Nathan Luscher

Ben Salah Mohamed

Projet CSF: Maison connecté

Sommaire:

1) Introduction
2) Matériel utilisé
3) Programmation
4) Conclusion

Introduction: Ce projet à pour objectif d'améliorer le conforts des utilisateurs et de renforcer la sécurité et l'autonomie des personnes dépendantes ou à mobilité réduite

Problématique: Comment peut – on automatiser des éléments sans passer par les grandes entreprises et ainsi à économiser de l'argent ?

Ce qui nous a motivé a faire ce projet, c'est de pouvoir créer plus de conforts dans un habitat en le faisant soi-même (motivation)

2) Matériel utilisé: Pour réalisé ce projet nous avons eu besoin de: carte programmable, Un servomoteur, capteur de présence , plus la boite pour les capteurs, et les files..

3) Programmation: Code intérieur

```
#include <SPI.h>
#include <LoRa.h>
int counter = 0;

int txPower = 14; // from 0 to 20, default is 14
int spreadingFactor = 12; // from 7 to 12, default is 12
long signalBandwidth = 125E3; // 7.8E3, 10.4E3, 15.6E3, 20.8E3, 31.25E3,41.7E3,62.5E3,125E3,250E3,500e3, default is 125E3
int codingRateDenominator=5; // Numerator is 4, and denominator from 5 to 8, default is 5
int preambleLength=8; // from 2 to 20, default is 8
String payload = "hello"; // you can change the payload



#include "FastLED.h"
#include <Servo.h>
#define NUM_LEDS 9
CRGB leds[NUM_LEDS];
int lum;
#define DATA_PIN 4


Servo monServo;
int potpin = 1;


int capteur_lum = A0; // capteur branché sur le port 0
int analog_lum; // valeur analogique envoyé par le capteur

#define Broche_Echo A3 // Broche Echo du HC-SR04 sur D7 //
#define Broche_Trigger A2 // Broche Trigger du HC-SR04 sur D8 //
#define servomoteur D6
long Duree;
long Distance;

#define SS 10
#define RST 8
#define DI0 3
#define BAND 865E6  // Here you define the frequency carrier
```

Code intérieur 2

```
void setup() {

  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Receiver");
  Serial.print("SetFrequency : ");
  Serial.print(BAND);
  Serial.println("Hz");
  Serial.print("SetSpreadingFactor : SF");
  Serial.println(spreadingFactor);

  SPI.begin();
  LoRa.setPins(SS,RST,DI0);

  if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
 LoRa.setTxPower(txPower,1);
 LoRa.setSpreadingFactor(spreadingFactor);
 LoRa.setSignalBandwidth(signalBandwidth);
 LoRa.setCodingRate4(codingRateDenominator);
 LoRa.setPreambleLength(preambleLength);


  Serial.begin(9600); // démarrer la liaison série
  FastLED.addLeds<WS2812, DATA_PIN, RGB>(leds, NUM_LEDS);

pinMode(Broche_Trigger, OUTPUT); // Broche Trigger en sortie //
pinMode(Broche_Echo, INPUT); // Broche Echo en entree //
Serial.begin (9600);
monServo.attach(A1);
}
```

Code intérieur 3:

```
void loop() {
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // received a packet
    Serial.print("Received packet '");
    // read packet
    while (LoRa.available()) {
      Serial.print((char)LoRa.read());
    }
    // print RSSI of packet
    Serial.print("' with RSSI ");
    Serial.println(LoRa.packetRssi());
  }
  analog_lum = analogRead(capteur_lum); // lecture de la valeur analogique, qu'on enregistre dans analog_lum
  Serial.print("Valeur luminosité = ");
  Serial.print(analog_lum);
  Serial.println("");
  delay(1000);

 if (analog_lum >= 0; analog_lum <= 1023);{
   lum =(255 - (analog_lum * 0.25));
  Serial.print ("Intensitée des leds = ");
  Serial.print (lum);
  Serial.println("");
  delay(1000);
}
  leds[0] = CRGB(lum,lum,lum);
  leds[1] = CRGB(lum,lum,lum);
  leds[2] = CRGB(lum,lum,lum);
  leds[3] = CRGB(lum,lum,lum);
  leds[4] = CRGB(lum,lum,lum);
  leds[5] = CRGB(lum,lum,lum);
  leds[6] = CRGB(lum,lum,lum);
  leds[7] = CRGB(lum,lum,lum);
  leds[8] = CRGB(lum,lum,lum);
  FastLED.show();

// Debut de la mesure avec un signal de 10 µS applique sur TRIG //

digitalWrite(Broche_Trigger, LOW); // On efface l'etat logique de TRIG //
delay(100);

digitalWrite(Broche_Trigger, HIGH); // On met la broche TRIG a "1" pendant 10µS //
delay(100);
digitalWrite(Broche_Trigger, LOW); // On remet la broche TRIG a "0" //

// On mesure combien de temps le niveau logique haut est actif sur ECHO //
Duree = pulseIn(Broche_Echo, HIGH);
// Calcul de la distance grace au temps mesure //
Distance = Duree*0.034/2; // *** voir explications apres l'exemple de code *** //



  // Affichage dans le moniteur serie de la distance mesuree //
  Serial.print("Distance mesuree :");
  Serial.print(Distance);
  Serial.println("cm");

  delay(1000); // On ajoute 1 seconde de delais entre chaque mesure //

if ( Distance >= 25){
  monServo.write(0);
}
else{
  monServo.write(200);

}
}
```

Code extérieur:

```
#include <SPI.h>
#include <LoRa.h>

int counter = 0;

// Parameters you can play with :

int txPower = 14; // from 0 to 20, default is 14
int spreadingFactor = 12; // from 7 to 12, default is 12
long signalBandwidth = 125E3; // 7.8E3, 10.4E3, 15.6E3, 20.8E3, 31.25E3,41.7E3,62.5E3,125E3,250E3,500e3, default is 125E3
int codingRateDenominator=5; // Numerator is 4, and denominator from 5 to 8, default is 5
int preambleLength=8; // from 2 to 20, default is 8
String payload = "2610"; // you can change the payload

#define SS 10
#define RST 8
#define DI0 3
#define BAND 865E6  // Here you define the frequency carrier

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Sender");
  Serial.print("SetFrequency : ");
  Serial.print(BAND);
  Serial.println("Hz");
  Serial.print("SetSpreadingFactor : SF");
  Serial.println(spreadingFactor);

  SPI.begin();
  LoRa.setPins(SS,RST,DI0);



  if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
 LoRa.setTxPower(txPower,1);
 LoRa.setSpreadingFactor(spreadingFactor);
 LoRa.setSignalBandwidth(signalBandwidth);
 LoRa.setCodingRate4(codingRateDenominator);
 LoRa.setPreambleLength(preambleLength);
// LoRa.setPolarity(1);
 //LoRa.setFSK();

}

void loop() {

  // send packet

  LoRa.beginPacket();
  LoRa.print(payload);

  LoRa.endPacket();
  counter++;

  Serial.print("Sending packet with payload {");
  Serial.print(payload);
  Serial.print("} N°");
  Serial.println(counter);


  delay(100);
}
```

4) Conclusion:

Domoticz