

```
1 # Blink LED
2 |
3 from machine import Pin
4 import time
5 led=Pin(13,Pin.OUT)
6 while True:
7     led.value(1)
8     time.sleep(5)
9     led.value(0)
10    time.sleep(5)
11
```

```
#Push Button (Toggle Switch)

from machine import Pin
led = Pin(2, Pin.OUT)
switch = Pin(15, Pin.IN, Pin.PULL_UP)
led_state = False
last_switch = switch.value()

while True:
    current_switch = switch.value()
    if last_switch == 1 and current_switch == 0:
        led_state = not led_state
        led.value(led_state)
    last_switch = current_switch
```

```
# Potentiometer values on serial monitor

from machine import Pin,ADC
from time import sleep
pot=ADC(Pin(34))
pot.atten(ADC.ATTN_11DB)
pot.width(ADC.WIDTH_12BIT)
while True:
    pot_value=pot.read()
    print("pot value:",pot_value)
    sleep(0.5)
```

```
1 # Controlling Brightness of led with potentiometer
2
3 from machine import ADC,Pin,PWM
4 import time
5
6 pot=ADC(Pin(34))
7 pot.atten(ADC.ATTN_11DB)
8 pot.width(ADC.WIDTH_10BIT)
9
10 led=PWM(Pin(15))
11 led.freq(60)
12
13 while True:
14     pot_value=pot.read()
15     led.duty(pot_value)
16     print(pot_value)
17
18     time.sleep(0.5)
19
```

```
1 # Display String on LCD
2
3 import machine
4 from machine import Pin, SoftI2C
5 from lcd_api import LcdApi
6 from i2c_lcd import I2cLcd
7 from time import sleep
8
9 I2C_ADDR = 0x27
10 totalRows = 2
11 totalColumns = 16
12
13 i2c = SoftI2C(scl=Pin(22), sda=Pin(21), freq=100000)      #I2C for ESP32
14 #i2c = I2C(scl=Pin(5), sda=Pin(4), freq=100000)          #I2C for ESP8266
15
16 lcd = I2cLcd(i2c, I2C_ADDR, totalRows, totalColumns)
17
18 while True:
19     lcd.putstr("I2C LCD TEST")
20     sleep(2)
21     lcd.move_to(0,1)
22     lcd.putstr("Hello World!")
23     sleep(5)
24     lcd.clear()
25 |
```

```
1 # Interface a DHT11 sensor and display the values on an LCD
2
3 import machine
4 from machine import Pin, SoftI2C
5 from lcd_api import LcdApi
6 from i2c_lcd import I2cLcd
7 from time import sleep
8 import dht
9
10 I2C_ADDR = 0x27
11 totalRows = 2
12 totalColumns = 16
13
14 i2c = SoftI2C(scl=Pin(22), sda=Pin(21), freq=100000)
15 lcd = I2cLcd(i2c, I2C_ADDR, totalRows, totalColumns)
16
17 dht11 = dht.DHT11(Pin(13,Pin.IN))
18
19 while True:
20     dht11.measure()
21     temp = dht11.temperature()
22     humid = dht11.humidity()
23     print("temperature=", temp)
24     print("Humidity=", humid)
25
26     lcd.putstr("Temperature="+str(temp)+"c")
27     lcd.move_to(0,1)
28     lcd.putstr("Humidity="+str(humid)+"%")
29     sleep(1)
30     lcd.clear()
31
```

```
# Log DHT11 sensor data to the ThingSpeak server with MQTT Module

from machine import Pin
import network
from time import sleep
import sys
from umqtt.simple import MQTTClient
from dht import DHT11

SSID='iot1'
PWD='iot12345'
clientId='MwMKGisTDx8eDzkLKDMShgs'
server='mqtt3.thingspeak.com'
username='MwMKGisTDx8eDzkLKDMShgs'
password='uPKe7Xp3kYluTEDX7Fs81ZMJ'
channelId='2676576'

topic='channels/'+channelId+'/publish'
topic=bytes(topic,'utf-8')
d=DHT11(Pin(15))

client=MQTTClient(client_id=clientId,server=server,user=username,password=password)

def dhtData():
    d.measure()
    t=d.temperature()
    h=d.humidity()
    return t,h

def connectWifi():
    wifi = network.WLAN(network.STA_IF)
    wifi.active(False)
    sleep(0.5)
    wifi.active(True)
    wifi.connect('iot1','iot12345')
    sleep(2)
```

```
36     sleep(2)
37     if wifi.isconnected():
38         print('Connected to Wi-Fi')
39         ip = wifi.ifconfig()
40         print(f"IP Address: {ip}")
41     else:
42         print('Wi-Fi connection failed')
43         sys.exit()
44
45 connectWifi()
46
47 try:
48     client.connect()
49     print('client connected to mqtt server')
50 except:
51     print('client is not connected')
52     sys.exit()
53
54 while True:
55     temp,hum=dhtData()
56     msg='&field1={}&field2={}'.format(temp,hum)
57     msg=bytes(msg,'utf-8')
58     client.publish(topic,msg)
59     print(temp,hum)
60     sleep(20)
```

```
1 #Log DHT11 sensor data to the ThingSpeak server with urequests module
2
3 from machine import Pin
4 from dht import DHT11
5 import network
6 import urequests
7 import sys
8 from time import sleep
9
10 SSID = 'iot'
11 PWD = 'iot12345'
12 WRITE_API_KEY = '9N38LS2WISRZQP0R'
13 BASE_URL = "https://api.thingspeak.com/update"
14
15 d = DHT11(Pin(15))
16
17 def dhtData():
18     d.measure()
19     t = d.temperature()
20     h = d.humidity()
21     return t, h
22
23 def connectWifi():
24     wifi = network.WLAN(network.STA_IF)
25     wifi.active(False)
26     sleep(0.5)
27     wifi.active(True)
28     wifi.connect('iot','iot12345')
29     sleep(2)
30     if wifi.isconnected():
```

```
29 sleep(2)
30 if wifi.isconnected():
31     print('Connected to Wi-Fi')
32     ip = wifi.ifconfig()
33     print(f"IP Address: {ip}")
34 else:
35     print('Wi-Fi connection failed')
36
37 def sendData(temp, humid):
38     url = f"{BASE_URL}?api_key={WRITE_API_KEY}&field1={temp}&field2={humid}"
39     response = urequests.get(url)
40     response.close()
41
42 connectWifi()
43
44 while True:
45     temp, humid = dhtData()
46     print(f"Temp: {temp}, Humid: {humid}%")
47     sendData(temp, humid)
48     sleep(15)
```

```
# Control an LED using the Adafruit-Io server

from machine import Pin
import network
from time import sleep
import sys
from umqtt.simple import MQTTClient

SSID='iot1'
PWD='iot12345'

clientId='cont12345'
server='io.adafruit.com'
username='shriyanarahari'
password='aio_ZNHS4183kUdJ9H4vTcLxav1uXUPs'
feedkey='LED'

topic=username+'/feeds/'+feedkey
topic=bytes(topic,'utf-8')

led=Pin(2,Pin.OUT)
client=MQTTClient(client_id=clientId,server=server,user=username,password=password)

def CBF(topic,msg):
    msg=str(msg,'utf-8')
    if msg=='1':
        led.on()
        print('led is on')
    else:
        led.off()
        print('led is off')

def connectWifi():
    wifi = network.WLAN(network.STA_IF)
    wifi.active(False)
    sleep(0.5)
    wifi.active(True)
```

```
37 wifi.active(True)
38 wifi.connect('iot1','iot12345')
39 sleep(2)
40 if wifi.isconnected():
41     print('Connected to Wi-Fi')
42     ip = wifi.ifconfig()
43     print(f"IP Address: {ip}")
44 else:
45     print('Wi-Fi connection failed')
46     sys.exit()
47
48 connectWifi()
49
50 try:
51     client.connect()
52     print('client connected to mqtt server')
53 except:
54     print('client is not connected')
55     sys.exit()
56
57 client.set_callback(CBF)
58 client.subscribe(topic)
59 while True:
60     client.check_msg()
```