

SPRINT DELIVERY – 3

TEAM ID : PNT2022TMID19314

PROJECT NAME: PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF RELIANT

SIMULATION USING ESP32:

The lcd displays the medicine name when the time arrives.

CODE:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#define LED 1
#include <LiquidCrystal_I2C.h> LiquidCrystal_I2C
lcd(0x27,16,2); void callback(char* subscribetopic, byte*
payload, unsigned int payloadLength);
//-----credentials of IBM Accounts-----
#define ORG " 711i15" //IBM ORGANITION ID
#define DEVICE_TYPE "Iotsensors" //Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "Anandh@1973" //Token
String data3,light; float h, t;
#define BUZZER_PIN 19 // ESP32 GIOP21 pin connected to Buzzer's pin
//----- Customise the 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
char subscribetopic[] = "iot-2/cmd/test/fmt/string"; // cmd REPRESENT command
type
char authMethod[] = "use-token-auth"; // authentication method char
token[] = TOKEN; char clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID; //client id
//-----
WiFiClient wifiClient; // creating the instance for wificlient PubSubClient
client(server, 1883, callback ,wifiClient); //calling the predefinedclient id
by passing parameter like server id,portand wificredential void setup()//
configuring the ESP32
{
  Serial.begin(115200);
  Serial.begin(9600); //
  dht.begin();
  pinMode(LED,OUTPUT);
  pinMode(BUZZER_PIN,
  OUTPUT); delay(10);
  lcd.init();
  lcd.clear();
```

```

lcd.backlight();
Serial.println();
wificonnect();
mqttconnect();
} void loop()// Recursive
Function
{   digitalWrite(BUZZER_PIN,
HIGH); delay(1000); if
(!client.loop()) { mqttconnect();
} } void mqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to "); Serial.println(server);
while (!!!client.connect(clientId, authMethod, token)) {
Serial.print("."); delay(500); }
initManagedDevice();
Serial.println();
} } void wificonnect() //function defination for
wificonnect {
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection while (WiFi.status() != WL_CONNECTED) { delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
} void initManagedDevice()
{
if (client.subscribe(subscribetopic)) {
Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
} } void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength) {
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
light=(char)payload[0];
for (int i = 1; i < payloadLength; i++) {
Serial.print((char)payload[i]); data3 +=
(char)payload[i];
}
// Make sure backlight is on
Serial.println("data: "+ data3); if(light=="n")
{
digitalWrite(BUZZER_PIN, HIGH);
Serial.println(data3); digitalWrite(LED,HIGH);

```

```

// Print a message on both lines of the LCD.
lcd.setCursor(2,0); //Set cursor to character 2 on line 0
lcd.print("Take now"); lcd.setCursor(2,1); //Move cursor
to character 2 on line 1 lcd.print(data3); delay(3000);
digitalWrite(BUZZER_PIN, LOW); digitalWrite(LED,LOW);
lcd.clear();
} else
{
digitalWrite(BUZZER_PIN, LOW);
Serial.println(data3);
digitalWrite(LED,LOW); lcd.clear();
} data3="";
}

```

NODE RED DASHBOARD:

The person enters the medicine name,date and time.It is stored in cloudant database.

It checks which medicine has to be taken at that time.

The screenshot shows a web browser window displaying a Node-RED dashboard. The dashboard has a blue header with the word 'Home'. Below the header, there is a form titled 'Medicine reminder'. The form contains three input fields: 'Medicine name' with the value 'Dolo', 'Time(HH:MM)' with the value '11:16', and 'Date(YYYY-MM-DD)' with the value '2022-11-19'. At the bottom of the form are two buttons: 'SUBMIT' and 'CANCEL'. The browser's address bar shows a local IP address and a socket ID. The Windows taskbar is visible at the bottom of the screen.

MEDICINE DATABASE:

The screenshot shows the IBM Watson IoT Platform dashboard for a project named 'medicine'. A table displays a list of documents, each containing a timestamp, a date, and a medicine name. The table has columns for '_id' and 'name'. The documents listed are:

_id	name
Time:07:08 Date:2022:11:19	{ "name": "metformin" }
Time:08:30 Date:2022:11:23	{ "name": "Plegitazone" }
Time:09:00 Date:2022:11:24	{ "name": "Nateinde" }
Time:11:16 Date:2022:11:19	{ "name": "Dolo" }
Time:17:09 Date:2022:11:22	{ "name": "Repaglinde" }
Time:18:09 Date:2022:11:18	{ "name": "paracetamol" }

The dashboard also includes a sidebar with navigation options like 'All Documents', 'Query', 'Permissions', 'Changes', and 'Design Documents'. At the bottom, there's a status bar showing 'Showing 2 of 3 columns' and 'Showing document 1 - 6'.

When the medicine details is added it sends command to ibm iot platform.ESP32 displays the medicine name in lcd display.

The screenshot shows the Wokwi IDE interface with a C++ sketch for an ESP32. The sketch includes libraries for WiFi, MQTT, and LCD. It defines a device type 'Iotsensors' and a device ID '12345'. The code sets up a WiFi client and an MQTT client, and it configures an LCD display and a buzzer. The simulation window shows the ESP32 board connected to an LCD and a buzzer. The LCD displays the text 'take dolo now'.

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #define LED 1
4 #include <LiquidCrystal_I2C.h>
5 LiquidCrystal_I2C lcd(0x27,16,2);
6 void callback(char* topic, byte* payload, unsigned int payloadLength);
7 //-----credentials of IBM Accounts-----
8 #define ORG "P11115"/IBM ORGANIZATION ID
9 #define DEVICE_TYPE "Iotsensors" //device type mentioned in ibm watson IOT Platform
10 #define DEVICE_ID "12345" //device ID mentioned in ibm watson IOT Platform
11 #define TOKEN "Anandh@1973" //Token
12 String data3,light;
13 float h, t;
14 #define BUZZER_PIN 19 // ESP32 GPIO21 pin connected to Buzzer's pin
15 //----- Customise the above values -----
16 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
17 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event
18 char subscribeTopic[] = "iot-2/cmd/test/fmt/string"; // cmd REPRESENT command type
19 char authMethod[] = "use-token-auth"; // authentication method
20 char token[] = TOKEN;
21 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
22 //-----
23 WiFiClient wificlient; // creating the instance for wificlient
24 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client
25 void setup() // configuring the ESP32
26 {

```

