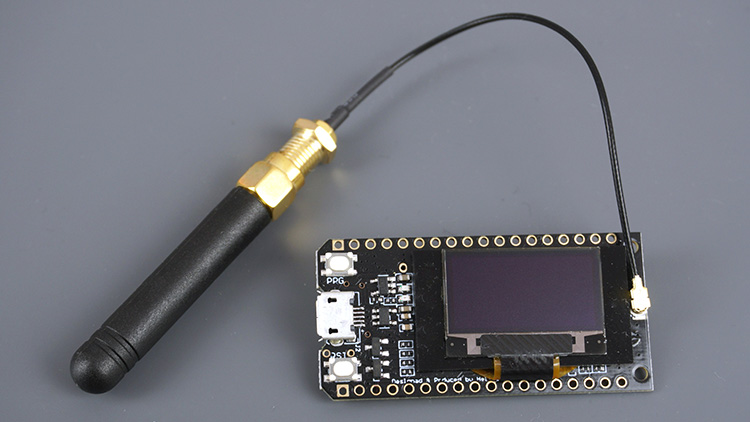
**<https://randomnerdtutorials.com/ttgo-lora32-sx1276-arduino-ide/>**

**TTGO LoRa32 SX1276 OLED Board:**

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.

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**“:

([Guide for OLED display with ESP32](https://randomnerdtutorials.com/esp32-ssd1306-oled-display-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 |

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, if you haven’t already:

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

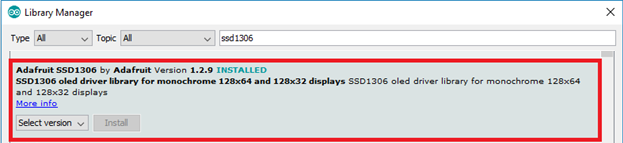
**Installing OLED Libraries**

In this tutorial we’ll use two Adafruit libraries:

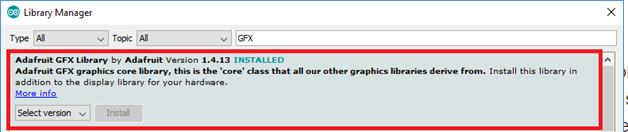
* [Adafruit\_SSD1306 library](https://github.com/adafruit/Adafruit_SSD1306)
* [Adafruit\_GFX library](https://github.com/adafruit/Adafruit-GFX-Library).

1. In Arduino IDE and go to **Sketch**> **Include Library** > **Manage Libraries**.

2. Type “**SSD1306**” in the search box



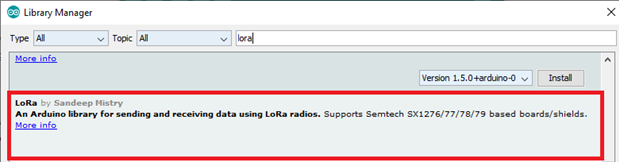
3. Type “**GFX**” in the search box and install the library.



**Installing LoRa Library**

We’ll be using the [arduino-LoRa library by sandeep mistry](https://github.com/sandeepmistry/arduino-LoRa" \t "_blank).

Search for “**LoRa**“. Select the LoRa library highlighted in the figure below, and install it.



After installing the libraries, restart your Arduino IDE.

**LoRa Sender Sketch**

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

#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

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

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

// Counter variable to keep track of the number of LoRa packets sent.

int counter = 0;

//Adafruit\_SSD1306 object called display.

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

void setup() {

//initialize Serial Monitor

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(); // En otro caso 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 SENDER ");

display.display();

Serial.println("LoRa Sender 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.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);

}

In the setup(), to start using the OLED you need to do a manual reset via software using the RST pin

pinMode(OLED\_RST, OUTPUT);

digitalWrite(OLED\_RST, LOW);

delay(20);

digitalWrite(OLED\_RST, HIGH);

Start an I2C communication using the defined OLED\_SDA and OLED\_SCL pins using Wire.begin().

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

En otros casos es sin parametros

Wire.begin();

Initialize the display with the following parameters. The parameters set as false ensure that the library doesn’t use the default I2C pins and use the pins defined in the code (GPIO 4 and GPIO 15).

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3c, false, false)) { // Address 0x3C for 128x32

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

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

}

Then, you can use the methods from the Adafruit library to interact with the OLED display. To learn more you can read our [tutorial about the I2C OLED display with the ESP32](https://randomnerdtutorials.com/esp32-ssd1306-oled-display-arduino-ide/).

Write the message “LORA SENDER” to the display.

display.clearDisplay();

display.setTextColor(WHITE);

display.setTextSize(1);

display.setCursor(0,0);

display.print("LORA SENDER ");

display.display();

Initialize the serial monitor for debugging purposes.

Serial.begin(115200);

Serial.println("LoRa Sender Test");

Define the SPI pins used by the LoRa chip.

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

And set up the LoRa transceiver module.

LoRa.setPins(SS, RST, DIO0);

Finally, initialize the LoRa transceiver module using the begin() method on the LoRa object and pass the frequency as argument.

if (!LoRa.begin(BAND)) {

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

while (1);

}

If we succeed in initializing the display, we write a success message on the OLED display.

display.setCursor(0,10);

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

display.display();

**loop()**

In the loop() is where we’ll send the packets. You initialize a packet with the beginPacket() method.

LoRa.beginPacket();

You write data into the packet using the print() method. As you can see in the following two lines, we’re sending a hello message followed by the counter.

LoRa.print("hello ");

LoRa.print(counter);

Then, close the packet with the endPacket() method.

LoRa.endPacket();

Next, write the counter on the OLED display

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();

After this, the counter message is incremented by one in every loop, which happens every 10 seconds.

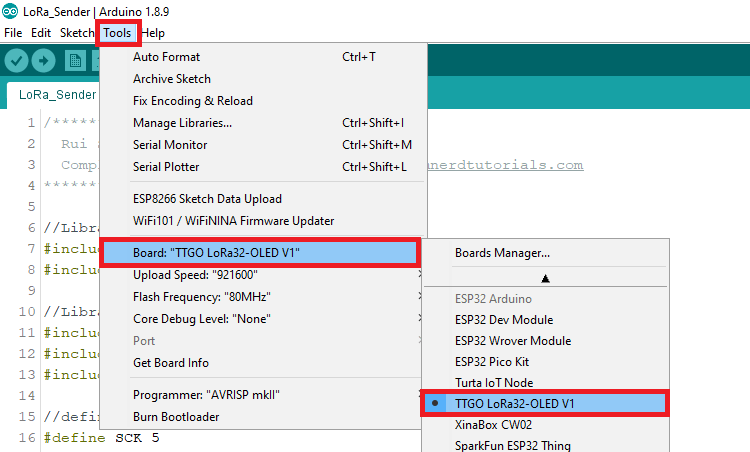
counter++;

delay(10000);

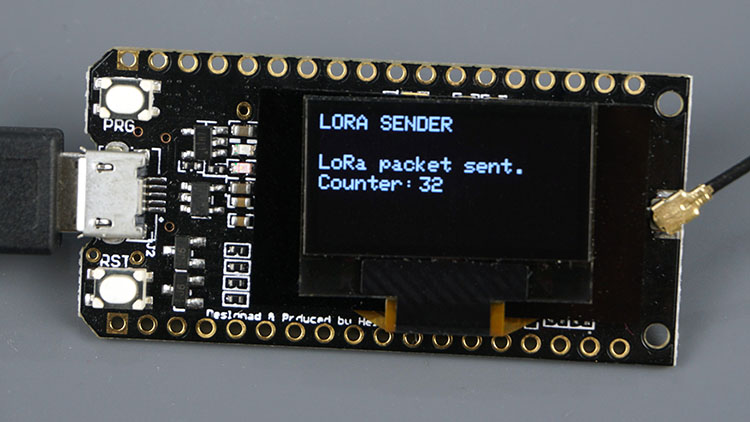
**Testing the LoRa Sender**

Upload the code to your board. You need to select the right board and COM port you’re using.

To select the board, in the Arduino IDE, go to **Tools**> **Board**and select the **TTGO LoRa32-OLED V1** board.



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



**LoRa Receiver Sketch**

Now, upload the receiver sketch to another TTGO LoRa32 OLED board. This sketch listens for LoRa packets within its range and prints the content of the packets on the OLED, as well as the RSSI (relative received signal strength).

//Libraries for LoRa

#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

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

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

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

String LoRaData;

void setup() {

//initialize Serial Monitor

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);

// Dsiplay information

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();

}

}

[View raw code](https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/raw/master/Projects/ESP32/ESP32_TTGO_LoRa_Receiver.ino)

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();

[](https://www.mediavine.com/)

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) {

//received a packet

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);

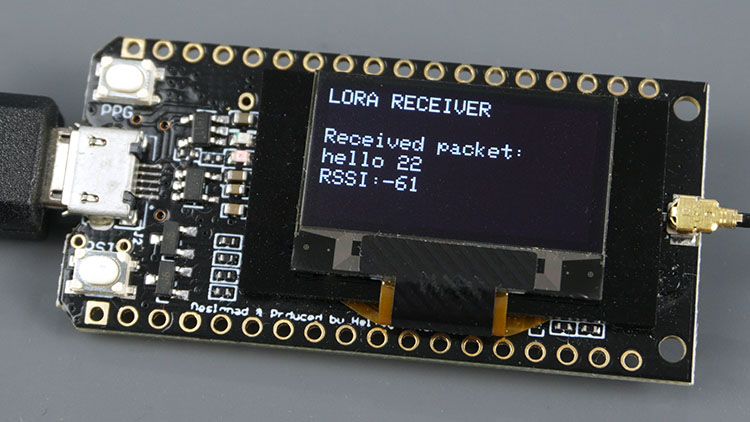
display.print(rssi);

display.display();

**Testing the LoRa Receiver**

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.



void FxMostar(String Msg)

{

display.clearDisplay();

display.setCursor(0,0);

display.println(mensaje);

display.display();

}

FxMostar(“Encendido”);

FxMostrar(“Apagado”);