

```

5 SETUP CLOUD PLATFORM TO LOG DATA

import network
import urequests
import utime
import machine
import dht
SSID = "Wokwi-GUEST"
PASSWORD = ""
API_KEY = "6LZYV2GJUJ58G4CZ"
dht_sensor = dht.DHT22(machine.Pin(16))

def connect_wifi():
    """Connects the Raspberry Pi Pico W to the specified Wi-Fi network."""
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        print(f"Connecting to Wi-Fi network '{SSID}'...")
        wlan.connect(SSID, PASSWORD)
        # Wait for connection with a timeout
        max_attempts = 20
        while not wlan.isconnected() and max_attempts > 0:
            utime.sleep(0.5)
            print(".", end="")
            max_attempts -= 1
        print()
    if wlan.isconnected():
        print("Connected!")
        print('Network configuration:', wlan.ifconfig())
    else:
        print("Failed to connect to Wi-Fi.")

def send_data():
    """Measures T/H and sends the data to ThingSpeak via HTTP GET request."""
    try:
        dht_sensor.measure()
        temp = dht_sensor.temperature()
        humidity = dht_sensor.humidity()
    except OSError as e:
        print(f"Error reading DHT22 sensor: {e}. Check wiring.")
        return
    url =
    f"https://api.thingspeak.com/update?api_key={API_KEY}&field1={temp}&field2={humidity}"
    try:
        response = urequests.get(url)
        if response.text == '0':
            print(f"Failed to log data to ThingSpeak. Check your ThingSpeak Channel setup.")
        else:
            print(f"Data Logged Successfully. Entry: {response.text}. Temp={temp}°C, Humidity={humidity}%")
            response.close()
    except Exception as e:
        print(f"Network error sending data: {e}")
    # --- Main Program Execution ---
    if __name__ == '__main__':
        connect_wifi()
        if
network.WLAN(network.STA_IF).isconnect
ed():
        while True:
            send_data()
            utime.sleep(15) # Wait 15 seconds before the next reading
        else:
            print("Program halted. Wi-Fi connection failed.")
-----
```

**6)LOG DATA USING RASPBERRY PI UPLOAD TO CLOUD PLATFORM**

```

#include <WiFi.h>
#include "DHTesp.h"

#include "ThingSpeak.h"
const int DHT_PIN = 15;
const char* WIFI_NAME = "Wokwi-GUEST";
const char* WIFI_PASSWORD = "";
const int myChannelNumber = your
channel;
const char* myApiKey = "your key"; //
const char* server = "api.thingspeak.com";
DHTesp dhtSensor;
WiFiClient client;
void setup() {
    Serial.begin(115200);
    dhtSensor.setup(DHT_PIN,
DHTesp::DHT22);
    WiFi.begin(WIFI_NAME,
WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED){
        delay(1000);
        Serial.println("Wifi not connected"); // Print a message if WiFi is not connected }
    Serial.println("Wifi connected !"); // Print a message if WiFi is connected
    Serial.println("Local IP: " +
String(WiFi.localIP()));
    WiFi.mode(WIFI_STA);
    ThingSpeak.begin(client); }
void loop() {
    TempAndHumidity data =
dhtSensor.getTempAndHumidity();
    ThingSpeak.setField(1,data.temperature);
    ThingSpeak.setField(2,data.humidity);
    int status =
ThingSpeak.writeFields(myChannelNumber
,myApiKey);
    Serial.println("Temp: " +
String(data.temperature, 2) + "°C"); // Print the temperature value with 2 decimal places
    Serial.println("Humidity: " +
String(data.humidity, 1) + "%"); // Print the humidity value with 1 decimal place
    if(status == 200){
```

```

Serial.println("Data pushed
successfully"); // Print a message if the
data was successfully pushed to
ThingSpeak

}else{

Serial.println("Push error" +
String(status)); }

Serial.println("---");
delay(10000);

7) SOIL MOISTURE

#define sensorPin A0#define sensorPower
7#define LED_PIN 8

#define WET_VALUE 0  #define
DRY_VALUE 511 void setup() {

pinMode(sensorPower, OUTPUT);
pinMode(LED_PIN, OUTPUT);
Serial.begin(9600);}

void loop() {

int val = readSensor();
Serial.print("Analog Value: ");
Serial.print(val);

int constrainedVal = constrain(val,
WET_VALUE, DRY_VALUE);

int moisturePercent =
map(constrainedVal, DRY_VALUE,
WET_VALUE, 0, 100);
Serial.print(", Moisture: ");

Serial.print(moisturePercent);
Serial.println("%");

if (moisturePercent < 30) {
Serial.println("Low moisture detected!
Water the plant");

digitalWrite(LED_PIN, HIGH);
} else {
digitalWrite(LED_PIN, LOW);
} delay(1000);

Int readSensor() {

digitalWrite(sensorPower, HIGH);
delay(10);

int val = analogRead(sensorPin);

```

---

```

digitalWrite(sensorPower, LOW);
return val;}

9)HEALTHCARE SYSTEM

#include "DHTesp.h"
const int DHT_PIN = 12;
const int pulsePin = 34;
const int ledPin = 2;
DHTesp dhtSensor;
void setup() {
Serial.begin(115200);
dhtSensor.setup(DHT_PIN,
DHTesp::DHT22);
pinMode(ledPin, OUTPUT);void loop() {

TempAndHumidity data =
dhtSensor.getTempAndHumidity();
int pulseValue = analogRead(pulsePin);
int heartRate = map(pulseValue, 0, 4095,
60, 120);
if (heartRate > 90) digitalWrite(ledPin,
HIGH);
else digitalWrite(ledPin, LOW);
Serial.println("== Remote Patient
Monitoring ==");
Serial.print("Temp: ");
Serial.print(data.temperature, 1);
Serial.println("°C");
Serial.print("Humidity: ");
Serial.print(data.humidity, 1);
Serial.println("%");
Serial.print("Heart Rate: ");
Serial.print(heartRate); Serial.println("BPM");
Serial.println("-----");
DELAY(2000);

```

---

#### **8) VEHICLE PARKING SYSTEM**

```

#define TRIG_PIN 9
#define ECHO_PIN 10
#define LED_PIN 6
void setup() {
pinMode(TRIG_PIN, OUTPUT);

```

```

pinMode(ECHO_PIN, INPUT);
pinMode(LED_PIN, OUTPUT);
Serial.begin(9600);

}

void loop() {
long duration;
int distance;
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
duration = pulseIn(ECHO_PIN, HIGH);
distance = duration * 0.034 / 2;
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
if (distance > 0 && distance < 10) {
digitalWrite(LED_PIN, HIGH);
} else {
digitalWrite(LED_PIN, LOW);}
delay(500);}
delay(2000);}


```

---

#### **10) DATA ACQUISITION (temp,hum,gas,buzzer)**

```

#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
#define GAS_SENSOR A0
#define LED 6
#define BUZZER 7
void setup() {
Serial.begin(9600);
dht.begin();
pinMode(GAS_SENSOR, INPUT);

```

```

pinMode(LED, OUTPUT);
pinMode(BUZZER, OUTPUT);
Serial.println("IoE Environmental
Monitoring System Started...");

Serial.println("Reading Temperature,
Humidity, and Gas levels...");

void loop() {
    // Read Sensor Values
    float temperature =
dht.readTemperature();

    float humidity = dht.readHumidity();

    int gasValue = analogRead(GAS_SENSOR);

    Serial.print("Temperature: ");
    Serial.print("30.63");

    Serial.print(" °C | Humidity: ");
    Serial.print("69.00");
    Serial.print(" % | Gas: ");
    Serial.println(gasValue);

    if (temperature > 40 || humidity > 70 || gasValue > 300) {
        digitalWrite(LED, HIGH);
        digitalWrite(BUZZER, HIGH);

        Serial.println("⚠️ ALERT! Unsafe
conditions detected!");
    } else {
        digitalWrite(LED, LOW);
        digitalWrite(BUZZER, LOW);}

        delay(2000);
    }

```

### **3)RASPERRY PI LED BLINK**

```

//Task #1 Blink an LED Using Wokwi
Simulation
void setup() { pinMode(17, OUTPUT);
Serial1.begin(115200); Serial1.println("LED
Blink Start"); } void loop() { digitalWrite(17,
HIGH); delay(1000); digitalWrite(17, LOW);
delay(1000); } //Task #2 BlinkTwo LEDs
AlternatelyUsing Wokwi void setup() {
pinMode(17, OUTPUT); pinMode(18,
OUTPUT); } void loop() {
digitalWrite(17, HIGH); digitalWrite(18,
LOW); delay(1000); digitalWrite(17,
LOW); digitalWrite(18, HIGH); delay(1000);
} //Task #3 - Control an LED

```

**using a Push Button (Raspberry Pi Pico - Wokwi)**

```

void setup() { pinMode(15, OUTPUT);
pinMode(14, INPUT_PULLUP); void
loop() { int buttonState = digitalRead(14); if
(buttonState == LOW) { digitalWrite(15,
HIGH); } else { digitalWrite(15, LOW); }
delay(100)

```

**2) STEPPER MOTOR SERVO** void setup() {  
pinMode(8, OUTPUT); pinMode(9,
OUTPUT); pinMode(10, OUTPUT);
pinMode(11, OUTPUT);} void loop() {
digitalWrite(8, HIGH);
delay(5);digitalWrite(8,
LOW);delay(5);digitalWrite(
9,HIGH);delay(5);digitalWrit
e(9,
LOW);delay(5); digitalWrite(10,
HIGH);delay(5);digitalWrite( 10, LOW);
delay(5);digitalWrite(11, HIGH); delay(5);
digitalWrite(11, LOW); delay(5);

**To interface a temperature sensor (e.g.,
LM35) with Arduino** const int sensorPin =
A0; void setup() { Serial.begin(9600);
Serial.println("Temperature Sensor
Interface Starting...");} void loop() { int
sensorValue = analogRead(sensorPin); float
voltage = sensorValue \* (5.0 / 1023.0);
float temperature = voltage \* 100.0;
Serial.print("Temperature:")
Serial.print(temperature); Serial.println(
"°C"); delay(1000);}

**Buzzer and sensor** const int sensorPin =
A0; const int buzzerPin = 8; float threshold
= 100.0; void setup() { Serial.begin(9600);
pinMode(buzzerPin, OUTPUT);
Serial.println("LM35 Temperature Sensor
with Buzzer Alert System Starting...");} void
loop() { int sensorValue =
analogRead(sensorPin); float voltage =
sensorValue \* (5.0 / 1023.0); float
temperature = voltage \* 100.0;
Serial.print("Temperat");
Serial.print(temperature); Serial.println(
"°C"); if (temperature > threshold) {
digitalWrite(buzzerPin, HIGH);
Serial.println("ALERT: High Temperature!");}
else{digitalWrite(buzzerPi n,
LOW);}delay(1000);

### **4) SENSORS AND ACTUATORS**

#### **a )BLINK LED USING WOKWI**

```

import machine import utime LED_PIN =
machine.Pin(17, machine.Pin.OUT) while
True: LED_PIN.value(1) utime.sleep(1)
LED_PIN.value(0) utime.sleep(1) b)
INTERFACE US SENSOR (HC-SR04) TO
MEASURE DISTANCE import
machine import utime trigger =

```

```

machine.Pin(2, machine.Pin.OUT) echo =
machine.Pin(3, machine.Pin.IN) def
get_distance(): trigger.low()
utime.sleep_us(2) trigger.high()
utime.sleep_us(10) trigger.low() while
echo.value() == 0: signaloff =
utime.ticks_us() while echo.value() == 1:
signalon = utime.ticks_us() timepassed =
signalon - signaloff distance = (timepassed
* 0.0343) / 2 return distance while True:
dist = round(get_distance(), 2)
print("Distance:", dist, "cm")
utime.sleep(1) c)CONTROL SERVO
USING RASPEERRY PI
import machine import utime servo =
machine.PWM(machine.Pin(15))
servo.freq(50) def set_angle(angle): duty =
int(((angle / 180) * 5000) + 2500)
servo.duty_u16(duty) utime.sleep(1) while
True: set_angle(0) utime.sleep(1)
set_angle(90) utime.sleep(1)
set_angle(180) utime.sleep(1)
d)READ HUMI AND TEMP USING DHT11
SENSOR import
machine import utime import dht
dht_sensor = dht.DHT22(machine.Pin(16))
while True: dht_sensor.measure() temp =
dht_sensor.temperature() humidity =
dht_sensor.humidity()
print("Temperature:", temp, "°C")
print("Humidity:", humidity, "%")
utime.sleep(2)

```

### **EXP 1)**

**Blink the onboard LED in Uno** void setup() { pinMode(13, OUTPUT);} void loop() {
digitalWrite(13, HIGH); delay(1000);
digitalWrite(13, LOW); delay(1000);} **Blink
the user LED in Uno using the port D8** void
setup() { pinMode(8, OUTPUT);} void loop() {
{ digitalWrite(8, HIGH); delay(1000);
digitalWrite(8, LOW); delay(1000);} **Blink
the Two user LED inUno using the port
D8,D9** void setup() { pinMode(8, OUTPUT);
pinMode(9, OUTPUT); }void loop() {
digitalWrite(8, HIGH); digitalWrite(9, LOW);
delay(1000); digitalWrite(8, LOW);
digitalWrite(9, HIGH) delay(1000); }