# Einführung

Mit dem WLAN steht eine häufig eingesetzte Funkübertragungstechnik zur Verfügung. In dem Skript ***WLAN-Experimente mit dem ESP32-Board TTGO T-Display*** [WLN] habe ich dargestellt, wie man mit dieser Technologie nicht nur einfache Daten wie z. B. einen Temperaturwert oder eine Zeitangabe, sondern auch umfangreiche Dateien wie z. B. ganze Webseiten übertragen kann. Ähnliches gilt auch für **Bluetooth** [BLE]..

Neben dem WLAN und Bluetooth ist uns noch das ***HANDY-Netz (GSM/LTE/5G)*** geläufig. Auch dieses kann nicht nur kleine Datenmengen (wie z. B. SMS), sondern auch große Dateien wie Videos übertragen.

### Wozu jetzt also LoRaWAN als weitere Funkübertragungstechnik?

Die oben genannten Techniken haben unterschiedliche Nachteile: WLAN und Bluetooth haben nur eine geringe Reichweite. Im Freien kommen wir beim WLAN auf bis zu 100 m; in Gebäuden werden meist nur ein bis zwei Mauern durchdrungen. Beim Handy-Netz ist die Reichweite deutlich höher; hier werden (im Freien) Strecken von mehreren Kilometern überwunden. Allerdings ist hier der Energieaufwand deutlich höher als beim WLAN. Außerdem gibt es (neben der Anschaffung) Kosten für die Nutzung des Netzes. LoRaWAN hat nun folgende Vorteile:

- **Geringer Energieeinsatz**: Im Leerlauf brauchen diese Geräte praktisch keine Energie, wenn man die Geräte so programmiert, dass sie nur wenn erforderlich aktiv werden. Dies kann man z. B. durch die Festlegung bestimmter Zeitschlitze erreichen.

- **Hohe Reichweite**: Ein Grund für die höhere Reichweite ist die niedrigere Frequenz: Grob gesagt ist die Reichweite um so größer, je kleiner die Frequenz ist. Hinzu kommt aber auch eine spezielle Modulationstechnik. Aufgrund dieser beiden Faktoren können nun deutlich größere Reichweiten erzielt werden als bei WLAN oder Bluetooth: In Gebäuden können häufig mehrere Mauern und Decken überwunden werden. Im Freien sind Reichweiten von mehreren Kilometern möglich. Die Abkürzung LoRa beschreibt genau diese Eigenschaft: *Lo*ng *Ra*nge (große Reichweite).

- **Keine zusätzlichen Kosten**: Die Benutzung des 868MHz-Bandes (SRD-Band - **S**hort **R**ange **D**evice) ist frei; insbesondere ist keine Anmeldung erforderlich. Jedoch gibt es einige Vorgaben: So soll man z. B. (bei der in diesem Skript benutzten Frequenz von 868,5 MHz) nicht mehr als 1% der Zeit senden, vgl. Kapitel 6.

All diese Vorteile werden allerdings erkauft durch eine geringe Datenrate. LoRaWAN ist prädestiniert für das **IoT** (**I**nternet **o**f **T**hings); hier werden meist nur kleine Datensätze übertragen, bei denen die Datenübertragungsrate nicht im Vordergrund steht. Damit können z. B. Füllstände, Temperaturwerte oder auch der Zählerstand von "Stromzählern" übermittelt werden.

Häufig werden die Abkürzungen **LoRa** und **LoRaWAN** verwechselt oder auch synonym verwendet. Dies ist eigentlich nicht korrekt: LoRa beschreibt die von SEMTECH entwickelte *physische Schicht*, d.h. die Modulationstechnik, mit der Kommunikationsverbindungen über Langstrecken erreicht werden. (Ihr wesentliches Merkmal ist die Benutzung von so genannten **Chirps**; hierbei handelt es sich um Signale mit einer ansteigenden Frequenz. Solche Chirps tauchen übrigens in akustischer Form auch in der Natur auf; sie werden z. B. von Delfinen und Fledermäusen benutzt.) LoRaWAN hingegen definiert das Kommunikationsprotokoll und die Systemarchitektur für das Netzwerk, welche die LoRa-Modulation in der physischen Schicht verwendet.

In diesem Skript werde ich im Wesentlichen nur auf **Peer-to-Peer-Übertragungen** (d. h. die Übertragung von einem LoRa-Modul zu einem anderen) eingehen. Es gibt auch die Möglichkeit, Daten per LoRaWAN an **LoRa-Gateways** zu übermitteln (vgl. Abb. 1). Diese senden dann die empfangenen Daten mittels TCP/IP an Server weiter, wo sie dann verarbeitet werden oder anderen Nutzern zur Verfügung gestellt werden.

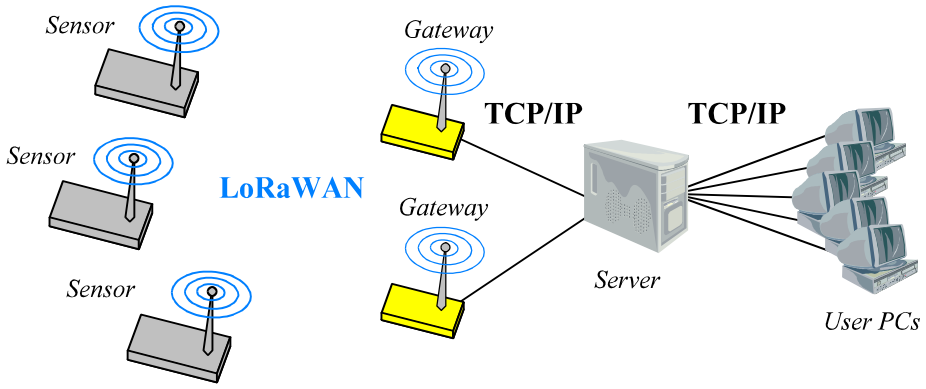


**Abb. 1**

## Das LoRa-Modul REYAX RYLR998

Das LoRa-Modul RYLR998 der Firma REYAX (Abb. 2) kann als Sender und auch als Empfänger benutzt werden. Gesteuert wird es durch so genannte **AT-Commands**. Solche AT-Commands sind schon vor Jahrzehnten - insbesondere bei Telefonen und Modems - zum Einsatz gekommen. Auch das weit verbreitete GSM-Modul SIM800 bedient sich solcher AT-Commands (s. [GSM]).


**Abb. 2**

Mit diesen AT-Commands können sowohl Einstellungen am LoRa-Modul (z. B. das benutzte Frequenz-Band) vorgenommen als auch Daten übertragen werden. Die AT-Commands werden über eine UART-Schnittstelle an das LoRa-Modul übertragen. Über diese Schnittstelle kann das Modul auch empfangene Botschaften ausgeben.

Das LoRa-Modul RYLR998 ist gerade für einen Einstieg in unser Thema gut geeignet: Es müssen keine zusätzlichen Installationen vorgenommen werden, die AT-Commands sind einfach strukturiert und ihre Anzahl ist gut überschaubar. Für unsere Experimente werden wir mit etwa einem Dutzend davon auskommen.

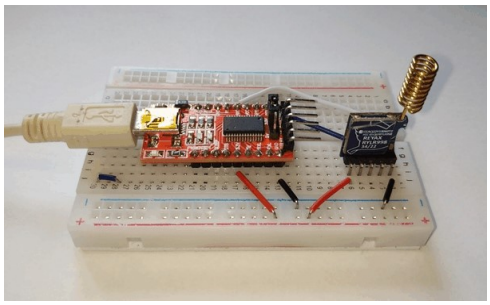

Abb. 3

An dieser Stelle soll aber nicht verschwiegen werden, dass diese Vorteile mit gewissen Einschränkungen verbunden sind: So kann man z. B. nicht alle LoRa-Parameter unabhängig von einander einstellen.

Um mit dem LoRa-Modul zu kommunizieren, wollen wir in diesem Skript zwei Wege benutzen.

**1. Weg:** Wir benutzen einen USB-UART-Konverter wie z. B. den FTDI232RL (s. Abb. 3). Durch diesen Konverter wird der LoRa-Baustein mit einem PC verbunden. Über diese Verbindung können wir nun mit Hilfe eines Terminal-Programms (z. B. *Hterm*) AT-Commands senden und auch Antworten vom LoRa-Baustein erhalten.

**2. Weg:** Wir können das LoRa-Modul direkt an einen Mikrocontroller (z. B. unseren TTGO T-Display) anschließen. Der Mikrocontroller kann dann über eine seiner UARTs durch entsprechende Programmierung AT-Commands an das LoRa-Modul senden und damit Daten senden und empfangen.

Bei diesem 2. Weg werden wir **Micropython** als Programmiesprache verwenden. Hierbei handelt es sich um eine Variante der Programmiersprache **Python**, die speziell für Mikrocontroller konzipiert wurde: Micropython benutzt dieselbe Syntax wie Python, aber im Kern besitzt sie nur eine Teilmenge der Python-Befehle; hinzu kommt allerdings noch eine Reihe von Mikrocontroller-spezifischen Befehlen.

Sie, lieber Leser, sollten einige Vorkenntnisse hinsichtlich der Programmierung mit Mikropython besitzen. In meinen Skripten "WLAN-Experimente mit dem ESP32-Board TTGO T-Display" [WLN]und "Bluetooth Low Energy: Eine praktische Einführung mit Micropython und dem ESP32-Board TTGO T-Display" [BLE] schildere ich diese genauer und biete dort auch neben einer Installationsanweisung für die Entwicklungsumgebung *Thonny* einen kleinen Micropython-Einführungskurs.

Die einzelnen Kapitel dieses LoRa-Skripts bauen aufeinander auf. Es bietet sich - insbesondere für Anfänger - deswegen an, diese Kapitel der Reihe nach durchzugehen. Ganz bewusst habe ich mich entschieden, das Thema nicht an einem einzigen Projekt abzuhandeln. Vielmehr lernen Sie Schritt für Schritt neue Eigenschaften des LoRa-Moduls und zugehörige Anwendungen kennen. Ganz wichtig war es für mich dabei, ab und zu einen *behutsamen Blick hinter die Kulissen von LoRa* zu gewähren: So lernen Sie z. B. wichtige Begriffe wie Modulation, Chirps, Bandbreite oder Spreizfaktor kennen. Damit können Sie dann eine Vorstellung davon erhalten, wie LoRa funktioniert.