

G V V Sharma*

Abstract—This manual shows how to teach Python programming to beginners using a Raspberry Pi and a seven segment display. All basic concepts like conditional statements, loops, arrays and functions are covered.

Problem 1. Install the RPi.GPIO library in the raspberry pi

Solution:

```
sudo apt-get update
sudo apt-get install rpi.gpio
```

Problem 2. Connect the Pi to the seven segment display in Fig. 2 using Fig. 2

Component	Value	Quantity
Breadboard		1
Resistor	$\geq 220\Omega$	1
Raspberry Pi	3	1
Seven Segment Display	Common Anode	1
Jumper Wires		20

TABLE 2

Problem 3. Write a program to display the number 1.

Solution: Save the following code in a file called **sevenseg.py** and run using

```
sudo python sevenseg.py
```

```
#Setup seven segment display
import RPi.GPIO as GPIO # RPi.GPIO can be referred as GPIO from now
```

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

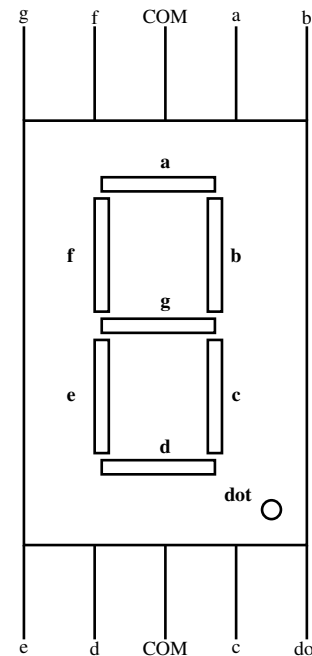


Fig. 2

```
GPIO.setmode(GPIO.BOARD) #
    GPIO Numbering of Pins
```

```
#Matching pins to segments
```

```
GPIO.setup(3, GPIO.OUT)#a
GPIO.setup(5, GPIO.OUT)#b
GPIO.setup(7, GPIO.OUT)#c
GPIO.setup(8, GPIO.OUT)#d
GPIO.setup(10, GPIO.OUT)#e
GPIO.setup(11, GPIO.OUT)#f
GPIO.setup(12, GPIO.OUT)#g
```

```
GPIO.output(3,1)
GPIO.output(5,0)
GPIO.output(7,0)
GPIO.output(8,1)
GPIO.output(10,1)
```

Raspberry Pi 3 Model B (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1			2
				5.0 VDC Power	
8	GPIO 8 SDA1 (I2C)	3			4
				5.0 VDC Power	
9	GPIO 9 SCL1 (I2C)	5			6
				Ground	
7	GPIO 7 GPCLK0	7			8
				GPIO 15 TxD (UART)	15
	Ground	9			10
				GPIO 16 RxD (UART)	16
0	GPIO 0	11			12
				GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	13			14
				Ground	
3	GPIO 3	15			16
				GPIO 4	4
	3.3 VDC Power	17			18
				GPIO 5	5
12	GPIO 12 MOSI (SPI)	19			20
				Ground	
13	GPIO 13 MISO (SPI)	21			22
				GPIO 6	6
14	GPIO 14 SCLK (SPI)	23			24
				GPIO 10 CE0 (SPI)	10
	Ground	25			26
				GPIO 11 CE1 (SPI)	11
30	SDA0 (I2C ID EEPROM)	27			28
				SCL0 (I2C ID EEPROM)	31
21	GPIO 21 GPCLK1	29			30
				Ground	
22	GPIO 22 GPCLK2	31			32
				GPIO 26 PWM0	26
23	GPIO 23 PWM1	33			34
				Ground	
24	GPIO 24 PCM_FS/PWM1	35			36
				GPIO 27	27
25	GPIO 25	37			38
				GPIO 28 PCM_DIN	28
	Ground	39			40
				GPIO 29 PCM_DOUT	29

Attention! The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Fig. 2

```
GPIO.output(11,1)
GPIO.output(12,1)
```

Problem 4. Modify the program in Problem 3 to generate all numbers between 0-9.

Problem 5. Run the following program and test for dec=0 and dec=8.

Solution:

```
#if-else
import RPi.GPIO as GPIO # RPi.GPIO
    can be referred as GPIO from
    now
```

```
GPIO.setmode(GPIO.BOARD) #
    GPIO Numbering of Pins
```

#Matching pins to segments

```
GPIO.setup(3, GPIO.OUT)#a
GPIO.setup(5, GPIO.OUT)#b
GPIO.setup(7, GPIO.OUT)#c
GPIO.setup(8, GPIO.OUT)#d
GPIO.setup(10, GPIO.OUT)#e
GPIO.setup(11, GPIO.OUT)#f
GPIO.setup(12, GPIO.OUT)#g
```

```
def sevenseg(a,b,c,d,e,f,g):
    GPIO.output(3,a)
    GPIO.output(5,b)
    GPIO.output(7,c)
    GPIO.output(8,d)
    GPIO.output(10,e)
    GPIO.output(11,f)
    GPIO.output(12,g)
    return
```

```
dec = 8
```

```
if dec == 8:
    sevenseg(0,0,0,0,0,0,0)
```

```
elif dec == 0:
    sevenseg(0,0,0,0,0,0,1)
```

Problem 6. Extend the previous program for all numbers between 0-9. Print E on the display if the input is not in this range.

Problem 7. Write a function for writing a decimal number to the seven segment display.

Problem 8. Using the function in problem 7 and a for loop, implement a decade counter.

Solution:

```
#bcd-sevenseg
import RPi.GPIO as GPIO # RPi.GPIO
    can be referred as GPIO from
    now
import time

GPIO.setmode(GPIO.BOARD) #
    GPIO Numbering of Pins
```

```

#Matching pins to segments
GPIO.setup(3, GPIO.OUT)#a
GPIO.setup(5, GPIO.OUT)#b
GPIO.setup(7, GPIO.OUT)#c
GPIO.setup(8, GPIO.OUT)#d
GPIO.setup(10, GPIO.OUT)#e
GPIO.setup(11, GPIO.OUT)#f
GPIO.setup(12, GPIO.OUT)#g

def sevenseg(a,b,c,d,e,f,g):
    GPIO.output(3,a)
    GPIO.output(5,b)
    GPIO.output(7,c)
    GPIO.output(8,d)
    GPIO.output(10,e)
    GPIO.output(11,f)
    GPIO.output(12,g)
    return

def bcd_sevenseg(dec):
    if dec == 0:
        sevenseg
            (0,0,0,0,0,0,1)
        return
    elif dec == 1:
        sevenseg
            (1,0,0,1,1,1,1)
        return
    elif dec == 2:
        sevenseg
            (0,0,1,0,0,1,0)
        return
    elif dec == 3:
        sevenseg
            (0,0,0,0,1,1,0)
        return
    elif dec == 4:
        sevenseg
            (1,0,0,1,1,0,0)
        return
    elif dec == 5:
        sevenseg
            (0,1,0,0,1,0,0)
        return
    elif dec == 6:
        sevenseg
            (0,1,0,0,0,0,0)
        return
    elif dec == 7:

```

```

        sevenseg
            (0,0,0,1,1,1,1)
        return
    elif dec == 8:
        sevenseg
            (0,0,0,0,0,0,0)
        return
    elif dec == 9:
        sevenseg
            (0,0,0,0,1,0,0)
        return
    else:
        sevenseg
            (0,1,1,0,0,0,0)
        return

while True:
    for i in range(10):
        bcd_sevenseg(i)
        time.sleep(1.0)

```

Problem 9. Repeat problem 7 using a list for pin numbering.

Problem 10. Repeat problem 8 by using a while loop.

Problem 11. Implement decimal to binary conversion and test your logic using the display.

Problem 12. Implement decimal to binary conversion using a for loop and a list.

Problem 13. Repeat the above exercise by using a function.