chip

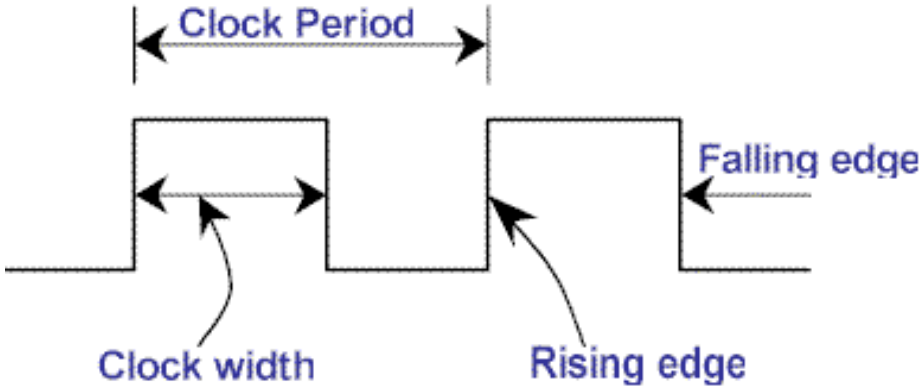
Open Drain/Collector Output





- Open Drain (or Open Collector) Output can be in one of two states,
  - Low
  - High Impedance (non-connected)
- This means that, when it is in non-connected state, it should be interpreted as high
- This means we need a pull up resistor to force that

Clock Signal



- A lot of times we need a clock signal to synchronize different components in the system (to add a rhythm)
- Sometimes actions are triggered by the rising edge of the clock
- Other times actions are triggered by the falling edge
- Sometimes some action is triggered by the rising edge, while others are triggered by the falling edge, to make sure that they don't overlap
- The frequency of the clock defines the speed of the circuit (number of actions that can be performed per second)

Linux 4 Embedded Systems

http://Linux4EmbeddedSystems.com