

SPRINT DELIVERY – 3

TEAM ID : PNT2022TMID25288

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 GPIO21 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 predefined client id
by passing parameter like server id,port and 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")

```

```

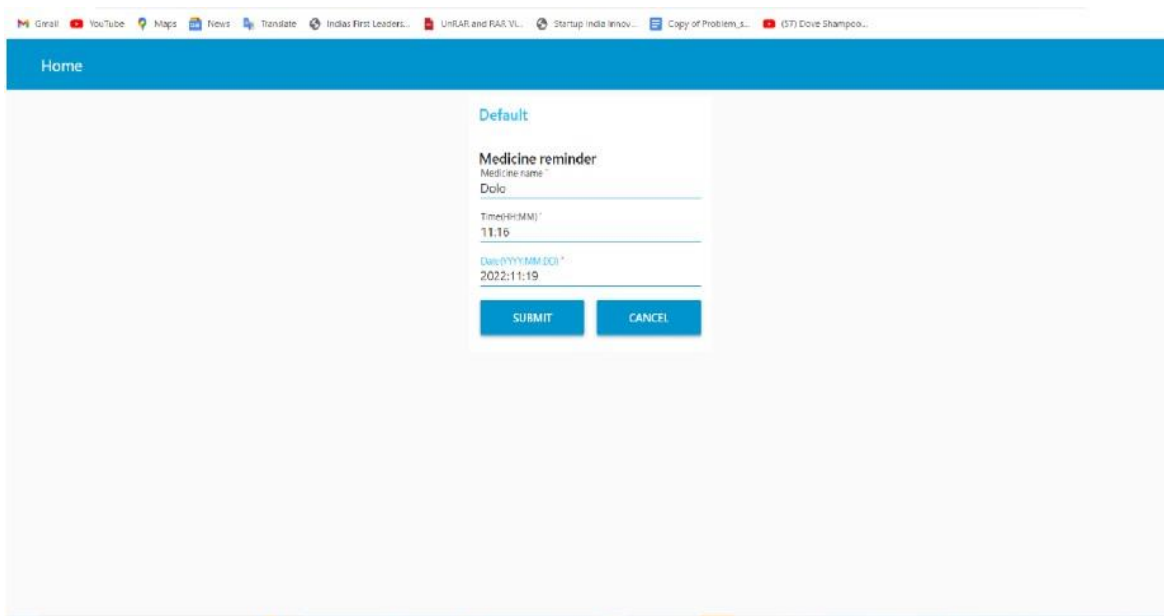
{ digitalWrite(BUZZER_PIN,
HIGH); Serial.println(data3);
digitalWrite(LED,HIGH);
// Print a message on both lines of the LCD.
lcd.setCursor(2,0); //Set cursor to character 2 on line 0
lcd.print("Take now"); lcd.setCursor(2,1); //Move cursor
to character 2 on line 1 lcd.print(data3); delay(3000);
digitalWrite(BUZZER_PIN, 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 with 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 has three input fields: 'Medicine name' with the value 'Dolo', 'Time(HH:MM)' with the value '11:16', and 'Date(YYYYMMDD)' with the value '2022-11-19'. At the bottom of the form, there are two buttons: 'SUBMIT' and 'CANCEL'.

MEDICINE DATABASE:

The screenshot shows the IBM Watson IoT Platform interface. On the left, there is a sidebar with navigation options: All Documents, Query, Permissions, Changes, and Design Documents. The main area displays a table of documents. The table has two columns: 'id' and 'name'. The documents listed are:

| id | name |
|----------------------------|---------------------------|
| Time:07:00 Date:2022-11-19 | { "name": "metformin" } |
| Time:08:30 Date:2022-11-23 | { "name": "Flagilazone" } |
| Time:09:00 Date:2022-11-24 | { "name": "Nadolol" } |
| Time:11:16 Date:2022-11-19 | { "name": "Dole" } |
| Time:17:00 Date:2022-11-22 | { "name": "Resaglide" } |
| Time:18:09 Date:2022-11-18 | { "name": "paracetamol" } |

At the bottom, it shows 'Showing 2 of 3 columns' and 'Showing document 1 of 6'.

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

