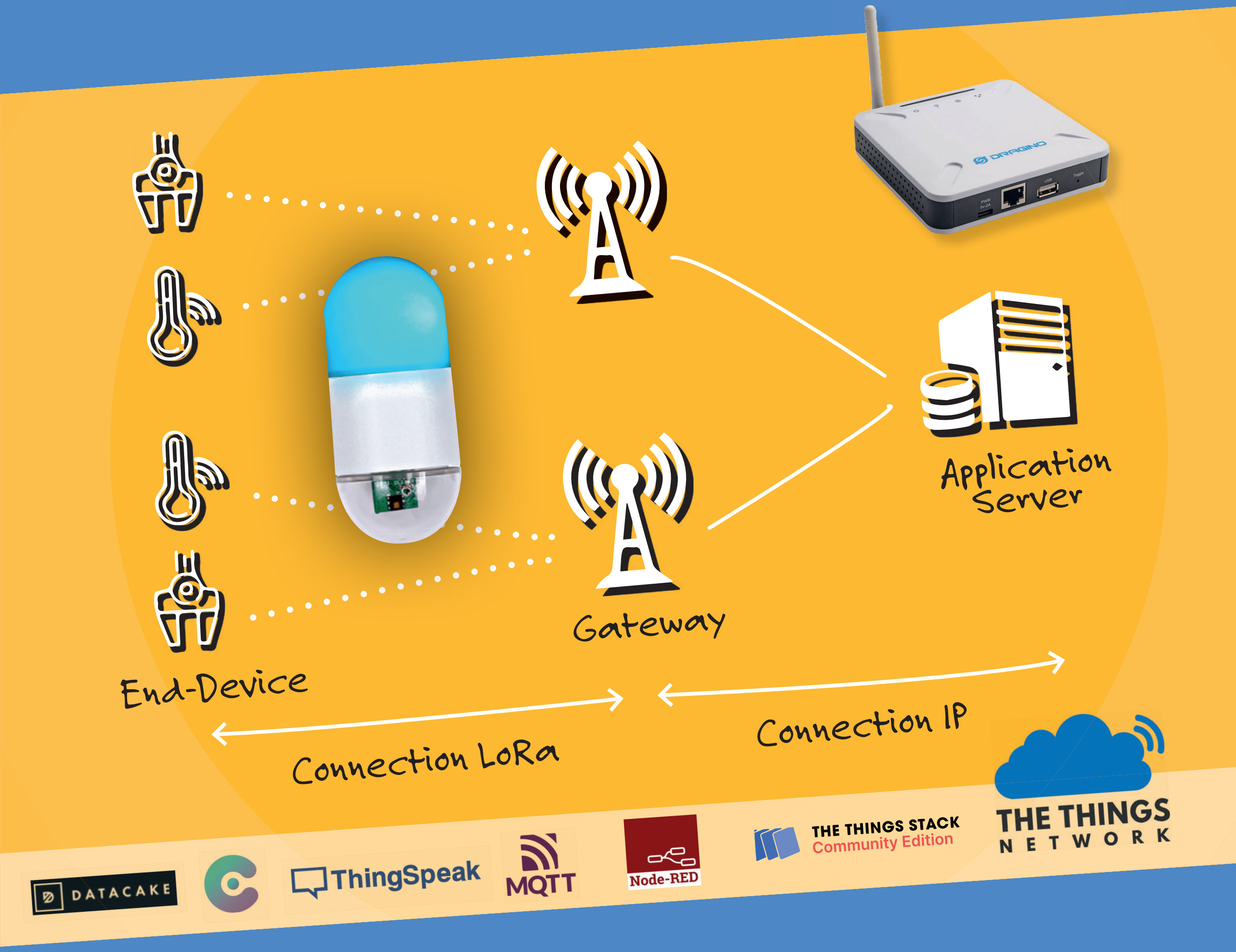
elektor books

Develop and Operate Your

LoRaWAN IoT Nodes

Ready-to-use devices and self-built Arduino nodes in the "The Things Network"



Claus Kühnel

elektor
design > share > earn

Develop and Operate Your LoRaWAN IoT Nodes

Ready-to-use devices and self-built Arduino nodes in the "The Things Network"

Dr. Claus Kühnel

elektor
design > share > sell

This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
PO Box 11, NL-6114-ZG Susteren, The Netherlands
Phone: +31 46 4389444

Declaration

The Author and Publisher have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, and hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident, or any other cause.
All the programs given in the book are Copyright of the Author and Elektor International Media. These programs may only be used for educational purposes. Written permission from the Author or Elektor must be obtained before any of these programs can be used for commercial purposes.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 978-3-89576-494-3 Print
ISBN 978-3-89576-495-0 eBook

© Copyright 2022: Elektor International Media B.V.
Prepress Production: D-Vision, Julian van den Berg
Printed in the Netherlands by Ipskamp Printing, Enschede

Chapter 1 • Preface

The Internet of Things (IoT) is on everyone's lips and promises massive changes in the technological landscape. Knowledge of every new technology is essential to avoid being unprepared. Such knowledge may be challenging to develop in retrospect.

IoT is defined by the networking of things (sensors, embedded systems) and the associated creation of added value, and finally, new business models based on this. The connecting element is recorded and transported data.

With the technologies available, this was in principle already possible, but not at acceptable costs or only with limited properties. If the IoT connection in the often-touted intelligent refrigerator is more expensive than the refrigerator itself, then acceptance of such a product is hard to be expected. The basic properties of an IoT node, as we will henceforth refer to our intelligent refrigerator and similar components that need to be networked, i.e., things in themselves, are

- great coverage
- wireless connection to the network
- battery operation with long battery life
- low price.

IoT nodes incorporate various objects, sensors, measuring devices, monitoring systems, and security systems.

Low Power Wide Area Network (LPWAN) is a generic term for many different communication protocols. As you can see in Figure 1.1, there are others for LPWAN connections worldwide. In addition to LoRa or LoRaWAN, which are in focus here, NB-IoT, Sigfox, LTE-M, Weightless, Symphony Link, and a few others compete.

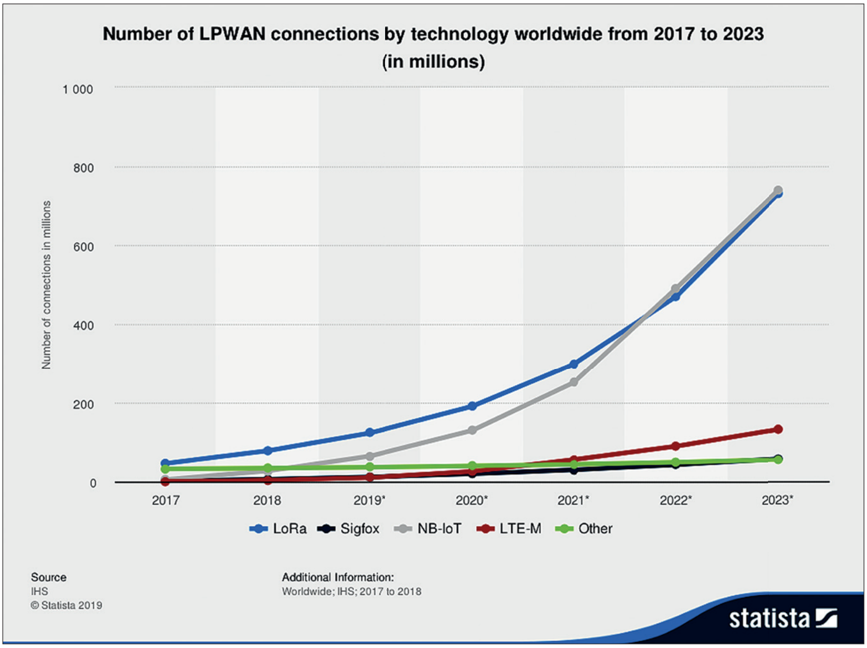


Figure 1.1: Number of LPWAN connections worldwide.

In Figure 1.1, the dominance of LoRaWAN and NB-IoT over the other LPWAN technologies is clear to see.

LoRaWAN is an open LPWAN system architecture developed and standardized by the LoRa Alliance^{TM}, a non-profit association of more than 500 member companies.

Near the end of 2021, the International Telecommunication Union (ITU), i.e., the United Nations specialized agency for information and communication technologies (ICTs), officially approved LoRaWAN as a standard.

The standard is titled Recommendation ITU-T Y.4480 "Low power protocol for wide-area wireless networks" and is under the responsibility of Study Group 20 of the ITU Telecommunication Standardization Sector (ITU-T), ITU's standardization expert group for "Internet of Things and smart cities and communities" (https://www.eenewseurope.com/news/lorawan-recognized-itu-standard).

LoRa is based on the Chirp Spread Spectrum (CSS) modulation technique, enabling long-range low-power communication. I will go into this in more detail later.

NB-IoT, on the other hand, works in the licensed spectrum (i.e., not free of charge) and uses LTE, Frequency Division Multiple Access (FDMA) in the uplink, orthogonal FDMA (OFDMA) in the downlink, and quadrature phase-shift keying (QPSK) modulation.

Both technologies can compete in Quality of Service (QoS). IoT applications requiring more communication are better served by NB-IoT, as this has no work cycle restrictions in the licensed spectrum. However, NB-IoT points higher total cost of ownership (TCO) than LoRaWAN.

Don't be afraid of the terms used here. You can find explanations in the glossary at the end of the book. So much for the distinction between LoRaWAN and NB-IoT!

If you want to deepen the comparison between the two LPWAN technologies, I will refer you to the white paper entitled "LoRaWAN and NB-IoT: Competitors or Complementary?" [1].

LoRaWAN is a widespread and, under certain conditions, accessible and free option for everyone wanting to transmit data from an IoT node. It is thereforethe focus of our considerations.

Unlike other protocols, the LoRa standard is open source and not proprietary – one reason for the rapid growth of LoRaWAN networks across entire countries, starting in metropolitan areas.

This book uses commercially available LoRaWAN sensor nodes developed using simple means and at low costs. These LoRaWAN nodes send data to a LoRaWAN server. They can then be called up and integrated into any application.

You are well prepared for this promising task if you have gained your first experience with an Arduino so far. The Arduino IDE covers all microcontrollers of different architecture considered here so you do not have to work with different development environments.

On the base of an ESP8266 microcontroller from Espressif, I had already shown that we could set up a WiFi-compatible IoT node for a very low cost (US $ 15) [2].

At the time, I worked with an ESP8266 Node MCU obtained directly from China. There isnow enough availability here, but not at comparable prices.

My studies have shown that WiFi is only suitable to a limited extent for a battery-operated IoT node due to its short range and relatively high power requirement [3] [4].

The LoRaWAN nodes presented here open up entirely new possibilities, whereby the costs are not significantly higher.

The Internet of Things (IoT) requires a global infrastructure that connects IoT nodes for information exchange. We already encounter these developments daily.

Think of your fitness tracker, which reminds you to complete your exercises, or your bathroom scale, which warns you of weight gain. Your wristwatch mercilessly reveals a possible sleep deficit, and your mobile phone may be equipped with a Corona warning app to help limit Covid-19 infection chains.

With these few arbitrarily selected examples, the complexity of the applications becomes evident, which lie above the technical levels of connection establishment and data exchange. This book on LoRa & LoRaWAN is about the latter.

LoRaWAN has developed excellently as a communication solution in the IoT. The Things Network (TTN) has significantly contributed to this, both for makers and commercial use.

TTN is a global collaborative IoT ecosystem that creates networks, devices, and solutions with LoRaWAN. With The Things Stack Community Edition (TTS (CE)), TTN now operates an open and decentralized LoRaWAN network. This network is an excellent opportunity to start testing devices, applications, and integrations and becomes familiar with LoRaWAN.

With the knowledge conveyed here, you will register end devices and gateways in the TTS (CE) or migrate your end devices already in the TTN V2 to TTS (CE).

For better readability, I observe the following conventions for the textual presentation:

- Commands and outputs to the console are in `Courier New`.
- Entries via the console are in **`Courier New`** in **bold**.
- Labels, programs, and file names appear in *italics*.

I often change long web addresses (URLs) using the URL shortener Bitly (https://bitly.com).

This edition is a translation of my German book "LoRaWAN-Knoten im IoT" published by Elektor in November 2021 (ISBN 978-3-89576-467-7, LoRaWAN-Knoten im IoT - Elektor). This is why some German sources appear in the references.

I have checked all existing links at the end of 2021. As the Internet is constantly changing, I cannot guarantee these links will work or lead to the same content at the time of inclusion. Please inform me about broken links!

The program examples presented here can be downloaded from Github at the URL

https://github.com/ckuehnel/LoRaWAN-Node/tree/master/---%20Elektor%20---.

Altendorf, December 2021

Claus Kühnel

Chapter 2 • LoRa

Semtech developed LoRa to fulfill the requirements of the IoT. The following features characterize LoRa:

- LoRa can transmit small amounts of data over several kilometers by radio.
- The considerable transmission distance is achieved with minimal transmission power.
- Due to the low energy requirement, sensor nodes equipped with batteries can function autonomously for several years.
- LoRa enables bi-directional communication with data rates from 0.3 to 50 kbps. Data transfers are possible in an urban environment over 2 to 5 km distances. In a suburban environment, you achieve transmissions of up to 15 km.
- LoRa's modulation type is derived from Chirp Spread Spectrum (CSS) encoding technology. This technology uses all of the allocated bandwidth to send a signal to spread it over a wider band of the spectrum. This means such a system can also operate with a low signal-to-noise ratio (SNR). Therefore, LoRa has a high level of robustness against noise.
- LoRa uses the license-free ISM bands at 433 MHz and 868 MHz (ISM = Industrial Scientific & Medical) in Europe.
- Thanks to integrated encryption technology, the data transmission is securely and optimally protected against unauthorized access.

In the following chapters, I will show you how these properties are achieved and which customization options LoRa offers.

2.1 LoRa Basics

LoRa is a proprietary and patented technology developed by Semtech Corp., which works based on Direct Sequence Spread Spectrum Modulation (DSSS) with Chirp Spread Spectrum (CSS) modulation.

Modulating the data stream (bit sequence) with a code sequence (pre-defined bit pattern) generates a significantly higher bit rate and an output signal with a greatly increased bandwidth. The bits of the code sequence are called chirps to distinguish them from the bits of the digital data stream. Figure 2.1 shows DSSS modulation and the associated spreading of the frequency band to be transmitted.

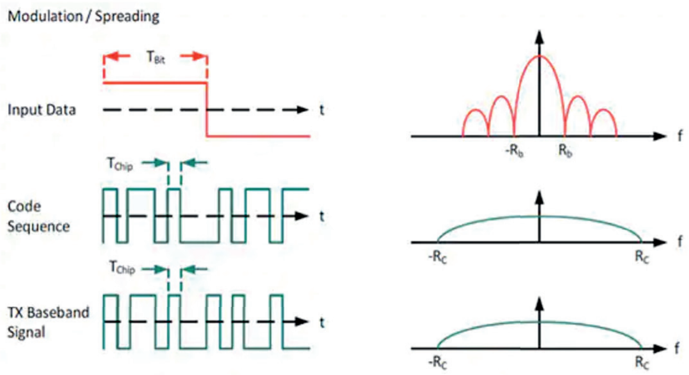


Figure 2.1: Direct Sequence Spread Spectrum Modulation (Image: Semtech).

Figure 2.1 demonstrates that the chirp rate determines bandwidth. The higher the chirp rate (i.e., the smaller the chirp time T_{Chip}), the greater the resulting bandwidth:

$$BW=R_C=chip\,rate\in(\frac{chirps}{s})$$

The disadvantage of such a DSSS system is that it requires a highly accurate reference clock.

Semtech's LoRa Chirp Spread Spectrum (CSS) technology offers a cost-effective and power-saving yet robust DSSS alternative that does not require a highly accurate reference clock. Generating a chirp signal that continuously changes its frequency spreads the signal spectrum. Figure 2.2 shows the chirp signals used.

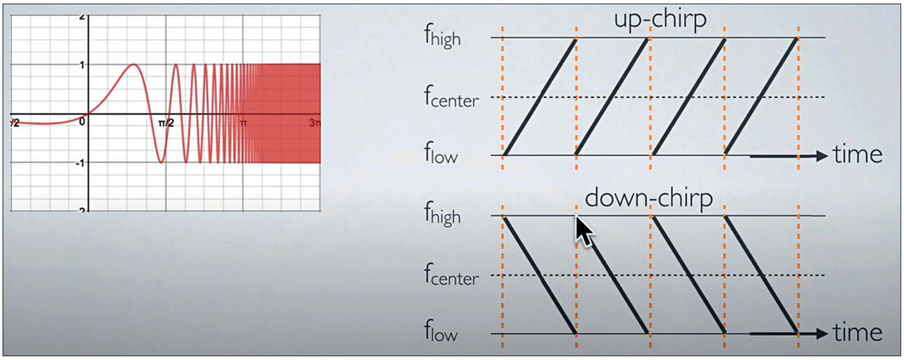


Figure 2.2: Chirp Signal for LoRa Modulation (Image: mobilefish.com).

Figure 2.3 now shows an unmodulated and a modulated signal in comparison.

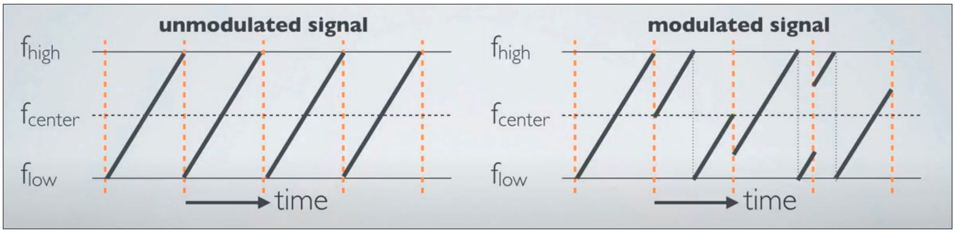


Figure 2.3: LoRa Modulation (Image: mobilefish.com).

With the unmodulated signal, the frequency change progresses steadily from f_{low} to f_{high}. However, in the case of a modulated signal, the frequency hop marks the symbol to be transmitted. A symbol represents one or more data bits.

The Spreading Factor (SF) indicates how many chirps encode a symbol. The following relationships apply to the symbol rate R_S and the bit rate R_b:

$$R_S = \frac{BW}{2^{SF}} \quad \text{and} \quad R_b = SF\,\frac{BW}{2^{FS}}$$

Figure 2.4 shows a complete LoRa message. The LoRa message begins with a preamble (introduction) with eight up chirps, followed by two down chirps for synchronization (marked by the arrow). The modulated data follows this.

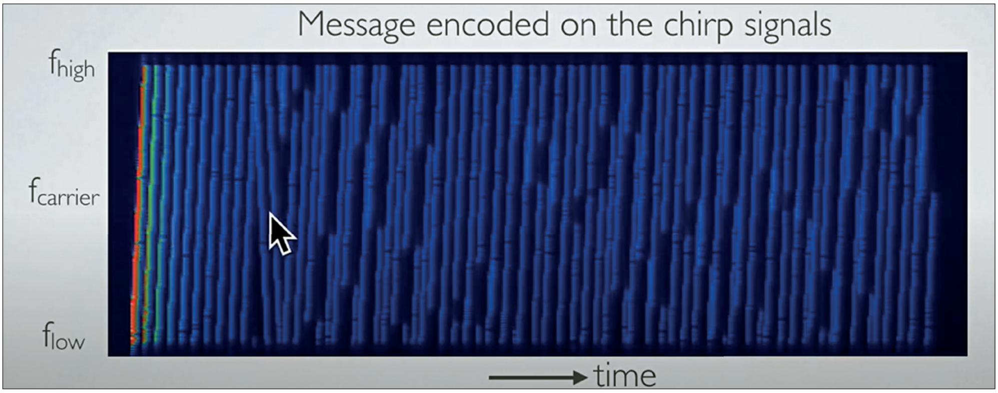


Figure 2.4: LoRa Message (Image: mobilefish.com).

Figure 2.5 shows the format of a complete LoRa message.

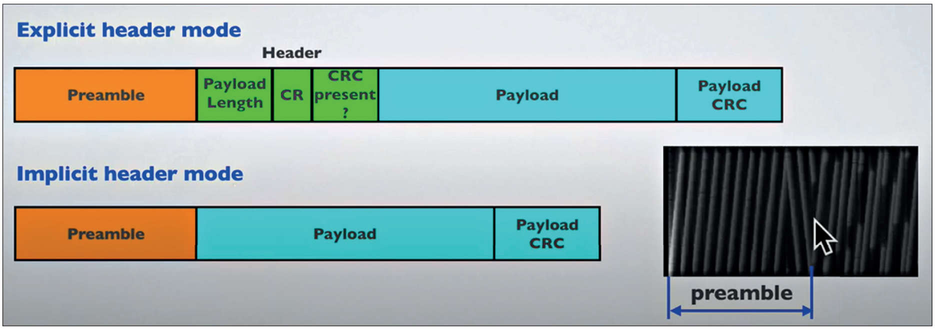


Figure 2.5: LoRa Message Format (Image: mobilefish.com).

It is interesting how long it takes to transmit such a LoRa message. The following relationships calculate the so-called Time-on-Air (ToA) or Air Time:

$$ToA = T_{Packet} = T_{Preamble} + T_{Payload}$$

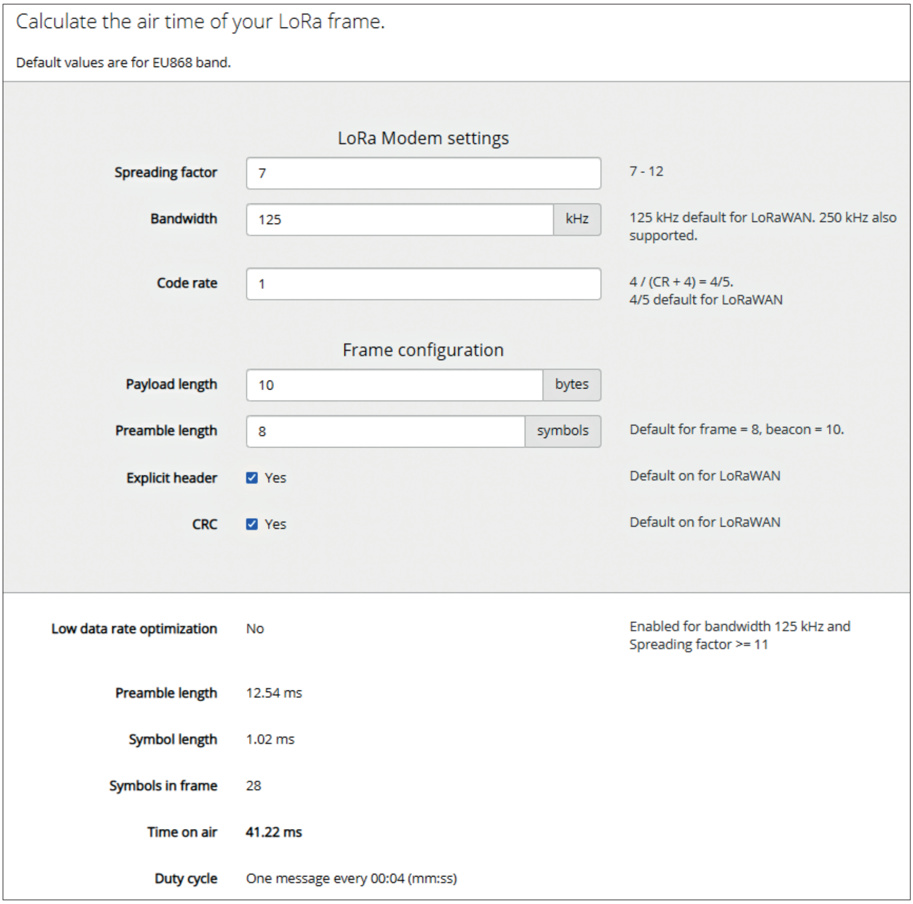
$$T_{Preamble} = 12.25\, T_S (for\ EU\ 868) \quad \text{with} \quad T_S = \frac{2^{FS}}{BW}$$

The calculation of the payload time is somewhat more complex, and the easiest way to do this is with an appropriate tool. You can find an Air Time Calculator at the URL

https://www.loratools.nl/#/airtime.

With this Air Time Calculator, I have calculated the expected air time for SF7 (Figure 2.6) and SF12 (Figure 2.7) with a payload of 10 bytes.

The calculations appling to LoRaWAN will be considered later. Therefore the corresponding defaults have been taken into account.



Calculate the air time of your LoRa frame.

Default values are for EU868 band.

LoRa Modem settings

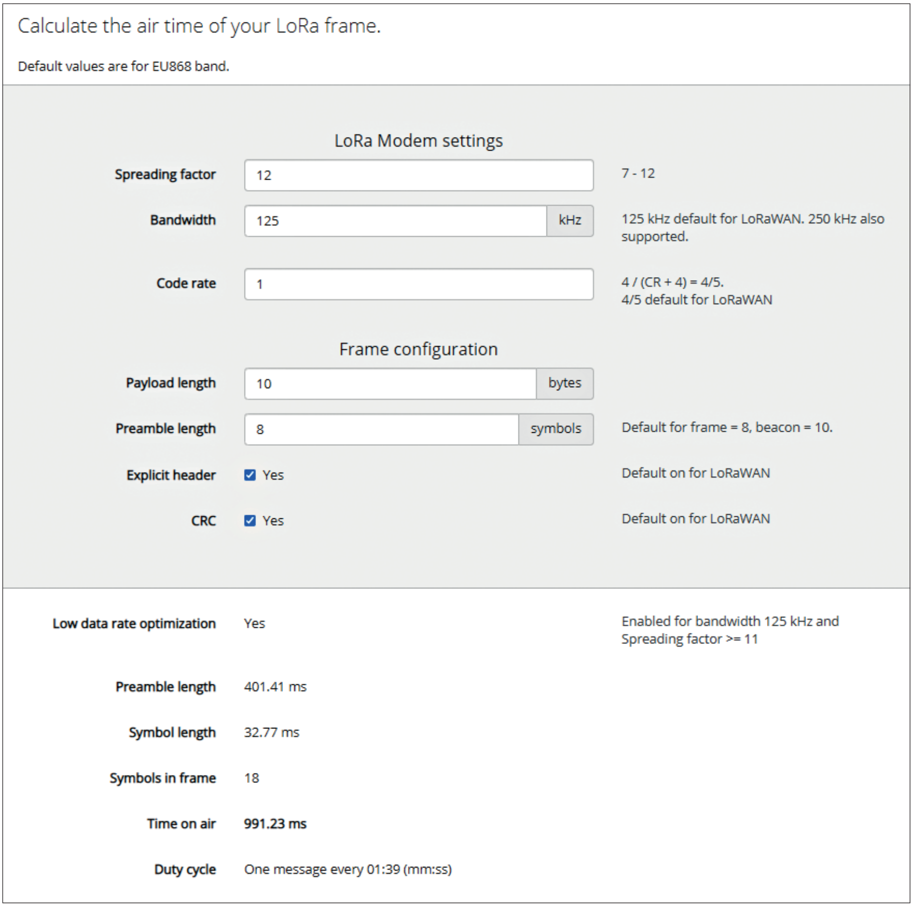
Spreading factor	7	7 - 12
Bandwidth	125 kHz	125 kHz default for LoRaWAN. 250 kHz also supported.
Code rate	1	4 / (CR + 4) = 4/5. 4/5 default for LoRaWAN

Frame configuration

Payload length	10 bytes	
Preamble length	8 symbols	Default for frame = 8, beacon = 10.
Explicit header	☑ Yes	Default on for LoRaWAN
CRC	☑ Yes	Default on for LoRaWAN

Low data rate optimization	No	Enabled for bandwidth 125 kHz and Spreading factor >= 11
Preamble length	12.54 ms	
Symbol length	1.02 ms	
Symbols in frame	28	
Time on air	**41.22 ms**	
Duty cycle	One message every 00:04 (mm:ss)	

Figure 2.6: Air Time Calculation for SF7 and 10 Byte Payload.



Calculate the air time of your LoRa frame.

Default values are for EU868 band.

LoRa Modem settings

Spreading factor	12	7 - 12
Bandwidth	125 kHz	125 kHz default for LoRaWAN. 250 kHz also supported.
Code rate	1	4 / (CR + 4) = 4/5. 4/5 default for LoRaWAN

Frame configuration

Payload length	10 bytes	
Preamble length	8 symbols	Default for frame = 8, beacon = 10.
Explicit header	☑ Yes	Default on for LoRaWAN
CRC	☑ Yes	Default on for LoRaWAN

Low data rate optimization	Yes	Enabled for bandwidth 125 kHz and Spreading factor >= 11
Preamble length	401.41 ms	
Symbol length	32.77 ms	
Symbols in frame	18	
Time on air	**991.23 ms**	
Duty cycle	One message every 01:39 (mm:ss)	

Figure 2.7: Air Time Calculation for SF12 and 10 Byte Payload.

As expected, the times differ significantly. Table 1 shows the compared results.

Time	SF7	SF12
Preamble Length	12.54 ms	401.41 ms
Symbol Time	1.02 ms	32.77 ms
Time on Air	41.22 ms	991.23 ms
Duty Cycle	0:04 min	1:39 min

Table 1: Air Time Calculation Results.

You can see that with a high spreading factor, the transmission of a LoRa message takes a correspondingly long time and will therefore also have an increased energy requirement.

Due to the duty cycle limitation in LoRaWAN, you may send LoRa messages less often. I will go into this later.

Table 2 shows a summary of the LoRa specifications for Europe EU868.

Parameter	Value
Frequency band	867–869 MHz
Channels	10
Channel bandwidth upstream	125/250 kHz
Channel bandwidth downstream	125 kHz
TX power upstream	+14 dBm
TX power downstream	+14 dBm
Spreading factor upstream	7 to 12
Data rate	250 Bit/s–50 Kbit/s
Link budget upstream	155 dB
Link budget downstream	155 dB

Tabelle 2: LoRa Specifications for Europe EU868.

2.2 LoRa Transmission Range

The link budget indicates the quality of radio transmission channel. Using a simple model, you can calculate the link budget. The link budget is composed of the transmitter power Tx, the receiver sensitivity Rx, antenna gains on both sides, cable losses, and the free space path loss (FSPL) (Figure 2.8) [5].

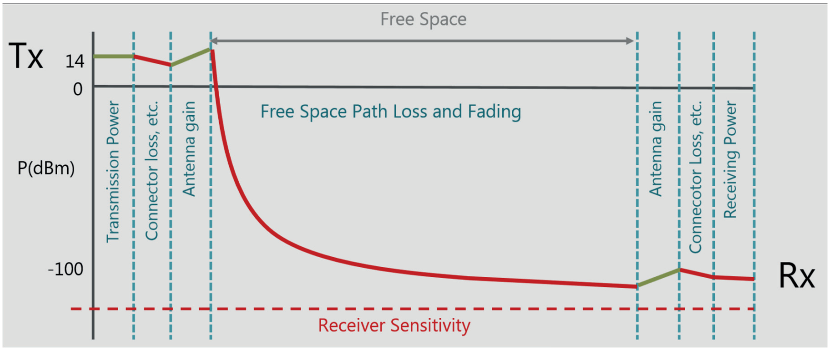


Figure 2.8: LoRa Link Budget (Image: https://www.waziup.io).

The free space path loss (FSPL) states how much energy is lost in free space over a distance between transmitter Tx and receiver Rx. The further the distance between Tx and Rx, the lower the remaining power.

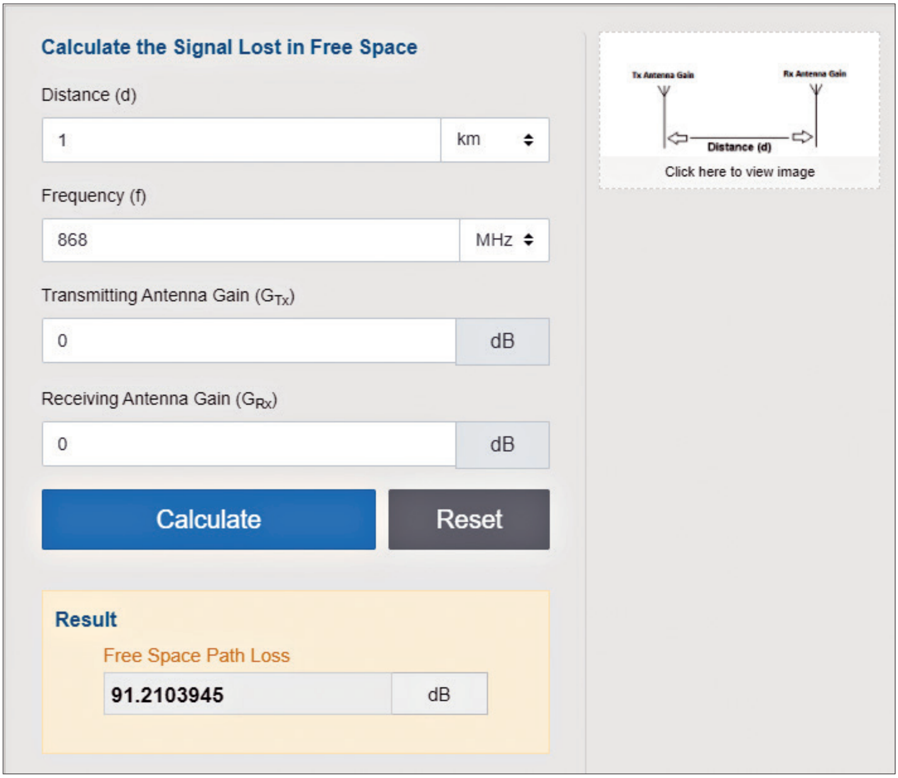
FSPL attempts to predict the strength of an RF signal at a certain distance. It is a theoretical value as many obstacles, reflections, and losses in the real world need to be considered when estimating the signal at a location.

However, the FSPL is a good approximation for estimating signal loss as it propagates through free space. The equation applies:

$$FSPL = 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{c}\right) - G_{Tx} - G_{Rx}$$

With d = distance between transmitting and receiving antenna, f = transmission frequency, G_{Tx} = antenna gain of the transmitting antenna, and G_{Rx} = antenna gain of the receiving antenna.

With the FSPL Calculator (https://www.everythingrf.com/rf-calculators/free-space-path-loss-calculator), the calculation is straightforward. For example, this is how you get a theoretical FSPL of approx. 91 dB at a distance of 1 km between the transmitting and receiving antenna (Figure 2.9).



Calculate the Signal Lost in Free Space

Distance (d)

1 km

Frequency (f)

868 MHz

Transmitting Antenna Gain (G_{Tx})

0 dB

Receiving Antenna Gain (G_{Rx})

0 dB

Result

Free Space Path Loss

91.2103945 dB

Figure 2.9: FSPL Calculator.

On the receiver side (Rx), the receiver's sensitivity is the parameter that influences the link budget. So called Rx sensitivity describes the minimum possible reception power and tolerance for thermal noise. The following equation calculates the RX sensitivity:

$$Sense_{Rx,max} = -174 + 10\log_{10}(BW) + NF + SNR$$

BW = bandwidth in Hz, NF = noise factor in dB, and SNR = signal-to-noise ratio, which indicates how far the signal must be above the noise.

With the values from [5]

Tx power = 14 dBm
BW = 125 kHz = 10 $\log_{10}$ (125000) = 51
NF = 6 dB (the gateways in LoRaWAN networks have lower NF values)
SNR = −20 (for SF = 12)

we get an Rx sensitivity = −174 + 51 + 6 − 20 = −137 dBm.

The following applies to the link budget LB:

$$LB = Sense_{Rx,max} - Sense_{Tx,max}$$

And with the aforementioned values, a link budget of −151 dB follows.

The Adaptive Data Rate (ADR) mechanism, which controls the transmission parameters of an end device, is used to optimize data rates, transmission time, and energy consumption in the network:

- Spreading factor
- Bandwidth
- Transmission performance

ADR can optimize the power consumption of devices while ensuring gateways still receive messages.

Using ADR, the network server indicates to the end device to reduce the transmission power or increase the data rate. End devices close to gateways should use a lower spreading factor and a higher data rate, while devices far away should use a high spreading factor as they require a higher link budget.

ADR should always be activated when an end device has sufficiently stable HF conditions. ADR should be active for static devices.

The end devices decide whether ADR should be used or not, not the application or the network.

The antennas used can bring an additional antenna gain to contribute to the link budget. Rod antennas usually have a radiation pattern reminiscent of a donut (Figure 2.10).

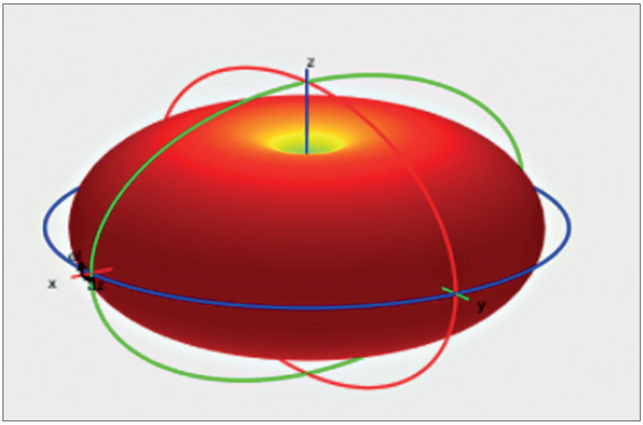


Figure 2.10: Radiation Pattern of a Rod Antenna.

The transmitting antenna radiates the majority of its power to the side, and the radiation is lower upwards and downwards. The sensitivity of the receiving antenna is comparable.

Therefore, transmitting and receiving antennas should be arranged roughly in one plane.

In summary, we can state the following:

- The link budget determines the maximum range of a LoRa data transmission.
- The free space path loss (FSPL) limits the transmission range.
- Obstacles that cause reflections and refractions further shorten the transmission range.
- Antenna gain helps to increase the transmission range.

When aligning the antennas, pay attention to the antenna characteristics to secure the coverage.

The presentation made here is purely theoretical. You can find practical in-depth information in [6]. It shows how the transfer behavior changes in a natural environment and what needs to be considered a priority, if necessary.

2.3 LoRa Communication

Based on basic LoRa nodes, you can build a simple peer-to-peer network in which all nodes have equal rights. Figure 2.11 shows a network where each node receives all communications from other nodes.

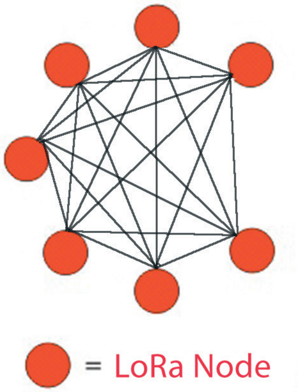


Figure 2.11: Peer-to-Peer Network.

In my Arduino handbook [7], I described a pure LoRa data transmission with the components shown in Figure 2.12.

For the test, I used a sensor node that records temperature and humidity with a DHT11 sensor and sends these values to two receivers via LoRa as an example. The addressing of the recipients is not provided (broadcasting).

In principle, you can use the same hardware for the receiver node as for the sensor node. For practical reasons, I used Wemos TTGO devices as receivers.

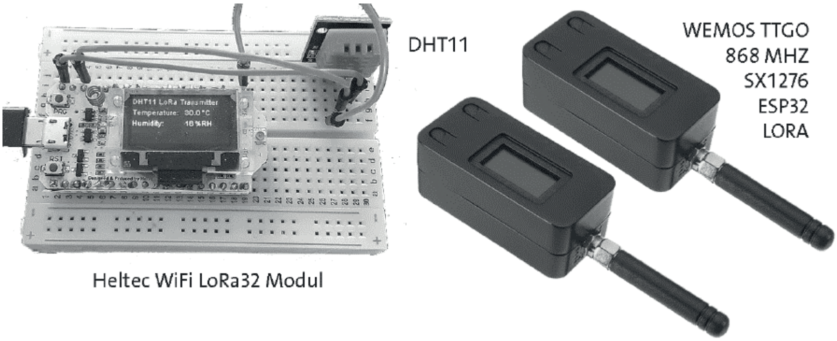


Figure 2.12: LoRa Data Transmission.

After embedding the ESP32 into the Arduino IDE, you must integrate the LoRa Node and the esp8266-oled-ssd1306 libraries.

You can use the programs *Heltec_LoRa32_Transmit.ino* and *Heltec_LoRa32_Receive.ino* from the repository to test the LoRa connection. If you want to do this without sensors and OLEDs, you can also use the *LoRaSender.ino* and *LoRaReceiver.ino* programs.