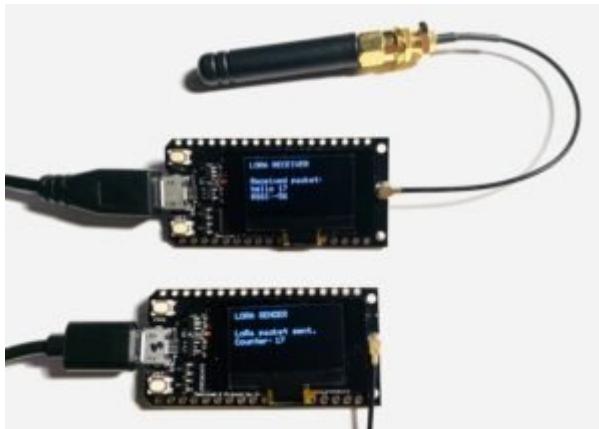


Daten per LoRa senden und empfangen mit dem TTGO LoRa32 Board



In diesem Tutorial lernst du, wie du zwei Microcontroller über LoRa miteinander kommunizieren lässt – genauer gesagt zwei Boards des Typs TTGO LoRa32. Die Kommunikation über LoRa bietet sich immer dann an, wenn dir kein WLAN zur Verfügung steht oder die Microcontroller für Bluetooth zu weit auseinander liegen.

Hierbei handelt es sich um einen Funkstandard, den du frei verwenden darfst. [Auf Wikipedia erfährst du alles Wissenswerte.](#)

Die theoretische Reichweite des Signals wird für den ländlichen Raum oft mit circa 10 Kilometern angegeben. Wir selbst haben LoRa unserem Stadtviertel ausprobiert, in dem hauptsächlich Altbauten mit dicken Mauern stehen. Hier konnten wir lediglich eine Reichweite von 200 Metern erreichen.

Wie weit das Signal reicht, ist also stark von der Umgebung abhängig – das solltest du bei deinen Projektideen bedenken.

Das benötigst du für dieses Tutorial

Um über LoRa Daten senden zu können, benötigst du einen [ESP32*](#) und das Modul [RFM95*](#). Damit du diese Daten auch empfangen kannst, ist es natürlich sinnvoll, hiervon gleich jeweils zwei zu haben – ein Mal für den Sender und ein Mal für den Empfänger.

Da der Aufbau jedoch recht aufwändig ist, verwenden wir an dieser Stelle zwei fertige Board des Typs [TTGO LoRa32*](#). Hier ist das LoRa-Modul schon fest verbaut. Im Lieferumfang dieser Boards befindet sich meist bereits eine richtige Antenne und sie besitzen ebenso ein fest verbautes OLED-Display, das du ganz leicht mit den üblichen Bibliotheken ansteuern kannst.

Hinweise:

LoRa nutzt regional unterschiedliche Frequenzbereiche – in Europa ist das zum Beispiel die Frequenz 868 Mhz. Achte bei deinem Kauf darauf, dass du ein Board bzw. Modul bestellst, dass in diesem Bereich sendet.

Betreibe dein LoRa-Board immer nur mit angeschlossener Antenne! Ansonsten kann es Schaden nehmen.

Die Arduino IDE vorbereiten

Du kannst dein LoRa Board wie gewohnt über die Arduino IDE programmieren. Hierfür musst du jedoch zunächst die Boards des Typ ESP32 verfügbar machen. Wie das geht, erfährst du in diesem [Tutorial auf heise.de](#)

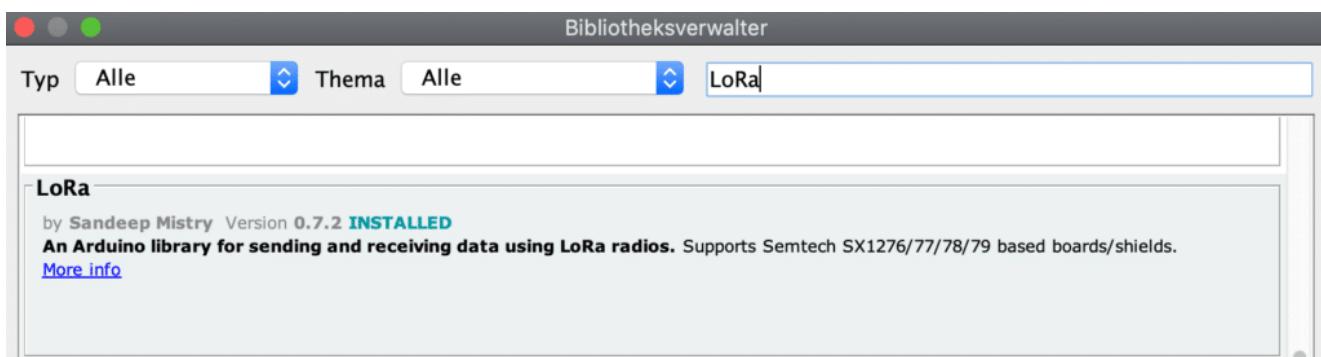
Die benötigten Bibliotheken

installieren

Für das TTGO LoRa32 benötigst du neben den vorinstallierten Standard-Bibliotheken SPI und Wire außerdem einige weitere Bibliotheken – für LoRa und für das OLED-Display. Öffne also deine Bibliotheksmanager, suche und installiere dir zunächst diese Bibliotheken fürs Display:

- Adafruit SSD1306
- Adafruit GFX Library

Für die Kommunikation über LoRa benötigst du die gleichnamige Bibliothek von Sandeep Mistry:



Die passende LoRa Bibliothek

Sobald du diese Bibliotheken installiert hast, starte deine Arduino IDE neu. Anschließend geht es mit dem Code weiter.

Der Sketch für den Sender

Beginnen wir zunächst mit dem Board, mit dem du Daten senden wirst. Kopiere dir folgenden Code und lade ihn auf dein Sender-Board hoch:

```
*****  
 LoRa Sender  
 pollux labs  
 https://polluxlabs.net  
*****/
```

```
//Bibliotheken für LoRa
```

```
#include <SPI.h>
#include <LoRa.h>

//Bibliotheken für das OLED-Display
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//Pins, an denen das LoRa-Modul am ESP32 angeschlossen ist
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DI00 26

//Frequenzbereich für Europa
#define BAND 866E6

//Pins, an denen das OLED-Display angeschlossen ist
#define OLED_SDA 4
#define OLED_SCL 15
#define OLED_RST 16

//Displaygröße
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

//Counter für die gesendeten Datenpakete
int counter = 0;

//OLED-Display instanzieren
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RST);

void setup() {

    //Reset des OLED-Displays
    pinMode(OLED_RST, OUTPUT);
    digitalWrite(OLED_RST, LOW);
    delay(20);
```

```

digitalWrite(OLED_RST, HIGH);

//Display initialisieren
Wire.begin(OLED_SDA, OLED_SCL);
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3c, false, false))
{ // I2C-Adresse: 0x3c
    Serial.println(F("Display nicht gefunden."));
    for(;); //Programm nicht weiter ausführen, wenn das
Display nicht gefunden wird
}
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(1);
display.setCursor(0,0);
display.print("LoRa SENDER ");
display.display();
//Seriellen Monitor starten
Serial.begin(115200);
//Pins für LoRa definieren und Kommunikation starten
SPI.begin(SCK, MISO, MOSI, SS);
LoRa.setPins(SS, RST, DI00);
if (!LoRa.begin(BAND)) {
    Serial.println("LoRa konnte nicht gestartet werden!");
    while (1);
}
Serial.println("LoRa OK!");
display.setCursor(0,10);
display.print("LoRa OK!");
display.display();
delay(2000);
}

void loop() {
    Serial.print("Sende Paket: ");
    Serial.println(counter);

    //Datenpaket senden
    LoRa.beginPacket();
    LoRa.print("Paket Nr. ");
    LoRa.print(counter);
    LoRa.endPacket();
}

```

```

display.clearDisplay();
display.setCursor(0,0);
display.println("LoRa SENDER");
display.setCursor(0,20);
display.setTextSize(1);
display.print("Gesendete Pakete: ");
display.setCursor(105,20);
display.print(counter);
display.display();

counter++;
delay(10000);
}

```

Wenn die Antenne richtig angeschlossen ist, sollte dein Board LoRa erfolgreich initialisieren und umgehend mit dem Versenden der Datenpakete beginnen. Zeit für einen Empfänger!

Der Sketch für den Empfänger

Der Code für den Empfänger unterscheidet sich nicht allzu sehr von jenem des Senders. Du verwendest lediglich ein paar andere Funktionen. Kopiere folgenden Sketch und lade ihn auf dein zweites Board hoch, das die Daten empfangen soll.

```

*****
LoRa Receiver
pollux labs
https://polluxlabs.net
*****/

//Bibliotheken für LoRa
#include <SPI.h>
#include <LoRa.h>

//Bibliotheken für das OLED-Display
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

```

```
//Pins, an denen das LoRa-Modul am ESP32 angeschlossen ist
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DI00 26

//Frequenzbereich für Europa
#define BAND 866E6

//Pins, an denen das OLED-Display angeschlossen ist
#define OLED_SDA 4
#define OLED_SCL 15
#define OLED_RST 16

//Displaygröße
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

//Counter für die gesendeten Datenpakete
int counter = 0;

//OLED-Display instanzieren
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RST);

//Variable für die empfangenen Daten
String LoRaData;

void setup() {

    //Reset des OLED-Displays
    pinMode(OLED_RST, OUTPUT);
    digitalWrite(OLED_RST, LOW);
    delay(20);
    digitalWrite(OLED_RST, HIGH);

    //Display initialisieren
    Wire.begin(OLED_SDA, OLED_SCL);
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3c, false,
```

```
    false)) { // I2C-Adresse: 0x3C
        Serial.println(F("Display nicht gefunden."));
        for (;;) { //Programm nicht weiter ausführen, wenn das
Display nicht gefunden wird
    }

    display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print("LoRa RECEIVER ");
    display.display();

    //Seriellen Monitor starten
    Serial.begin(115200);

    //Pins für LoRa definieren und Kommunikation starten
    SPI.begin(SCK, MISO, MOSI, SS);
    LoRa.setPins(SS, RST, DI00);

    if (!LoRa.begin(BAND)) {
        Serial.println("LoRa konnte nicht gestartet werden!");
        while (1);
    }
    Serial.println("LoRa OK!");
    display.setCursor(0, 10);
    display.println("LoRa OK!");
    display.display();
}

void loop() {

    //Datenpaket parsen
    int packetSize = LoRa.parsePacket();
    if (packetSize) {
        Serial.print("Paket empfangen ");

        //Paket auslesen
        while (LoRa.available()) {
            LoRaData = LoRa.readString();
            Serial.print(LoRaData);
        }
    }
}
```

```

}

//RSSI ausgeben
int rssi = LoRa.packetRssi();
Serial.print(" mit RSSI ");
Serial.println(rssi);

// Infos auf Display ausgeben
display.clearDisplay();
display.setCursor(0, 0);
display.print("LoRa RECEIVER");
display.setCursor(0, 20);
display.setTextSize(1);
display.print(LoRaData);
display.setCursor(0, 30);
display.print("empfangen");
display.setCursor(0, 45);
display.print("RSSI:");
display.setCursor(30, 45);
display.print(rssi);
display.display();
}

}

```

Wenn nun beide LoRa Boards ihren jeweiligen Sketch erhalten haben, kannst du jetzt testen. Versorge beide Boards mit Strom und schaue, ob dein Empfänger alle 10 Sekunden ein Datenpaket vom Sender erhält.

Neben der Paketnummer gibt dein Empfänger auch die Signalstärke RSSI aus. Das ist ein relativer Wert, der dir helfen kann, die maximale Entfernung deiner Boards zu bestimmen.

Falls dein Aufbau nicht funktioniert, prüfe, ob die Antennen richtig angeschlossen sind. **Noch einmal der Hinweis: Betreibe die Boards niemals ohne korrekt angeschlossene Antenne, da sie ansonsten Schaden nehmen können.**

Wie geht es weiter?

LoRa ist dann sinnvoll, wenn du über eine einigermaßen weite Strecke Daten übertragen möchtest und hierbei nicht auf WLAN oder Bluetooth zurückgreifen kannst. Das können z.B. [Sensordaten](#) aus einem Gewächshaus sein, das bei dir im Garten steht.

Aber es gibt auch Projekte ganz anderen Kalibers, wie z.B. dieser [LoRa-Kommunikator](#), der auch nach dem Weltuntergang (bzw. wenn kein Internet mehr verfügbar ist) funktioniert. ☺

**Amazon Affiliate Link – wenn du dort bestellst, erhalten wir eine kleine Provision.*