



TRAFFIC SIGNAL AS A USE CASE OF PYTHON WITH RASPBERRY PI

DR. NANASAHEB M. HALGARE

Asst. Professor

M. S. Bidve Engineering College, Latur.

(MS) INDIA

ABSTRACT

As we know that there are many resources and technologies are launched now a days in computer society like R+,Jquery ,etc and all these languages are execute on Computers as well as but it is quiet hard to maintain the computer and laptop for the students because of its higher weight ,larger size, high cost and to overcome all these factors Creator Eben Upton's launch a device which is comparatively small in size ,low in cost ,and lesser in weight callas Raspberry pi.in this paper we are dealing with what is raspberi pi and how python programming works with raspberry pi.

General Terms: language,LED,

Keywords: Raspberry pi ,Python.

1 INTRODUCTION

A Raspberry Pi is a debit card-sized computer newly designed for education, encouraged by the 1981 BBC Micro. Creator Eben Upton's aim was to launch a low-cost device that would enhance programming skills and hardware understanding at the pre-university level. But thanks to its small size and affordable price, it was rapidly accepted by tinkerers, makers, and electronics enthusiasts for projects that need more than a basic microcontroller (such as Arduino devices).The Raspberry Pi is slower than a modern laptop or desktop but is still a complete Linux computer and can accommodate all the expected capabilities that implies, at a dipressed-power consumption level.

Python is an easy to learn, competent programming language. It has efficient high-level data structures and a simple but powerful approach to object-oriented programming. Python's appropriate syntax and dynamic typing, together with its explained nature, make it an perfect language for scripting and ready application development in many areas on most platforms.

The Python interpreter is easily continued with advanced functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for designing applications.

This paper encompasses

1. What is raspberry pi and python
2. Example that show how can python can work on raspberry pi .

2. Raspberry Pi open hardware

The Raspberry Pi is nothing but a open hardware, with the passing over of the primary chip on the Raspberry Pi, the Broadcom SoC (System on a Chip), which executes many components of the board—CPU, like graphics, memory, the USB controller, etc. Number of the projects built with a Raspberry Pi are open and well-documented as well and are things you can build and modify easily.

3. What are the differences in Raspberry Pi models?

The Raspberry Pi launched recently a new model, the Raspberry Pi 2, which replaces some of the previous boards, even if the earlier boards will still be formed as long as there is a call for them. It is generally rearward adaptable with previous versions of the board, so any projects assignment you see which were built for a earlier version of the board should still work.

There are a two Raspberry Pi models, the A and the B, named after the aforementioned BBC Micro, which was also released in a Model A and a Model B. The A comes with 256MB of RAM and one USB port. It is cheaper and uses less power than the B. The current model B comes with a second USB port, an Ethernet port for connection to a network, and 512MB of RAM. The Raspberry Pi A and B boards been upgraded to the A+ and B+ respectively.

4. What kind of operating system does the Raspberry Pi run?

Basically The Raspberry Pi was constructed for the Linux operating system, and many Linux dispersion have a version optimized for the Raspberry Pi.

There are Two options first one is Raspbian, which is based on the Debian operating system, and second one is Pidora, which is based on the Fedora operating system. For newcomers, either of these two work well.

5. What is python language

Python is an explained, object-oriented, high-level programming language with dynamic meaning. Its high-level in built data structures, combined with dynamic typing and dynamic binding, make it very beautiful for fast Application Development, as well as for use as a scripting or glue language to bridge existing components together. Python's lucid, royal to learn syntax highlight creditability and therefore curtail the cost of program maintenance. Python guide modules and packages, which boost program structure and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Because of no compilation step, the edit-test-debug cycle is fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

6. Smart traffic system using Raspberry Pi and Python

To implement such system there is need to see working of GPIO

GPIO.BCM	Function	GPIO.BOARD	Function	GPIO.BCM
<50mA	3V3	1 2	5V	
BCM GPIO02	SDA1 ARM	3 4	5V	
BCM GPIO03	SCL1 ARM	5 6	GND	
BCM GPIO04		7 8	TX	BCM GPIO14
	GND	9 10	RX	BCM GPIO15
BCM GPIO17	SPI1 CE1	11 12	PWM0/SPI1 CE0	BCM GPIO18
BCM GPIO27		13 14	GND	
BCM GPIO22		15 16		BCM GPIO23
<50mA	3v3	17 18		BCM GPIO24
BCM GPIO10	SPI0 MOSI	19 20	GND	
BCM GPIO09	SPI0 MISO	21 22		BCM GPIO25
BCM GPIO11	SPI0 SCLK	23 24	SPI0 CE0	BCM GPIO08
	GND	25 26	SPI0 CE1	BCM GPIO07
BCM GPIO00	SDA0 VC	27 28	SCL0 VC	BCM GPIO01
BCM GPIO05		29 30	GND	
BCM GPIO06		31 32	PWM0	BCM GPIO 12
BCM GPIO13	PWM1	33 34	GND	
BCM GPIO19	SPI1 MISO/PWM1	35 36	SPI1 CE2	BCM GPIO16
BCM GPIO26		37 38	SPI1 MOSI	BCM GPIO20
	GND	39 40	SPI1 SCLK	BCM GPIO21

Table 1

6.1 Prerequisite

6.1.1 Hardware

As well as a Raspberry Pi with an SD card and the usual peripherals, you'll also need:

- 1x Solderless breadboard
- All kind of jumper leads
- 1x Tactile button
- 3x LEDs (Red, Green and Yellow)
- 3x 330 ohm Resistor
- Buzzer
- Button

6.1.2 Software

Raspberry Pi require NOOBS OS. It is a way of making setting up a Raspberry Pi for the first time .

6.1.3 GPIO pins

One competent feature of the Raspberry Pi is the row of GPIO pins along the top edge of the board. GPIO stands for General-Purpose Input/Output. These pins are a physical interface between the Raspberry Pi and the outside world.

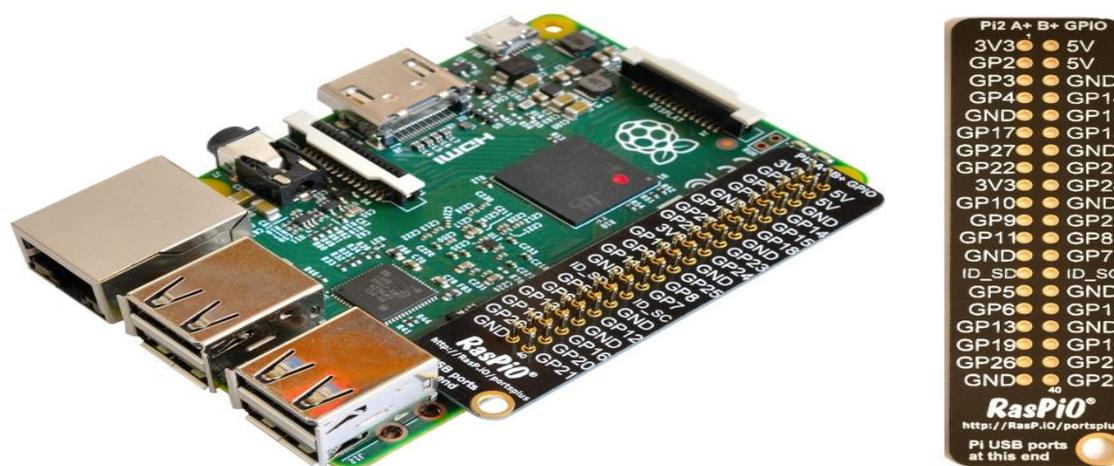


Fig 6.1

If don't have a pin label, then this can help you to identify the pin numbers

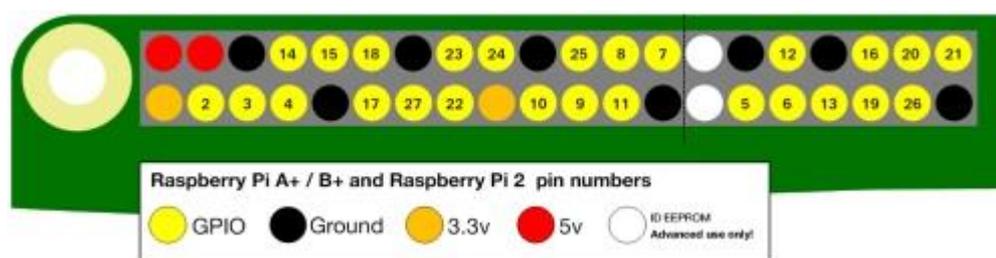


Fig 6.2

3.3 volts

Everything connected to these pins will always get 3.3 v of power.

5 volts

Everything connected to these pins will always get 5 v of power.

Gnd

Zero volts, used to complete a circuit

GPIO

These pins are used for general-purpose use and can be configured as input or output pins

ID_SC/ID_SD/DNC

Special purpose pins.

Lighting an LED

LEDs are delicate little things. if you put too much current through them they will pop. To limit the current going through the LED, you should always use a resistor in series with it.

Long leg an LED to the Pi 3.3v and the short leg to a GND pin. Automatically LED will turn ON

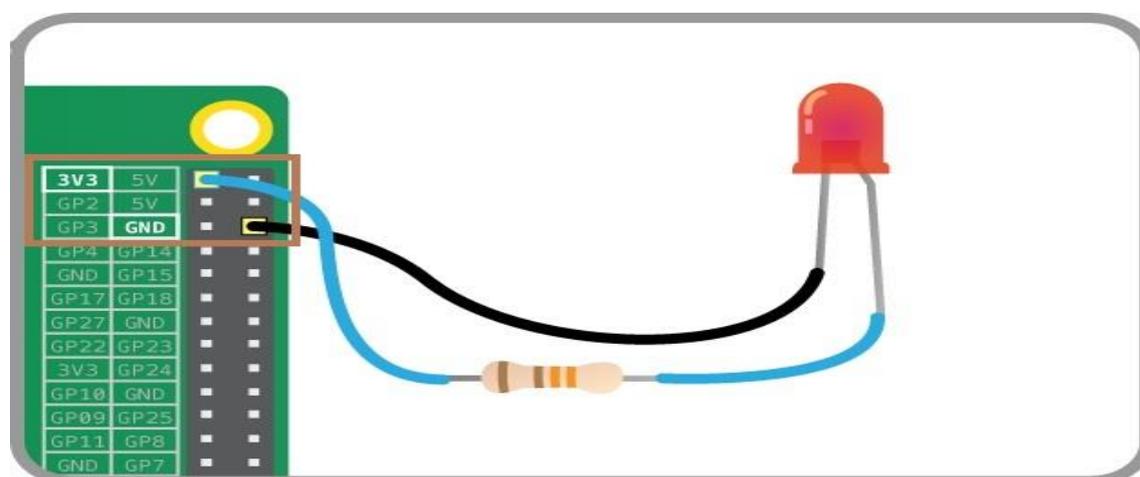


Fig 6.3

If you connect special purpose(GPIO) pin are connect to long leg, make sure to write the import code.

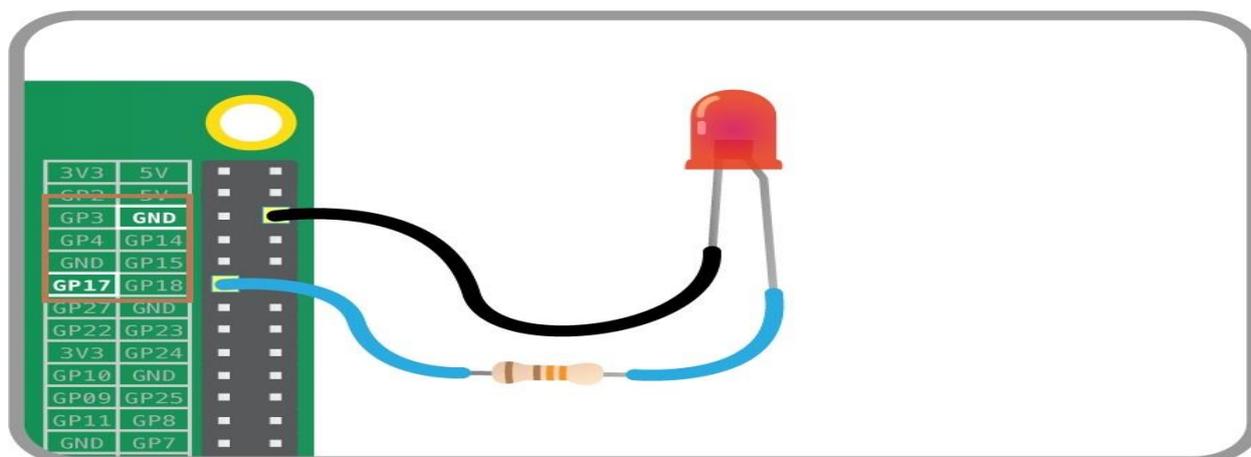


Fig 6.4

Switching an LED on and off

GPIO Zero is a new Python library which provides a simple interface to everyday GPIO component. It comes installed by default in Raspbian.

Open IDLE(Integrated Development Environment), which you can use to write and run code.

Raspbian Menu Icon >> Programming >> Python 3 (IDLE).



Fig 6.5

- To create a new file in IDLE, You can click on File and then New File in IDLE's menu bar.

- Create a new file by clicking File >> New File



Fig 6.6

- Save the new file by clicking File >> Save. Save the file as *traffic.py*
- You'll need the LED Class, and to tell it that the LED is on pin 17. Write the following code in your new file.
 1. `from gpiozero import LED`
 2. `led = LED(17)`
- To make the LED switch on, type the following and press Enter
 1. `led.on()`
- To make it switch off you can type
 1. `led.off()`

Your LED should switch on and then off again. But that's not all you can do. Similarly check the Buzzer and Button. Just import a Buzzer and Button for header file.

Making Traffic Light

We need a breadboard, three LEDs, a button, a buzzer and the necessary jumper cables and registers.

Wiring

First, we need to understand how each component is connected.

- A push button requires 1 ground pin and 1 GPIO pin
- An LED requires 1 ground pin and 1 GPIO pin, with a current limiting resistor
- A buzzer requires 1 ground pin and 1 GPIO pin

Place the components on the breadboard and connect them to the Raspberry Pi GPIO pins, according to the following diagram.

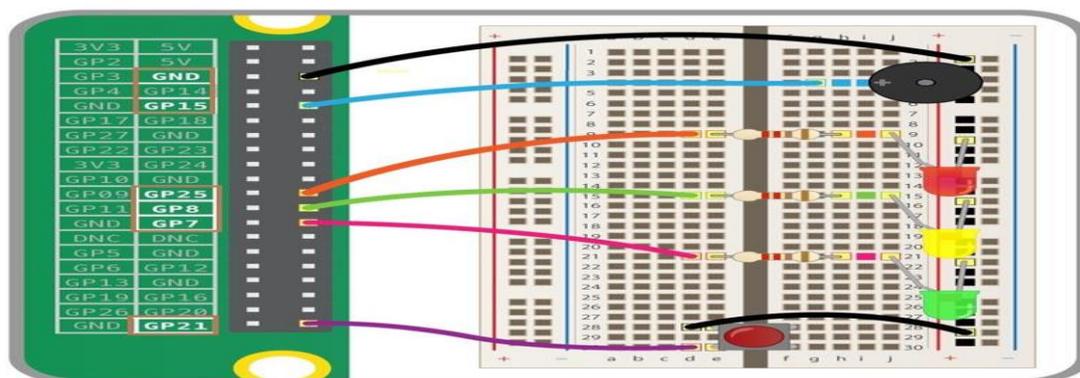


Fig 6.7

Component GPIO pin

Button	21
Red LED	25
Yellow LED	8
Green LED	7
Buzzer	15

This is the same as the Switching and LED on and off step

- Open Python 3 from the main menu
- Create a new file just save with the project name.py

Add TrafficLight, Button and Buzzer Code

1. from gpiozero import Button, TrafficLights, Buzzer
2. from time import sleep
- 3.
4. buzzer = Buzzer(15)
5. button = Button(21)
6. lights = TrafficLights(25, 8, 7)

- 7.
8. while True:
9. button.wait_for_press()
10. buzzer.on()
11. light.green.on()
12. sleep(1)
13. lights.amber.on()
14. sleep(1)
15. lights.red.on()
16. sleep(1)
17. lights.off()
18. buzzer.off()

7.Result



Fig 7 yellow light glowing



Fig 7.1 red light glowing

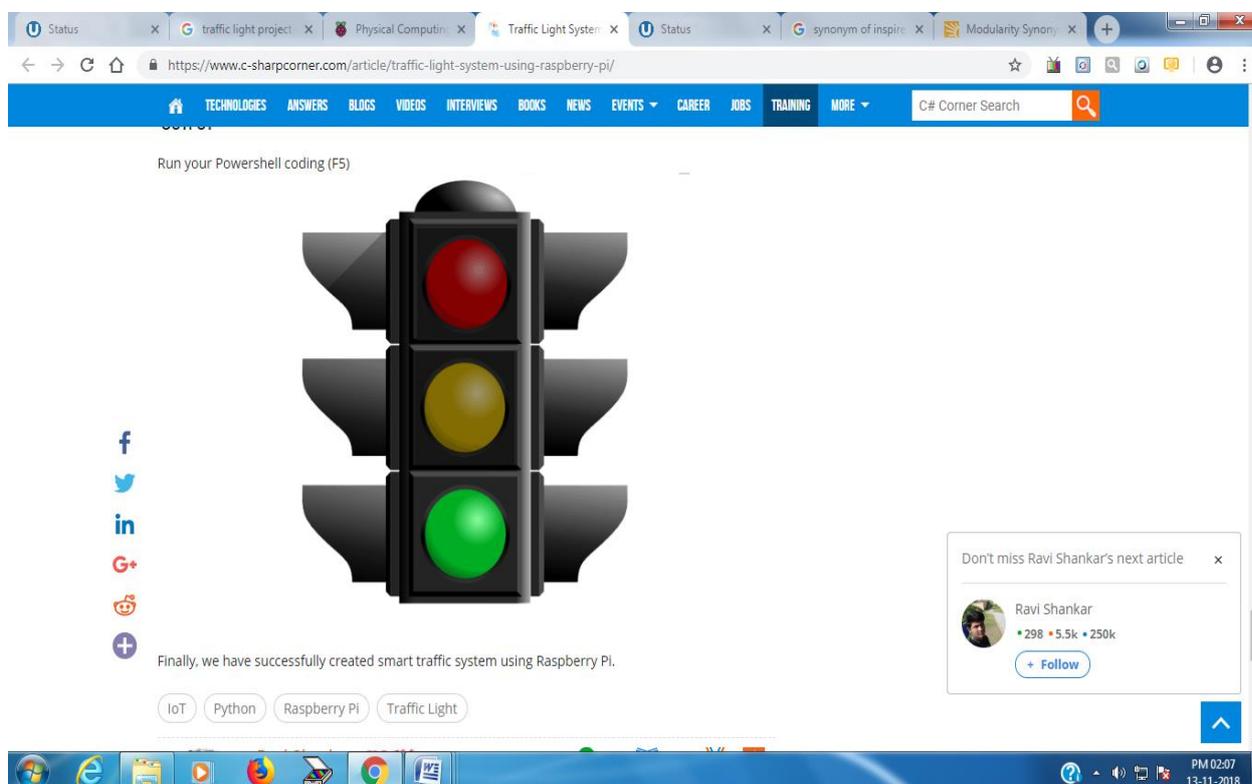


Fig 7.2 green light glowing



CONCLUSION:

Raspberry pi and python is recently launched resource and technology and in this paper we implemented traffic light system using Raspberry pi and python as on of of the use case.

REFERENCE

References:

1. <https://www.raspberrypi.org/>
2. [Python.org](https://python.org).
3. <http://www.engpaper.com/raspberry-pi-2018>