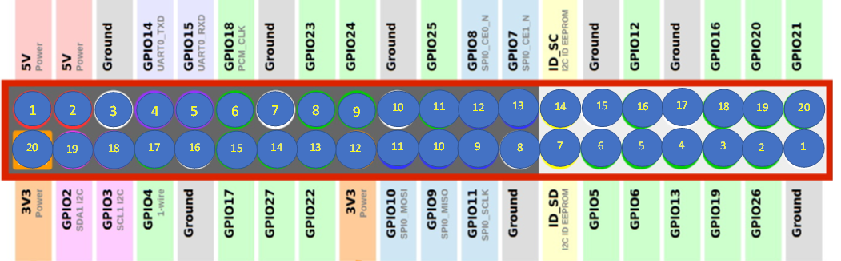
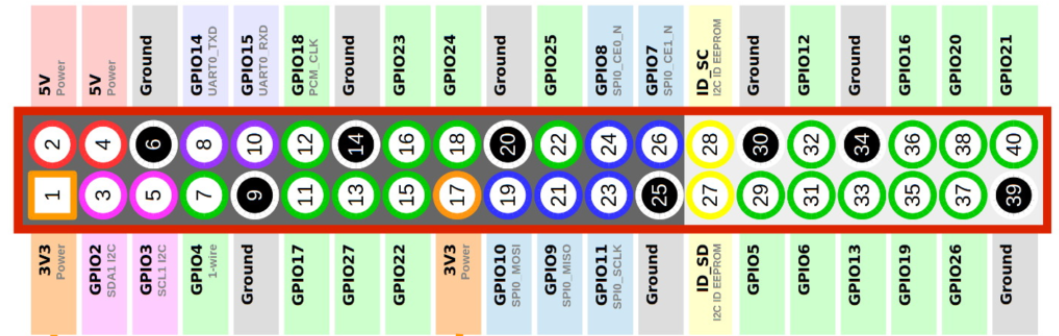
**RECAP OF LESSON 1**

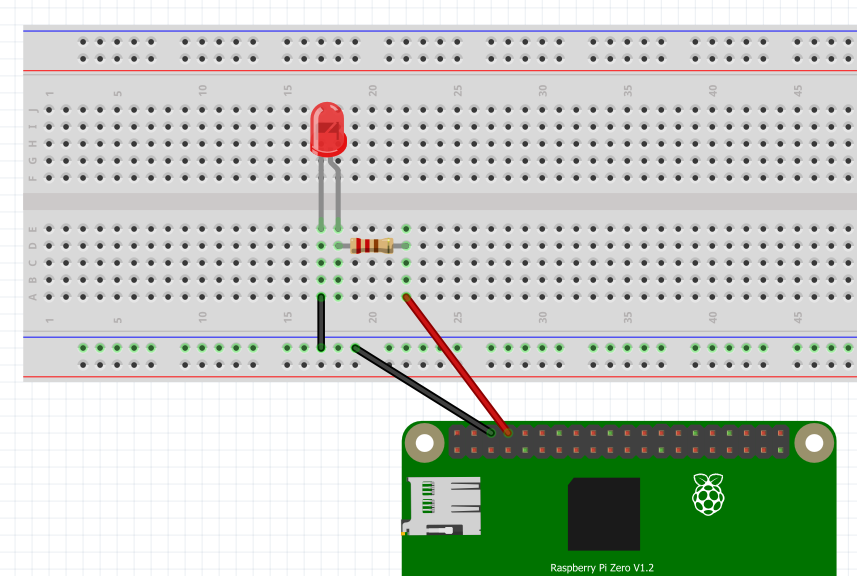
**RASPBERRY PI PIN DIAGRAM**

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



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





|  |  |
| --- | --- |
| from gpiozero import LED | We are “borrowing” as module LED from the library called gpiozero |
| led\_red=LED(14) | We give our red coloured LED a name called led\_red |
| led\_red.on() | Turn on LED. Set Pin 14 High |
| led\_red.off() | Turn off LED. Set Pin 14 Low |
| led\_blink(on\_time=.5, off\_time=.5, n=5) | Blink LED five times(n=5)  Each on\_time is half a second.  Each off time is half a second |

We did this is 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:

**R**ead the user input (your Python commands).

**E**valuate your code (to work out what you mean).

**P**rint any results (so you can see the computer’s response).

**L**oop back to step 1 (to continue the conversation).

**Pin 14**

**PATROLCAR Example**

|  |  |
| --- | --- |
| **patrolcar.py** | **patrolcar2.py** |
| 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) | 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()** |
|  | \*\*\*  **def flash():** is called a **python function** |