

IOT LAB PROGRAMS

Lab program 1a

A Program to read a name and print Hello <name>

```
#print("Please Enter Your Name Here")  
  
name = input("Please Enter Your Name Here:\n") #get the input from console  
  
print("Hello " + name)
```

Lab program 1b

A Program to read two numbers and display sum, difference, product and division

```
print("Enter First Number")  
  
num1 = input()  
  
  
print("Enter Second Number")  
  
num2 = input()  
  
  
print("*****")  
print("Sum of Given numbers is : "+str(int(num1)+int(num2)))  
print("Difference of Given numbers is : "+str(int(num1)-int(num2)))  
print("Product of Given numbers is : "+str(int(num1)*int(num2)))  
print("Division of Given numbers is : "+str(int(num1)/int(num2)))  
print("*****")
```

Lab program 1C

A Program to calculate number of words and characters of a given string

```
print("Enter a sentence ")
```

```
sentence = input()
```

```
words = sentence.split()
```

```
word_count = 0
```

```
character_count = 0
```

```
for word in words:
```

```
    word_count += 1
```

```
    character_count += len(word)
```

```
print("Total Numbers of Words in the sentence are : ",word_count)
```

```
print("Total Numbers of characters in the sentence excluding spaces are :  
",character_count)
```

```
print("Total Numbers of characters in the sentence including spaces are :  
",character_count+word_count-1)
```

Lab program 1d

A Program to get Area of a selected shape.

'''

1. Rectangle

2. Triangle

3. Circle

'''

while True:

```
print("*****")
```

```
print("Select the Shape that you want to calculate Area")
```

```
print("""      1. Rectangle
```

```
      2. Triangle
```

```
      3. Circle
```

```
      4. Exit """)
```

```
choice = input()
```

```
if(choice == '1'):
```

```
    print("Enter the Width of the Rectangle in meters")
```

```
    width = int(input())
```

```
    print("Enter the height of the Rectangle in meters")
```

```
    height = int(input())
```

```
    print("The area of a Given Rectangle is ", width*height , " square  
meters ")
```

```
    continue
```

```
elif(choice == '2'):
```

```
    print("Enter the Base value of the Triangle in meters")
```

```
        base = int(input())
        print("Enter the height of the Triangle in meters")
        height = int(input())
        print("The area of a Given Rectangle is ", 0.5*base*height , " square
meters ")

        continue

    elif(choice == '3'):

        print("Enter the Radius of the Circle in meters")
        radius = int(input())
        print("The area of a Given Circle is ", 3.14*radius*radius , " square
meters ")

        continue

    elif(choice == '4'):

        break

    else:

        print("Please enter a valid number from the menu")

        continue

print("Thank You")
```

lab program 1e

A Program to print a name n times where name and n values has to be input from standard input

```
print("Enter Your name : ")
name = input()
print("Enter How many times you want to print your name")
n = int(input())
for i in range(n):
    print(name)
```

lab program 1f

A Program to Handle Divide By Zero Exception

```
print("Enter Numerator Value : ")
num1 = int(input())

print("Enter Denominator value : ")
num2 = int(input())

try:
    result = num1/num2
    print("The Division of Given Numbers is : ", result)
except ZeroDivisionError:
    print("Divide By zero Error. The Denominator should not be Zero")
```

lab program 1g

A Program to print current time with an interval of 10 seconds

```
import time
```

```
# seconds passed since epoch
```

```
for i in range(10):
```

```
    seconds = time.time() #it will give to you as a float values
```

```
    local_time = time.ctime(seconds) #it will gives to you as a current local time
```

```
    print("Local time:", local_time)
```

```
    time.sleep(10)
```

lab program 1h

```
## A Program to Read a file and print No of Words in each Line
```

```
file1 = open('myfile.txt', 'r') #to read file
```

```
Lines = file1.readlines() #to read the each line of file1 ,if we want only one line  
of file1 then write readlines(1)
```

```
#print(Lines) #we can able print also
```

```
i=0
```

```
for line in Lines:
```

```
    i += 1
```

```
    count = len(line.split())
```

```
    print("Line ", i, "No of Words ", count)
```

lab program 2

Program to Take input from two switches and switch on corresponding LEDs.

Connect J4A PIN-1,2,3,4,5 TO J4 PIN-1,2,3,4,5

Dont forget to short the jumper JP1

import time

import RPi.GPIO as gpio

gpio.setwarnings(False)

gpio.setmode(gpio.BOARD)

led1 = 15 #P15 # pin is connected to LED and it should be OUT

led2 = 13 #P13 # pin is connected to LED and it should be OUT

switch1 = 37 #P37 # pin is connected to SWITC and it should be IN

switch2 = 35 #P35 # pin is connected to SWITC and it should be IN

gpio.setup(led1,gpio.OUT,initial=1)

gpio.setup(led2,gpio.OUT,initial=1)

gpio.setup(switch1,gpio.IN)

gpio.setup(switch2,gpio.IN)

def glow_led(event):

if event == switch1 :

gpio.output(led1, False)

```

        time.sleep(3)

        gpio.output(led1, True)

    elif event == switch2 :

        gpio.output(led2, False)

        time.sleep(3)

        gpio.output(led2, True)

gpio.add_event_detect(switch1, gpio.RISING , callback = glow_led, bouncetime =
1)

gpio.add_event_detect(switch2, gpio.RISING , callback = glow_led, bouncetime =
1)

try:

    while(True):

        #to avoid 100% CPU usage

        time.sleep(1)

except KeyboardInterrupt:

    #cleanup GPIO settings before exiting

    gpio.cleanup()

```

lab program 3

ALS Bengaluru

Program to switch ON and Switch off led for corresponding delay times provided in file.

Connect J4A PIN-1,5 TO J4 PIN-1,5

Dont forget to short the jumper JP1

#OnTime and OffTime values can be read from ledintervals.txt file

```
import time
```

```
import RPi.GPIO as gpio
```

```
gpio.setwarnings(False)
```

```
gpio.setmode(gpio.BOARD)
```

```
led1 = 15    #P15    # pin is connected to LED and it should be OUT
```

```
gpio.setup(led1,gpio.OUT,initial=1)
```

```
file1 = open('ledintervals.txt', 'r')
```

```
Lines = file1.readlines()
```

```
ON_TIME = int(Lines[0].split("=")[1])
```

```
OFF_TIME = int(Lines[1].split("=")[1])
```

```
try:
```

```
    while(True):
```

```
        gpio.output(led1,False)
        time.sleep(ON_TIME)
        gpio.output(led1,True)
        time.sleep(OFF_TIME)
except KeyboardInterrupt:
    #cleanup GPIO settings before exiting
    gpio.cleanup()
```

lab program 4

```
# Program to switch on relay at given time using cron.
# Set crontab -e.
#minute Hour Day of the month month Day of the week <command to execute>
#example
#0 12 13 * * python3 /home/pi/MCASyllabus/10.py
```

```
import time

import RPi.GPIO as gpio

gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)

relay1 = 38    #P38

gpio.setup(relay1,gpio.OUT,initial=0)
```

try:

```
    gpio.output(relay1, True)
    print("Relay is Switched On. Please Press ctrl+c to exit")
    time.sleep(15)
    print("Relay is Switched Off.")
    gpio.output(relay1, False)
```

except KeyboardInterrupt:

```
    gpio.cleanup()
    print("Program Exited")
```

lab program 5

Program to capture a image form pi camera

#connect a Pi camera. and Enable camera from Raspberry pi configuration

#All the captured images will be stored at /home/pi/MCASyllabus/images folder.

```
from picamera import PiCamera
```

```
from time import sleep
```

```
import datetime
```

```
camera = PiCamera()
```

```
camera.start_preview()
current_date = datetime.datetime.now().strftime('%d-%m-%Y %H:%M:%S')
sleep(3)
camera.capture('/home/pi/Desktop/MCASyllabus/images/'+current_date+'.jpg')
camera.stop_preview()
print("Image captured")
#sleep(2)
```

Lab program 6

Program to controlling a light source using web page

#install the following command in terminal

#sudo apt-get install python3-flask

Create a folder "templates" and write your html pages in this folder.

```
import RPi.GPIO as GPIO
```

```
import time
```

```
import datetime
```

```
led = 13 #P15
```

```
GPIO.setmode(GPIO.BOARD)
```

```
GPIO.setwarnings(False)
```

```
GPIO.setup(led, GPIO.OUT,initial=1) #Initialising GPIO pins as output
GPIO.setup(led,GPIO.OUT)
```

```
from flask import Flask, render_template #Importing Flask
```

```
app = Flask(__name__)
```

```
@app.route('/') #Simple Hello World Route
```

```
def hello_world():
```

```
    return render_template('web.html')
```

```
@app.route('/redledon') #Route for Turning RedLed On
```

```
def redledon():
```

```
    GPIO.output(13, GPIO.LOW) #Turning Pin 31 --> Red Led HIGH
```

```
    now = datetime.datetime.now()
```

```
    timeString = now.strftime("%Y-%m-%d %H:%M")
```

```
    templateData = {
```

```
        'status' : 'ON',
```

```
        'time': timeString
```

```
    }
```

```

return render_template('web.html', **templateData)

@app.route('/redledoff') #Route for Turning RedLed Off
def redledoff():
    GPIO.output(13, GPIO.HIGH) #Turning Pin 31 --> Red Led LOW
    now = datetime.datetime.now()
    timeString = now.strftime("%Y-%m-%d %H:%M")
    templateData = {
        'status': 'OFF',
        'time': timeString
    }
    return render_template('web.html', **templateData)

if __name__ == "__main__":
    app.run(debug = True, port = 4000, host='172.20.10.5')

```

lab program 7

```

# Program to implement intruder system that sends an alarm to given gmail.
# Commands to install
#sudo apt-get install ssmtp
#sudo apt-get install mailutils
#Please use General email for sending emails as it is having few security issues.

```

```
import RPi.GPIO as gpio
```

```
import picamera
```

```
import time
```

```
import smtplib
```

```
from email.mime.multipart import MIMEMultipart
```

```
from email.mime.text import MIMEText
```

```
from email.mime.base import MIMEBase
```

```
from email import encoders
```

```
from email.mime.image import MIMEImage
```

```
fromaddr = "alsdatasave@gmail.com" # change the email address accordingly
```

```
toaddr = "alsdatasave@gmail.com"
```

```
mail = MIMEMultipart()
```

```
mail['From'] = fromaddr
```

```
mail['To'] = toaddr
```

```
mail['Subject'] = "Attachment"
```

```
body = "Please find the attachment"
```

```
led=15
```

```

pir=12
HIGH=1
LOW=0

gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)
gpio.setup(led, gpio.OUT)      # initialize GPIO Pin as outputs
gpio.setup(pir, gpio.IN)      # initialize GPIO Pin as input
data=""

def sendMail(data):
    mail.attach(MIMEText(body, 'plain'))
    print(data)
    dat='%s.jpg'%data
    print(data)
    attachment = open(dat, 'rb')
    image=MIMEImage(attachment.read())
    attachment.close()
    mail.attach(image)
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(fromaddr, "alsdatasave2017")
    text = mail.as_string()
    server.sendmail(fromaddr, toaddr, text)

```



```
server.quit()
```

```
def capture_image():
```

```
    data= time.strftime("Image was captured on %H:%M:%S|%d_%b_%Y")
```

```
    camera.start_preview()
```

```
    time.sleep(5)
```

```
    print(data)
```

```
    camera.capture('%s.jpg'%data)
```

```
    camera.stop_preview()
```

```
    time.sleep(1)
```

```
    sendMail(data)
```

```
gpio.output(led , 0)
```

```
camera = picamera.PiCamera()
```

```
camera.rotation=180
```

```
camera.awb_mode= 'auto'
```

```
camera.brightness=55
```

```
while 1:
```

```
    if gpio.input(pir)==1:
```

```
        gpio.output(led, HIGH)
```

```
        capture_image()
```

```
        while(gpio.input(pir)==1):
```

```
            time.sleep(1)
```

```
else:
```

```
    gpio.output(led, LOW)
```

```
    time.sleep(0.01)
```

Lab program 8

Program to Read Light Status from remote place.

```
#sudo apt-get install python-flask
```

```
import time
```

```
import RPi.GPIO as gpio
```

```
from flask import Flask, render_template #Importing Flask
```

```
import datetime
```

```
app = Flask(__name__)
```

```
gpio.setwarnings(False)
```

```
gpio.setmode(gpio.BOARD)
```

```
led1 = 13 # pin is connected to LED and it should be OUT
```

```
switch1 = 35 # pin is connected to SWITC and it should be IN
```

```
gpio.setup(led1,gpio.OUT,initial=1)
```

```
gpio.setup(switch1,gpio.IN)
```

```
light_status = "OFF"
```

```
def glow_led(event):
```

```
    print("Entered Here")
```

```
    global light_status
```

```
    if event == switch1 and light_status == "OFF":
```

```
        gpio.output(led1, False)
```

```
        light_status = "ON"
```

```
    elif event == switch1 and light_status == "ON":
```

```
        gpio.output(led1, True)
```

```
        light_status = "OFF"
```

```
@app.route('/') #Route for Turning RedLed On
```

```
def ledstatus():
```

```
    now = datetime.datetime.now()
```

```
    timeString = now.strftime("%H:%M %d-%m-%Y")
```

```
    templateData = {
```

```
'status' : light_status,  
  
'time': timeString  
  
}  
  
return render_template('lightstatus.html', **templateData)
```

```
gpio.add_event_detect(switch1, gpio.RISING , callback = glow_led, bouncetime =  
100)  
  
app.run(debug = True, port = 4000, host='169.254.21.219')
```

lab program 9 a

```
#!/usr/bin/env python3  
  
#server program to read gas values  
  
#sudo pip3 install Adafruit_MCP3008
```

```
import socket  
  
import Adafruit_MCP3008  
  
import Adafruit_GPIO.SPI as SPI  
  
import time
```

```
HOST = '169.254.21.219' # Standard loopback interface address (localhost)  
  
PORT = 4000      # Port to listen on (non-privileged ports are > 1023)
```

```
SPI_DEVICE = 0
```

```
SPI_PORT = 0
```

```
mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_PORT,  
SPI_DEVICE))
```

```
try:
```

```
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
```

```
        s.bind((HOST, PORT))
```

```
        s.listen()
```

```
        conn, addr = s.accept()
```

```
        with conn:
```

```
            print('Connected by', addr)
```

```
            while True:
```

```
                value = mcp.read_adc(0)
```

```
                print("Gas Value ", value , "units")
```

```
                if(value >300):
```

```
                    data = "Alert".encode('utf-8')
```

```
                    conn.sendall(data)
```

```
                time.sleep(3)
```

```
except KeyboardInterrupt:
```

```
    s.close()
```

```
    GPIO.cleanup()
```

Lab program 9b

#client program to alert through buzzer.

#sudo pip3 install Adafruit_MCP3008

import socket

import RPi.GPIO as GPIO

import time

Buzzer = 36

HOST = '169.254.233.6' # The server's hostname or IP address

PORT = 4000 # The port used by the server

GPIO.setmode(GPIO.BOARD)

GPIO.setup(36, GPIO.OUT)

GPIO.setwarnings(False)

try:

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:

s.connect((HOST,PORT))

while True:

```
data = s.recv(1024).decode('utf-8')
print(data)
if(str(data) == 'Alert'):
    print("ALert! Gas Leakage detected")
    GPIO.output(36, True)
    time.sleep(3)
    GPIO.output(36, False)
    time.sleep(3)
except KeyboardInterrupt:
    s.close()
    GPIO.cleanup()
```