Displaying Time over 4-Digit 7-Segment Display Using Raspberry Pi.

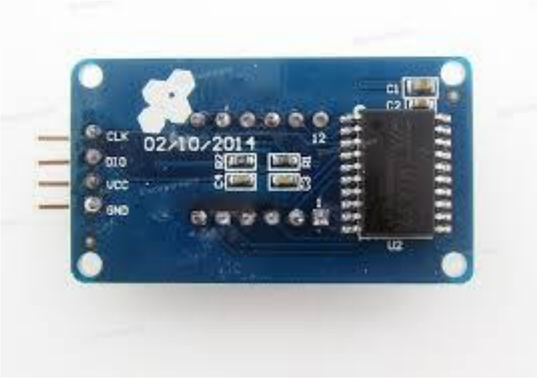
Installation Manual

To display small amount of data with Raspberry Pi, we can use 4 digit 7-segment Display. 7 Segment Display has seven segments in it and each segment has one LED inside it to display the numbers by lighting up the corresponding segments.

Hardware Requirements

1. Raspberry Pi Model A/B/B+
2. 4 digit 7 Segment Display
3. Jumper wires (Female to Female)

Here, I am using **4 digits-7 segments LED** display with TM1637 controller

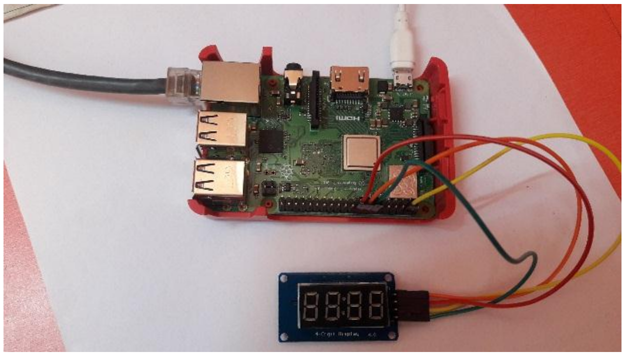


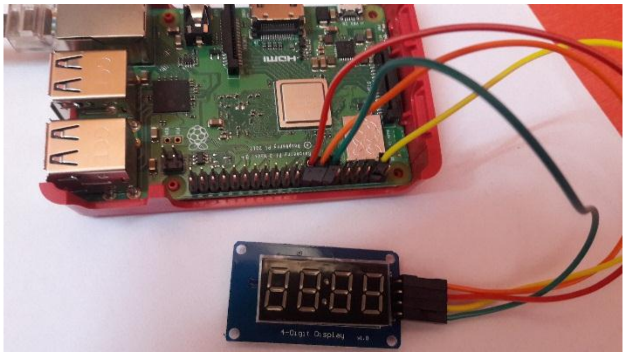
- **Software Requirements**

  1. Raspbian Stretch OS

1. Connect your 4 digit 7 segment display with Raspberry Pi's GPIO Pins.

TM1637 Board Pin	Function	RPI Physical Pin	Raspberry Function
GND	Ground	14	GND
VCC	+ 5V Power	4	5V
DI0	Data In	18	GPIO 24
CLK	Clock	16	GPIO 23

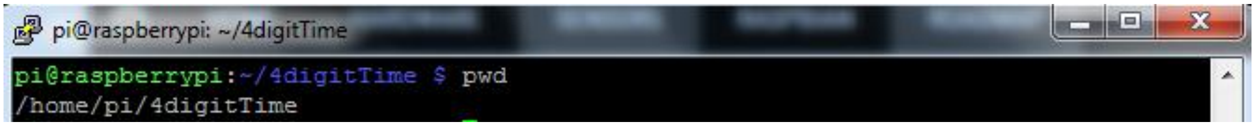




Step 1: Download Python Script

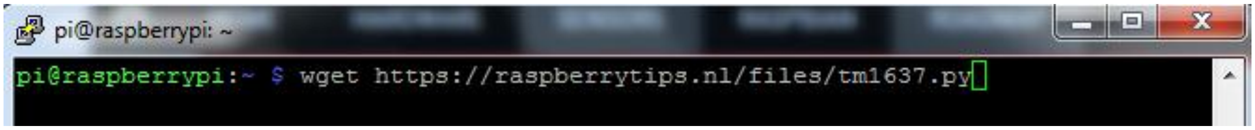
In order to control the LED, using a special script with pre-defined functions. Various functions are available in the script, for example, to display numbers and adjust the intensity of the LEDs.

Create a folder 4digitTime under /home/pi.

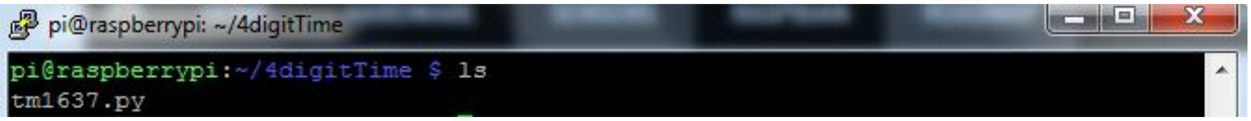


```
pi@raspberrypi:~/4digitTime $ pwd
/home/pi/4digitTime
```

Download the script using wget command.



```
pi@raspberrypi:~ $ wget https://raspberrytips.nl/files/tm1637.py
```



```
pi@raspberrypi:~/4digitTime $ ls
tm1637.py
```

Note: This Script file contains some of the important functions, which are required to add in our Python script.

Step 2: Write Python Script to display Time (e.g clock.py)

```
import sys
import time
import datetime
import RPi.GPIO as GPIO
import tm1637

#CLK -> GPIO23 (Pin 16)
#Di0 -> GPIO24 (Pin 18)
```

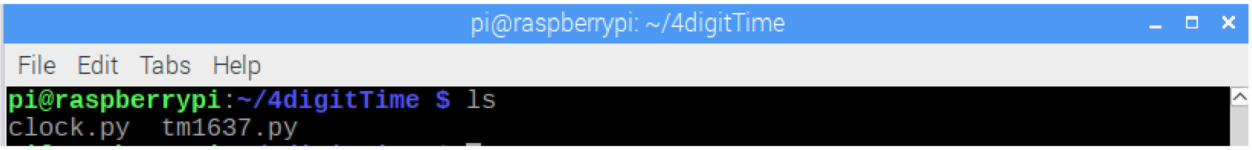
```
Display = tm1637.TM1637(23,24,tm1637.BRIGHT_TYPICAL)

Display.Clear()
Display.SetBrightnes(1)
while(True):
   now = datetime.datetime.now()
   hour = now.hour
   minute = now.minute
   second = now.second
   currenttime = [ int(hour / 10), hour % 10, int(minute / 10), minute % 10 ]

   Display.Show(currenttime)
   Display.ShowDoublepoint(second % 2)

   time.sleep(1)
```

The above script needs the **tm1637.py** script to work, so place both files in the same folder.



```
pi@raspberrypi:~/4digitTime $ ls
clock.py   tm1637.py
```

Script functions

The clock script uses the following functions, defined in tm1637.py:

Display. Clear () - Clears the display if individual LEDs are still active.

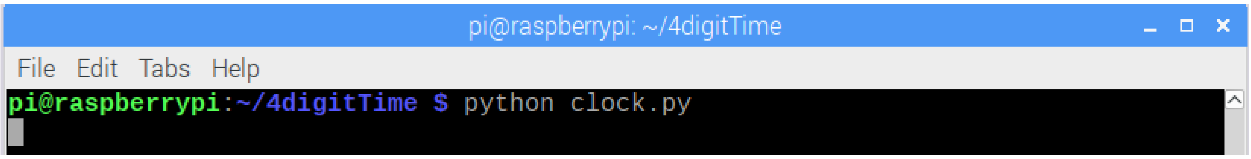
Display.SetBrightnes(x) - After this you can adjust the brightness of the display, at least 0 and maximum 7.

Display. Show(x,x,x,x) - Show the actual 4 digits (digits), x can be 0 to 9.

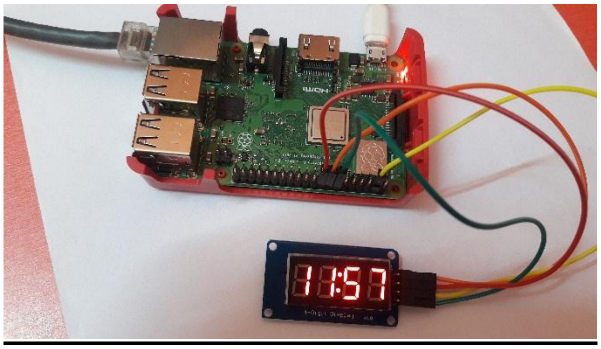
Display.ShowDoublepoint (status) - Controlling the ':' between the second and third digit, true (1) = on / false (0) = off.

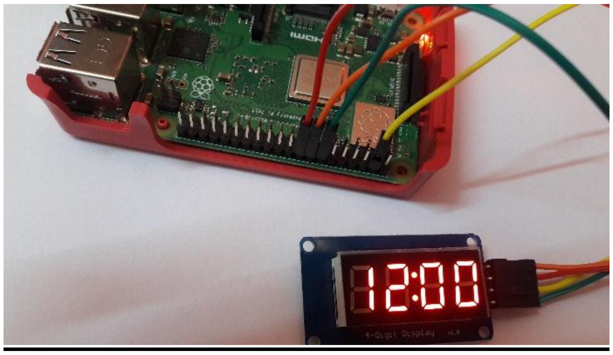
To know more about TM1637 controller, check http://www.microcontroller.it/english/Tutorials/Elettronica/componenti/TM1637.htm

Step 3: Start the script with following command

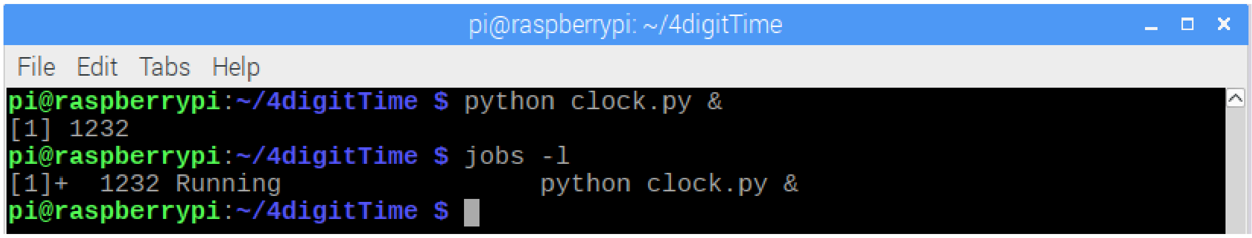


```
pi@raspberrypi:~/4digitTime $ python clock.py
```





To run the script in background you can use following command:



```
pi@raspberrypi:~/4digitTime $ python clock.py &
[1] 1232
pi@raspberrypi:~/4digitTime $ jobs -l
[1]+  1232 Running                 python clock.py &
pi@raspberrypi:~/4digitTime $ 
```

That's all !!!

Thank you….