Emision y recepción LoRa

David Serrano Moreno 2021

ÍNDICE

PRESENTACIÓN

En este trabajo se han realizado dos aparatos basados en Arduino y el modulo de radio LoRa sx1278. Un emisor y un receptor.

El emisor consiste en un teclado (key pad) del kit para Arduino Elegoo, el propio Arduino, un display LCD con un módulo SPI y el módulo sx1278. El emisor espera la entrada por teclado (alfanumérico) y al pulsar la tecla ‘#’ , se envía el mensaje.

El receptor consiste en un display LCD con un módulo SPI, el propio Arduino y el módulo sx1278. El receptor espera la recepción del mensaje y lo muestra en el LCD.

SISTEMA LoRa

LoRaWAN™ es una especificación para **redes** de baja potencia y área amplia, LPWAN (en inglés, *Low Power Wide Area Network*), diseñada específicamente para dispositivos de bajo consumo de alimentación, que operan en redes de alcance local, regional, nacionales o globales.

El estándar de red LoRaWAN apunta a requerimientos característicos de Internet de las Cosas, tales como conexiones bidireccionales seguras, bajo consumo de energía, largo alcance de comunicación, bajas velocidades de datos, baja frecuencia de transmisión, movilidad y servicios de localización. Permite la interconexión entre objetos inteligentes sin la necesidad de instalaciones locales complejas, y además otorga amplia libertad de uso al usuario final, al desarrollador y a las empresas que quieran instalar su propia red para Internet de las Cosas.

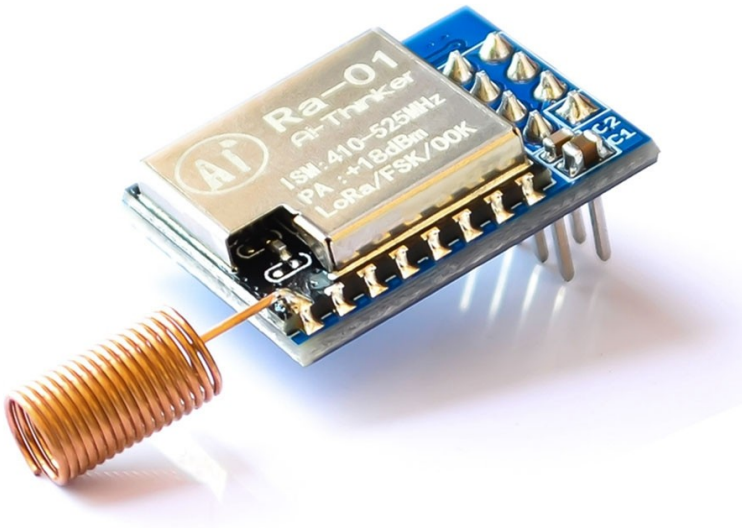
Las velocidades de datos se encuentran en el rango de 0.3 kbps a 50 kbps.

Wikipedia.org

MÓDULO DE RADIO LoRa
sx1278 433mhz

El módulo SX1278 LoRa es uno de los últimos productos de la tecnología RF de largo alcance. SX1278 utiliza el protocolo de comunicación SPI para la comunicación de datos con el controlador principal. El módulo usa una antena para la comunicación de radio frecuencia. Utiliza múltiples modulaciones que se pueden seleccionar para la comunicación de radio frecuencia. Tiene un rango de hasta 5km-10km. Utiliza el espectro de comunicación LoRa el cual extiende su rango hasta los 10km pero requiere un canal específico de 1mhz de ancho de banda.

https://microcontrollerslab.com/sx1278-lora-rf-module-pinout-arduino-interfacing-datasheet/



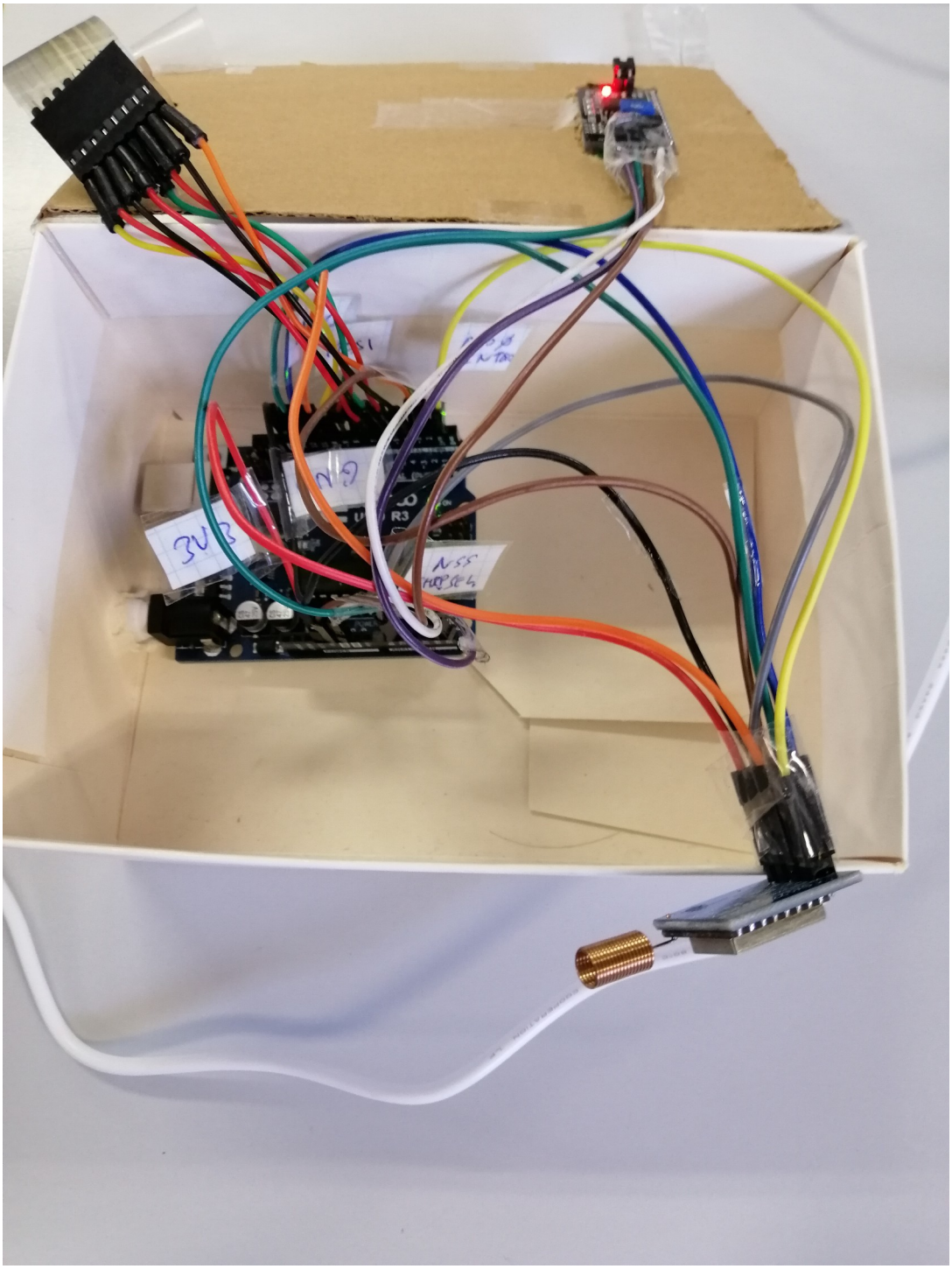
para la realización del proyecto se ha utilizado una librería sacad de internet.

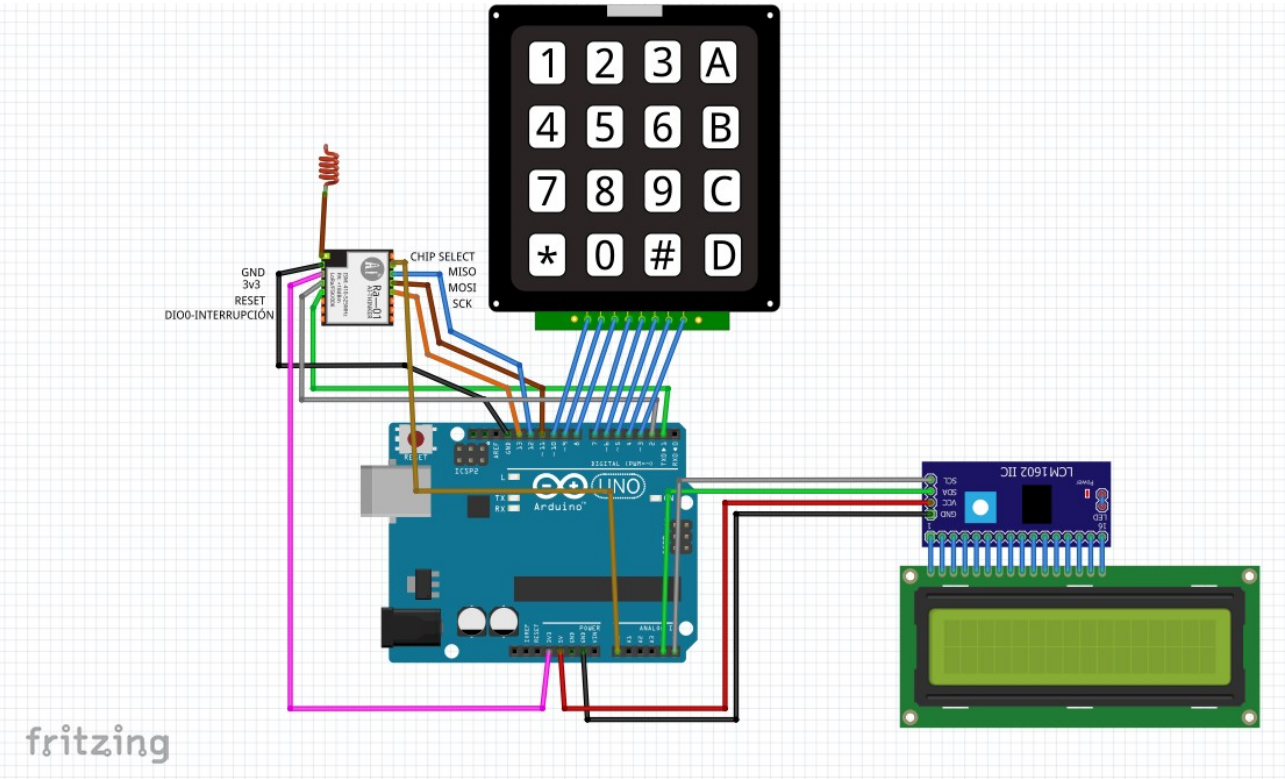
En esta librería hay unos programas de ejemplo que se han tenido que corregir para que funcionaran: LoRaReceiver y LoRaSender.

librería:
https://github.com/sandeepmistry/arduino-LoRa.git

api:
https://github.com/sandeepmistry/arduino-LoRa/blob/master/API.md

EMISOR


3V3
GND
R3


1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D
GND
3v3
RESET
DIO0-INTERRUPCIÓN
CHIP SELECT
MISO
MOSI
SCK
LCM1602 IIC
fritzing

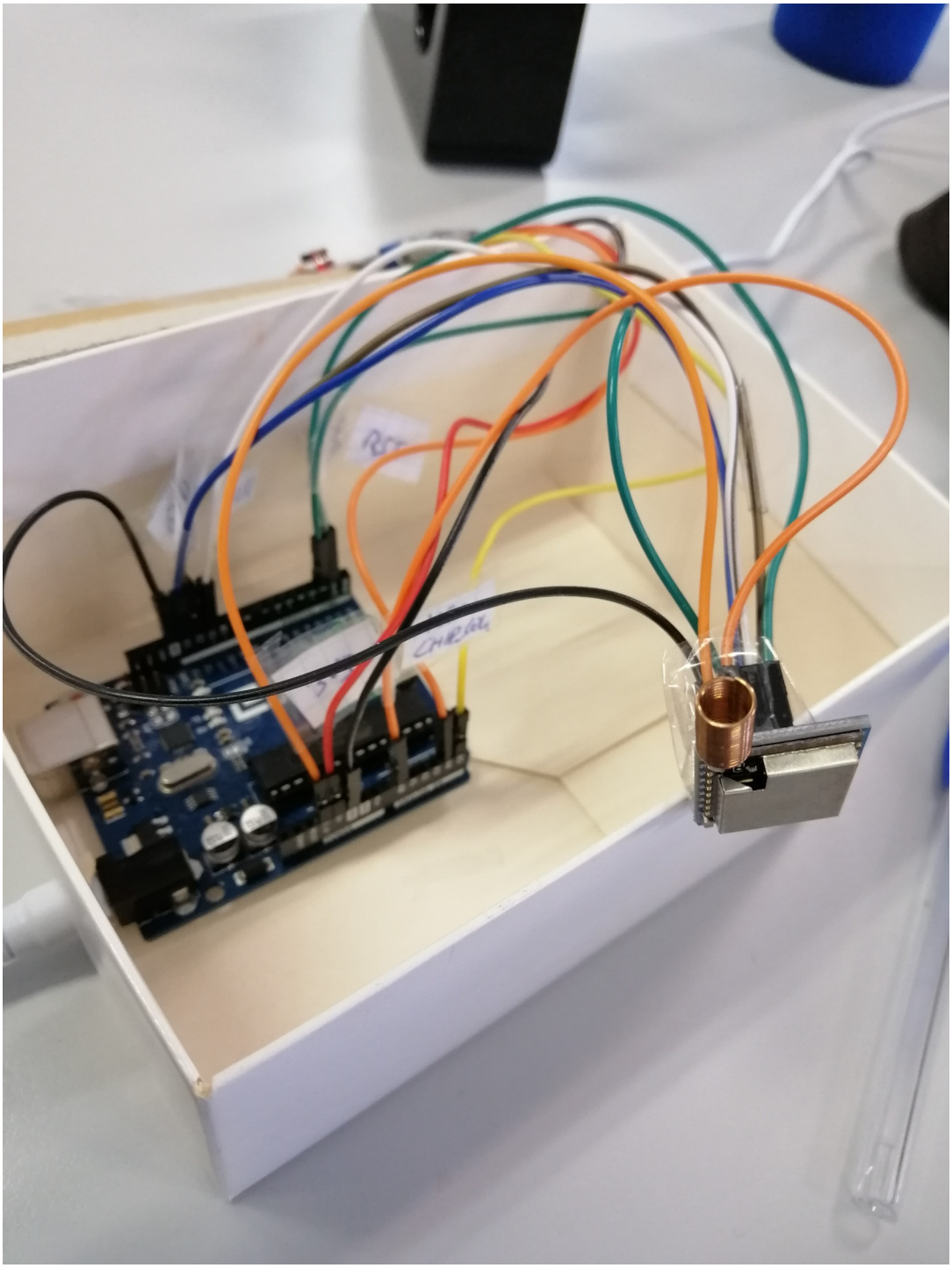
PROGRAMA EMISOR

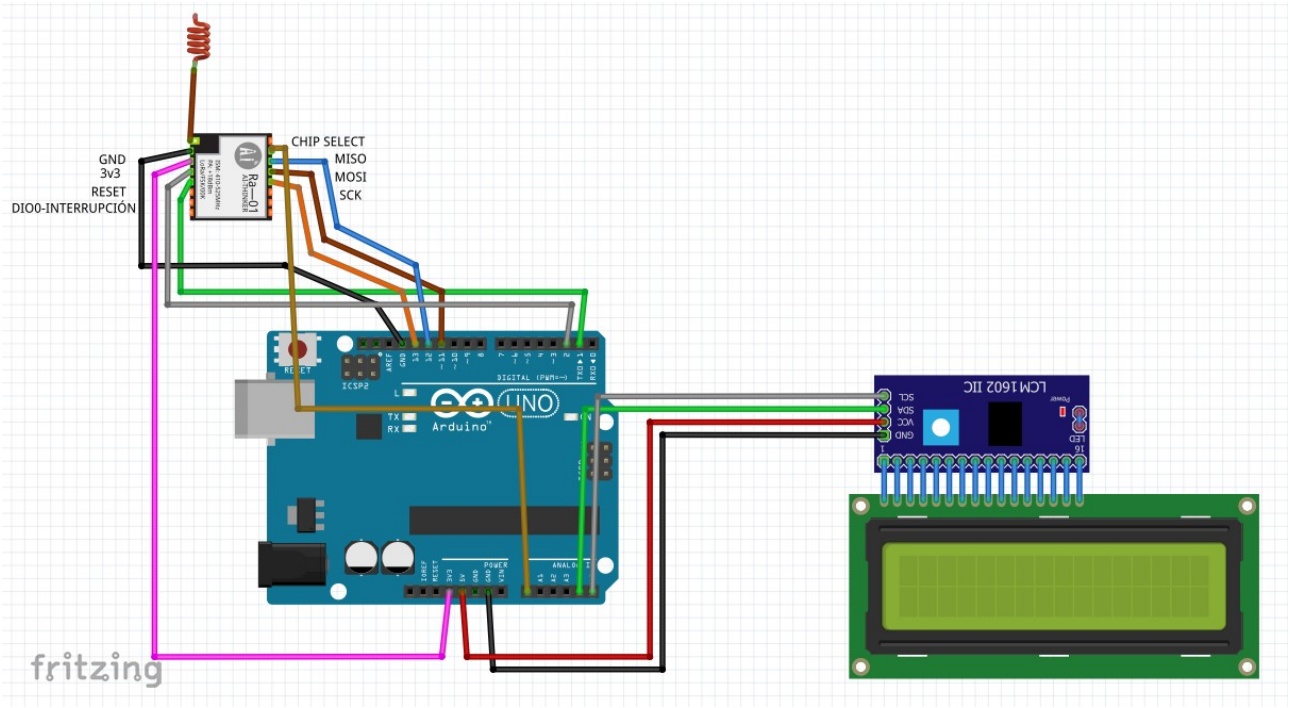
El programa emisor consiste en un bucle que comprueba constantemente si se pulsa una tecla. Si se pulsa una tecla se almacena en una variable y si se pulsa la tecla '#' se envía el contenido de dicha variable. El dato que se va ha enviar se muestra por el LCD y por consola.

```
/*

Emisor.

pines del Arduino:
TECLADO: columnas 3, 4, 5, 6 filas 7, 8, 9, 10.
I2C: sda A4, scl A5.
SPI: mosi 11, miso 12, sck 13, ss (no se tiliza).
MODULO RADIO: chip-select A0 (pin NSS del modulo de radio), interrupción 1 (pin DIO0 del módulo radio) reset 2.

*/

//librería para el teclado
#include <Keypad.h>
const byte COLS = 4; //cuatro columnas
const byte ROWS = 4; //cuatro filas
byte colPins[COLS] = {6, 5, 4, 3}; //pines de conexión a las columnas del teclado
byte rowPins[ROWS] = {10, 9, 8, 7}; //pines de conexión a las filas del teclado
//definicion de los simbolos de los botones
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

//inicializa una instancia de clase Keypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

//libreria para la comunicacion I2C con el LCD
//En Arduino Uno, Nano y Mini Pro, SDA es el pin A4 y el SCK el pin A5.
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Inicia el LCD en la dirección 0x27, con 16 caracteres y 2 líneas

//librería para la comunicaión SPI para comunicarse con el módulo de radio.
//en Arduino Uno, MOSI es el pin 11, MISO es el pin 12, SCK es el pin 13, y SS es el pin 10
#include <SPI.h>

//librería que resuelve la comunicíon entre el módulo y el Arduino.
//la librería no utiliza encriptación de datos. para  enviar datos encriptados hay que encriptarlos antes de mandarlos a la libreria.
#include <LoRa.h>

String mensaje = "";  //datos a enviar

void setup() {
  //inicializa el LCD
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);

   //inicializa el puerto serie del arduino a 9600 baudios
  Serial.begin(9600);
  Serial.println("LoRa Sender");
```

```
  //se tienen que definir los pines ChipSelect, Reset e Interrupcion en el Arduino para manejar el con el módulo de radio.
  int ChipSelectPin = A0; //pin NSS del módulo LoRa
  int ResetPin = 2; //pin RST del módulo LoRa
  int InterruptPin = 1; //pin DIO0 del modólo LoRa
  LoRa.setPins(ChipSelectPin, ResetPin, InterruptPin);

  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Fallo en LoRa ");
    while (1);
  }

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("LoRa iniciado");
  delay(1000);
}

void loop() {

  //consulta si se ha pulsado una tecla
  char customKey = customKeypad.getKey();
  //si se pulsa una tecla del teclado
  if (customKey){
    if(customKey == '#'){
      // envia el paquete
      LoRa.beginPacket();
      LoRa.print(mensaje);
      LoRa.endPacket();

      //muestra el resultado por consola
      Serial.print("Sending packet: ");
      Serial.println(mensaje);

      //muestra el resultado por LCD
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Enviado: ");
      lcd.setCursor(0,1);
      lcd.print(mensaje);
      delay(2000);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("LoRa iniciado");

      //limpia el mensaje
      mensaje = "";

    }else{
        //acumula las teclas pulsadas en el mensaje
        mensaje = mensaje + customKey;

        //muestra los datos por la consola
        Serial.print(customKey);
        // muestra los datos por el LCD
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("A enviar.");
        lcd.setCursor(0,1);
        lcd.print(mensaje);
      }
  }
}
```

RECEPTOR




GND
3v3
RESET
DIO0-INTERRUPCIÓN
CHIP SELECT
MISO
MOSI
SCK
LCM1602 IIC
fritzing

PROGRAMA RECEPTOR

El programa receptor consiste en un bucle que esta comprobando continuamente si se ha recibido un dato. En caso de que se reciba se puestra por el LCD y por consola.

```
/*

Receptor.

pines del Arduino:
I2C: sda A4, scl A5.
SPI: mosi 11, miso 12, sck 13, ss (no se tiliza).
MODULO RADIO: chip-select A0 (pin NSS del modulo de radio), interrupcion 1 (pin DIO0 del módulo radio) reset 2.

*/

//libreria para la comunicacion I2C con el LCD
//En Arduino Uno, Nano y Mini Pro, SDA es el pin A4 y el SCK el pin A5.
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Inicia el LCD en la dirección 0x27, con 16 caracteres y 2 líneas

//librería para la comunicaión SPI para comunicarse con el módulo de radio.
//en Arduino Uno, MOSI es el pin 11, MISO es el pin 12, SCK es el pin 13, y SS es el pin 10
#include <SPI.h>

//librería que resuelve la comunicíon entre el módulo y el Arduino.
//la librería no utiliza encriptación de datos. para  enviar datos encriptados hay que encriptarlos antes de mandarlos a la libreria.
#include <LoRa.h>

void setup() {
  //inicializa el LCD
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);

  //inicializa el puerto serie a 9600 baudios
  Serial.begin(9600);
  Serial.println("LoRa Receiver");

  //se tienen que definir los pines ChipSelect, Reset e Interrupcion en el Arduino para manejar el con el módulo de radio.
  int ChipSelectPin = A0; //pin NSS del módulo LoRa
  int ResetPin = 2; //pin RST del módulo LoRa
  int InterruptPin = 1; //pin DIO0 del modólo LoRa
  LoRa.setPins(ChipSelectPin, ResetPin, InterruptPin);

  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Fallo en LoRa ");
    while (1);
  }

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("LoRa iniciado");
  delay(1000);
}
```

```
void loop() {
  char paquete[16] = "";      //contendrá los datos recibidos
  int packetSize = LoRa.parsePacket();   // comprueba que ha recibido algo

  //si ha recibido algo
  if (packetSize) {
    int i = 0;
    while (LoRa.available()) {   //mientras hay datos recibidos
      paquete[i] = LoRa.read();      // lee lo que ha recibido
      i++;                        //lo acumula en "paquete"
    }
    int potencia = LoRa.packetRssi();  //lee la potecia recibida

    //muestra el resultado por consola
    Serial.print("Received packet '");
    Serial.print(paquete);
    Serial.print("' with RSSI ");
    Serial.println(potencia);

    //muestra el resultado por LCD
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Recibido:");
    lcd.setCursor(0,1);
    lcd.print(paquete);

    //limpia la variable paquete
    for(int j = 0; j < 16; j++){
      paquete[j] = ' ';
    }

    delay(1000); //necesita un retardo
  }
}
```

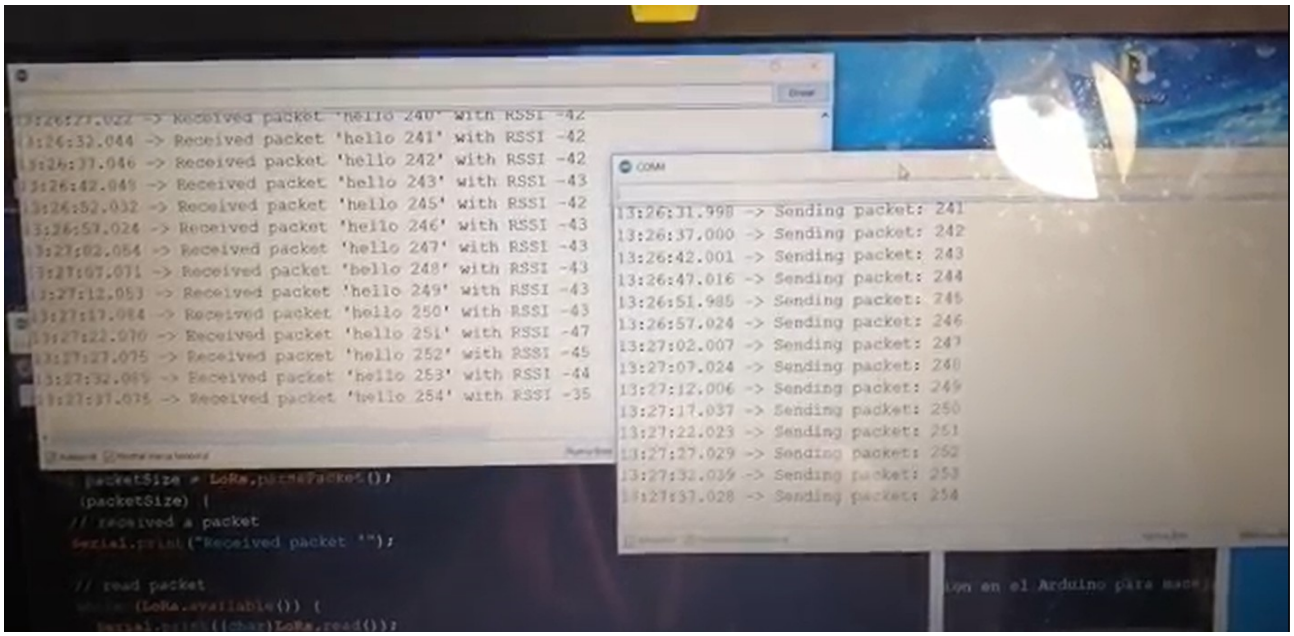
pruebas de prototipo

prueba emision y recepcion

se envia una cadena de caracteres

pruebas de emisión y recepción
se emite una cadena de caracteres ‘hello nº’ cada 5 segundos.

programas:
LoRaSender-modificado-LCD-keypad
LoRaReceiver-modificado-LCD



LoRaSender-modificado-LCD-keypad

```
/*

Emisor.

pines del Arduino:
TECLADO: columnas 3, 4, 5, 6 filas 7, 8, 9, 10.
I2C: sda A4, scl A5.
SPI: mosi 11, miso 12, sck 13, ss (no se tiliza).
MODULO RADIO: chip-select A0 (pin NSS del modulo de radio), interrupcion 1 (pin DIO0 del módulo radio) reset 2.

*/

//librería para el teclado
#include <Keypad.h>
const byte COLS = 4; //cuatro columnas
const byte ROWS = 4; //cuatro filas
byte colPins[COLS] = {6, 5, 4, 3}; //pines de conexión a las columnas del teclado
byte rowPins[ROWS] = {10, 9, 8, 7}; //pines de conexión a las filas del teclado
//definicion de los simbolos de los botones
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
//inicializa una instancia de clase Keypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

//libreria para la comunicacion I2C con el LCD
//En Arduino Uno, Nano y Mini Pro, SDA es el pin A4 y el SCK el pin A5.
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Inicia el LCD en la dirección 0x27, con 16 caracteres y 2 líneas

//librería para la comunicaión SPI para comunicarse con el módulo de radio.
//en Arduino Uno, MOSI es el pin 11, MISO es el pin 12, SCK es el pin 13, y SS es el pin 10
#include <SPI.h>

//librería que resuelve la comunicíon entre el módulo y el Arduino.
//la librería no utiliza encriptación de datos. para  enviar datos encriptados hay que encriptarlos antes de mandarlos a la libreria.
#include <LoRa.h>

String mensaje = "";  //datos a enviar

void setup() {
  //inicializa el LCD
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);

   //inicializa el puerto serie del arduino a 9600 baudios
  Serial.begin(9600);
  Serial.println("LoRa Sender");

  //se tienen que definir los pines ChipSelect, Reset e Interrupcion en el Arduino para manejar el con el módulo de radio.
  int ChipSelectPin = A0; //pin NSS del módulo LoRa
  int ResetPin = 2; //pin RST del módulo LoRa
  int InterruptPin = 1; //pin DIO0 del modólo LoRa
  LoRa.setPins(ChipSelectPin, ResetPin, InterruptPin);
```

```
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Fallo en LoRa ");
    while (1);
  }

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("LoRa iniciado");
  delay(1000);
}

void loop() {

  //consulta si se ha pulsado una tecla
  char customKey = customKeypad.getKey();
  //si se pulsa una tecla del teclado
  if (customKey){
    if(customKey == '#'){
      // envia el paquete
      LoRa.beginPacket();
      LoRa.print(mensaje);
      LoRa.endPacket();

      //muestra el resultado por consola
      Serial.print("Sending packet: ");
      Serial.println(mensaje);

      //muestra el resultado por LCD
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Enviado: ");
      lcd.setCursor(0,1);
      lcd.print(mensaje);
      delay(2000);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("LoRa iniciado");

      //limpia el mensaje
      mensaje = "";

    }else{
        //acumula las teclas pulsadas en el mensaje
        mensaje = mensaje + customKey;

        //muestra los datos por la consola
        Serial.print(customKey);
        // muestra los datos por el LCD
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("A enviar.");
        lcd.setCursor(0,1);
        lcd.print(mensaje);
      }
  }
}
```

LoRaReceiver-modificado-LCD

```
/*

Receptor.

pines del Arduino:
I2C: sda A4, scl A5.
SPI: mosi 11, miso 12, sck 13, ss (no se tiliza).
MODULO RADIO: chip-select A0 (pin NSS del modulo de radio), interrupcion 1 (pin DIO0 del módulo radio) reset 2.

*/

//libreria para la comunicacion I2C con el LCD
//En Arduino Uno, Nano y Mini Pro, SDA es el pin A4 y el SCK el pin A5.
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Inicia el LCD en la dirección 0x27, con 16 caracteres y 2 líneas

//librería para la comunicaión SPI para comunicarse con el módulo de radio.
//en Arduino Uno, MOSI es el pin 11, MISO es el pin 12, SCK es el pin 13, y SS es el pin 10
#include <SPI.h>

//librería que resuelve la comunicíon entre el módulo y el Arduino.
//la librería no utiliza encriptación de datos. para  enviar datos encriptados hay que encriptarlos antes de mandarlos a la libreria.
#include <LoRa.h>

void setup() {
  //inicializa el LCD
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);

  //inicializa el puerto serie a 9600 baudios
  Serial.begin(9600);
  Serial.println("LoRa Receiver");

  //se tienen que definir los pines ChipSelect, Reset e Interrupcion en el Arduino para manejar el con el módulo de radio.
  int ChipSelectPin = A0; //pin NSS del módulo LoRa
  int ResetPin = 2; //pin RST del módulo LoRa
  int InterruptPin = 1; //pin DIO0 del modólo LoRa
  LoRa.setPins(ChipSelectPin, ResetPin, InterruptPin);

  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Fallo en LoRa ");
    while (1);
  }

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("LoRa iniciado");
  delay(1000);
}
```

```
void loop() {
  char paquete[16] = "";      //contendrá los datos recibidos
  int packetSize = LoRa.parsePacket();   // comprueba que ha recibido algo

  //si ha recibido algo
  if (packetSize) {
    int i = 0;
    while (LoRa.available()) {   //mientras hay datos recibidos
      paquete[i] = LoRa.read();      // lee lo que ha recibido
      i++;                    //lo acumula en "paquete"
    }
    int potencia = LoRa.packetRssi();  //lee la potecia recibida

    //muestra el resultado por consola
    Serial.print("Received packet '");
    Serial.print(paquete);
    Serial.print("' with RSSI ");
    Serial.println(potencia);

    //muestra el resultado por LCD
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Recibido:");
    lcd.setCursor(0,1);
    lcd.print(paquete);

    //limpia la variable paquete
    for(int j = 0; j < 16; j++){
      paquete[j] = ' ';
    }

    delay(1000); //necesita un retardo
  }
}
```