#include <LiquidCrystal.h>

#include <Bridge.h>

#include <Process.h>

#include <SoftwareSerial.h>

#include <SPI.h>

/\*

to send data to library the data type shud be a array and not a pointer

\*/

char \*server = "riot4.azure-mobile.net";//\*/"myiot.azure-mobile.net";

char \*table\_name\_otp = "env";//\*/"iotarduino\_data";

char \*table\_name\_data = "env\_settings";

char \*ams\_key = "pDTJmShRzBgzhPDJQhQrkHSRKTlsCp39"; //\*/"mhvusfpprlFQcZEgByItejyhdNKOLw24";

char \*pri\_key\_value = "1315";

char \*buff;

char buffer1[200];

char response[370];

boolean responseFlag;

char query[100];

int i;

int otp;

int temperature;

int intensity;

int l;

char str[10];

char update = '0';

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

Process p1, p2;

SoftwareSerial mySerial(9, 10);

boolean toggle = false;

char ch;

boolean uflag = false;

int otpPin = 7;

void setup() {

// Starting USB Serial, Bluetooth Serial, Initialize Bridge Connection

Serial.begin(9600);

mySerial.begin(9600);

temperature = 25;

intensity = 50;

toggle = false;

Bridge.begin();

while (!Serial);

Serial.println("Starting Bridge...");

lcd.begin(16, 2);

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("IoT DEVICE");

lcd.setCursor(1, 1);

lcd.print("CHIP ID 1315");

delay(3000);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("MICROSOFT IoT");

lcd.setCursor(5, 1);

lcd.print("TEAM A");

pinMode(otpPin, INPUT);

pinMode(8, OUTPUT);

digitalWrite(8, HIGH);

pinMode(A0, OUTPUT);

pinMode(A2, OUTPUT);

pinMode(A1, INPUT);

digitalWrite(A0, HIGH);

digitalWrite(A2, LOW);

}

void loop() {

// put your main code here, to run repeatedly:

boolean pin = digitalRead(otpPin);

if (pin) // FOR OTP GENERATION AND TEMP VALUE UPDATION

{

toggle = !toggle;

Serial.println("Reseting OTP...");

lcd.clear();

lcd.setCursor(3, 0);

lcd.print("Setting Up");

lcd.setCursor(0, 1);

lcd.print(" New OTP ");

otp = millis() % 10000;

if (otp < 1000)

otp = (random(8) + 1) \* 1000 + otp;

//DO not Change the Below LINE...

sprintf(query, "{\\\"OTP\\\" : %d}", otp);

sprintf(buffer1, "curl -H \"Content-type:application/json\" -H X-ZUMO-APPLICATION:%s -X PATCH -d \"%s\" http://%s/tables/%s/%s", ams\_key, query, server, table\_name\_otp, pri\_key\_value);

Serial.println(buffer1);

p1.runShellCommand(buffer1);

while (p1.running());

Serial.println("Reading OTP Value: ");

while (p1.available())

{

ch = p1.read();

Serial.print(ch);

}

p1.flush();

p1.close();

lcd.print(otp);

lcd.setCursor(1, 0);

lcd.print("CHIP ID - 1315");

while (digitalRead(otpPin));

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("MS TEAM A");

lcd.setCursor(0, 1);

lcd.print("Temp ");

lcd.print(temperature);

lcd.print("`C ");

lcd.print(intensity);

lcd.print("%");

}

else

{

toggle = !toggle;

Serial.println("Checking For UPDATE");

sprintf(buffer1, "curl --header X-ZUMO-APPLICATION:%s \"http://%s/tables/%s/%s\"", ams\_key, server, table\_name\_otp, pri\_key\_value);

Serial.println(buffer1);

p2.runShellCommand(buffer1);

while (p2.running());

Serial.print("Update:");

uflag = false;

while (p2.available())

{

ch = p2.read();

if (uflag)

{

if (ch >= '0' && ch <= '1')

update = ch;

}

if (ch == ':')

uflag = true;

}

p2.flush();

p2.close();

if (update == '1')

{

Serial.println("Making HTTP Request...");

sprintf(buffer1, "curl --header X-ZUMO-APPLICATION:%s \"http://%s/tables/%s\"", ams\_key, server, table\_name\_data);

Serial.println(buffer1);

p2.runShellCommand(buffer1);

while (p2.running());

Serial.println("Reading Values: ");

while (p2.available())

{

ch = p2.read();

mySerial.write(ch);

}

p2.flush();

p2.close();

sprintf(query, "{\\\"changed\\\" : %d}", 0);

sprintf(buffer1, "curl -H \"Content-type:application/json\" -H X-ZUMO-APPLICATION:%s -X PATCH -d \"%s\" http://%s/tables/%s/%s", ams\_key, query, server, table\_name\_otp, pri\_key\_value);

Serial.println(buffer1);

p1.runShellCommand(buffer1);

while (p1.running());

Serial.println("Response: ");

while (p1.available())

{

ch = p1.read();

Serial.print(ch);

}

p1.flush();

p1.close();

}

Serial.println("Received Values From Server");

temperature = (analogRead(A1) \* 500) / 1023;

intensity = analogRead(A4)/10;

if (temperature < 0)

temperature = -1 \* temperature;

if (intensity < 0)

intensity = -1 \* intensity;

//DO not Change the Below LINE...

sprintf(query, "{\\\"Temp\\\" : %d,\\\"Intensity\\\" : %d}", temperature, intensity);

sprintf(buffer1, "curl -H \"Content-type:application/json\" -H X-ZUMO-APPLICATION:%s -X PATCH -d \"%s\" http://%s/tables/%s/%s", ams\_key, query, server, table\_name\_otp, pri\_key\_value);

Serial.println(buffer1);

p1.runShellCommand(buffer1);

while (p1.running());

Serial.println("Reading Value: ");

while (p1.available())

{

ch = p1.read();

Serial.print(ch);

}

p1.flush();

p1.close();

delay(10);

}

Serial.println("--------------DONE------------------");

}