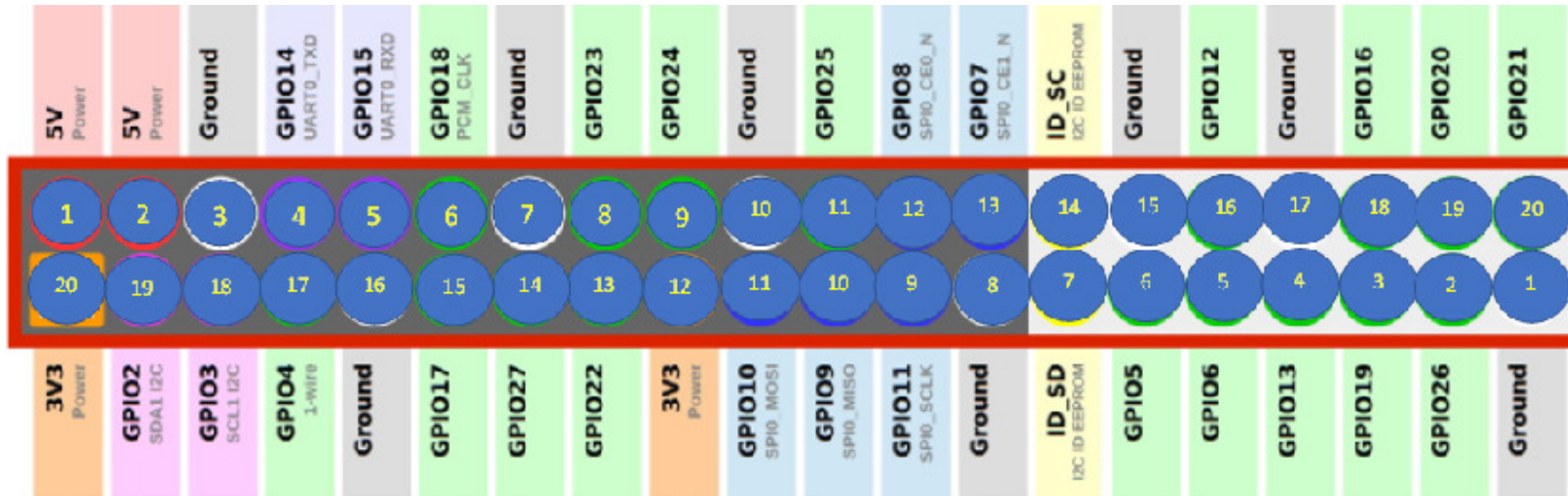


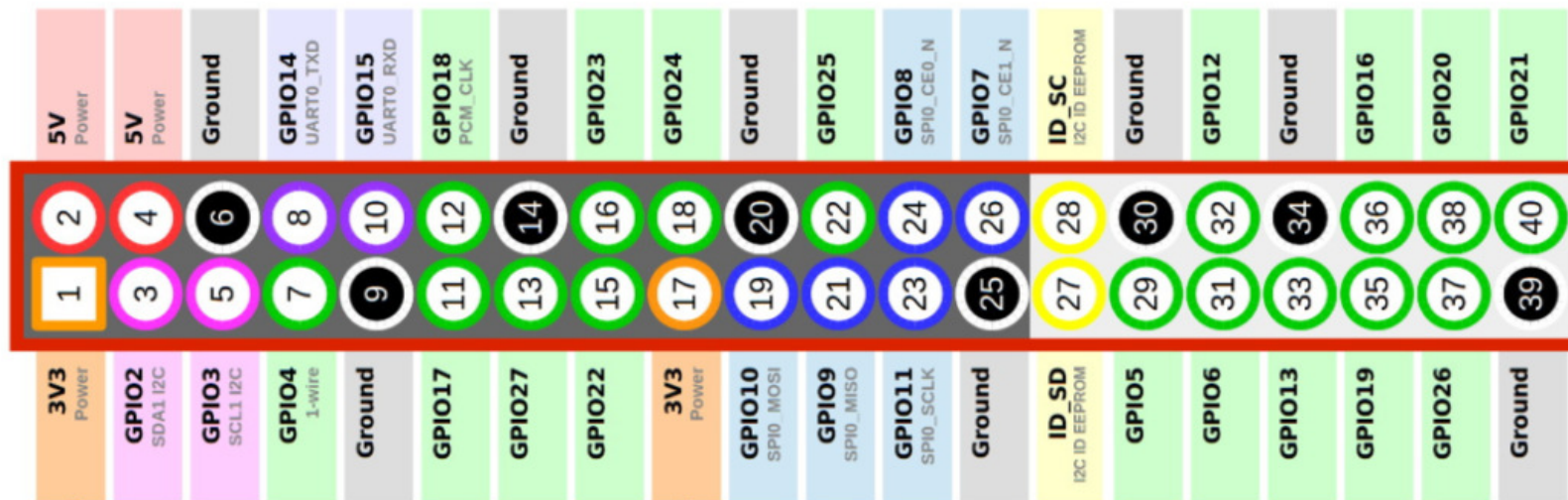
RECAP OF LESSON 1

RASPBERRY PI PIN DIAGRAM

WE ARE USING **BROADCOM (BCM)** PIN NUMBERING SYSTEM FOR OUR LESSONS. THE DEFAULT FOR **gpiozero** Library



BOARD – PIN NUMBERING SYSTEM using **RPi.GPIO** Library



<code>from gpiozero import LED</code>	We are “borrowing” as module LED from the library called gpiozero
<code>led_red=LED(14)</code>	We give our red coloured LED a name called led_red
<code>led_red.on()</code>	Turn on LED. Set Pin 14 High
<code>led_red.off()</code>	Turn off LED. Set Pin 14 Low
<code>led_blink(on_time=.5, off_time=.5, n=5)</code>	Blink LED five times(n=5) Each on_time is half a second. Each off time is half a second

We did this in the shell (**REPL**) of Thonny.
Program here is not permanent.

What is a **REPL**?

A REPL (say it, “REP-UL”) is an interactive way to talk to your computer in Python. To make this work, the computer does four things:

- Read the user input (your Python commands).
- Evaluate your code (to work out what you mean).
- Print any results (so you can see the computer’s response).
- Loop back to step 1 (to continue the conversation).

PATROLCAR Example

patrolcar.py	patrolcar2.py
<pre>from gpiozero import LED from time import sleep red_led=LED(14) red_led.off() blue_led=LED(12) blue_led.off() while True: red_led.blink(on_time=.1 , off_time=.1, n=5) sleep(1) blue_led.blink(on_time=.1, off_time=.1,n=5) sleep(1)</pre>	<pre>from gpiozero import LED from time import sleep red_led=LED(14) red_led.off() blue_led=LED(12) blue_led.off() def flash(): red_led.blink(on_time=.1 , off_time=.1, n=5) sleep(1) blue_led.blink(on_time=.1, off_time=.1,n=5) sleep(1) while True: flash()</pre>
	<pre>*** def flash(): is called a python function</pre>

Notes repository for this course

https://github.com/ssgoh/Intro_To_Raspberry_Pi