ANEXO III

PROGRAMACIÓN BASADA EN ARDUINO

TFG

GRADO EN SISTEMAS DE TELECOMUNICACIONES E INFORMÁTICOS

AUTOR: PÉREZ ARANDA, ALEJANDRO

TUTOR: BONILLA FERNÁNDEZ, D. ANTONIO

IES LAS FUENTEZUELAS

AÑO 2024

Tabla de contenido

1. Arduino.

1.1. ¿Qué es Arduino?

Arduino es una plataforma de hardware y software de código abierto diseñada para la creación de proyectos electrónicos interactivos. Está compuesta por una placa de desarrollo, como Arduino Uno o Arduino Mega, en un entorno de programación, y una comunicad activad de usuarios.

1.2. Características y aplicaciones.

Principalmente se utiliza para crear prototipos de dispositivos electrónicos y sistemas embebidos, permitiendo la creación de una amplia gama de aplicaciones.

Algunas de las características son:

- **Prototipado Rápido:** permite probar conceptos antes de invertir en el hardware personalizado.
- **Automatización del hogar:** permite crear sistema IoT.
- **Robótica:** permite la construcción de robots con diversas funciones.
- **Electrónica educativa:** siendo una herramienta para enseñar electrónica a los más pequeños.

Y un largo etcétera, abarcando el ámbito industrial y el ámbito de la agricultura, así como el ámbito médico y de la salud.

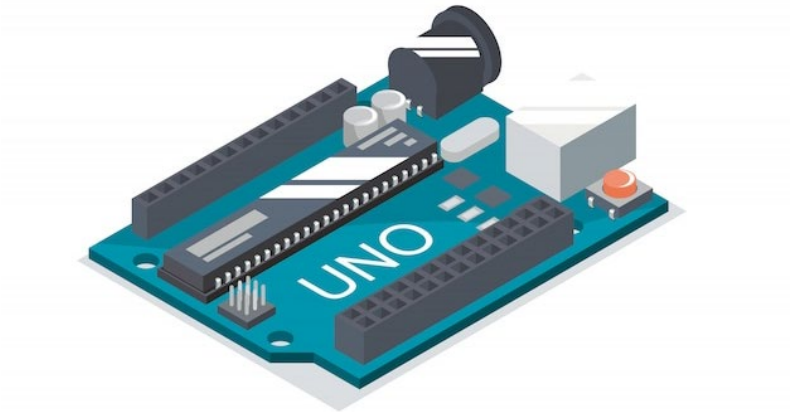

Figura 1: ARDUINO UNO

2. Código basado en Arduino – Módulos Individuales.

Conociendo que es Arduino y para que se utiliza, aplicamos su lenguaje en C# para la creación de los códigos necesarios para hacer transmitir y recibir la señal de nuestro satélite.

2.1. Código valores analógicos.

2.1.1. FIRE – Detector de llama.

```
*/GNU GENERAL PUBLIC LICENSE v3.0
REALIZACIÓN DE UN SATÉLITE METEOROLÓGICO BASADO EN ARDUINO
PROYECTO REALIZADO POR ALEJANDRO PÉREZ ARANDA.
https://github.com/alejandroperez9/ARDUSAT
https://linkedin.com/in/alejandro-p←rez-aranda-a91012278

---------- LECTURA LLAMA DE FUEGO (FIRE) ----------
*/
const int sensorPin = A0;

void setup(){

  Serial.begin(9600);

}

void loop(){

  int llama = analogRead(sensorPin);
  Serial.println(llama);

  if (llama < 500){
    Serial.println("Detección");
      //SE AÑADEN LAS MEDIDAS
  }

  delay(1000);
}
```

*NOTA: El código comentado se encuentra en GITHUB.

Así como las librerías y recursos utilizados.

2.1.2. LDR – Intensidad de luz obtenida.

```
/*
GNU GENERAL PUBLIC LICENSE v3.0
REALIZACIÓN DE UN SATÉLITE METEOROLÓGICO BASADO EN ARDUINO
PROYECTO REALIZADO POR ALEJANDRO PÉREZ ARANDA.
https://github.com/alejandroperez9/ARDUSAT
https://linkedin.com/in/alejandro-p←rez-aranda-a91012278

---------- LECTURA LDR ----------
*/

void setup(){

  Serial.begin(9600);
  pinMode(A0,INPUT);

}

void loop(){

  int valorLDR = analogRead(A0);
  Serial.println(valorLDR);

  }
```

*NOTA: El código comentado se encuentra en GITHUB.

Así como las librerías y recursos.

2.2. Códigos valores digitales.

2.2.1. DHT – Sensor Temperatura y Humedad.

```
/*
GNU GENERAL PUBLIC LICENSE v3.0
REALIZACIÓN DE UN SATÉLITE METEOROLÓGICO BASADO EN ARDUINO
PROYECTO REALIZADO POR ALEJANDRO PÉREZ ARANDA.
https://github.com/alejandroperez9/ARDUSAT
https://linkedin.com/in/alejandro-p←rez-aranda-a91012278

---------- LECTURA DHT ----------
*/

#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup(){

  Serial.begin(9600);
  dht.begin();

}

void loop() {

  float h = dht.readHumidity();
  float t = dht.readTemperature();

  Serial.print(Temp: );
  Serial.print(t);
  Serial.print(" Hum: ");
  Serial.println(h);

}
```

*NOTA: El código comentado se encuentra en GITHUB.

Así como las librerías y recursos.

2.2.2. MPU6050 – Giroscopio y Acelerómetro.

```
/*
GNU GENERAL PUBLIC LICENSE v3.0
REALIZACIÓN DE UN SATÉLITE METEOROLÓGICO BASADO EN ARDUINO
PROYECTO REALIZADO POR ALEJANDRO PÉREZ ARANDA.
https://github.com/alejandroperez9/ARDUSAT
https://linkedin.com/in/alejandro-p←rez-aranda-a91012278

---------- LECTURA MPU6050 ----------
*/

#include <Wire.h>
#include <MPU6050.h>

const int mpuAddress = 0x68;
MPU6050 mpu(mpuAddress);

void setup() {

  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();

}

void loop() {

  int16_t ax, ay, az;
  int16_t gx, gy, gz;

  mpu.getRotation(&gx, &gy, &gz);
  mpu.getAcceleration(&ax, &ay, &az);
  // AQUÍ PUEDE AÑADIRSE LA IMPRESIÓN EN EL PUERTO SERIE DE CADA
VALOR.

}
```

*NOTA: El código comentado se encuentra en GITHUB.

Así como las librerías y recursos.

2.3. Código del Módulo de Comunicación.

2.3.1. Módulo Emisor.

```
/*
GNU GENERAL PUBLIC LICENSE v3.0
REALIZACIÓN DE UN SATÉLITE METEOROLÓGICO BASADO EN ARDUINO
PROYECTO REALIZADO POR ALEJANDRO PÉREZ ARANDA.
https://github.com/alejandroperez9/ARDUSAT
https://linkedin.com/in/alejandro-p←rez-aranda-a91012278

---------- EMISOR LoRa ----------
*/

#include <LoRa.h>

void setup(){

  Serial.begin(9600);
  if (!LoRa.begin(433E6)){
    Serial.println("Error al iniciar el módulo LoRa.");
    while (1);
  }

}

void loop(){

  int PAQUETE = "VALORES A ENVIAR";

  LoRa.beginPacket();
  LoRa.print(PAQUETE);
  LoRa.endPacket();

  delay(5000);
}
```

*NOTA: El código comentado se encuentra en GITHUB.

Así como las librerías y recursos.

2.3.2. Módulo Receptor.

```
/*
GNU GENERAL PUBLIC LICENSE v3.0
REALIZACIÓN DE UN SATÉLITE METEOROLÓGICO BASADO EN ARDUINO
PROYECTO REALIZADO POR ALEJANDRO PÉREZ ARANDA.
https://github.com/alejandroperez9/ARDUSAT
https://linkedin.com/in/alejandro-p←rez-aranda-a91012278

---------- RECEPTOR LoRa ----------
*/

#include <LoRa.h>

void setup(){

  Serial.begin(9600);
  if (!LoRa.begin(433E6)){
    Serial.println("Error al iniciar el módulo LoRa.");
    while (1);
  }

}

void loop(){

  if (LoRa.parsePacket()){
    float PAQUTE = LoRa.parseFloat();
    Serial.print("Recibido: ");
    Serial.println(PAQUETE);
  }
}
```

*NOTA: El código comentado se encuentra en GITHUB.

Así como las librerías y recursos.

3. Código basado en Arduino – Emisor.

```
/*
GNU GENERAL PUBLIC LICENSE v3.0
REALIZACIÓN DE UN SATÉLITE METEOROLÓGICO BASADO EN ARDUINO
PROYECTO REALIZADO POR ALEJANDRO PÉREZ ARANDA.
https://github.com/alejandroperez9/ARDUSAT
https://linkedin.com/in/alejandro-p←rez-aranda-a91012278

---------- EMISOR ----------
*/

#include <DHT.h>
#include <LoRa.h>
#include <I2Cdev.h>
#include <MPU6050.h>
#include <Wire.h>

#define DHTPIN 12
#define DHTTYPE DHT11

#define DHTPINA 11
#define DHTTYPEA DHT11

#define DHTPINB 10
#define DHTTYPEB DHT11

DHT dht(DHTPIN, DHTTYPE);
DHT dhtA(DHTPINA, DHTTYPEA);
DHT dhtB(DHTPINB, DHTTYPEB);

const int ledPIN1 = 6;
const int ledPIN2 = 5;
const int ledPIN3 = 4;
const int ledPIN4 = 3;
const int ledPIN5 = 2;

const int LDR_A = A0;
const int LDR_B = A1;
const int FUEGO = A2;

const int mpuAddress = 0x68;
MPU6050 mpu(mpuAddress);
```

```
void setup(){

  Serial.begin(9600);

  if (!LoRa.begin(433E6)){
    digitalWrite(ledPIN5, HIGH);
    while (1);
  }else{
    digitalWrite(ledPIN5, LOW);
  }

  Wire.begin();
  mpu.initialize();

  dht.begin();
  dhtA.begin();
  dhtB.begin();

  pinMode(ledPIN1, OUTPUT);
  pinMode(ledPIN2, OUTPUT);
  pinMode(ledPIN3, OUTPUT);
  pinMode(ledPIN4, OUTPUT);
  pinMode(ledPIN5, OUTPUT);

  pinMode(LDR_A, INPUT);
  pinMode(LDR_B, INPUT);
  pinMode(FUEGO, INPUT);

}



void loop(){

  int LDRA = analogRead(LDR_A);
  int LDRB = analogRead(LDR_B);

  if (isnan(LDRA) || isnan(LDRB)){
    digitalWrite(ledPIN3, HIGH);
  }else{
    digitalWrite(ledPIN3, LOW);
  }
```

```
int Lectura_Solar = (LDRA + LDRB)/2;
int ALERTA_FUEGO = analogRead(FUEGO);

if (ALERTA_FUEGO < 500){
  digitalWrite(ledPIN4, HIGH);
}else{
  digitalWrite(ledPIN4, LOW);
}

float Temp_Int = dht.readTemperature();

if (isnan(Temp_Int)){
  digitalWrite(ledPIN1, HIGH);
  return;
}else{
  digitalWrite(ledPIN1, LOW);
}

if (Temp_Int >= 22){
  digitalWrite(9, HIGH);
}else{
  digitalWrite(9, LOW);
}

float Temp_ExtA = dhtA.readTemperature();
float Hum_ExtA = dhtA.readHumidity();

if (isnan(Temp_ExtA) || isnan(Hum_ExtA)){
  digitalWrite(ledPIN2, HIGH);
  return;
}else{
  digitalWrite(ledPIN2, LOW);
}

float Temp_ExtB = dhtB.readTemperature();
float Hum_ExtB = dhtB.readHumidity();

if (isnan(Temp_ExtB) || isnan(Hum_ExtB)){
  digitalWrite(ledPIN2, HIGH);
  return;
}else{
  digitalWrite(ledPIN2, LOW);
  }

  float Temp_Ext = (Temp_ExtA + Temp_ExtB)/2;
  float Hum_Ext = (Hum_ExtA + Hum_ExtB)/2;
```

```
int16_t ax, ay, az;
int16_t gx, gy, gz;

mpu.getRotation(&gx, &gy, &gz);
mpu.getAcceleration(&ax, &ay, &az);

String strV1 = String(Temp_Int, 2);
String strV2 = String(Temp_Ext, 2);
String strV3 = String(Hum_Ext, 2);
String strV4 = String(Lectura_Solar, 2);
String strV5 = String(ALERTA_FUEGO, 2);
String strax = String(ax, 2);
String stray = String(ay, 2);
String straz = String(az, 2);
String strgx = String(gx, 2);
String strgy = String(gy, 2);
String strgz = String(gz, 2);

String cadenaTransmitir = strV1 + "," + strV2 + "," + strV3 + "," +
strV4 + "," + strV5 + "," + strax + "," + stray + "," + straz + ","
+ strgx + "," + strgy + "," + strgz;


LoRa.beginPacket();
LoRa.print(cadenaTransmitir);
LoRa.endPacket();

delay(7000);

}
```

*NOTA: El código comentado se encuentra en GITHUB.

Así como las librerías y recursos.

4. Código basado en Arduino – Receptor.

```
/*
GNU GENERAL PUBLIC LICENSE v3.0
REALIZACIÓN DE UN SATÉLITE METEOROLÓGICO BASADO EN ARDUINO
PROYECTO REALIZADO POR ALEJANDRO PÉREZ ARANDA.
https://github.com/alejandroperez9/ARDUSAT
https://linkedin.com/in/alejandro-p←rez-aranda-a91012278
---------- RECEPTOR ----------
*/

#include <LoRa.h>

void setup(){

  Serial.begin(9600);
  if (!LoRa.begin(433E6)){
    Serial.println("Error al iniciar el módulo LoRa.");
    while (1);
  }
}

void loop(){

  if (LoRa.parsePacket()){
    String data = LoRa.readString();
    Serial.print("Datos: ");
    Serial.println(data);

    float values[11];
    char *token = strtok(data.c_str(), ",");
    int i = 0;

    while (token != NULL && i < 11){
      values[i] = atof(token);
      Serial.print("Valor: ");
      Serial.print(i);
      Serial.print(": ");
      Serial.println(values[i]);
      token = strtok(NULL, ",");
      i++;
    }
  }
}
```

5. Consultas.

Para cualquier duda o consulta acerca de este ***ANEXO III*** o sobre el proyecto en general, contacte conmigo a través de los siguientes medios:

MAIL: alejandroperezaranda99@gmail.com

GITHUB: Https://www.github.com/alejandroperez9/ARDUSAT

Proyecto realizado por: **ALEJANDRO PÉREZ ARANDA**.

Trabajo Fin de Grado en Técnico Superior de Sistemas de Telecomunicaciones e Informáticos.

LINKEDIN: www.linkedin.com/in/alejandro-pérez-aranda-a91012278

GITHUB: https://github.com/alejandroperez9

INSTAGRAM: https://www.instagram.com/aleejandropeerez/?hl=es