EXPERIMENT : No.

Page No.
Date

1. Introduction to Raspberry pi

The Raspberry pi 3 model B is a single board computer developed by the raspberry pi foundation. This board consist of 1.2 GHz 64 Bit Quadcore ARM processor

Specifications of raspberry pi 3 board

1) Quadcore 1.2 GHz Broadcom BCM2837 64 bit CPU
2) 1 GB RAM
3) Micro SD port
4) Micro USB power port
5) 2 USB ports
6) 1 ethernet port
7) Stereo output & composite video port
8) Camera board
9) 1 HDMI port
10) 40 pin extended GPIO.

* GPIO pins (General Purpose Input output)

A powerful feature of raspberry pi is the row of GPIO pins along the extreme right edge of the board. GPIO pins don't have specific function and can be customized using the software.

1) Powerpins : The board consists of 2 5V pins, 2 3.3 V pins and 7 Gnd pins which are unconfigurable. The 5V pins directly deliver the 5V supply coming from the main adaptor. This pin can use to powerup raspberry pi & it can also use to powerup other 5V devices. The 3V pin is there to offer a stable 3.3V supply to power the components.

Teacher's Sign. : _____

2) Input output pins: A GPIO pin that is set as an input will allow a signal to be recieved by the raspberry pi that is sent by a device connected to this pin. A GPIO pin set as an output will send the voltage signal to a device connected to this pin. Along with simple function of I/O pins. The GPIO pins can also perform a variety of alternative functions.

3) SPI pins (Serial peripheral Interface): It is a protocol used for master slave communication. It is used by raspberry pi board to quickly communicate between one or more peripheral devices. There are 5 pins involved in SPI communication: GND, SCLK, MOSI, MISO, CE

4) I2C pins (Inter Integrated circuit): It is used by raspberry pi board to communicate with devices that are compatible with interintegrated circuit. It uses 2 pins: SCL and SDA

5) UART pins (Universal Asynchronous reciever transmittor): UART pins provide a way to communicate between 2 Micro controller or the computer. Tx pin is used to transmit the serial data and Rx pin is used to receive serial data coming from different serial device.

* prepare raspberry pi

1) Download raspbian operating system
   https://www.raspberrypi.org/downloads

2) Download & install SD card Farmatter
   https://www.SDcard.org/downloads

3) Download & install etcher
   https://www.etcher.io.

EXPERIMENT: No.

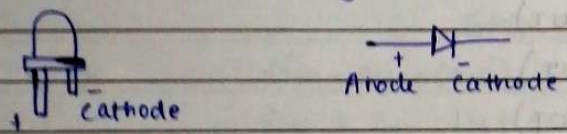
Page No.
Date

2. Generating different LED patterns

Component used

1) 5 LEDs
2) 5 330 Ω register
3) jumper wires

LED (Light emitting diode)



The raspberry pi GPIO pins can only supply about 60 MA LEDs are ready to draw as much as current possible from source If we don't resist it will damage raspberry pi so weed need to to calculate the resistor value which will make less current drawing from LED.

$$V = IR$$

$$R = \frac{V}{I}$$

$$= V_{\text{power source}} - V_{\text{forward voltage}} / I$$

$$= 5v - 1.9v / 10mA$$

$$= 3.1v / 0.01A$$

$$= 310 \text{ ≅ } 330\ \Omega$$

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(5, GPIO.OUT)
```

Sundaram

Teacher's Sign.: _____

```
While True:
        GPIO.output (5, GPIO.HIGH)
        time.sleep (0.5)
        GPIO.output (5, GPIO.LOW)
        time.sleep (0.5)
```

Pattern 1

```
import RPi.GPIO as GPIO
import time
GPIO.setmode (GPIO.BCM)
GPIO.setwarnings (False)
GPIO.setup (5, GPIO.OUT)
GPIO.setup (6, GPIO.OUT)
GPIO.setup (13, GPIO.OUT)
GPIO.setup (19, GPIO.OUT)
GPIO.setup (26, GPIO.OUT)
list = [5, 6, 13, 19, 26]
While True:
        for i in list
                GPIO.output (i, GPIO.HIGH)
                time.sleep (0.5)
                GPIO.output (i, GPIO.LOW)
                time.sleep(0.5)
```

Pattern 2

```
import RPi.GPIO as GPIO
import time
GPIO.setmode (GPIO.BCM)
GPIO.setwarnings (False)
GPIO.setup (5, GPIO.OUT)
GPIO.setup (6, GPIO.OUT)
GPIO.setup (13, GPIO.OUT)
GPIO.setup (19, GPIO.OUT)
GPIO.setup (26, GPIO.OUT)
While True:
```

EXPERIMENT : No.

Page No.
Date

```
            GPIO.output(list, GPIO.HIGH)
            time.sleep(0.5)
            GPIO.output(list, GPIO.LOW)
            time.sleep(0.5)
```

Another way from while loop:

```
while True:
        for i in list:
                GPIO.output(i, GPIO.HIGH)
        time.sleep(0.5)
        for i in list:
                GPIO.output(i, GPIO.LOW)
        time.sleep(0.5)
```

Pattern 3

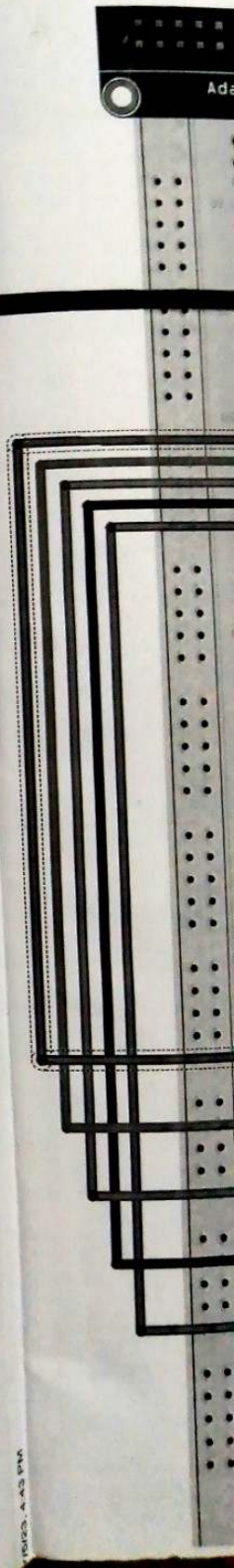
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(5, GPIO.OUT)
GPIO.setup(6, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)
GPIO.setup(19, GPIO.OUT)
GPIO.setup(26, GPIO.OUT)
list = [5, 6, 13, 19, 26]
while True:
    for i in list:
        GPIO.output(i, GPIO.HIGH)
        time.sleep(0.5)
        GPIO.output(i, GPIO.LOW)
```

Teacher's Sign. :

```
            time.sleep(0.5)
    for i in list[::-1]:
            GPIO.output(i, GPIO.HIGH)
            time.sleep(0.5)
            GPIO.output(i, GPIO.LOW)
            time.sleep(0.5)
```

Pattern 4

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(5, GPIO.OUT)
GPIO.setup(6, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)
GPIO.setup(19, GPIO.OUT)
GPIO.setup(26, GPIO.OUT)
list = [5, 6, 13, 19, 26]
While True:
        for i in range(int(len(list)/2+1)):
                GPIO.output(list[len(list)-i-1], GPIO.HIGH)
                GPIO.output(list[len(list)
                time.sleep(0.5)
                GPIO.output(list[i], GPIO.LOW)
                GPIO.output(len(list)-i-1], GPIO.LOW)
                time.sleep(0.5)
```



1-LED-Pattern_bb.png

15/23, 4:43 PM

ps://drive.google.com/drive/my-drive

Bharat Rai

TO·HIGH)

_OW)

EXPERIMENT : No.

Page No.
Date

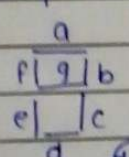
8. Displaying time over 4 digit 7 segment display using raspberry pi

Introduction:

There are many types of 4 digit 7 segment display available in market. You can get them in various packages including I2C backpack, 12 pins 16 pins, resistor built-in, common anode, common cathode etc. The one which we are using is 12 pins 4 digits 7 segment common anode resistor built-in.

Component used:

1) 4 digit 7 segment display
2) Jumper wire



Display model included in the kit has 12 pins out of which 8 pins are attached to 7 segment of all digits and single decimal point other 4 are used as high for every digit - If you connect pin 9 to high second digit will be activated same for pin 8 and 6 so if we have 12, 9, 8, 6 all connected to high and 7, 4 also high then all four digit would display numbers one.

To display seperate digits like 1, 2, 3, 4 on a display we need to enable certain segments depending on the digit we display for a small amount of time on a first digit

Teacher's Sign. :

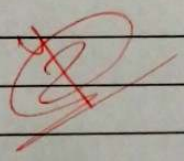
and go to next digit. So basically we only have 1 digit turned on at a time our eyes cannot see those on and offs because they are happening so fast

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
segments = (17, 27, 22, 23, 24, 25, 5, 6)
for segment in segments:
        GPIO.setup(segment, GPIO.OUT)
        GPIO.output(segment, GPIO.HIGH)
digits = (16, 20, 21, 12)
for digit in digits:
        GPIO.setup(digit, GPIO.OUT)
        GPIO.output(digit, GPIO.HIGH)
num = {' ' : (0,0,0,0,0,0,0,0),
       '0' : (1,1,1,1,1,1,1,0),
       '1' : (0,1,1,0,0,0,0,0),
       '2' : (1,1,0,1,1,0,1),
       '3' : (1,1,1,1,0,0,1),
       '4' : (0,1,1,0,0,1,1),
       '5' : (1,0,1,1,0,1,1),
       '6' : (1,0,1,1,1,1,1),
       '7' : (1,1,1,0,0,0,0);
       '8' : (1,1,1,1,1,1,1),
       '9' : (1,1,1,1,0,1,1) }
while True:
    n = time.ctime()[11:13] + time.ctime()[14:16]
    s = str(n)
    for digit in range(4):
        for loop in range(0,7):
            if num[s[digit]][loop]==0:
```

EXPERIMENT : No.

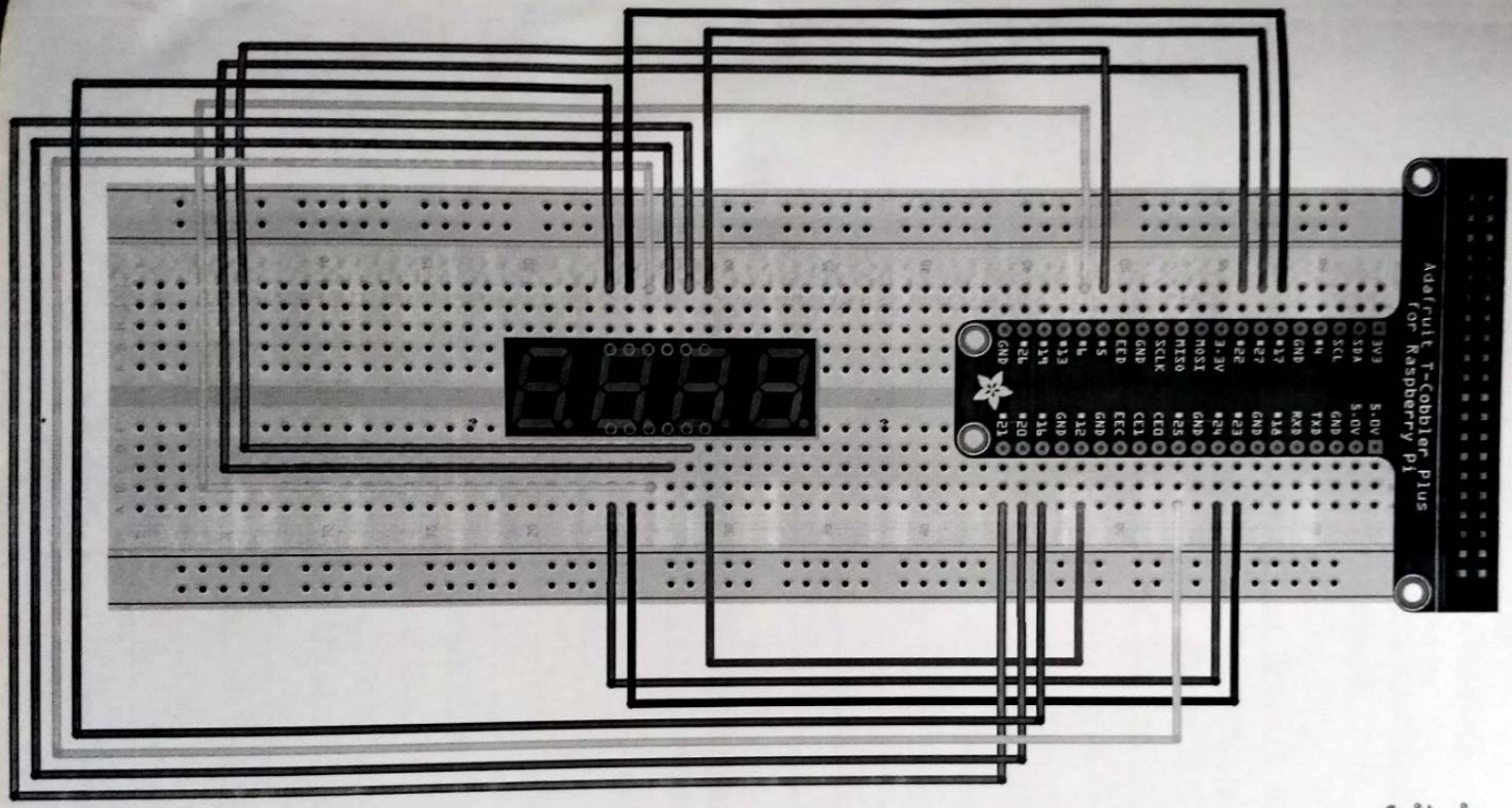
Page No.
Date

```
            GPIO.output (segments[loop], 1)
        else:
            GPIO.Output (segments [loop], 0)
        if (int (time.ctime() [18:19]) % 2 == 0) and (digit)==1):
            GPIO.output (6, 0)
        else :
            GPIO.Output (6,1)
    GPIO.output (digits [digit], 1)
    time.sleep (0.001)
    GPIO.output (digits [digit], 0]
```



Sundaram

Teacher's Sign. : _____



fritzing

Bharat Rai

EXPERIMENT: No.

Page No.
Date

4. Interfacing Raspberry Pi with RFID

The RFID RC522 is very low cost RFID (Radio Frequency identification) reader & writer that is based on the MFRC522 Microcontroller provides its data through the SPI protocol & works by creating 13.56 Hz electromagnetic field that it uses to communicate with RFID tags.

Component used:

1) RFID 13.56 Hz tags
2) RFID RC522
3) Jumper wire

Steps

1) Enable SPI

Start menu → Preferences → RPi Configuration
Select interface tab → Select SPI enable → Click OK → Reboot

2) Install dependencies

```
Sudo apt-get update
sudo apt-get upgrade
sudo apt-get install python2.7-dev
```

3) Install SPI-Py library

```
cd /home/raspberrypi/batch3/RFID
```

Next, clone SPI-Py git folder for library by running following command

```
git clone https://github.com/          /SPI-Py.git
```

Sundaram

Teacher's Sign.: _____

```
cd SPI-Py
Sudo python setup.py install
```

4) Install pimylifeup MFRC522 library

```
cd ..
```

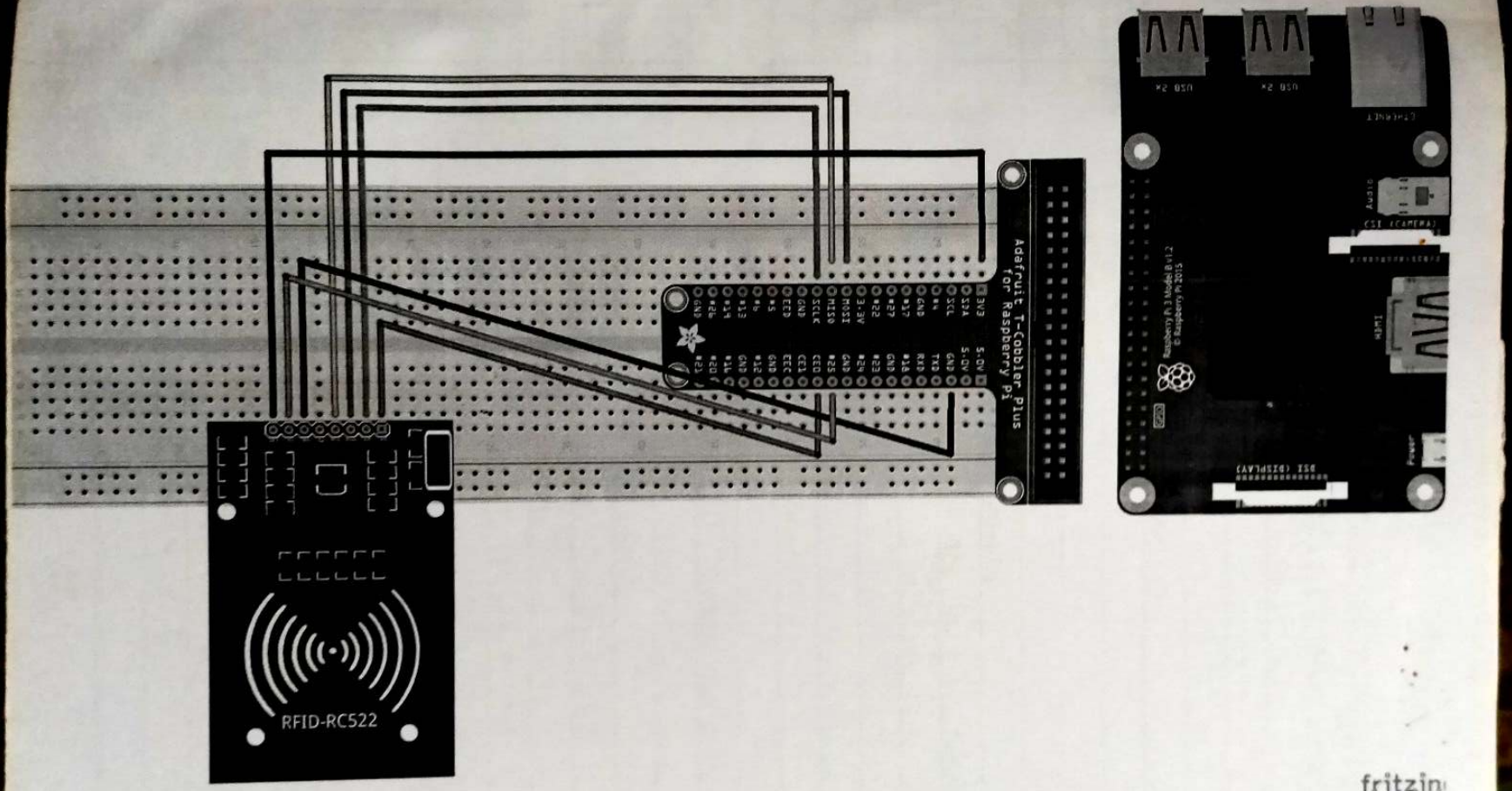
Next, clone MFRC522 Gitfolder for the library by running Following command

```
git clone https://github.com/pimylifeup/MFRC522-python.git

cd /home/raspberrypi/batch3/RFID/MFRC522-python

Sudo nano write.py

import RPi.GPIO as GPIO
From mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
try:
    text = input ('Enter new data:')
    print ('Now place your tag to write')
    reader.write (text)
    print ('written')
finally:
    GPIO.cleanup()

Ctrl + x then y
  Sudo python write.py


import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
try:
    id, text = reader.read()
    print (id)
    print (text)
Finally:
    GPIO.cleanup()
```

10-RFID_bb.png

11/23, 4:26 PM

11/23, 4:26 PM 10-RFID_bb.png



fritzin

Bharat Rai

https://drive.google.com/drive/my-drive

1/2

EXPERIMENT : No.

Page No.
Date

5. Controlling raspberry pi with telegram messenger

Component used

1) 4 LED
2) 4 3 Ω resistors & Jumper wire
3) Smartphone

Steps

1) Prepare telegram on our mobile
   Install & open telegram app in your system / mobile
   i) Search BotFather
   ii) Open BotFather
   iii) Start BotFather :
   iv) Create new bot (/newbot)
   v) give bot name
   vi) Choose bot username (it should end with bot)

2) Install dependencies
   i) sudo apt-get update
   ii) sudo apt-get upgrade
   iii) sudo apt-get install python3-pip
   iv) sudo pip install telepot

```
import RPi.GPIO as GPIO
import time
import telepot
from telepot.loop import MessageLoop
GPIO.setmode (GPIO.BCM)
GPIO.setwarnings (False)
```

Sundaram

Teacher's Sign. : _____

```
        green = 6
        red = 13
        yellow = 19
        White = 26
GPIO.setup (green, GPIO.OUT)
GPIO.output (green, GPIO.LOW)
GPIO.setup (red, GPIO.OUT)
GPIO.output (red, GPIO.LOW)
GPIO.setup (yellow, GPIO.OUT)
GPIO.output (yellow, GPIO.LOW)
GPIO.setup (white, GPIO.OUT)
GPIO.output (White, GPIO.LOW)
def action (msg):
        Chat_id = msg['chat']['id']
        Command = msg['text']
        print ('Recieved:' command]
        if 'on' in command:
            Message = 'Turned on'
            if 'red' in command:
                GPIO.output (red, GPIO.HIGH)
                Message = message + 'red led'
            if 'green' in command:
                GPIO.output (green, GPIO.HIGH)
                Message = Message + 'green led'
            if 'yellow' in command:
                GPIO.output (yellow, GPIO.HIGH)
                Message = Message + 'yellow led'
            if 'white' in command:
                GPIO.output (White, GPIO.HIGH)
                Message = message + 'White led'
            if 'all' in command:
                GPIO.output (red, GPIO.HIGH)
                GPIO.output (green, GPIO.HIGH)
                GPIO.output (yellow, GPIO.HIGH)
                GPIO.output (White, GPIO.HIGH)
```

EXPERIMENT : No. Page No. Date

```
                Message = Message + 'all led'
        telegram_bot.sendMessage(chat_id, Message)

if 'off' in command:
        Message = 'Turned off'
        if 'red' in command:
                GPIO.output(red, GPIO.LOW)
                Message = Message + 'red led'
        if 'green' in command:
                GPIO.output(green, GPIO.LOW)
                Message = Message + 'green led'
        if 'yellow' in command:
                GPIO.output(yellow, GPIO.LOW)
                Message = Message + 'yellow led'
        if 'white' in command:
                GPIO.output(white, GPIO.LOW)
                Message = Message + 'white led'
        if 'all' in command:
                GPIO.output(red, GPIO.LOW)
                GPIO.output(green, GPIO.LOW)
                GPIO.output(yellow, GPIO.LOW)
                GPIO.output(white, GPIO.LOW)
                Message = Message + 'all led'
        telegram_bot.sendMessage(chat_id, Message)
telegram_bot = telepot.Bot('
print(telegram_bot.getMe())
MessageLoop(telegram_bot, action).run_as_thread()
print('up and Running ----')
while(1):
    time.sleep(10)
```

Sundaram

Teacher's Sign. :

EXPERIMENT: No. Page No. Date

6. Raspberry Pi based Oscilloscope

Introduction

The Oscillosope is an electronic test instrument that allows visualization & observation of varying signal voltages. Usually as a 2 Dimensional plot with one or more signal plotted against time.

In the practical we will replicate the signal visualization capabilities of oscilloscope using Raspberry Pi and analog to digital converter module.

Components Used

1) 10K Potentiometer
2) 16 Bit ADS1115
3) Jumper wires

Steps

1) Enable I2C
   Start Menu → Preferences → RPi Configuration
   Select interfaces tab, Select I2C enable → Click OK → Reboot

2) Install dependencies
   i) Sudo apt-get update
   ii) Sudo apt-get upgrade
   iii) Sudo apt-get install build-essential python-dev python3-smbus git

```
git clone https://github.com/adafruit/Adafruit_Python_ADS1x15.git
cd Adafruit_Python_ADS1x15
sudo python3 setup.py install
```

Sundaram

Teacher's Sign. :

4 Test the library & I2C communication

Before we proceed it is important to test library & ensure the ADC can communicate with Raspberry Pi over I2C to do this. we will use an example script that comes with library

```
cd examples
```

Next run Simpletest.py example which displays value of four Channels on ADC with tabular form

```
Python3 Simpletest.py
```

5 Install Mat-Plotlib

To visualize the data we need to install Mat-Plotlib which is used to plot all kinds of graph in python

```
sudo apt-get install python3-matplotlib
```

6 Install DrawNow python Module

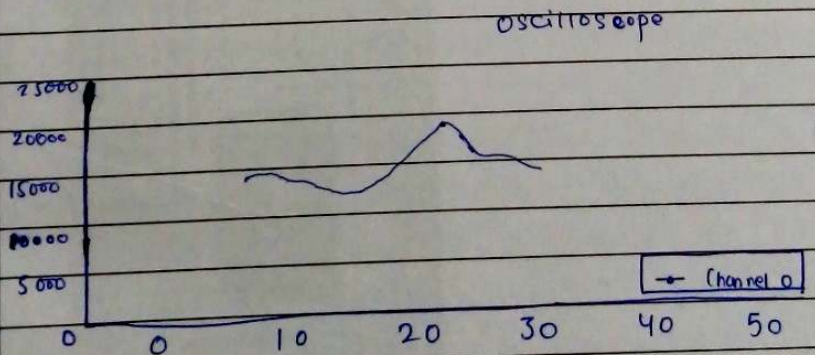
```
sudo apt-get install python3-pip
sudo pip install drawnow
```

```
sudo nano scope.py
import time
import matplotlib.pyplot as plt
from drawnow import *
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1
val = []
int = 0
plt.ion()
adc.start_adc(0, gain = GAIN)
print('Reading ADS1x15 channel 0')
```

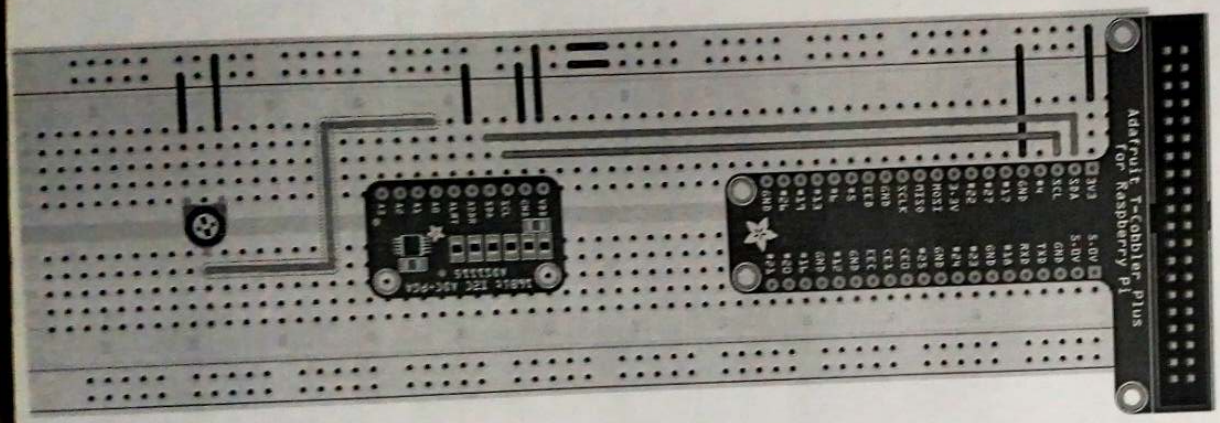
EXPERIMENT : No.

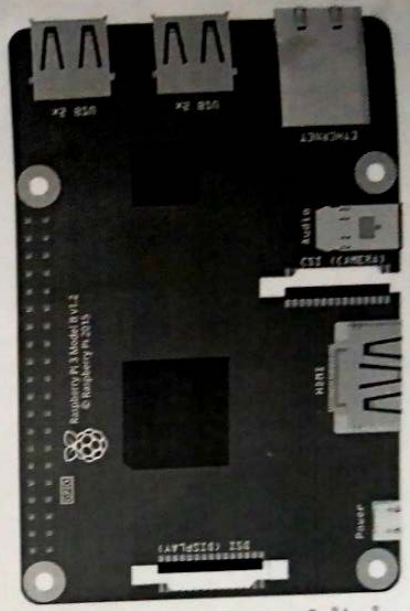
Page No.
Date

```
def makefig():
        plt.ylim(-5000, 5000)
        plt.title('Oscilloscope')
        plt.grid(True)
        plt.ylabel('ADC outputs')
        plt.plot(val, 'ro-', label='Channel 0')
        plt.legend(loc='lower right')
While True:
        value = adc.get_last_result()
        print('Channel 0: {0}'.format(value))
        time.sleep(0.5)
        val.append(int(value))
        drawnow(makefig)
        plt.pause(.00001)
        cnt = cnt+1
        if(cnt > 50):
                val.pop(0)
```



Teacher's Sign. :


Adafruit T-Cobbler Plus
for Raspberry Pi


USB 2x
USB 2x
ETHERNET
Audio
CSI (CAMERA)
HDMI
DSI (DISPLAY)
Power
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015

fritzing

EXPERIMENT: No.

Page No.
Date

7. Raspberry Pi GPS Module Interfacing

GPS stands for (Global positioning system) and used to detect the latitude and longitude of any location on the earth with exact UTC time (Universal time co-ordinates).

GPS Module send the data related to tracking position in the real-time and it sends so many data in NMEA format.

NMEA format consists several sentences in which we only we only need one sentence.

This sentence start from $GPGGA and contains the co-ordinates time and other useful information.

This GPGGA is referred to GPS fixed data.

- Component used
  1. Neo 6m GPS Module with an Antenna

- Steps

Step 1 Setting UP UART

The Raspberry Pi has 2 built-in UARTs PL011 and mini UART.

They are implemented using different hardware block On Raspberry Pi 3 the Wireless Bluetooth module is connected to PL011 UART while the mini UART is used for the linux console output

Teacher's Sign.:

In this practical we will deactivate the bluetooth module from PL011 UART using on overlay.

The First thing is to edit /boot/config.txt

→ Create GPS Folder in Batch3 folder

→ In terminal

sudo nano /boot/config.txt

→ In end of config.txt

dtparam = spi = ON

dtoverlay = pi3-disable-bt

Core_Frequency = 250

enable UART = 1

force-turbo = 1

Step 2: UART setup section

Sudo cp/boot/cmdline.txt/boot/cmdline - bak.txt

Sudo nano /boot/cmdline.txt

dwc_otg.lpm_enable=0 console = tty1 root = dev/mmcblk0p2

rootfstyle = ext4 elevator = deadline. fsck.repair = yes

→ Sudo reboot

Step 3: disable Raspberry Pi Serial getty service

Sudo system Ctl disable serial-getty@S0.Servia

sudo reboot

Step 4: Activating ttyAMA0

Sudo SystemCtl enable Serial-getty@ttyAMA0.Servico

Step 5: Install minicom and Pynmea 2

Description :- We will install minicom to connect to the GPS module and make sense of the data

> Sudo apt-get install minicom.

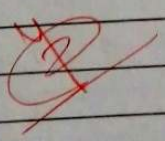
> Sudo pip install pynmea2

EXPERIMENT : No.

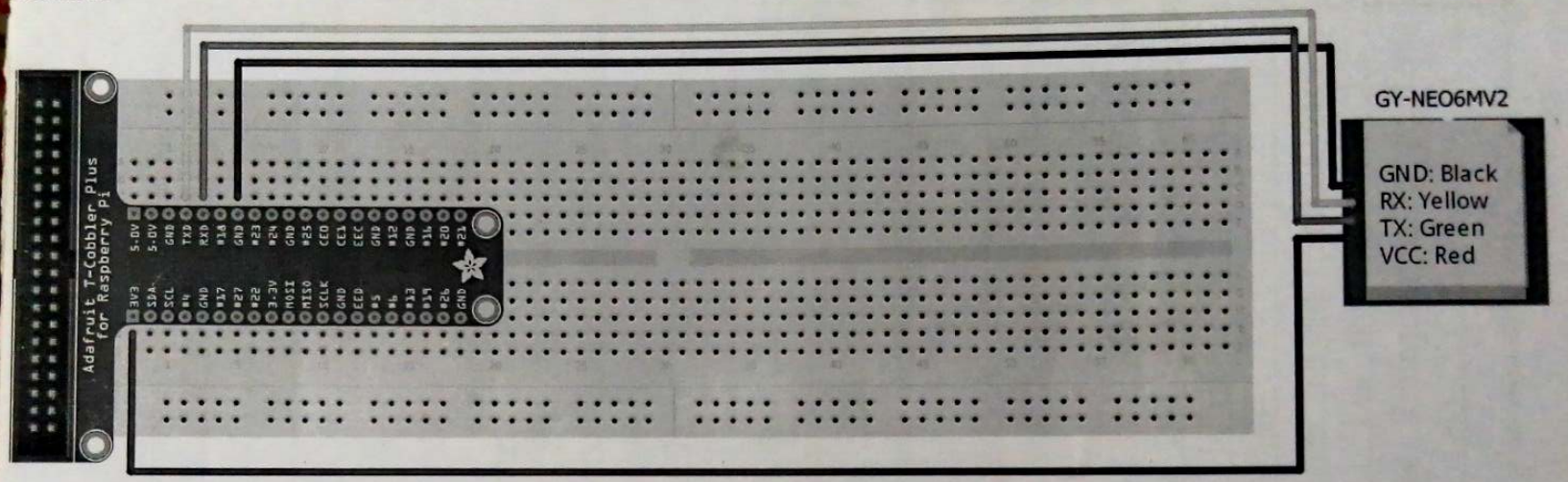
Page No.
Date

Code :

```
import time
import serial
import string
import pynmea2
port = "/dev/ttyAMA0"
ser = serial.Serial (port, baurate = 9600, timeout = 0.5)
while 1 :
    try:
        data = ser.readline()
        if data[0:6] = = '$GPGGA'
            msg = pynmea.parse(data)
            latval = msg.lat
            concatlat = "lat:" + str(latval)
            print (concatlat)
            lonval = msg.lon
            concatlon = "lon:" + str(lonval)
            print (concatlon)
    Finally:
        print ("loading")
```



Teacher's Sign. :


Adafruit T-Cobbler Plus
for Raspberry Pi
GY-NEO6MV2
GND: Black
RX: Yellow
TX: Green
VCC: Red

fritzing

EXPERIMENT: No. _____ Page No. Date

8. Installing Windows 10 IOT core on Raspberry Pi

Introduction

Windows 10 IOT core is an operating system which is built for small, Secured, Smart devices.

Steps:

1) To prepare Windows IOT core installation the SD card is prepared on windows 10 pc visit https://developer.microsoft.com/en-us/windows/iot/Download, Url and download, window 10 IOT Core Dashboard.
2) Run the dashboard setup and install Windows 10 IOT core dashboard.
3) Run the dashboard after instau and select "setup a new device".
4) Select the device type as Raspberry Pi 2 and 3, OS build and arrive to prepare installation. Add Administrator password except the license terms and click on "Download and install".
5) It will download and then install the windows IOT core onto the SD card, Onee card is prepared the dashboard will display message "Your SD Card is ready". At this point remove SD card and put it in Raspberry pi.
6) Once power is on Windows IOT core will do many things in the background for first run, it will take time and on successful loading it will show a screen to choose the language, select desired Language and click next.
7) Windows IOT Core will boot into desktop it shows device information Provide command line and browser.
8) We can change the required Setting from setting tab

Teacher's Sign. : _____

EXPERIMENT : No. _____ Page No. Date

9. Interfacing Picamera with Raspberry Pi

Settings

> sudo raspi - config

After that → "Select 'interface option 1" → then select "legacy system' Enable / Disable legacy camera Support → then "Yes".

For capturing the image

Camera.Py

```
From picamera import PiCamera
From time import Sleep
Camera = PiCamera()
Camera.Start_preview()
time.sleep(0.5)
Camera.capture('/home/raspberryPi/tyitb3/myimage.jpg')
Camera.Stop-preview()
```

For capturing the video

video.Py

```
From picamera import PiCamera
From time import Sleep
Camera = PiCamera()
camera.Start_preview()
camera.Start_recording('/home/raspberryPi/tyitb3/myvideo.h264')
time.sleep(10)
Camera.Stop_recording()
Camera.Stop_preview()
```

> Sudo python3 camera.py

> sudo python3 video.py

Sundaram

Teacher's Sign. : _____