(1) 868mhz / 915MhzOLED LoRaSender
#include
#include
#include
#include "SSD1306.h"
#include "images.h"

#define SCK 5 // GPIO5 - SX1278's SCK
#define MISO 19 // GPIO19 - SX1278's MISO
#define MOSI 27 // GPIO27 - SX1278's MOSI
#define SS 18 // GPIO18 - SX1278's CS
#define RST 14 // GPIO14 - SX1278's RESET
#define DI0 26 // GPIO26 - SX1278's IRQ (interrupt request)
#define BAND 868E6 // 915E6

unsigned int counter = 0;

SSD1306 display (0x3c, 4, 15);
String rssi = "RSSI -";
String packSize = "-";
String packet;

void setup () {
  pinMode (16, OUTPUT);
  pinMode (2, OUTPUT);

  digitalWrite (16, LOW); // set GPIO16 low to reset OLED
  delay (50);
  digitalWrite (16, HIGH); // while OLED is running, GPIO16 must go high

  Serial.begin (9600);
  while (! Serial);
  Serial.println ();
  Serial.println ("LoRa Sender Test");

  SPI.begin (SCK, MISO, MOSI, SS);
  LoRa.setPins (SS, RST, DI0);
  if (! LoRa.begin (868)) {
    Serial.println ("Starting LoRa failed!");
    while (1);
  }
  //LoRa.onReceive(cbk);
// LoRa.receive ();
  Serial.println ("init ok");
  display.init ();
  display.flipScreenVertically ();
  display.setFont (ArialMT\_Plain\_10);
  delay (1500);
}

void loop () {
  display.clear ();
  display.setTextAlignment (TEXT\_ALIGN\_LEFT);
  display.setFont (ArialMT\_Plain\_10);

  display.drawString (0, 0, "Sending packet:");
  display.drawString (90, 0, String (counter));
  display.display ();

  // send packet
  LoRa.beginPacket ();
  LoRa.print ("hello");
  LoRa.print (counter);
  LoRa.endPacket ();

  counter ++;
  digitalWrite (2, HIGH); // turn the LED on (HIGH is the voltage level)
  delay (1000); // wait for a second
  digitalWrite (2, LOW); // turn the LED off by making the voltage LOW
  delay (1000); // wait for a second
}

(2) 868mhz / 915Mhz OLED LoRaSender
#include
#include
#include
#include "SSD1306.h"
#include "images.h"

#define SCK 5 // GPIO5 - SX1278's SCK
#define MISO 19 // GPIO19 - SX1278's MISO
#define MOSI 27 // GPIO27 - SX1278's MOSI
#define SS 18 // GPIO18 - SX1278's CS
#define RST 14 // GPIO14 - SX1278's RESET
#define DI0 26 // GPIO26 - SX1278's IRQ (interrupt request)
#define BAND 868E6 // 915E6

SSD1306 display (0x3c, 4, 15);
String rssi = "RSSI -";
String packSize = "-";
String packet;

void loraData () {
  display.clear ();
  display.setTextAlignment (TEXT\_ALIGN\_LEFT);
  display.setFont (ArialMT\_Plain\_10);
  display.drawString (0, 15, "Received" + packSize + "bytes");
  display.drawStringMaxWidth (0, 26, 128, packet);
  display.drawString (0, 0, rssi);
  display.display ();
}

void cbk (int packetSize) {
  packet = "";
  packSize = String (packetSize, DEC);
  for (int i = 0; i <packetsize; br="" i="" lora.read="" packet="" style="box-sizing: border-box;">  rssi = "RSSI" + string (LoRa.packetRssi (), DEC);
  loraData ();
}

void setup () {
  pinMode (16, OUTPUT);
  digitalWrite (16, LOW); // set GPIO16 low to reset OLED
  delay (50);
  digitalWrite (16, HIGH); // while OLED is running, GPIO16 must go high,

  Serial.begin (9600);
  while (! Serial);
  Serial.println ();
  Serial.println ("LoRa Receiver Callback");
  SPI.begin (SCK, MISO, MOSI, SS);
  LoRa.setPins (SS, RST, DI0);
  if (! LoRa.begin (868E6)) {
    Serial.println ("Starting LoRa failed!");
    while (1);
  }
  //LoRa.onReceive(cbk);
  LoRa.receive ();
  Serial.println ("init ok");
  display.init ();
  display.flipScreenVertically ();
  display.setFont (ArialMT\_Plain\_10);

  delay (1500);
}

void loop () {
  int packetSize = LoRa.parsePacket ();
  if (packetSize) {cbk (packetSize); }
  delay (10);
}<packetsize; br="" i="" lora.read="" packet="" style="box-sizing: border-box;">