#include "thingProperties.h"

#include <HTTPClient.h>

const char\* ntpServer = "gr.pool.ntp.org";//"gr.ntp.grnet.org";

const long gmtOffset\_sec = 7200;/\* +2 7200sec\*/

const int daylightOffset\_sec = 3600;/\*3600 1 hour daylight offset\*/

int buzzer = 2, LedPin = 32, Movesensor = 35, WaterPin = 33, flamePin = 15, reedPin = 12, door = 0;

bool motioncheck, Watercheck, flamecheck, reedcheck;

char \*email, \*msg;

String IFTTT\_URL = "https://maker.ifttt.com/trigger/email/with/key/cFiKgFaqDSoeeNR4mqFUYJ";

void setup()

{

pinMode(buzzer, OUTPUT);

pinMode(LedPin, OUTPUT);

pinMode(Movesensor, INPUT);

pinMode(WaterPin, INPUT);

pinMode(flamePin, INPUT);

pinMode(reedPin, INPUT);

Serial.begin(9600);

delay(1500);

initProperties();

ArduinoCloud.begin(ArduinoIoTPreferredConnection);

setDebugMessageLevel(2);

ArduinoCloud.printDebugInfo();

}

void loop()

{

ArduinoCloud.update();

delay(500);//0.5sec

if (motioncheck)

{

PIRSensor();

}

if (Watercheck)

{

WaterSensor();

}

if (flamecheck)

{

flameSensor();

}

if (reedcheck)

{

reedSensor();

}

}

void PIRSensor()

{

int pinStateCurrent = LOW, pinStatePrevious = LOW;

email = "Motion was detected";

msg = "Motion detected at: ";

pinStatePrevious = pinStateCurrent;

pinStateCurrent = digitalRead(Movesensor);

if (pinStatePrevious == LOW && pinStateCurrent == HIGH)

{

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

tone(buzzer, 500, 100);

sendEmai();

}

else if (pinStatePrevious == HIGH && pinStateCurrent == LOW)

{

noTone(buzzer);

}

}

void WaterSensor()

{

email = "Water leak was detected";

msg = "Water leak detected at: ";

if (analogRead(WaterPin) > 300)

{

tone(buzzer, 500, 100);

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

sendEmai();

}

else

{

noTone(buzzer);

}

}

void flameSensor()

{

int fire = digitalRead(flamePin);

email = "Fire was detected";

msg = "Fire detected at: ";

if ( fire == HIGH)

{

tone(buzzer, 500, 100);

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

sendEmai();

}

else

{

noTone(buzzer);

}

}

void reedSensor()

{

int proximity = digitalRead(reedPin);

if ((proximity == LOW && door == 2) or (proximity == LOW && door == 0))

{

msg = ("Door closed at: ");

email = "The door closed";

tone(buzzer, 500, 50); //100

door = 1;

sendEmai();

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

}

else if ((proximity == HIGH && door == 1) or (proximity == HIGH && door == 0))

{

email = "The door opened";

msg = ("Door opened at: ");

noTone(buzzer);

door = 2;

sendEmai();

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

}

}

void onMotionDetChange()

{

if (motion\_det)

{

email = "Motion detection is enabled";

msg = "Motion detection turned on at: ";

motioncheck = true;

}

else

{

email = "Motion detection is disabled";

msg = "Motion detection turned off at: ";

motioncheck = false;

}

sendEmai();

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

OnOffLed();

}

void onWaterDetChange()

{

if (water\_det)

{

email = "Water detection is enabled";

msg = "Water detection turned on at: ";

Watercheck = true;

}

else

{

email = "Water detection is disabled";

msg = "Water detection turned off at: ";

Watercheck = false;//Waterchek

}

sendEmai();

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

OnOffLed();

}

void onFlameDetChange()

{

if (flame\_det)

{

email = "Flame detection is enabled";

msg = "Flame detection turned on at: ";

flamecheck = true;

}

else

{

email = "Flame detection is disabled";

msg = "Flame detection turned off at: ";

flamecheck = false;

}

sendEmai();

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

OnOffLed();

}

void onReedDetChange()

{

if (reed\_det)

{

email = "Door detection is enabled";

msg = "Door detection turned on at: ";

reedcheck = true;

}

else

{

email = "Door detection is disabled";

msg = "Door detection turned off at: ";

reedcheck = false;

}

sendEmai();

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

printLocalTimeAndMsg(msg);

OnOffLed();

}

void OnOffLed()

{

digitalWrite(LedPin, HIGH);

delay(500);

digitalWrite(LedPin, LOW);

}

void printLocalTimeAndMsg(char\* mesg)

{

char printTime[50], buf[70];

time\_t rawtime;

struct tm timeinfo;

if (!getLocalTime(&timeinfo))

{

Serial.println("Failed to obtain time");

return;

}

strftime(printTime, sizeof(printTime), "%d, %B %Y %H:%M:%S", &timeinfo);

strcpy(buf, mesg);

strcpy(buf + strlen(mesg), printTime);

messages = buf;

}

void sendEmai()

{

String url = IFTTT\_URL;

url.replace("email", "email");

url.replace("cFiKgFaqDSoeeNR4mqFUYJ", "cFiKgFaqDSoeeNR4mqFUYJ");

HTTPClient http;

http.begin(url);

http.addHeader("Content-Type", "application/json");

String payload = "{\"value3\":\"body\"}";

payload.replace("body", email);

int httpResponseCode = http.POST(payload);

if (httpResponseCode > 0)

{

String response = http.getString();

Serial.println(httpResponseCode);

Serial.println(response);

}

else

{

Serial.print("Error sending HTTP POST request: ");

Serial.println(httpResponseCode);

}

http.end();

}

void onMessagesChange(){

}

