**<https://randomnerdtutorials.com/ttgo-lora32-sx1276-arduino-ide/>**

**TTGO LoRa32 SX1276 OLED Board:**

|  |  |
| --- | --- |
| TTGO LoRa32 OLED SX1276 Overview | The **TTGO LoRa32 SX1276 OLED** is an ESP32 development board with a built-in LoRa chip and an SSD1306 0.96 inch OLED display. In this guide, we’ll show you how to: send and receive LoRa packets (point to point communication) and use the OLED display with Arduino IDE. |

The board also features several GPIOs to connect peripherals, PRG (BOOT) and RST buttons, and a lithium battery connector. For a more in-depth overview of this board, read: [TTGO LoRa32 SX1276 OLED Review](https://makeradvisor.com/esp32-sx1276-lora-ssd1306-oled/).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The OLED displays communicates using [I2C communication protocol](https://randomnerdtutorials.com/esp32-i2c-communication-arduino-ide/). It is internally connected to the ESP32 on the following pins: | |  |  | | --- | --- | | **OLED** (built-in) | **ESP32** | | **SDA** | GPIO 4 | | **SCL** | GPIO 15 | | **RST** | GPIO 16 | |
| The SX1276 LoRa chip communicates via SPI communication protocol, and it is internally connected to the ESP32 on the following GPIOs: | |  |  | | --- | --- | | **SX1276 LoRa** | **ESP32** | | MISO | GPIO 19 | | MOSI | GPIO 27 | | SCK | GPIO 5 | | CS | GPIO 18 | | IRQ | GPIO 26 | | RST | GPIO 14 | |

**Install ESP32 Boards on Arduino IDE**

To program the TTGO LoRa32 board, we’ll use Arduino IDE. So, you must have Arduino IDE installed as well as the ESP32 add-on. Follow the next guide to install the ESP32 package on Arduino IDE:

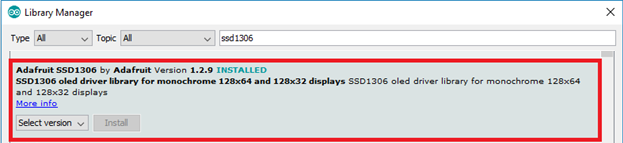
[Installing the ESP32 Board in Arduino IDE (Windows, Mac OS X, Linux)](https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/)

* File / Preferences
* Enter **https://dl.espressif.com/dl/package\_esp32\_index.json** into the “Additional Board Manager URLs”
* https://dl.espressif.com/dl/package\_esp32\_index.json, http://arduino.esp8266.com/stable/package\_esp8266com\_index.json
* Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager…**
* Search for **ESP32** and press install button for the “**ESP32 by Espressif Systems**“:

**Installing Libraries**

1. Open your Arduino IDE and go to **Sketch**> **Include Library** > **Manage Libraries**. The Library Manager should open.

|  |  |  |
| --- | --- | --- |
| **Library** | **Buscar** | **Para** |
| Adafruit SSD1306 | SSD1306 | Display |
| Adafruit GFX | GFX | Display |
| LoRa | LoRa | Lora |



After installing the libraries, restart your Arduino IDE.

**LoRa Sender Sketch**

Copy the following code to your Arduino IDE. This code sends a “hello” message followed by a counter via LoRa every 10 seconds. It also displays the counter on the OLED display.

//Libraries for LoRa

#include <SPI.h>

#include <LoRa.h>

//Libraries for OLED Display, to interface with I2C OLED Display

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

//define the pins used by the LoRa transceiver module

#define SCK 5

#define MISO 19

#define MOSI 27

#define SS 18

#define RST 14

#define DIO0 26

//433E6 for Asia

//866E6 for Europe

//915E6 for North America

#define BAND 866E6

//OLED pins

#define OLED\_SDA 4

#define OLED\_SCL 15

#define OLED\_RST 16

// OLED Size

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

//packet counter

int counter = 0;

// Adafruit\_DDS1306 object called display

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RST);

void setup() {

Serial.begin(9600);

//reset OLED display via software, set RST as outout

// then LOW few milliseconds and then HIgH

pinMode(OLED\_RST, OUTPUT);

digitalWrite(OLED\_RST, LOW);

delay(20);

digitalWrite(OLED\_RST, HIGH);

//initialize OLED, start I2C

Wire.begin(OLED\_SDA, OLED\_SCL);

// initialize display, set false to avoid default I2C pins

// and use the pins set for GPIO4 and GPIO15

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3c, false, false)) {

// Address 0x3C for 128x32

Serial.println(F("SSD1306 allocation failed"));

for(;;); // Don't proceed, loop forever

}

// Write as a test “LORA SENDER”

display.clearDisplay();

display.setTextColor(WHITE);

display.setTextSize(1);

display.setCursor(0,0);

display.print("LORA SENDER ");

display.display();

//SPI LoRa pins

SPI.begin(SCK, MISO, MOSI, SS);

//setup LoRa transceiver module

LoRa.setPins(SS, RST, DIO0);

if (!LoRa.begin(BAND)) {

Serial.println("Starting LoRa failed!");

while (1);

}

display.setCursor(0,10);

display.print("LoRa Initializing OK!");

display.display();

delay(2000);

}

void loop() {

Serial.print("Sending packet: ");

Serial.println(counter);

//Send LoRa packet to receiver

LoRa.beginPacket();

LoRa.print("hello ");

LoRa.print(counter);

LoRa.endPacket();

display.clearDisplay();

display.setCursor(0,0);

display.println("LORA SENDER");

display.setCursor(0,20);

display.setTextSize(1);

display.print("LoRa packet sent.");

display.setCursor(0,30);

display.print("Counter:");

display.setCursor(50,30);

display.print(counter);

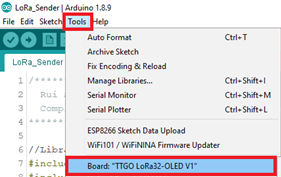
display.display();

counter++;

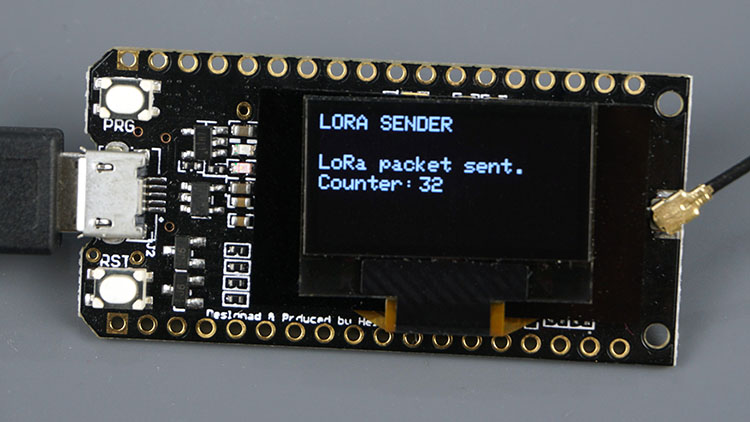
delay(10000);

}

**Tools**> **Board**and select the **TTGO LoRa32-OLED V1** board.



After uploading the code to your board, it should start sending LoRa packets.



Now, upload the receiver sketch to another TTGO LoRa32 OLED board.

#include <SPI.h>

#include <LoRa.h>

//Libraries for OLED Display

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

//define the pins used by the LoRa transceiver module

#define SCK 5

#define MISO 19

#define MOSI 27

#define SS 18

#define RST 14

#define DIO0 26

//433E6 for Asia

//866E6 for Europe

//915E6 for North America

#define BAND 866E6

//OLED pins

#define OLED\_SDA 4

#define OLED\_SCL 15

#define OLED\_RST 16

Adafruit\_SSD1306 display(128, 64, &Wire, OLED\_RST);

String LoRaData;

void setup() {

Serial.begin(115200);

//reset OLED display via software

pinMode(OLED\_RST, OUTPUT);

digitalWrite(OLED\_RST, LOW);

delay(20);

digitalWrite(OLED\_RST, HIGH);

//initialize OLED

Wire.begin(OLED\_SDA, OLED\_SCL);

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3c, false, false)) {

// Address 0x3C for 128x32

Serial.println(F("SSD1306 allocation failed"));

for(;;); // Don't proceed, loop forever

}

display.clearDisplay();

display.setTextColor(WHITE);

display.setTextSize(1);

display.setCursor(0,0);

display.print("LORA RECEIVER ");

display.display();

Serial.println("LoRa Receiver Test");

//SPI LoRa pins

SPI.begin(SCK, MISO, MOSI, SS);

//setup LoRa transceiver module

LoRa.setPins(SS, RST, DIO0);

if (!LoRa.begin(BAND)) {

Serial.println("Starting LoRa failed!");

while (1);

}

Serial.println("LoRa Initializing OK!");

display.setCursor(0,10);

display.println("LoRa Initializing OK!");

display.display();

}

void loop() {

//try to parse packet

int packetSize = LoRa.parsePacket();

if (packetSize) {

//received a packet

Serial.print("Received packet ");

//read packet

while (LoRa.available()) {

LoRaData = LoRa.readString();

Serial.print(LoRaData);

}

//print RSSI of packet

int rssi = LoRa.packetRssi();

Serial.print(" with RSSI ");

Serial.println(rssi);

display.clearDisplay();

display.setCursor(0,0);

display.print("LORA RECEIVER");

display.setCursor(0,20);

display.print("Received packet:");

display.setCursor(0,30);

display.print(LoRaData);

display.setCursor(0,40);

display.print("RSSI:");

display.setCursor(30,40);

display.print(rssi);

display.display();

}

}

This sketch is very similar with the previous one. We just need to modify some lines to receive LoRa packets instead of sending.

In the loop(), we check if there are new packets to receive using the parsePacket() method.

int packetSize = LoRa.parsePacket();

If there’s a new packet, we’ll read its content. To read the incoming data, use the readString() method. The data received is saved on the LoRaData variable.

if (packetSize) {

Serial.print("Received packet ");

//read packet

while (LoRa.available()) {

LoRaData = LoRa.readString();

Serial.print(LoRaData);

}

We also get the RSSI of the received packet by using the packetRSSI() method.

int rssi = LoRa.packetRssi();

Finally, display the received message, as well as the RSSI.

display.clearDisplay();

display.setCursor(0,0);

display.print("LORA RECEIVER");

display.setCursor(0,20);

display.print("Received packet:");

display.setCursor(0,30);

display.print(LoRaData);

display.setCursor(0,40);

display.print("RSSI:");

display.setCursor(30,40);

display.print(rssi);

display.display();

Upload the code to your board.

Don’t forget you need to select the **TTGO LoRa32-OLED V1** in the Boards menu.

After uploading the code, it should start receiving the LoRa packets from the other board.