

Practical 2:

Displaying different LED patterns with Raspberry Pi.

Components Required:

Here we are using Raspberry Pi 2 Model B with Raspbian OS. All the basic Hardware and Software requirements are previously discussed, you can look it up in the Raspberry Pi Introduction, other than that we need:

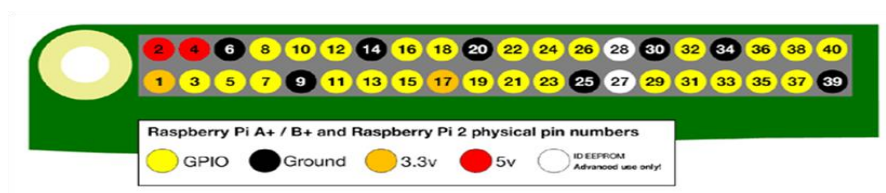
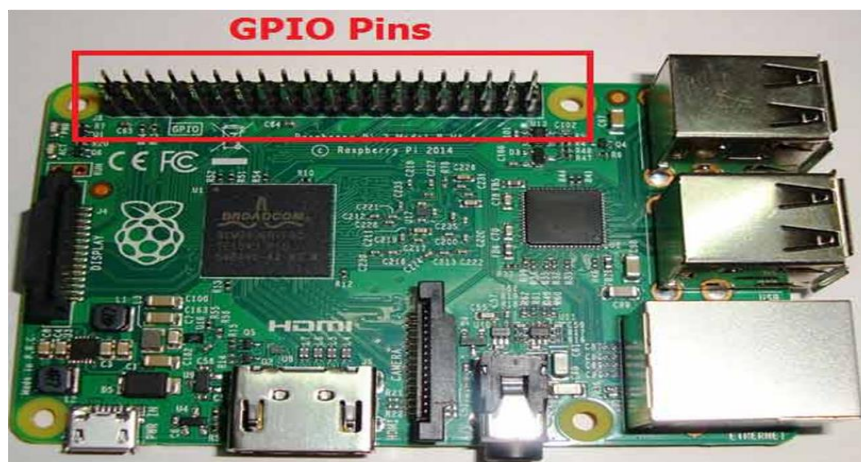
Connecting pins

220Ω or 1KΩ resistor

LED

Bread Board

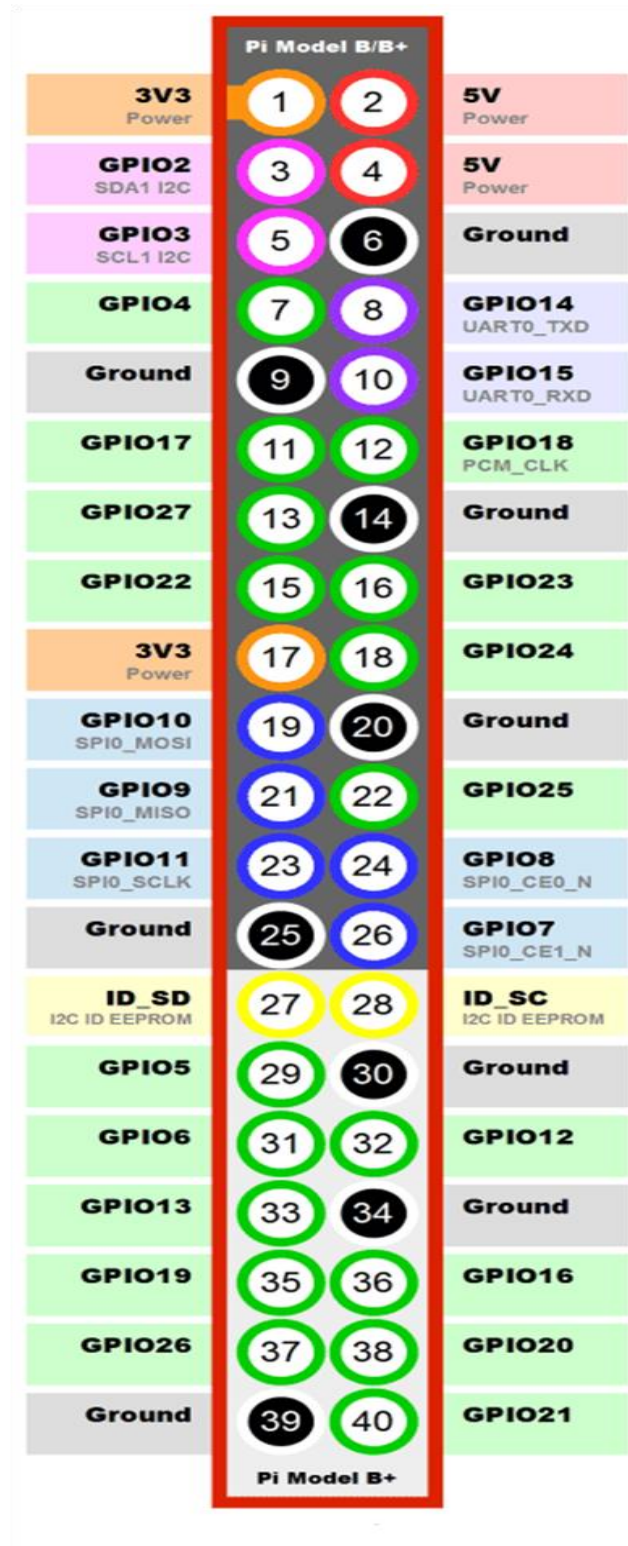
We will start with Blink LED using Raspberry Pi. Blinky is done by connecting an LED to one of GPIO pins of PI and turning it ON and OFF.

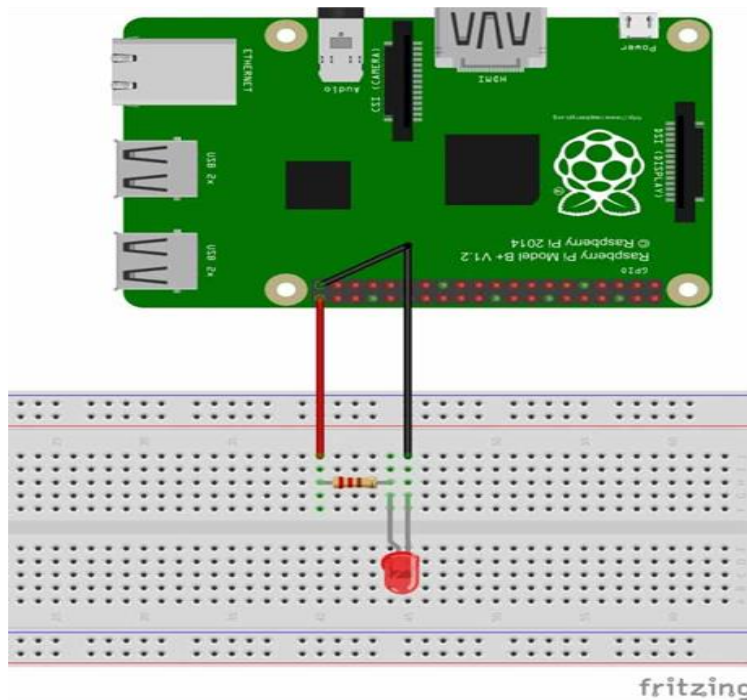


As shown in above figure, there are 40 output pins for the PI. But when you look at the second figure, you can see not all 40 pin out can be programmed to our use. These are only 26 GPIO pins which can be programmed. These pins go from GPIO2 to GPIO27.

These 26 GPIO pins can be programmed as per need. Some of these pins also perform some special functions, we will discuss about that later. With special GPIO put aside, we have 17 GPIO remaining (Light green Circle).

Each of these 17 GPIO pins can deliver a maximum of 15mA current. And the sum of currents from all GPIO cannot exceed 50mA. So we can draw a maximum of 3mA in average from each of these GPIO pins. So one should not tamper with these things unless you know what you are doing.



Circuit Explanation:

As shown in the circuit diagram we are going to connect an LED between PIN40 (GPIO21) and PIN39 (GROUND). As said earlier, we cannot draw more than 15mA from any one of these pins, so to limit the current we are connecting a 220 Ω or 1K Ω resistor in series with the LED.

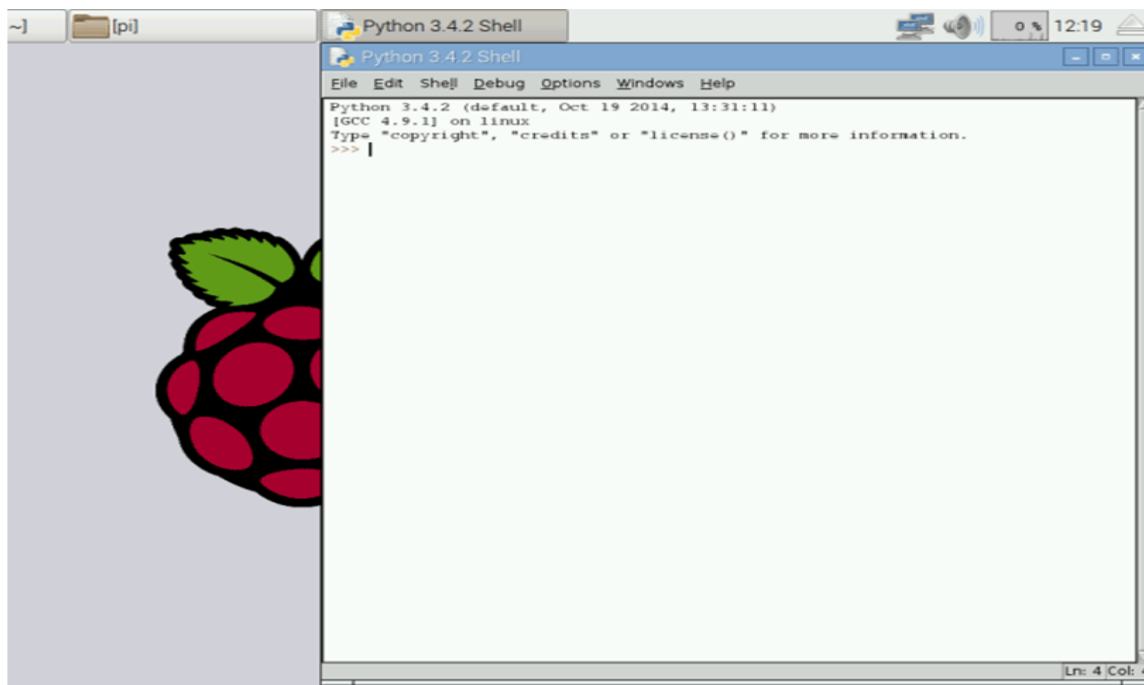
Working Explanation:

Since we have everything ready, turn ON your PI and go to the desktop.

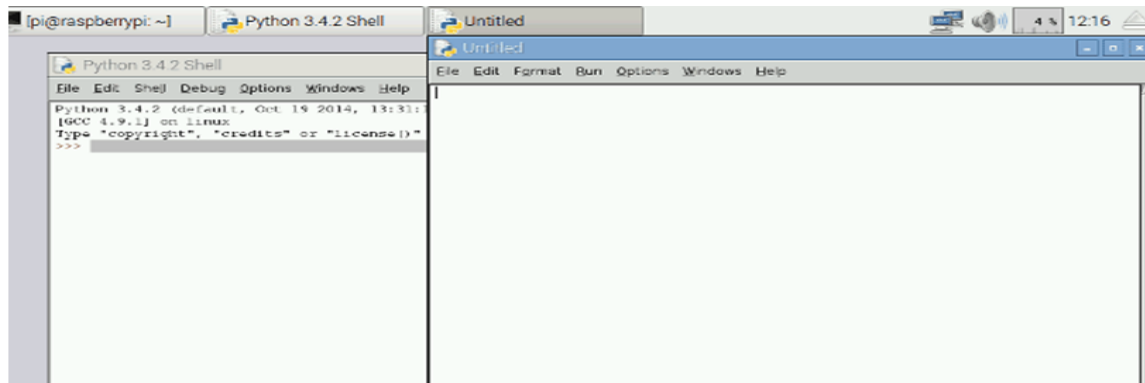
1. On the desktop, go the Start Menu and choose for the PYTHON 3, as shown in figure below.



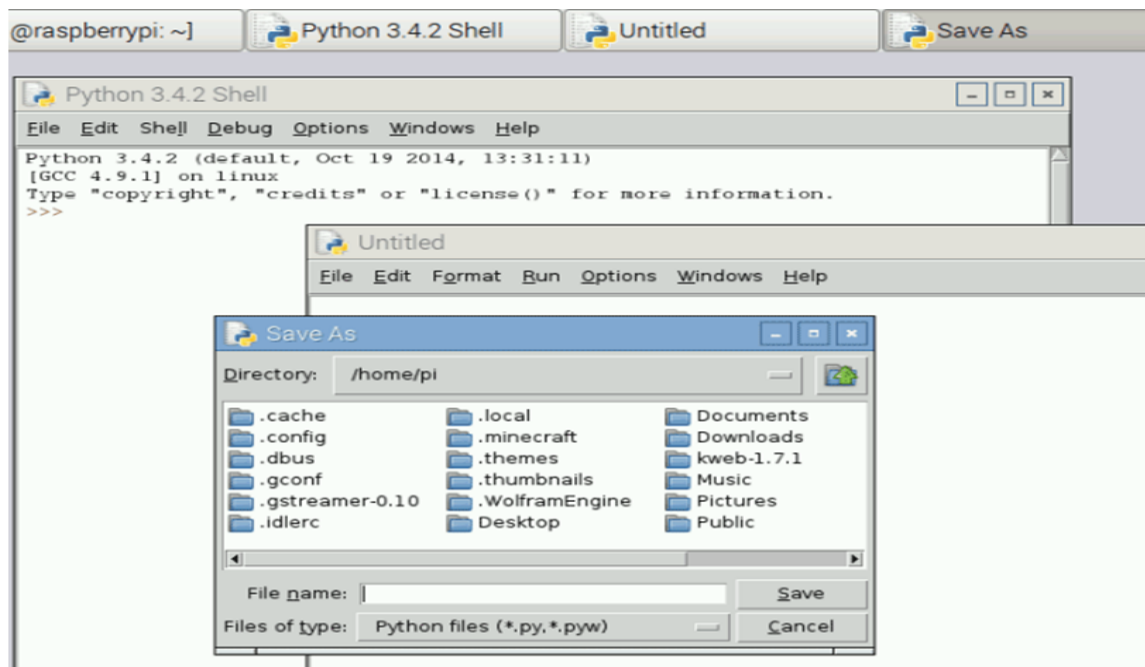
2. After that, PYTHON will run and you will see a window as shown in below figure.



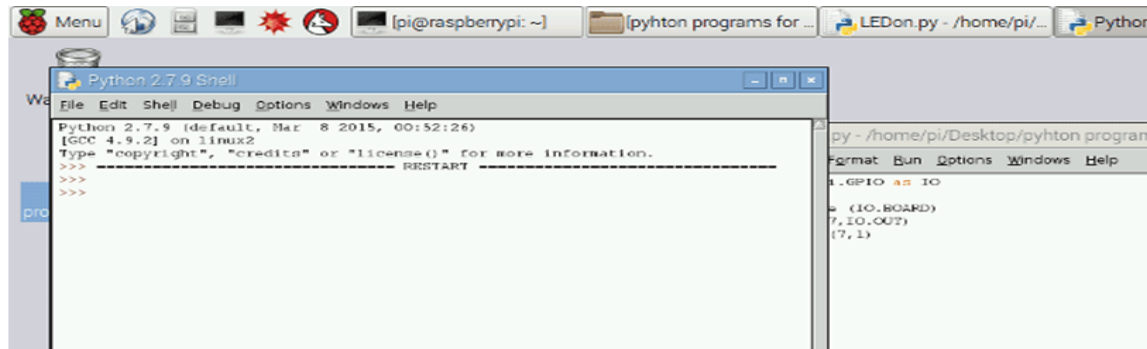
3. After that, click on New File in File Menu, You will see a new Window open,



4. Save this file as blinky on the desktop,



5. After that write the program for blinky as given below and execute the program by clicking on “RUN” on ‘DEBUG’ option.



If the program has no errors in it, you will see a “>>>”, which means the program is executed successfully. By this time you should see the LED blinking three times. If there were any errors in the program, the execution tells to correct it. Once the error is corrected execute the program again.

We will see the PYTHON program Code for LED Blinking, in detail, below.

Code

```
import RPi.GPIO as IO          # calling header file for GPIO's of PI

import time                    # calling for time to provide delays in program

IO.setmode (IO.BOARD)         # programming the GPIO by BOARD pin numbers, GPIO21 is called as
PIN40                           PIN40

IO.setup(40,IO.OUT)            # initialize digital pin40 as an output.

IO.output(40,1)                # turn the LED on (making the voltage level HIGH)

time.sleep(1)                  # sleep for a second
```

```
IO.cleanup()                # turn the LED off (making all the output pins LOW)
time.sleep(1)               #sleep for a second
```

#loop is executed second time

```
IO.setmode (IO.BOARD)
IO.setup(40,IO.OUT)
IO.output(40,1)
time.sleep(1)
IO.cleanup()
time.sleep(1)
```

#loop is executed third time

```
IO.setmode (IO.BOARD)
IO.setup(40,IO.OUT)
IO.output(40,1)
time.sleep(1)
IO.cleanup()
time.sleep(1)
```