CSLR61 – EMBEDDED SYSTEMS AND LABORATORY
COMPLETE CODE MANUAL
BY PRAJWAL SUNDAR, 106121092

1) Introduction

```
print("Welcome to GPIO Programming with Raspberry PI!")
```

2) LED Patterns

```
# import modules
import RPi.GPIO as GPIO
from time import sleep

# setup pins
GPIO.setmode(GPIO.BOARD)
leds = [3, 5, 7]
for led in leds:
    GPIO.setup(led, GPIO.OUT)

while True:
  for led in leds:
    GPIO.output(led, GPIO.HIGH)
    sleep(1)
  for led in leds:
    GPIO.output(led, GPIO.LOW)
    sleep(1)
```

3) Intensity Controlling

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

led = 3
GPIO.setup(led, GPIO.OUT)
pwm = GPIO.PWM(led, 100)
pwm.start(0)

while True:
  for i in range(100):
    pwm.ChangeDutyCycle(i)
    sleep(0.01)
  for i in range(100, 0, -1):
    pwm.ChangeDutyCycle(i)
    sleep(0.01)
```

4) Interrupt Programming

```
import RPi.GPIO as GPIO
from time import sleep
```

```
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

led = 3
GPIO.setup(led, GPIO.OUT)

button = 5
GPIO.setup(button, GPIO.IN, pull_up_down = GPIO.PUD_UP)

while True:
  GPIO.wait_for_edge(button, GPIO.RISING)
  GPIO.output(led, GPIO.HIGH)
  GPIO.wait_for_edge(button, GPIO.RISING)
  GPIO.output(led, GPIO.LOW)
```

5) Temperature Sensor

```
from adafruit_dht import DHT22
import board
from time import sleep

dht = DHT22(board.D22)

while True:
    try:
        t = dht.temperature
        h = dht.humidity
        if t is not None and h is not None:
            print("Temperature =", t, "and Humidity =", h, "%")
        else:
            print("Failed to retrieve readings correctly !")
    except RuntimeError as error:
        print(error.args[0])
    sleep(2)
```

6) 7-Segment Display

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

leds = [19, 21, 22, 23, 24, 26, 28]
for led in leds:
    GPIO.setup(led, GPIO.OUT)

M = [
    [1, 1, 1, 1, 1, 1, 0],
    [0, 1, 1, 0, 0, 0, 0],
    [1, 1, 0, 1, 1, 0, 1],
    [1, 1, 1, 1, 0, 0, 1],
    [0, 1, 1, 0, 0, 1, 1],
    [1, 0, 1, 1, 0, 1, 1],
    [1, 0, 1, 1, 1, 1, 1],
    [1, 1, 1, 0, 0, 0, 0],
    [1, 1, 1, 1, 1, 1, 1],
    [1, 1, 1, 1, 0, 1, 1]
] # 7-segment display matrix

n = 0
while True:
    for led in leds:
        GPIO.output(led, GPIO.LOW)
    for i in range(8):
        if M[n][i] == 1:
            GPIO.output(leds[i], GPIO.HIGH)
    sleep(1)
    n = (n+1) % 10
```

7) PIR Sensor

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

pir = 17
GPIO.setup(pir, GPIO.IN)

while True:
    if GPIO.input(pir):
        print("Motion Detected")
    else:
        print("No Motion Detected")
    sleep(1)
```

8) Ultrasonic Sensor

```
import RPi.GPIO as GPIO
from time import sleep, time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

trigger = 23
GPIO.setup(trigger, GPIO.OUT)
echo = 24
GPIO.setup(echo, GPIO.IN)
```

```
while True:
    GPIO.output(trigger, GPIO.HIGH)
    sleep(0.01)
    GPIO.output(trigger, GPIO.LOW)

    while GPIO.input(echo) == GPIO.LOW:
        start = time()
    while GPIO.input(echo) == GPIO.HIGH:
        end = time()

    roundtime = end-start
    dis = round(roundtime*17150, 2)
    print("Distance:", dis, "cm")
```

9) Smoke Sensor

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

smoke = 7
GPIO.setup(smoke, GPIO.IN)

while True:
    level = GPIO.input(smoke)
    print("Smoke Level:", level)
    sleep(1)
```

10) LCD Display

```
import RPi.GPIO as GPIO
from RPi_LCD import LCD
from time import sleep

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

RS, E, D4, D5, D6, D7, BCK = 22, 18, 16, 11, 12, 15, 13
LINES, WIDTH = 2, 16

lcd = LCD(RS, E, D4, D5, D6, D7, BCK, LINES, WIDTH)
lcd.clear()

lcd.display_string("HELLO WORLD", 1)
lcd.display_string("LCD DISPLAY", 2)
```

11) SQLite Database

```
import sqlite3
conn = sqlite3.connect('database.db')
```

```
c = conn.cursor()
c.execute("INSERT INTO `users` (`rno`, `name`) VALUES ('106121092',
'Prajwal')")
c.commit()
```

12) Web-Interface

```
from flask import Flask, render_template
from flask_cors import CORS

import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
led = 3
GPIO.setup(led, GPIO.OUT)

app = Flask(__name__)
cors = CORS(app)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/e')
def e():
    GPIO.output(led, GPIO.HIGH)
    return "Light Turned ON !"

@app.route('/d')
def d():
    GPIO.output(led, GPIO.LOW)
    return "Light Turned OFF !"

app.config['CORS_HEADERS'] = 'Content-Type'
if __name__ == '__main__':
    app.run(debug = True, port = 8000, host = '127.0.0.1')
```

index.html (Within folder ‘templates’)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>LED Controller</title>
</head>
<body>
    <h1>LED Controller</h1>
    <button id="on">Turn ON</button>
    <button id="off">TURN OFF</button>
    <script>
```

```
        document.getElementById('on').addEventListener('click', function() {
            const xhttp = new XMLHttpRequest()
            xhttp.onload = function() {
                alert(this.responseText)
            }
            xhttp.open("GET", "/e", true)
            xhttp.send()
        })
        document.getElementById('off').addEventListener('click', function() {
            const xhhtp = new XMLHttpRequest()
            xhttp.onload = function() {
                alert(this.responseText)
            }
            xhttp.open("GET", "/d", true)
            xhttp.send()
        })
    </script>
</body>
</html>
```