Practical 1: led lights-29,31,33,35 gnd-7,9,14(any one) power-19

Code:

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

led1=29

led2=31

led3=33

led4=35

GPIO.setup(led1,GPIO.OUT)

GPIO.setup(led2,GPIO.OUT)

GPIO.setup(led3,GPIO.OUT)

GPIO.setup(led4,GPIO.OUT)

GPIO.output(led1,False)

GPIO.output(led2,False)

GPIO.output(led3,False)

GPIO.output(led4,False)

def ledpattern(v1,v2,v3,v4):

GPIO.output(led1,v1)

GPIO.output(led2,v2)

GPIO.output(led3,v3)

GPIO.output(led4,v4)

try:

while True:

for I in range(0,4):

ledpattern(1,0,0,0)

time.sleep(0.5)

ledpattern(0,1,0,0)

time.sleep(0.5)

ledpattern(0,0,1,0)

time.sleep(0.5)

ledpattern(0,0,0,1)

time.sleep(0.5)

finally:

GPIO.cleanup()

Practical 2: gnd:14 vcc:4 Dio:18 clk:16

-Create a folder in home dir

-Use cd command to change dir

-cd foldername/

-pwd

-wget <https://raspberrytips.nl/files/tm1637.py>

-pip install raspberry.tm1637

Code: create a python file in the folder we created

import sys

import time

import datetime

import RPi.GPIO as GPIO

import tm1637

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)

Practical 3:

Pin- led-11 gnd -9

In cmd- sudo apt-get install python-pip

* sudo pip install telepot

Create bot on telegram

Code:

import sys

import time

import random

Import datetime

import telepot

import RPi.GPIO as GPIO

#LED

def on(pin):

GPIO.output(pin,GPIO.HIGH)

Return

def off(pin):

GPIO.output(pin,GPIO.LOW)

Return

# to use Raspberry Pi board pin numbers

GPIO.setmode(GPIO.BOARD)

# set up GPIO output channel

GPIO.setup(11, GPIO.OUT)

def handle(msg):

chat\_id = msg[‘chat’][‘id’]

command = msg[‘text’]

print (‘Got command: %s’ % command)

if command == ‘on’:

bot.sendMessage(chat\_id, on(11))

elif command ==’off’:

bot.sendMessage(chat\_id, off(11))

bot = telepot.Bot(‘6621527093:AAGsB7Vp4Yso2WdXPuwiKpVmu9pL1KOmdhc’)

bot.message\_loop(handle)

print (‘I am listening…’)

while 1:

time.sleep(10)

Practical 4: gnd-9 vdd-17 scl- 5 sda-3

* sudo raspi-config
* (Go to interfacing option→I2C→enable(yes)
* sudo apt-get update
* sudo apt-get upgrade
* Now install adafruit in home directory
* sudo apt-get install build-essential python-dev python3-smbus git
* sudo git clone <https://github.com/adafruit/Adafruit_Python_ADS1x15.git>
* cd /home/pi/Adafruit\_Python\_ADS1x15 (change into cloned file dir)
* sudo python3 setup.py install
* cd examples
* python simpletest.py
* sudo apt-get install python3-matplotlib

Code-

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import Adafruit\_ADS1x15

# Create an ADS1115 ADC (16-bit) instance.

adc = Adafruit\_ADS1x15.ADS1115()

GAIN = 1

val = []

# Start continuous ADC conversions on channel 0 using the previous gain value.

adc.start\_adc(0, gain=GAIN)

print(“Reading ADS1x15 channel 0”)

fig, ax = plt.subplots()

ax.set\_ylim(-5000, 5000)

ax.set\_title(‘Oscilloscope’)

ax.grid(True)

ax.set\_ylabel(‘ADC outputs’)

line, = ax.plot([], ‘ro-‘, label=’Channel 0’) # Fix for unpacking in the line assignment

ax.legend(loc=”lower right”) # Fix the typo in “loc”

def update(cnt):

# Read the last ADC conversion value and print it out.

value = adc.get\_last\_result()

print(‘Channel 0: {0}’.format(value))

# Set new data to line

line.set\_data(list(range(len(val))), val)

ax.relim()

ax.autoscale\_view()

val.append(int(value))

if cnt > 50:

val.pop(0)

ani = FuncAnimation(fig, update, interval=500)

plt.show()

Practical 5 :

-sudo apt-get install python-picamera

-sudo apt-get install pyhton3-picamera

-sudo pip install picamera

-sudo apt-get config

(Select interface option→camera option→enable→finish)ensure that camera option is enabled in a raspberry Pi configuration tool

Code:

import picamera

from time import sleep

#create object for PiCamera class

camera = picamera.PiCamera()

#set resolution

camera.resolution = (1024,768)

camera.brightness = 60

camera.start\_preview()

#add text on image

camera.annotate\_text = ‘Hi Pi User’

Sleep(5)

#store image

camera.capture(‘image1.jpg’)

camera.stop\_preview()

2)video recording

import picamera

from time import sleep

camera = picamera.PiCamera()

camera.resolution=(640,480)

print()

#start recording using pi camera

camera.start\_recording(“/home/pi/demo.h264”)

#wait for video to record

camera.wait\_recording(20)

#stop recording

camera.stop\_recording()

camera.close()

print(“video recording stopped”)

Practical 7: gps: vcc-1, gnd-39, txd-10, rxd-8

Led16x2 : rs-37, e- 35 , d4-33, d5-31, d6-29, d7-23 (connect with the help of breadboard)

-cmd

- sudo apt-get update

* sudo apt-get upgrade
* sudo nano /boot/config.txt
* Edit the txt
* Dtoverlay=pi3-disable-bt
* core\_freq=250
* force\_turbo= 1 (save with Ctrl x )
* Edit the cmdline
* sudo nano /boot/cmdline.txt
* Remove console serial0,115200 and modify root=/dev/mmcblk0ps
* sudo reboot
* sudo systemct1 stop [serial-getty@ttyS0.service](mailto:serial-getty@ttyS0.service)
* sudo systemct1 disable [serial-getty@ttyS0.service](mailto:serial-getty@ttyS0.service)
* sudo reboot
* sudo systemct1 enable [serial-getty@ttyAMA0.services](mailto:serial-getty@ttyAMA0.services)
* ls -l/dev (to verify)
* sudo apt-get install minicom
* sudo apt-get install pynmea2
* sudo minicom -D /dev/ttyAMA0 -b 9600 (press ctrl+a, press x and enter to exit)

Code:

import time

import serial

import string

import pynmea2

import RPi.GPIO as gpio

import Adafruit\_CharLCD as LCD

gpio.setmode(gpio.BCM)

lcd = LCD.Adafruit\_CharLCD(rs=26, en=19,

d4=13, d5=6, d6=5, d7=11, cols=16, lines=2)

lcd.message(“ Welcomes You”)

time.sleep(2)

lcd.clear()

lcd.message(“GPS Demo”)

time.sleep(2)

lcd.clear()

port = “/dev/ttyAMA0” # the serial port to which the pi is connected.

#create a serial object

ser = serial.Serial(port, baudrate = 9600, timeout = 0.5)

try:

while 1:

try:

data = ser.readline()

except:

print(“loading”)

#wait for the serial port to churn out data

If data[0:6] == ‘$GPGGA’:

# the long and lat data are always contained in the GPGGA string of the NMEA data

msg = pynmea2.parse(data)

latval = msg.lat

#parse the latitude and print

concatlat = “Lat:” + str(latval)

print(concatlat)

lcd.set\_cursor(0,0)

lcd.message(concatlat)

#parse the longitude and print

longval=msg.lon

concatlong=”Long:”+str(longval)

print(concatlong)

lcd.set\_cursor(0,1)

lcd.message(concatlong)

Time.sleep(0.5)

Except KeyboardInterrupt:

lcd.clear()

lcd.message(“Thank You”)

time.sleep(2)

Practical -8: +5v -4, i/p- 7, gnd-6

Connect led to breadboard positive-37 negetive-39

Command-

-pwd

wget hhtps://sourceforge.net/projects/webiopi/files/WebIOPi-0.7.1.tar.gz

tar xvzf WebIOPi-0.7.1.tar.gz

cd WebIOPi-0.7.1/

pwd

wget https://raw.githubusercontent.com/doublebind/raspi/master/webiopi-pi2bplus.patch

patch -p1 -I webiopi -po2bplus.patch

ls

sudo ./setup.sh

sudo reboot

sudo webiopi -d -c /etc/webiopi/config

(Now open browser and connect to <http://PI’s> ip add-192.168.1.36:8000)

After login click on gpio header link→set pin 37 as output

Pwd

ls

sudo nano /etc/webiopi/config

(Under http section of the config file, comment out doc-root line and change the path to your project file

)

sudo webiopi-passwd

sudo /etc/init.d/webiopi restart

sudo /etc/init.d/webiopi status

sudo /etc/init.d/webiopi stop

sudo update-rc.d webiopi defaults

sudo update-rc.d webiopi remove