

Description:

The 915MHz antenna needs to be in connection with the IPEX interface (if the antenna is not connected, it may damage the LoRa chip)

Lithium battery charge and discharge circuit, when the battery is full, the blue LED will stop working. When using, pay attention to the positive and negative of the battery, otherwise it will be damaged!

With IO port touch screen touch signal input, you need to add the 100nF pull-down capacitor to this pin!

Note:This product does not include the battery.

Example:

This product is a SX1276 chip based on ESP32 WIFI increased OLED, namely LoRa remote modem, 915MHz frequency, high sensitivity is over-148dBm, + 20dBm output power, high reliability, long transmission distance. the onboard 32MB Flash wi-fi antenna, 0.96 inch blue oled display, lithium battery charging circuit, CP2102 interface and USB serial chip, the perfect support for development environment, can be used for program checking and product development is very easy and fast.

Operating voltage: 3.3V to 7V

Operating temperature range: -40 ° C to + 90 ° C

Support for Sniffer software protocol analysis, Station, SoftAP, and Wi-Fi Direct modes

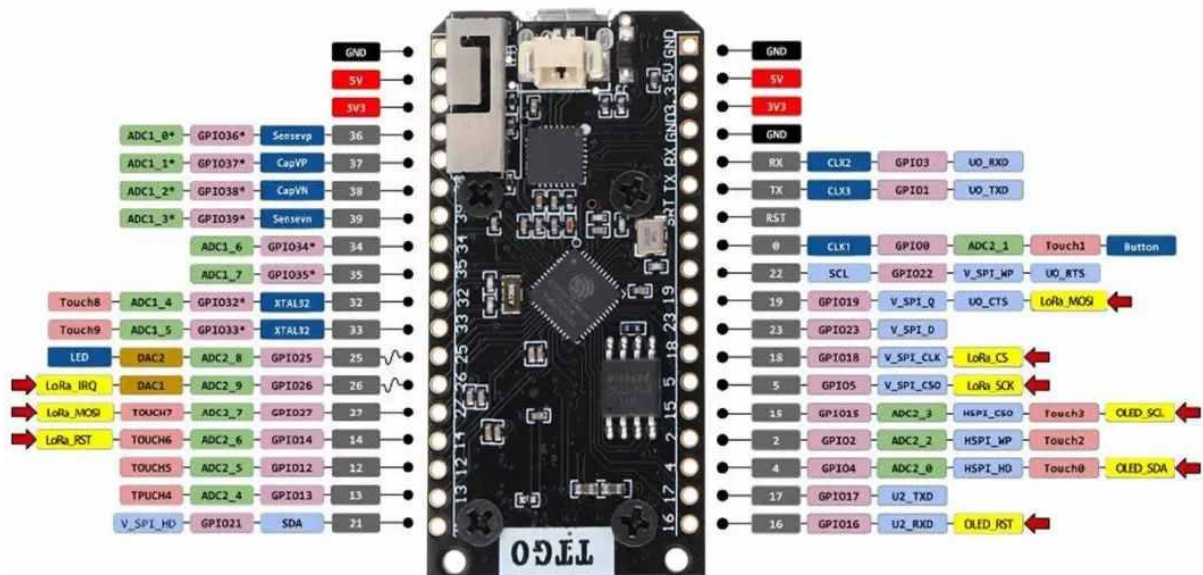
Data rates: 150 Mbps @ 11n HT40., 72 Mbps @ 11n HT20, 54 Mbps @ 11g, 11 Mbps @ 11b

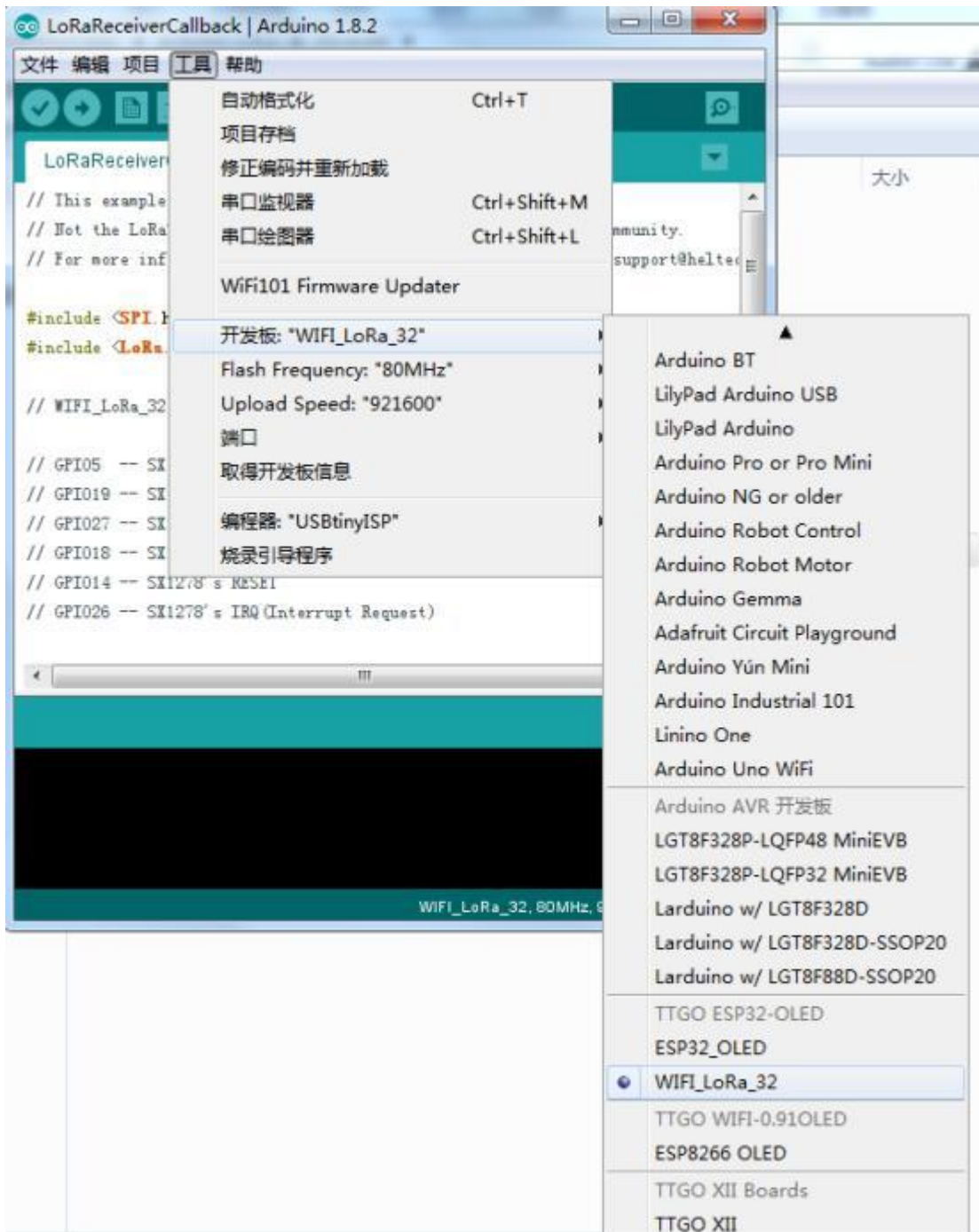
transmit power: 19.5 dBm @ 11b, 16.5 dBm @ 11g, 15.5 dBm @ 11n

receiver sensitivity up to -98 dBm

UDP sustained throughput of 135 Mbps

Pin diagram:





For example One:

(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 DIO 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 < 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, DIO);
  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);
}

```

Package included:

2 x ESP32 LoRa OLED Module
 2 x Line
 4 x Pin
 2 x 915mhz Spring Antenna

