SPRINT - 2

|  |  |
| --- | --- |
| Date | 8 NOV 2022 |
| Team ID | PNT2022TMID19565 |
| Project Name | Smart Waste Management |
| System for Metropolitan Cities |

# CODE FOR DETECTING BIN LEVEL AND DISPLAYING IT IN IBM CLOUD:

**esp32-dht22.ino:**

#include <WiFi.h> // library for wifi

#include <PubSubClient.h> // library for MQTT #include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 20, 4);

// credentials of IBM Accounts

---

#define ORG "mldk59" // IBM organisation id

#define DEVICE\_TYPE "pythoncode" // Device type mentioned in ibm watson iot platform

#define DEVICE\_ID "252525" // Device ID mentioned in ibm watson iot platform

#define TOKEN "QZqODYo6U\*Q6b+IpuC" // Token

// customise above values

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform and format in which data to be send

char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is test format of strings

char authMethod[] = "use-token-auth"; // authentication method char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID; //Client id

//

WiFiClient wifiClient; // creating instance for wificlient PubSubClient client(server, 1883, wifiClient);

#define ECHO\_PIN 12

#define TRIG\_PIN 13 float dist;

void setup()

{

**Serial**.begin(115200); pinMode(LED\_BUILTIN, OUTPUT); pinMode(TRIG\_PIN, OUTPUT); pinMode(ECHO\_PIN, INPUT);

//pir pin pinMode(34, INPUT);

//ledpins pinMode(23, OUTPUT); pinMode(2, OUTPUT); pinMode(4, OUTPUT); pinMode(15, OUTPUT);

lcd.init(); lcd.backlight(); lcd.setCursor(1, 0); lcd.print(""); wifiConnect(); mqttConnect();

}

float readcmCM()

{

digitalWrite(TRIG\_PIN, LOW); delayMicroseconds(2); digitalWrite(TRIG\_PIN, HIGH); delayMicroseconds(10); digitalWrite(TRIG\_PIN, LOW);

int duration = pulseIn(ECHO\_PIN, HIGH); return duration \* 0.034 / 2;

}

void loop()

{

lcd.clear();

publishData(); delay(500);

if (!client.loop())

{

mqttConnect(); // function call to

connect to IBM

}

}

/\* -retrieving to cloud

\*/

void wifiConnect()

{

**Serial**.print("Connecting to "); **Serial**.print("Wifi"); WiFi.begin("Wokwi-GUEST", "", 6); while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

**Serial**.print(".");

}

**Serial**.print("WiFi connected, IP address: ");

**Serial**.println(WiFi.localIP());

}

void mqttConnect()

{

if (!client.connected())

{

**Serial**.print("Reconnecting MQTT client to ");

**Serial**.println(server);

while (!client.connect(clientId, authMethod, token))

{

**Serial**.print("."); delay(500);

}

initManagedDevice();

**Serial**.println();

}

}

void initManagedDevice()

{

if (client.subscribe(topic))

{

**Serial**.println("IBM subscribe to cmd OK");

}

else

{

**Serial**.println("subscribe to cmd FAILED");

}

}

void publishData()

{

float cm = readcmCM();

if(digitalRead(34)) //pir motion detection

{

**Serial**.println("Motion Detected"); **Serial**.println("Lid Opened"); digitalWrite(15, HIGH);

if(digitalRead(34)== true)

{

if(cm <= 60) //Bin level

detection

{

digitalWrite(2, HIGH);

**Serial**.println("High Alert!!!,Trash bin is about to be full");

**Serial**.println("Lid Closed"); lcd.print("Full! Don't use"); delay(2000);

lcd.clear(); digitalWrite(4, LOW); digitalWrite(23, LOW);

}

else if(cm > 60 && cm < 120)

{

digitalWrite(4, HIGH);

**Serial**.println("Warning!!,Trash is about to cross 50% of bin level"); digitalWrite(2, LOW);

digitalWrite(23, LOW);

}

else if(cm > 120)

{

digitalWrite(23, HIGH); **Serial**.println("Bin is available"); digitalWrite(2,LOW); digitalWrite(4, LOW);

}

delay(10000);

**Serial**.println("Lid Closed");

}

else

{

**Serial**.println("No motion detected"); digitalWrite(2, LOW); digitalWrite(15, LOW); digitalWrite(4, LOW); digitalWrite(23, LOW);

}

}

else

{

digitalWrite(15, LOW);

}

if(cm <= 60)

{

digitalWrite(21,HIGH);

String payload = "{\"High\_Alert\":"; payload += cm;

payload += " }"; **Serial**.print("\n"); **Serial**.print("Sending payload: "); **Serial**.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) // if data is uploaded to cloud successfully,prints publish ok else prints publish failed

{

**Serial**.println("Publish OK");

}

}

else if(cm <= 120)

{

digitalWrite(22,HIGH);

String payload = "{\"Warning\":"; payload += cm ;

payload += " }"; **Serial**.print("\n"); **Serial**.print("Sending payload: "); **Serial**.println(payload);

if(client.publish(publishTopic, (char\*) payload.c\_str()))

{

**Serial**.println("Publish OK");

}

else

{

**Serial**.println("Publish FAILED");

}

}

else

{

**Serial**.println();

}

float inches = (cm / 2.54); //print on lcd lcd.setCursor(0,0);

lcd.print("Inches"); lcd.setCursor(4,0); lcd.setCursor(12,0); lcd.print("cm"); lcd.setCursor(1,1); lcd.print(inches, 1); lcd.setCursor(11,1); lcd.print(cm, 1); lcd.setCursor(14,1); delay(1000); lcd.clear();

}

# diagram.json:

{

"version": 1,

"author": "Uri Shaked", "editor": "wokwi", "parts": [

{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 42.67, "left": 54.67, "attrs": {} },

{ "type": "wokwi-pir-motion-sensor", "id": "pir1", "top": -88.9, "left": - 14.5, "attrs": {} },

{ "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": 95.1, "left": -140.5, "attrs": {} },

{

"type": "wokwi-lcd1602",

"id": "lcd1",

"top": 169.37,

"left": 232.3,

"attrs": { "pins": "i2c" }

},

{

"type": "wokwi-led",

"id": "led1",

"top": -70.9,

"left": 311.51,

"attrs": { "color": "green" }

},

{

"type": "wokwi-led",

"id": "led2",

"top": -23.57,

"left": 316.84,

"attrs": { "color": "yellow" }

},

{

"type": "wokwi-led",

"id": "led3",

"top": 82.44,

"left": 344.17,

"attrs": { "color": "blue" }

},

{ "type": "wokwi-led", "id": "led4", "top": 22.1, "left": 336.5, "attrs": { "color": "red" } },

{

"type": "wokwi-resistor", "id": "r1",

"top": -30.23,

"left": 250.17,

"attrs": { "value": "1000" }

},

{

"type": "wokwi-resistor",

"id": "r5",

"top": 11.77,

"left": 246.83,

"attrs": { "value": "1000" }

},

{

"type": "wokwi-resistor", "id": "r6",

"top": 67.1,

"left": 254.16,

"attrs": { "value": "1000" }

},

{

"type": "wokwi-resistor", "id": "r7",

"top": 124.44,

"left": 273.5,

"attrs": { "value": "1000" }

}

],

"connections": [

[ "esp:TX0", "$serialMonitor:RX", "", [] ],

[ "esp:RX0", "$serialMonitor:TX", "", [] ],

[ "pir1:OUT", "esp:D34", "green", [ "v0" ] ],

[ "esp:GND.2", "pir1:GND", "black", [ "h0" ] ],

[ "esp:3V3", "pir1:VCC", "red", [ "v-1", "h22.2", "v54", "h-161.33", "v-10" ]

],

[ "ultrasonic1:VCC", "esp:VIN", "red", [ "v0" ] ],

[ "ultrasonic1:TRIG", "esp:D13", "green", [ "v0" ] ],

[ "ultrasonic1:ECHO", "esp:D12", "yellow", [ "v0" ] ],

[ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ],

[ "lcd1:VCC", "esp:VIN", "red", [ "h-36", "v60.89", "h-164.67", "v-3.33" ] ],

[ "lcd1:SDA", "esp:D21", "cyan", [ "h-47.34", "v-111.94" ] ],

[ "lcd1:SCL", "esp:D22", "white", [ "h-28", "v-150.11", "h-0.67" ] ], [ "lcd1:GND", "esp:GND.1", "black", [ "h0" ] ],

[ "led1:A", "r1:2", "green", [ "v0" ] ],

[ "r1:1", "esp:D23", "green", [ "v2.06", "h-70", "v86.67", "h-12.67" ] ], [ "led2:A", "r5:2", "gold", [ "v0" ] ],

[ "r5:1", "esp:D4", "gold", [ "v2.73", "h-22.66", "v2.67" ] ], [ "led4:A", "r6:2", "red", [ "v14.07" ] ],

[ "r6:1", "esp:D2", "red", [ "v50.73", "h-86.66", "v45.33" ] ], [ "led3:A", "r7:2", "blue", [ "v0" ] ],

[ "r7:1", "esp:D15", "blue", [ "v0" ] ],

[ "led1:C", "esp:GND.1", "black", [ "v37.07", "h-121.01", "v188" ] ],

[ "led2:C", "esp:GND.1", "black", [ "v14.4", "h-116.34", "v160.67" ] ],

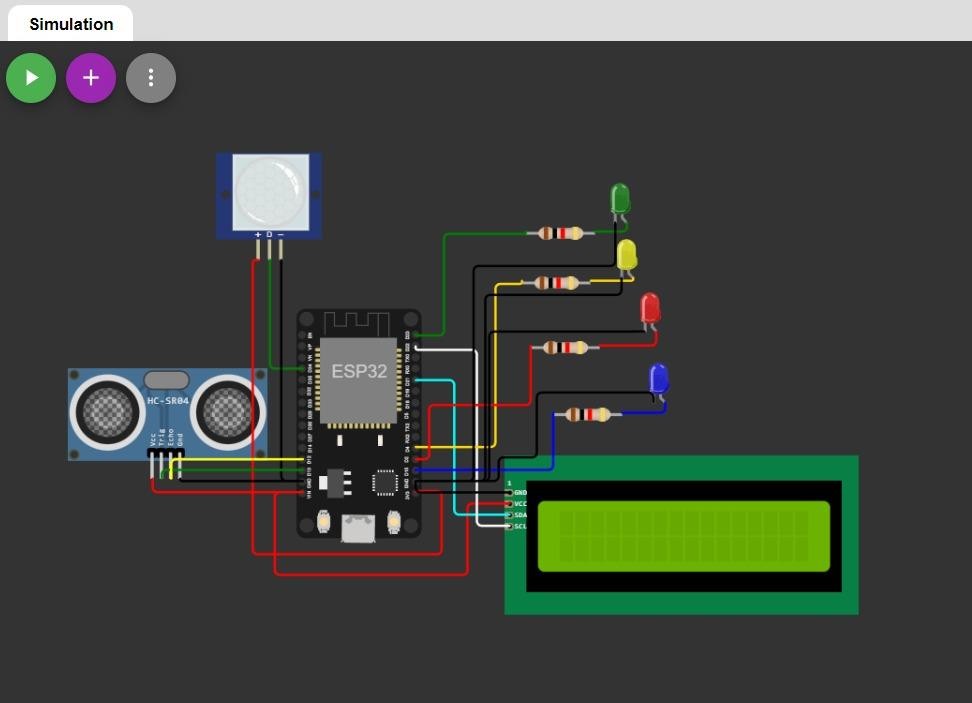
[ "led4:C", "esp:GND.1", "black", [ "v0.07", "h-132.67", "v125.33" ] ],

[ "led3:C", "esp:GND.1", "black", [ "v-8.27", "h-99.67", "v55.33", "h-32.67", "v17.33" ] ]

]

}

# CIRCUIT DIAGRAM:



**SIMULATION IN WOKWI:**

