# SOURCE CODE

| | |
|---|---|
| TEAM ID | PNT2022TMID39429 |
| PROJECT TITLE | PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF RELIANT |

## TINKERCAD CODE FOR MEDICAL REMAINDER

```
// PRESS PUSH BUTTONS FOR TEMPERATURE AND MEDICINE TIME REMAINDER


#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

float voltage;

int celsius;

int valPulse;

int pulsePin = A2;

const int temperaturePin = A0;

String seconds;

long duration;

int buzzerpin = 2,button_1=A3,button_2 = 4;

void setup()
```

```
{
  pinMode(temperaturePin, INPUT);
  pinMode(button_1,INPUT);
  pinMode(button_2,INPUT);
  lcd.begin(16, 2);
  Serial.begin(9600);
  lcd.print("MEDICAL REMAINDER");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.println("HEALTH MONITORING");
  lcd.setCursor(0, 2);
  lcd.print("SYSTEM");
  delay(3000);
  lcd.clear();
  pinMode(buzzerpin,OUTPUT);
}
void loop()
{
if(digitalRead(button_1)== HIGH)
```

```
{
  lcd.clear();
  lcd.print("Calculating....");
  delay(3000);
  voltage = analogRead(temperaturePin) * 0.004882814;
  celsius = (voltage - 0.5) * 100.0;
  lcd.clear();
  lcd.print("Body Temp. : ");
  lcd.print(celsius);
  lcd.print("C");
  delay(3000);
  if(celsius>38)
  {   lcd.clear();
     digitalWrite(buzzerpin, HIGH);
     lcd.print("High temp");
     delay(2000);
     lcd.clear();
     lcd.setCursor(0, 0);
     lcd.println("TAKE EMERGENCY");
     lcd.setCursor(0, 2);
```

```
    lcd.print("TIME MEDICINE");
    delay(3000);
    lcd.clear();
    digitalWrite(buzzerpin, LOW);
}
else
{
  lcd.clear();
  lcd.print("Normal Temp");
  delay(4000);
}
lcd.clear();
lcd.print("Temperature:");
lcd.print(celsius);
lcd.print("C");
lcd.setCursor(0,1);
lcd.print("Pulse : ");
lcd.print(valPulse);
while(digitalRead(button_1)== 0);
delay(250);
```

```
  lcd.clear();
}
if(digitalRead(button_2)== HIGH)
{
  while(1){
  Serial.println("(hours) : ");
  while (Serial.available()==0){}
  long int hours = Serial.parseInt();
  lcd.print(hours);
  lcd.print(" : ");
  Serial.println("(minutes) : ");
  while (Serial.available()==0){}
  long int mins = Serial.parseInt();
  lcd.print(mins);
  lcd.print(" : ");
  Serial.println("(seconds) : ");
  while (Serial.available()==0){}
  long int seconds = Serial.parseInt();
  lcd.print(seconds);
  lcd.clear();
```

```
long int current_time =(hours*3600)+(mins*60)+(seconds);
      lcd.print(current_time);
      delay(3000);
      lcd.clear();
long int i;

for(i= current_time ; i< (current_time+(86400)); i++){
if(i == 28800){
digitalWrite(buzzerpin,HIGH);
tone(buzzerpin, 100 );
lcd.print("Medicine time");
delay(5000);
lcd.clear();
digitalWrite(buzzerpin,LOW);
}
else if(i == 72000){
digitalWrite(buzzerpin,HIGH);
tone(buzzerpin, 100);
lcd.print("Medicine time");
delay(5000);
```

```
lcd.clear();
digitalWrite(buzzerpin,LOW);
}
else{
  lcd.setCursor(0,0);

    lcd.print("Your medicine");
    lcd.setCursor(0,1);
    lcd.print("time is after :");
     delay(1000);
  if(i<28800 ){
    lcd.clear();
     lcd.setCursor(0,0);
     lcd.print(28800-i );
     lcd.setCursor(0,1);
     lcd.print("seconds");
     delay(1000);
     lcd.clear();
     }
  else if (i>28000 && i<72000){
```

```
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print( 72000-i);
    lcd.setCursor(0,1);
    lcd.print("seconds");
    delay(1000);
    lcd.clear();
   }
    else if (i>72000){
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print( 86400-i + 28800);
     lcd.setCursor(0,1);
     lcd.print("seconds");
     delay(2000);
     lcd.clear();
    }
  }
   }
    }
```

```
 }
}
```

## PYTHON CODE FOR MEDICAL REMAINDER

```
import json
import wiotp.sdk.device
import time
import random
myConfig = {
"identity": {
"orgId": "tboyb4",
"typeId": "medicineremainder",
"deviceId": "19171603"
},
"auth": {
"token": "12345678"
}
}
```

```
client = wiotp.sdk.device.DeviceClient(config=myConfig,
logHandlers=None)

client.connect()

for i in range(0,20):

 tablet=["Amlodipine
Besylate","Azithromycin","Metformin","Amoxicillin","Cetiri
zine"]

 medicinetime=[12.00,1.00,2.00,3.00,5.00,18.00,20.00,7.00]

 name = "Rajammal"

 medicine=random.choice(tablet)

 medicinetime=random.choice(medicinetime)

 mydata = {'Patient Name': name, 'Medicine Name':
medicine, 'Time': medicinetime}

 client.publishEvent("IoTSensor", "json", data=mydata,
qos=0, onPublish=None)

 print("Data published to IBM IOT platform :", mydata)

 time.sleep(5)

client.disconnect()
```