



Harry Kellner

LoRaWAN für Maker

Sensoren auslesen, Daten teilen,
Projekte umsetzen

- ▶ Gateways, Netzwerkknoten und Sensoren
- ▶ IoT-Projekte mit ELV-Hardware und The Things Stack
- ▶ Mit vielen Beispielen und Heimautomationsideen

Kapitel 1

Einführung – Wer ist LoRa?

Dieses Kapitel gibt Ihnen einen Überblick über das Thema LoRa-WAN. Er stellt die wichtigsten Begriffe vor, die Sie kennen sollten, um sich selbst zu informieren.

Neben den allgemein bekannten drahtlosen Funkstandards *Wi-Fi* und *Bluetooth* verbreitet sich sehr rasant ein weiterer: *LoRaWAN*. Die Buchstaben *LoRa* stehen für *Long Range* und beschreiben bereits den ersten großen Vorteil dieses Funkstandards. Die Module von LoRaWAN haben Reichweiten bis zu 40 km und bieten faszinierende neue Möglichkeiten für die Übermittlung von Sensordaten. Daneben zeichnen sich diese Bausteine auch durch einen sehr stromsparenden Betrieb aus: Aufgrund eines geringen Datendurchsatzes und einer optimierten Funktechnik sind Batterielaufzeiten von zwei bis fünfzehn Jahren möglich. LoRa-Bausteine sind damit die ideale Grundlage für alle Maker und für das *Internet der Dinge (IoT, engl. Internet of Things)*.

1.1 Ein Protokoll erobert die Welt

Die Begriffe rund um LoRaWAN werden fälschlicherweise oft gleichbedeutend verwendet. Lassen Sie uns zu Beginn Folgendes klären:

- Wenn von LoRa die Rede ist, wird meist nicht der physikalische Layer mit dem Namen *LoRa* gemeint, sondern das *LoRaWAN-Protokoll*. *LoRa* ist ein Kunstwort aus den beiden englischen Wörtern *Long* und *Range* und ist ein patentiertes, leitungsloses Übertragungsverfahren auf der Bitübertragungsschicht (engl. *Physical Layer*).



Abbildung 1.1: Das LoRa-Logo

- Das Übertragungsverfahren basiert also auf dem *LoRa-Protokoll* und wurde von der Firma *Semtech* (<https://www.semtech.com>) als Marke registriert. Es steht symbolisch für »Große Reichweite« und kann heute dank Open-Source-Lizenzen von jedem Unternehmen genutzt werden.



Abbildung 1.2: Das Semtech-Logo

- LoRaWAN hingegen steht für *Long Range Wide Area Network* und ist eine weit verbreitete LPWA-Technologie. LPWA (engl. *Low Power Wide Area Network*) wiederum ist ein Kürzel, das eine Klasse von Netzwerkprotokollen beschreibt. Das Protokoll ist auf hohe Reichweite und geringen Energieverbrauch ausgelegt – ideal für die Anbindung von batteriebetriebenen Sensoren an einen Server.



Abbildung 1.3: Das LoRaWAN-Logo

- Die LoRaWAN-Spezifikationen werden von der LoRa-Alliance festgelegt. Sie sind frei verfügbar, und die Software-Grundmodule sind als Open-Source-Software unter <https://lora-alliance.org/> erhältlich.

Die LoRa-Alliance ist ein gemeinnütziger Verein, der sich zum Ziel gesetzt hat, durch die Entwicklung und die Förderung des LoRaWAN-Standards seine Bereitstellung in großem Maßstab zu ermöglichen. Sie standardisiert und akkreditiert Zertifizierungsschemata, um das Funktionieren und die Skalierung von LoRaWAN-Netzwerken sicherzustellen. Die Spe-

zifikationen sind auf große Reichweiten und Energieeffizienz ausgelegt, daher spricht die LoRa-Alliance auch von *Low Power Wide Area Networks*.



Abbildung 1.4: Das Logo der LoRa-Alliance

LoRaWAN basiert also auf dem LoRa-Protokoll und wurde für den Einsatz in Netzwerken konzipiert. Es handelt sich um ein Protokoll der MAC-Schicht (engl. *Medium Access Control*), dem sogenannten *Physical Layer* des OSI-Schichten-Modells (OSI, engl. *Open Systems Interconnection*).

Das OSI-Modell ist ein Referenzmodell für Netzwerkprotokolle, die in diesem Modell als eine Architektur aus aufeinander aufbauenden Schichten dargestellt werden. Der Sinn dieses Modells ist es, die Kommunikation über unterschiedlichste technische Systeme hinweg zu beschreiben. Es definiert sieben übereinanderliegende Schichten (engl. *Layers*) mit jeweils eng begrenzten Aufgaben. Die unterste Ebene (Schicht 1), ist der *Physical Layer*: Hier wird die digitale Bitübertragung auf einer leitungsgebundenen oder leitungslosen Übertragungsstrecke abgewickelt. Die oberste Ebene (Schicht 7) ist der *Application Layer*: Auf ihm werden die Funktionen für die Anwendungen zur Verfügung gestellt. Auf dieser Ebene findet auch die Dateneingabe und die Datenausgabe statt. Abbildung 1.5 gibt Ihnen einen Überblick:

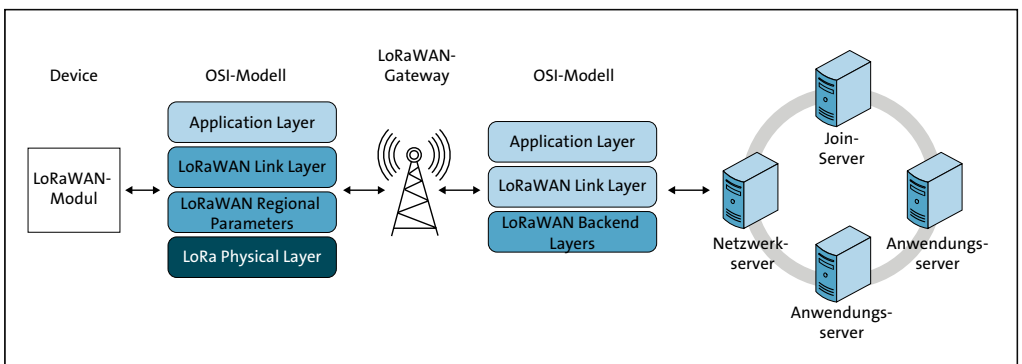


Abbildung 1.5: LoRaWAN im vereinfachten OSI-Schichten-Modell

1.2 Ein Exkurs in die Technik – keine Angst vor Fachbegriffen

Endgeräte und Gateways im LoRaWAN nutzen ein proprietäres und patentiertes Übertragungsverfahren, das auf einer *Chirp-Spread-Spectrum*-Modulationstechnik mit der Bezeichnung *LoRa* basiert. Das LoRa-Protokoll ist asymmetrisch und auf die energieeffiziente Uplink-Kommunikation für Reichweiten über 10 km ausgerichtet.

Hinweis: Uplink und Downlink

Als *Uplink* bzw. als *Upload* wird das Senden vom Endgerät an das Netz bezeichnet. Die Datenübertragungsrate reicht von 292 Bit/s bis 50 kBit/s. Verschiedene Betriebsabstufungen bis hin zu kontinuierlicher Downlink-Kommunikation sind möglich. Letztere geht allerdings zulasten der Energieeffizienz.

Als *Downlink* wird das Senden von Daten an das Endgerät bezeichnet. Dies geschieht meist, um Konfigurationsparameter einzustellen.

LoRa nutzt regional unterschiedliche Frequenzbereiche im ISM- und im SRD-Band. In Europa ist es das Frequenzband von 433,05 bis 434,79 MHz (*ISM-Band Region 1*) und von 863 bis 870 MHz (*SRD-Band Europa*). In Nordamerika ist das Frequenzband von 902 bis 928 MHz (*ISM-Band Region 2*) dafür freigegeben. Die Frequenzspreizung ermöglicht eine hohe Effizienz beim Datentransfer und beim Energieverbrauch. Die Datentransferrate zum Endgerät kann durch den LoRaWAN-Netzserver je nach Bedarf angepasst werden. Dies wird als *Adaptive Data Rate (ADR)* bezeichnet.

Die Reichweiten erstrecken sich von 2 km in Stadtgebieten bis zu 40 km in ländlichen Gebieten. Ein weiterer Vorteil ist die Gebäude-Durchdringung, sodass auch Kellerräume erreicht werden können. Der Strombedarf in Endgeräten beträgt rund 10 mA im Betrieb und nur 100 nA im Ruhemodus. Wenn wir die Selbstentladung der Batterie vernachlässigen, ergibt sich eine Batterielebensdauer von bis zu fünfzehn Jahren.

Die Kommunikation zwischen einem Endgerät und dem Gateway erfolgt auf verschiedenen Frequenzkanälen mit Datenraten von 0,3 bis 50 kBit/s. Die öffentlichen Frequenzen dürfen nicht überlastet werden, deshalb sollen LoRa-Knoten (engl. *Nodes*) möglichst nicht mehr als eine Nachricht pro ca. 5 Minuten senden. Dies schränkt den Einsatz als Realtime-Anwendungen natürlich ein.

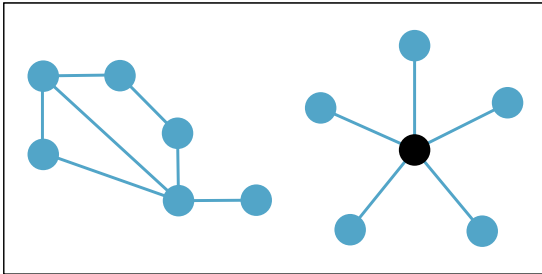


Abbildung 1.6: Maschennetz vs. Sternnetz

Im Gegensatz zum maschenförmigen Netz bei Wi-Fi und Bluetooth ist die LoRaWAN-Architektur sternförmig. Abbildung 1.6 zeigt das Prinzip auf der rechten Seite. LoRa-Endgeräte (engl. *End Devices*) kommunizieren mit LoRaWAN-Gateways, die die Datenpakete an einen LoRaWAN-Server senden. Der Server verbindet sich mit hoher Bandbreite über WiFi, Ethernet oder Mobilfunk mit einem Netzwerk für IoT-Plattformen und Applikationen. Die Kommunikation im LoRaWAN ist zweifach mit 128-Bit-AES (engl. *Advanced Encryption Standard*) verschlüsselt: zum einen bis zum Netzserver und zum anderen bis zu den Anwendungsservern. Alle Gateways leiten die Daten aller Endgeräte in Reichweite an einen LoRaWAN-Server weiter. Ein einziges Gateway kann Tausende Devices bedienen.

Abbildung 1.7 zeigt die Architektur in der Übersicht.

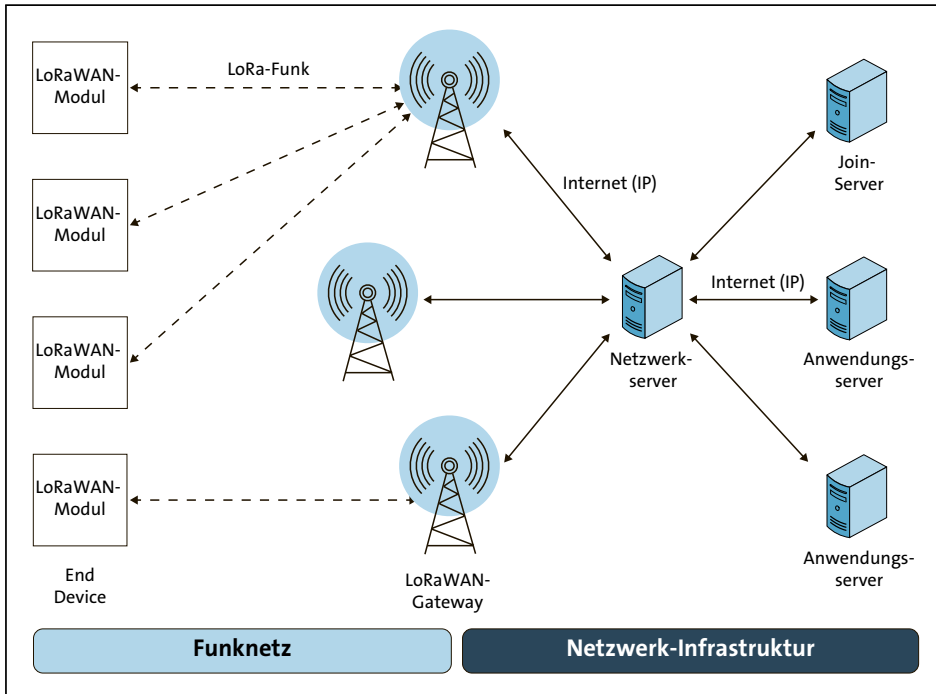


Abbildung 1.7: LoRaWAN-Architektur

Das zentrale Element eines LoRaWAN-Netzwerks ist der für die Verwaltungsfunktionen zuständige *Netzwerkserver* (engl. *Network Server*). Der *Join-Server* (von engl. *to join*, für dt. *beitreten*) kümmert sich um die Authentifizierung der Endgeräte. In einem LoRaWAN-Netzwerk gibt es zwei Authentifizierungsmethoden:

- personalisierte Aktivierung: ABP (engl. *Activation By Personalization*)
- gegenseitige Aktivierung: OTAA (engl. *Over The Air Activation*)

Bei *ABP* werden eine Geräteadresse und ein Session-Key fest vergeben. Bei *OTAA* führt das Endgerät eine Beitrittsprozedur (engl. *Join*) durch. Der *Anwendungsserver* (engl. *Application Server*) ist für die Weiterleitung aller vom Netzwerkserver empfangenen Pakete an die zugehörige Anwendung zuständig. In der anderen Richtung leitet er Nachrichten aus der Applikation an den Netzwerkserver weiter. Die Kommunikation zwischen den Servern erfolgt über das IP-Netzwerk per *Ethernet*, drahtlos per *WiFi* oder über den Mobilfunkstandard *LTE* (engl. *Long Term Evolution*).

Jedes LoRaWAN-Modul verfügt über mindestens einen LoRa-Chip sowie über eine Antenne, die für die Datenübertragung über weite Strecken optimiert wurde. Durch den sternförmigen Aufbau des LoRaWAN-Netzwerks ist sichergestellt, dass jeder Endpunkt über ein Gateway direkt mit dem Netzwerk verbunden ist. In diesem Netzwerk verteilt der zentrale Server die übermittelten, einzigartigen Endgeräte-IDs (*EUI*, engl. *Extended Unique Identifier*) und die Nutzdaten (engl. *Payload*) an die jeweilige Anwendung (engl. *Application*). Mit den eindeutigen IDs lassen sich alle Sensoren, Aktoren sowie Peripheriegeräte voneinander unterscheiden und die Daten lassen sich den richtigen Anwendungen zuordnen.

Die LoRaWAN-Spezifikation kennt drei Geräteklassen (siehe Abbildung 1.8). Grundsätzlich müssen alle Knoten der *Klasse A* entsprechen – Klasse B und Klasse C sind Erweiterungen der Spezifikation von Klasse A.

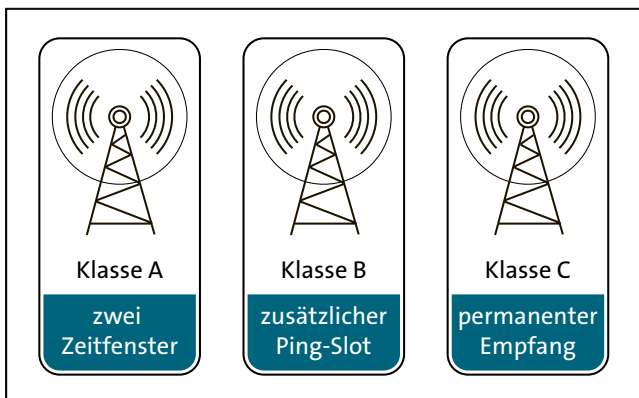


Abbildung 1.8: Geräteklassen der LoRaWAN-Spezifikation

- In der *Klasse A* findet eine bidirektionale Kommunikation zwischen einem Endgerät und dem Gateway statt. Die Übermittlung von Sensordaten (*Uplink*) ist jederzeit möglich. Danach öffnet das Gerät zwei Zeitfenster (*RX1-Delay* und *RX2-Delay*) und ermöglicht die Übermittlung von Werten (z. B. Konfigurationsdaten) vom Server an das Modul (*Downlink*). Die Geräte der Klasse A senden entweder regelmäßig entsprechend ihrer Konfiguration oder ereignisgesteuert beim Über- oder Unterschreiten bestimmter Schwellenwerte. Danach gehen sie batteriesparend in einen Power-down-Modus bis zur nächsten Übermittlung.

- Die erste Erweiterung der Klasse A ist die *Klasse B*. Hier werden in der Kommunikation periodische Empfangsfenster hinzugefügt. Das Gateway sendet alle 128 Sekunden ein kurzes Funktelegramm, ein sogenanntes *Beacon* (für dt. *Funkbake*). Durch den Empfang dieses Beacons können sich die Geräte mit dem Gateway synchronisieren. Das gesamte Zeitfenster für die Kommunikation wird als *Beacon-Periode* bezeichnet und die Zeit für den möglichen Downlink-Empfang wird *Ping-Slot* genannt. Nach dem Uplink sind auch noch die Zeitfenster gemäß Klasse A vorhanden. Im Gerät ist eine integrierte Echtzeituhr notwendig, damit es zu bestimmten Zeiten in Empfangsbereitschaft gehen kann.
- Besonders interessant sind Geräte, die der *Klasse C* zuzuordnen sind. Die Klasse C erweitert ebenfalls die Klasse A, indem hier ein permanentes Empfangsfenster offen gehalten wird. Die Geräte stehen ständig für Downlink-Daten zur Verfügung, das Zeitfenster schließt sich nur während einer Uplink-Übertragung. Dies ist energieaufwendig, und daher handelt es sich hier meist um Geräte, die an einer eigenen Stromversorgung hängen. Anwendungsbeispiele sind z. B. (Straßen-)Beleuchtungen, Bewässerungsanlagen, Ereignismelder, Fahrplananzeiger oder der IO-Controller, den Sie in Kapitel 4 kennenlernen: *DRAGINO LT-22222-L*. Tabelle 1.1 fasst die wesentlichen Informationen in einer Übersicht zusammen:

Klasse	Beschreibung	Anwendungsbeispiele
Klasse A	Das Gerät sendet Uplink-Daten, danach gibt es zwei Empfangsfenster für Downlink-Daten. Dieser Betrieb ermöglicht eine niedrige Energieaufnahme und lange Batterielaufzeit.	Alle LoRaWAN-Geräte unterstützen die Klasse A. Klasse-A-Geräte sind meist batteriebetriebene Knoten wie z. B. Fensterkontakte oder Füllstandssensoren und Leckagesensoren.
Klasse B	Zusätzlich zu den beiden Empfangsfenstern gemäß Klasse A gibt es einen Ping-Slot für den Empfang von Downlink-Daten. Dies erhöht den Stromverbrauch des Geräts, ermöglicht jedoch eine bessere Erreichbarkeit.	die Fernwartung von Maschinen oder Temperatur- und Feuchtesensoren mit einer tolerierbaren Übertragungsverzögerung von maximal 128 Sekunden

Klasse	Beschreibung	Anwendungsbeispiele
Klasse C	Es gibt ein permanentes Empfangsfenster für Downlink-Daten. Dies bietet die beste Erreichbarkeit und Zuverlässigkeit, geht jedoch mit einem höheren Energieverbrauch einher.	Beleuchtungen, Bewässerungsanlagen, Ereignismelder, Fahrplananzeiger, IO-Module, Relais, Stellglieder, Ventile

Tabelle 1.1: LoRaWAN-Geräteklassen

Mit den Geräten der Betriebsklasse A ist ein preiswerter und energiesparender Betrieb möglich. In der Klasse B ist eine etwas aufwendigere Zeitsynchronisation nötig und in Klasse C sollte eine feste Spannungsversorgung vorhanden sein. Der jeweilige Anwendungsfall entscheidet, welche Geräteklasse zum Einsatz kommen wird. In den allermeisten Fällen müssen Sie die *Latenzzeit* gegen einen eventuell höheren Stromverbrauch abwägen. LoRaWAN-Knoten können sowohl Sensoren (Uplink-lastig) als auch Aktuatoren (wenig energiesparend) sein.

Hinweis: Latenzzeit

Als *Latenzzeit* (auch: Verzögerungszeit) oder *Latenz* wird der Zeitraum zwischen einem Ereignis und einer Reaktion bezeichnet.

Das notwendige Netzwerk wird von verschiedenen Institutionen meist kostenlos zur Verfügung gestellt. Ein sehr bekanntes Netzwerk ist *The Things Stack* von *The Things Network* (siehe <https://www.thethingsnetwork.org/>).



Abbildung 1.9: Das Logo von »The Things Network«

Das *The Things Network* (TTN) ist eine offene, Community-basierte Initiative zur Errichtung eines energiesparenden Weitbereichsnetzwerks für das

Internet der Dinge und wurde 2015 gegründet. Es ist ein globales, offenes, kostenloses und dezentrales Projekt, bei dem mehr als 215.500 Freiwillige in 153 Ländern die Bereitstellung, die Errichtung und die Betreuung von über 21.000 LoRaWAN-Gateways übernehmen (Stand: Mai 2024). Auch The Things Network ist Mitglied der *LoRa-Alliance* (<https://lora-alliance.org/>).

Weitere Netzwerke stellen wir in Kapitel 6, »The Things Stack & Co.«, vor.

1.3 Der Einstieg für alle

Der Einstieg in die faszinierende LoRaWAN-Welt gestaltet sich für Maker sehr einfach. Die Module sind vielfältig erhältlich und auch relativ günstig. Sie brauchen nicht sofort Ihr eigenes Gateway, falls ein öffentliches Gateway erreichbar ist. Aber natürlich gibt es für erste eigene Versuche auch hier bezahlbare Hardware. Mehr darüber in Kapitel 5, »Das ELV-Modulsystem«.

Nach der Beschreibung der LoRaWAN-Grundlagen möchten wir Ihnen noch zwei praktische und absolut ernst gemeinte Warnhinweise mit auf den Weg geben:



Hier ein wichtiger Sicherheitshinweis, der für das ganze Buch gilt: Die Arbeit mit Modulen, die direkt im Stromnetz installiert werden, erfordert die Fachkenntnisse eines Elektrikers oder entsprechend qualifizierten Personals.



Daher der eindringliche Hinweis für Aktivitäten im 230-V-Umfeld: Führen Sie keine Reparaturen und Baseltarbeiten – selbst noch so einfacher Art – an elektrischen Geräten und Anlagen durch, wenn Sie keine ausreichenden Kenntnisse über die damit verbundenen Gefahren und die sichere Arbeitsweise besitzen!

4.3 DRAGINO LHT65 – Temperatur- und Luftfeuchtesensor

Ein weiterer LoRaWAN-Sensor der Firma DRAGINO ist der *LHT65*. Er ist ein Temperatur- und Luftfeuchtesensor, einfach in der Handhabung und weit verbreitet. Voll kompatibel mit dem *LoRaWAN v1.0.3 Class A*-Protokoll, arbeitet er mit jedem Standard-LoRaWAN-Gateway zusammen.

Das Gerät verfügt über einen integrierten *SHT20*-Temperatur- und Feuchtigkeitssensor. Über eine mehrpolige Klinkenbuchse lassen sich entweder der mitgelieferte externe Temperatursensor *DS18B20* oder ein Helligkeitssensor *BH1750* anschließen.

Der interne Temperatursensor hat eine Auflösung von 0,01 °C in einem Betriebsbereich von -40 bis +125 °C; der interne Luftfeuchtigkeitssensor hat eine Auflösung von 0,04 % RH (*Relative Humidity*) bei einem Betriebsbereich von 0 bis 96 % RH. Der externe Temperatursensor hat ein ca. 2 m langes Kabel und eine Auflösung von 0,0625 °C in einem Betriebsbereich von -40 bis +125 °C.



Abbildung 4.6: Der Temperatur- und Luftfeuchtesensor DRAGINO LHT65

Das Verbinden mit dem vierpoligen Programmier- und Konfigurationskabel erfolgt über einen magnetisch haftenden Anschlussstecker auf der Rückseite des Moduls. Der LHT65 verfügt über eine integrierte, austauschbare 2400-mAh-Batterie, das Öffnen des Gehäuses ist ohne Beschädigung jedoch etwas schwierig. In der Anleitung wird es so beschrieben: »The LHT65

has the top cover and bottom cover. User can use a knife to open the enclosure from the middle.« (Also auf Deutsch: »Der LHT65 verfügt über eine obere Abdeckung und eine untere Abdeckung. Der Benutzer kann das Gehäuse von der Mitte aus mit einem Messer öffnen.«). Passen Sie aber auf, dass Sie das Plastik dabei nicht beschädigen.

Der LHT65 wird inzwischen in weiteren Varianten vertrieben (z. B. als *LHT65N*, *LHT65S*). Bei meinem älteren Modell LHT65 waren ein E2-Programmierkabel und ein externer Temperatursensor im Lieferumfang enthalten. Mittlerweile ist das E2-Kabel optionales Zubehör, Sie brauchen es für Firmware-Updates und zur Konfiguration des Bausteins über *AT-Kommandos*. Die Klinkenbuchse wurde durch eine USB-C-Buchse ersetzt. Die nicht aufladbare LiMn-Batterie vom Typ *CR17450* (3 V, 2400 mAh) reicht laut Herstellerangabe bis zu 10 Jahre, abhängig von der Anzahl der Sendevorgänge und der Entfernung zum Gateway.



Abbildung 4.7: Der aktuelle Temperatur- und Luftfeuchtesensor DRAGINO LHT65N-E3

Hinweis: Der AT-Befehlssatz und das AT-Kommando

Der *AT-Befehlssatz* wurde ursprünglich von der Firma *Hayes Microcomputer Products* entwickelt. Die Befehle dienen zum Konfigurieren von Modems und sind mittlerweile zu einem Industriestandard geworden. Die Zeichen *AT* stehen dabei für das englische Wort *attention* (dt. *Achtung*) und müssen vor jedem Befehl gesendet werden.

Der neuere *LHT65N* hat eine Datalog-Funktion, die Daten aufzeichnet, wenn keine Netzabdeckung besteht. Damit kein Sensorwert verpasst wird, können die Sensorwerte dann später über LoRa abgerufen werden.

4.4 Seed Technology – Generic Node

Ein sehr interessantes Modul ist der *Generic Node* von *Seed Technology* – ein multifunktionaler LoRaWAN-Knoten mit vielen Anwendungsmöglichkeiten und einem extrem niedrigen Stromverbrauch. Die LoRaWAN-Antenne ist auf der Platine verbaut, ebenso wie der 3-Achsen-Beschleunigungssensor *LIS2DH12* von *STMicroelectronics* und der Temperatur- und Luftfeuchtigkeitssensor *SHTC3* von *Sensirion*. Ein User-Button auf der Platine erlaubt Eingaben, und eine RGB-Leuchtdiode und ein Piezo-Lautsprecher geben visuelles und Audio-Feedback.

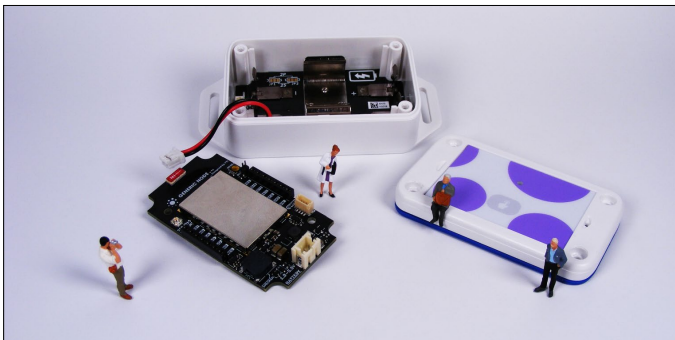


Abbildung 4.8: Der »Generic Node« von Seed Technology

Der interne Temperatursensor hat einen Betriebsbereich von -40 bis $+125$ °C mit einer Auflösung von $0,2$ °C. Der interne Luftfeuchtigkeitssensor deckt einen Betriebsbereich von 0 bis 100 % RH (*Relative Humidity*) ab und hat eine Auflösung von 2 % RH. Das Gehäuse erfüllt die Schutzart IP65, und die beiden austauschbaren, nicht im Lieferumfang enthaltenen AA-Batterien haben laut Hersteller eine Lebensdauer von über vier Jahren.

Zwei siebenpolige Pin-Leisten mit Programmier- und Debug-Anschlüssen sowie GPIO-Pins machen das Modul zum idealen Versuchsobjekt für jeden

```

lar system
*/
function Decoder(bytes, port) {
    var decoded = {};    // Container with the decoded out-
put
    var index = 5;      // Index variable for applicati-
on data in a bytes[] array
    var Temp_Value = 0; // Variable for temporarily calcu-
lated values
...

```

Listing 8.2: Auszug aus dem ELV-Parser V1.10.1.

Wie die umgewandelten Nutzdaten weiterverarbeitet werden können, erfahren Sie in den Abschnitt 8.4 und Abschnitt 8.5 sowie in den Anwendungsbeispielen in Kapitel 9.

8.3 Die Bausteine konfigurieren

Viele der LoRaWAN-Geräte können über das Senden von Konfigurations-Bytes per Downlink-Message konfiguriert werden. Ich zeige Ihnen dies bei einigen Geräten anhand der Screenshots von *The Things Stack*.

Hinweis: Bit und Byte

Das *Bit* ist die kleinste Informationseinheit eines Rechners und entspricht den Zuständen *Strom an* (1) und *Strom aus* (0). Das *Byte* ist eine Maßeinheit der Digitaltechnik und der Informatik; es steht für eine Folge von 8 Bit. Damit können vorzeichenlos die Werte 0 bis 255 dargestellt werden.

Eine Bitfolge mit 4 Zahlen wird (weniger gebräuchlich) als *Nibble* bezeichnet. Eine andere Bezeichnung für eine geordnete Zusammenstellung von 8 Bit ist *Oktett*. Mit zwei Bytes kann ein Wertebereich von 0 (0x0000) bis 65.535 (0xFFFF) dargestellt werden.

Zunächst sind die Bytes zu ermitteln, die gesendet werden sollen. In den Anleitungen zu den LoRa-Devices gibt es Parameter-Tabellen und meist auch Zahlenbeispiele.

Das erste Downlink-Beispiel ist für den LHT65 von DRAGINO. Bei ihm können Sie einige Parameter nicht nur durch AT-Befehle (siehe Abschnitt 4.3), sondern auch über einen Downlink einstellen. Standardmäßig sendet der LHT65 alle 20 Minuten (= 1200 Sekunden) seine Werte an den Server. Für die Umstellung auf ein 15-minütiges Sendeintervall muss das TDC (Transmit Time Interval) auf 900 Sekunden reduziert werden (siehe Abbildung 8.17).

The screenshot shows the Dragino website with the following content:

DRAGINO www.dragino.com

2.5 Downlink Payload
After each success downlink, the **PURPLE LED** will blink once.

Downlink Command	FPort	Header	Downlink payload format
TDC (Transmit Time Interval)	Any	01	Total bytes: 4. Unit: Second Example: 0x01000258: Set AT+TDC=0x258=600S Recommend: higher than 0x12C (5 minutes)
RESET	Any	04	Total bytes: 2 Example: 0x04FF, reset device
AT+CFM (Set Confirm status)	Any	05	Total bytes: 2 Example: 0x0501: AT+CFM=1, 0x0500: AT+CFM=0
AT+CHE	Any	07	Total bytes: 2 Example: 0x0702: AT+CHE=2
AT+DATE (Set time)	Any	A1	Total bytes: 7 Example: 0xA1190530162158: Set RTC time to 2019-5-30: 16:21:58
AT+EXT	Any	A2	Total bytes: 2 Example: 0xA201: Set external sensor type to E1
AT+CLRDTA (Clear Storage)	Any	A3	Total bytes: 2 Example: 0xA301: Clear data storage
AT+RTP (Set Record Time Period)	Any	A4	Total bytes: 3 unit: minutes Example: 0xA400A: Set record time period to 10 minutes

Abbildung 8.17: Downlink-Kommandos für der LHT65 von Dragino

Das erste Byte 0x01 ist der Header (dt. Kopf) des Kommandos, gefolgt von einem Nullwert 0x00 und zwei weiteren Bytes für die Anzahl der Sekunden. Durch die Umrechnung von dezimal auf hexadezimal mit einem Taschenrechner oder einer App erhalten Sie aus den gewünschten 900 Sekunden die Hex-Zahl 0x0384. Das Ergebnis besteht also aus insgesamt vier Bytes.

Im Menüpunkt **Messaging • Downlink** wird die Downlink-Maske angezeigt (siehe Abbildung 8.18). Der **FPort** ist hier willkürlich auf **4** eingestellt, aber der **LHT65** akzeptiert die Downlink-Daten auf einem beliebigen (engl. *any*) **FPort** – die komplette Byte-Folge lautet: 01 00 03 84.

Abbildung 8.18: Downlink-Payload für den LHT65 von DRAGINO

In der Anleitung wird empfohlen, mit dem Sendeintervall nicht unter fünf Minuten zu gehen: Das verlängert die Batterielaufzeit und hält die Datentransferrate niedrig. Nach einem Klick auf **Schedule downlink** werden die Werte in die Warteschlange eingereiht und warten auf die Übermittlung an das Gerät (siehe Abbildung 8.19). Um Energie zu sparen, werden bei **Klasse-A**-Knoten die Downlink-Werte immer erst nach dem Eintreffen von Uplink-Werten übermittelt. Dafür stehen zwei Zeitfenster zur Verfügung, in denen das Modul empfangsbereit ist (siehe Abschnitt 1.2).

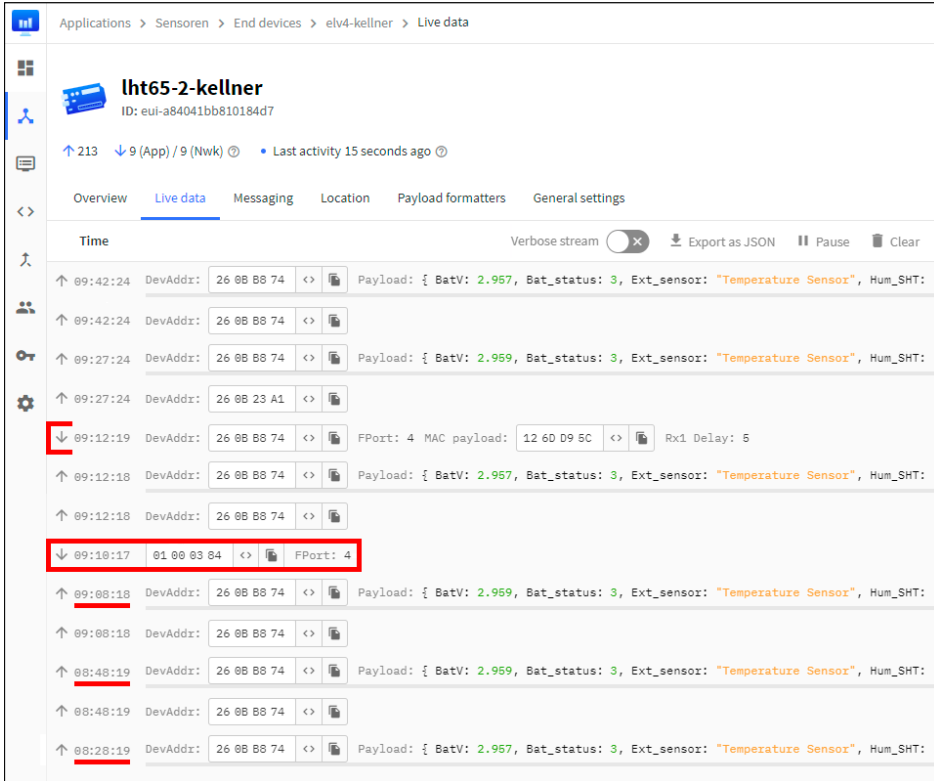


Abbildung 8.19: Das Protokoll von Up- und Downlink-Daten beim LHT65

Dies war ein erstes, einfaches Beispiel für den LHT65 von DRAGINO; bei anderen Geräten kann es etwas schwieriger werden. Je nach den Konfigurationsmöglichkeiten werden die Byte-Folgen auch länger und damit leider auch fehleranfälliger in der Bestimmung der Bytes.

Abbildung 8.20 zeigt ein zweites Beispiel für das Anwendungsmodul ELV-AM-AP.

Konfigurationsoptionen des ELV-AM-AP								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte0	Geräte-ID 0x0F							
Byte1	I ² C-Adresse: 0x46 oder 0x47, je nach Überbrückung der Lötbrücke J4 (von Bit7 bis Bit0)							
Byte2	x	x	x	x	x	Interrupt-Modus: mitlaufender Sollwert (Standardwert 1)	Interrupt-Modus aktiviert (Standardwert 0)	Zyklischer Modus aktiviert (Standardwert 1)
Byte3	Zeitintervall der zyklischen Messung in Minuten: Wertebereich 1-255, bei 0 wird der alte Wert beibehalten (Standardwert 0x0A → 10 Minuten)							
Byte4	Sollwert im Interrupt-Modus in Pa: Wertebereich 30000-125000, Werte außerhalb werden ignoriert (Standardwert 0x018BCD → 101325 Pa)							
Byte5								
Byte6								
Byte7	Abweichung vom Sollwert im Interrupt-Modus in Pa: Wertebereich 1-255, bei 0 wird dieses Feld ignoriert (Standardwert 100)							
Byte8	Anzahl an Messungen (bzw. 8-s-Intervallen), die die Schwellwerte überschreiten: Wertebereich = 0x1-0x4: 0x01 = 1 Messung, 0x02 = 3 Messungen, 0x30 = 7 Messungen, 0x04 = 15 Messungen (Standardwert 0x02)							

Abbildung 8.20: Downlink-Payload des ELV-AM-AP (Screenshot: ELV AG)

Die Downlink-Payload beginnt mit dem erstem Byte (*Byte0*) und beinhaltet die Geräte-ID (hier: $0 \times 0F$ für das *ELV-AM-AP*). Beachten Sie, dass die Nummerierung für das erste Byte mit 0 beginnt – das letzte, neunte Byte ist demnach *Byte8*. Bei den Anwendungsmodulen aus dem ELV-Modulsystem ist es grundsätzlich so, dass stets der komplette Downlink-String übermittelt wird (hier *Byte0* bis *Byte8*).

Im zweiten Byte geht es um die I²C-Adresse des Moduls – hier ändern wir nichts, es bleibt der Default-Wert 0×46 . Das dritte Byte legt den Modus fest, der Vorgabe-Wert ist 0×03 . Bei nächsten Byte sind wir beim Sendeintervall angekommen: Der Default-Wert sind 10 Minuten ($0 \times 0A$) – eine Änderung auf 15 Minuten bedeutet *Byte3* = $0 \times 0F$. Die weiteren Bytes bleiben wieder unverändert. Somit ergeben sich diese neun Hexadezimal-Zahlen als Download-Payload: $0F\ 46\ 03\ 0F\ 01\ 8B\ CD\ 64\ 02$ (siehe Abbildung 8.21).

Der **FPort** ist bei den ELV-Modulen in der Regel auf 10 einzustellen, bei anderen Herstellern kann er natürlich anders lauten.

The screenshot shows a configuration window for 'Schedule downlink'. The 'Downlink' tab is active. Under 'Insert Mode', 'Replace downlink queue' is selected. The 'FPort' field contains the value '10'. Under 'Payload type', 'Bytes' is selected. The 'Payload' field contains the hexadecimal string '0F 46 03 0F 01 8B CD 64 02', which is highlighted with a red box. Below the payload field, there is a checkbox for 'Confirmed downlink' which is unchecked. At the bottom, there is a blue button labeled 'Schedule downlink' with a mouse cursor pointing to it.

Abbildung 8.21: Downlink für das Anwendungsmodul ELV-AM-AP

Im Protokoll ist nach entsprechender Wartezeit wieder die Änderung sichtbar, wie Abbildung 8.22 beweist.

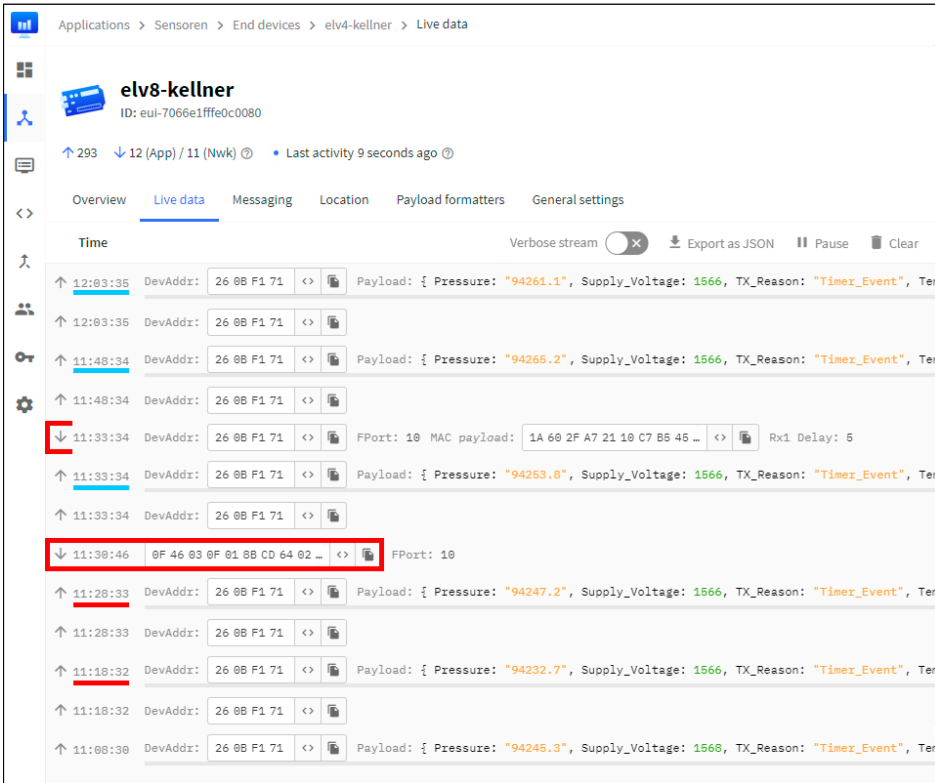


Abbildung 8.22: Das Protokoll von Up- und Downlink-Daten des ELV-AM-AP

Für erste Versuche können Sie das Sendeintervall zunächst auf eine Minute einstellen, denn so werden die Änderungen sehr schnell sichtbar. Für eine lange Batterielaufzeit empfehlen sich je nach Anwendungsfall allerdings höhere Werte.

Sehen wir uns jetzt ein weiteres Beispiel aus dem ELV-Modulsystem an: Das Sende-Intervall des Zweifach-Temperatursensors *ELV-AM-T2* (siehe Abbildung 5.5) ist standardmäßig auf fünf Minuten eingestellt, wie Sie in Abbildung 8.23 in der Zeile von *Byte1* erkennen. Um dieses Intervall auf fünfzehn Minuten zu erhöhen, sind 16 hexadezimale Bytes zu senden.

Konfigurationsoptionen des ELV-AM-T2 (Downlink)										
Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default	
Byte 0	0x13								Geräte ID	
Byte 1	1-255, 0 = kein zyklisches Sendeintervall								Sendeintervall in Minuten	0x05(5 min)
Byte 2	1-1500 = -499 bis 1000 (Auflösung 0,1°C) = -49,9 bis +100,0 °C								Obere Schwellwertgrenze von T1 in °C	0x0000 (aus)
Byte 3	0: Schwellwertgrenze aus/andere Werte werden ignoriert									
Byte 4	1-1500 = -499 bis 1000 (Auflösung 0,1°C) = -49,9 bis +100,0 °C								Untere Schwellwertgrenze von T1 in °C	0x0000 (aus)
Byte 5	0: Schwellwertgrenze aus/andere Werte werden ignoriert									
Byte 6	1-1500 = -499 bis 1000 (Auflösung 0,1°C) = -49,9 bis +100,0 °C								Obere Schwellwertgrenze von T2 in °C	0x0000 (aus)
Byte 7	0: Schwellwertgrenze aus/andere Werte werden ignoriert									
Byte 8	1-1500 = -499 bis 1000 (Auflösung 0,1°C) = -49,9 bis +100,0 °C								Untere Schwellwertgrenze von T2 in °C	0x0000 (aus)
Byte 9	0: Schwellwertgrenze aus/andere Werte werden ignoriert									
Byte 10	1-1500 = -499 bis 1000 (Auflösung 0,1°C) = -49,9 bis +100,0 °C								Obere Schwellwertgrenze von T1-T2 in °C	0x0000 (aus)
Byte 11	0: Schwellwertgrenze aus/andere Werte werden ignoriert									
Byte 12	1-1500 = -499 bis 1000 (Auflösung 0,1°C) = -49,9 bis +100,0 °C								Untere Schwellwertgrenze von T1-T2 in °C	0x0000 (aus)
Byte 13	0: Schwellwertgrenze aus/andere Werte werden ignoriert									
Byte 14	0-100 = -50 bis 50 (Auflösung 0,1°C) = -5,0 bis +5,0 °C								Temperaturoffset T1 in °C	0x32(0,0 °C)
Byte 15	0-100 = -50 bis 50 (Auflösung 0,1°C) = -5,0 bis +5,0 °C								Temperaturoffset T2 in °C	0x32(0,0 °C)

Abbildung 8.23: Konfigurationsoptionen des ELV-AM-T2 (Screenshot: ELV AG)

Die Downlink-Payload beginnt mit dem erstem Byte (*Byte0*) und beinhaltet wieder die Geräte-ID (hier: 0×13 für das ELV-AM-T2). Im zweiten Byte (*Byte1*) folgt das zyklische Sendeintervall für die periodische Übermittlung zum LoRaWAN-Server. Dieser Wert kann in einem Bereich von 0 bis 255 Minuten eingestellt werden, wobei der Wert 0×00 bedeutet, dass keine periodische Sendung der Messwerte stattfinden soll. Als Default-Wert sind fünf Minuten (0×05) vorgegeben, aber wir wollen das Sendeintervall auf 15 Minuten erhöhen, also lautet der neue Wert $0 \times 0F$. Die weiteren Bytes bleiben unverändert, hier werden die Vorgabe-Werte übermittelt. Für die Downlink-Payload werden die Bytes `13 0F 00 00 00 00 00 00 00 00 00 00 00 00 32 32` ermittelt (siehe Abbildung 8.24).

Uplink Downlink

Schedule downlink

Insert Mode

Replace downlink queue
 Push to downlink queue (append)

FPort*

10

Payload type

Bytes JSON

Payload

13 0F 00 00 00 00 00 00 00 00 00 00 00 00 32 32

The desired payload bytes of the downlink message

Confirmed downlink

Schedule downlink

Abbildung 8.24: Downlink für das Anwendungsmodul ELV-AM-T2

In Abbildung 8.25 sehen Sie wieder das Protokoll mit den Up- und Downlink-Daten.

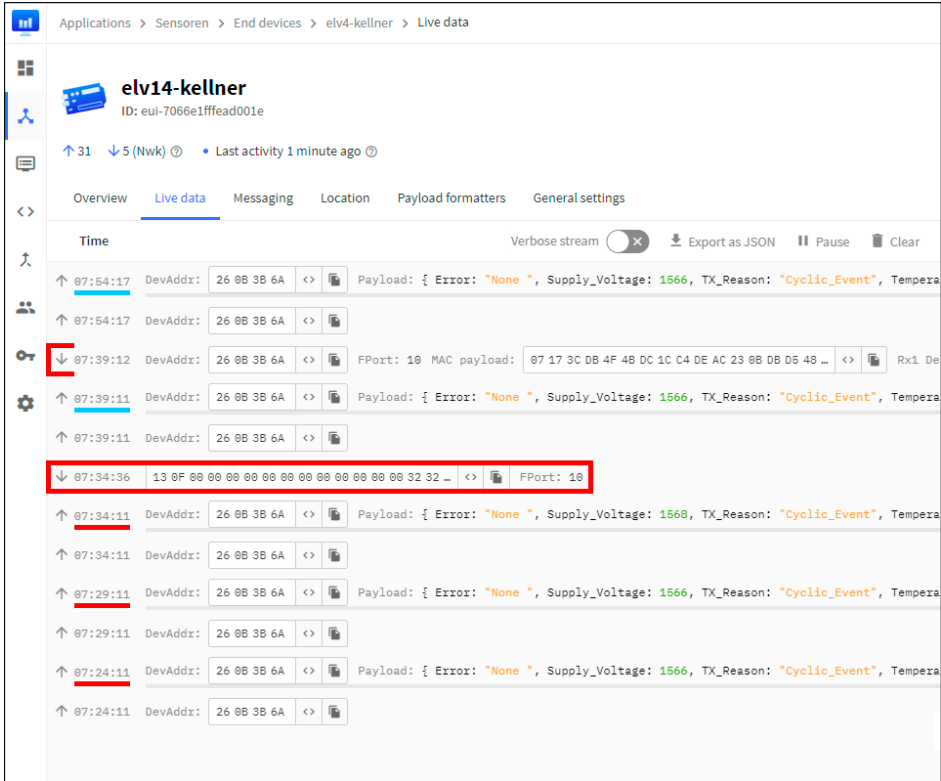


Abbildung 8.25: Das Protokoll von Up- und Downlink-Daten des ELV-AM-T2

Dies war ein einfaches Beispiel für das Anwendungsmodul *ELV-AM-T2*, in dem zwar sechzehn Bytes gesendet, aber nur eines geändert wurden. Bei anderen Geräten kann es aber aufwendiger werden!

Schauen Sie sich dazu Beispiel Nummer vier an: Bei der Interface-Platine *ELV-LW-INT1* aus Abbildung 5.36 ist der Konfigurationsmechanismus ein anderer: Hier müssen nicht immer alle Parameter übermittelt werden, sondern es können einzelne ausgewählt werden. Auch in diesem Beispiel soll wieder das Sendeintervall auf fünfzehn Minuten verlängert werden.

downlink frame types (FPort 10)	
ID	name
0x00	set single parameters
0x01	set following parameters
0x02	get config by channel
0x03	get firmware ID
0xf8	get spreading factor
0xf9	set spreading factor
0xfa	get rejoin
0xfb	set rejoin
0xfc	get config (complete)
0xfd	factory reset
0xfe	soft reset
0xff	get device info

Abbildung 8.26: Auswahl des Frame-Typs (Screenshot: ELV AG)

Für die Konfiguration des *INT1* werden verschiedene Frame-Typen angeboten (siehe Abbildung 8.26). Wir wollen einen einzelnen Parameter ändern, also wählen wir den Frame-Typ 0x00.

downlink header		
0	frame type	frame type
frame type = 0x00		
set single parameters		
1	channel	channel ID
2	parameter ID	ID of the parameter for selected channel
3..n	parameter value	value for the selected ID and channel (1 or several bytes)
..	parameter ID	ID of the parameter for selected channel
..	parameter value	value for the selected ID and channel (1 or several bytes)

Abbildung 8.27: Auswahl der Parameter (Screenshot: ELV AG)

Beim Frame-Typ 0x00 set single parameters müssen jeweils der Kanal, die Parameter-ID und die Werte angegeben werden (siehe Abbildung 8.27).

frame type	0x00	set single parameters
channel	0x01	DUS1
parameter ID	0x00	cyclic TX interval
parameter value	0x0F	15 Minuten
parameter ID	0x03	measurement interval
parameter value	0x85	5 Minuten
parameter ID	0x05	threshold level 1
parameter value	0x03E8	1000 mm

Abbildung 8.28: Konfiguration für drei Parameter (Screenshot: ELV AG)

Der Default-Wert für zyklische Geräte-Infos beträgt 30 Minuten und die zyklischen Sendungen für die Sensorkanäle sind komplett deaktiviert. Entweder müssen für die Sensorkanäle eigene zyklische Sendeintervalle aktiviert werden oder die Sensorkanäle werden einer Statusgruppe zugeordnet, die beim Senden der Gerätekanal-Statusmeldung mitgesendet wird. Weitere Infos finden Sie in den Fachbeiträgen zum *ELV-LW-INT1* und zu den Sensoren im kostenlosen *ELVjournal 2/2024*.

Der Downlink-String beginnt mit `0x00` und `0x01` für das Setzen eines einzelnen Parameter des *DUS1* (siehe Abbildung 8.28). Danach folgen zwei weitere Bytes mit der Parameter-ID und dem gewünschten Wert für fünfzehn Minuten. Der komplette String für das Sendeintervall lautet also `00 01 00 0F` (siehe Abbildung 8.29).

Abbildung 8.29: Downlink für das Anwendungsmodul *ELV-LW-INT1*

Auch in diesem Fall werden die Werte nach dem Klick auf **Schedule downlink** in die Warteschlange geschrieben. Nach dem Eintreffen von Uplink-

Werten wird die Downlink-Payload übermittelt (siehe Abbildung 8.30). Dies kann durch Drücken des User-Buttons oder durch kurzzeitiges Entnehmen der Batterien etwas beschleunigt werden.

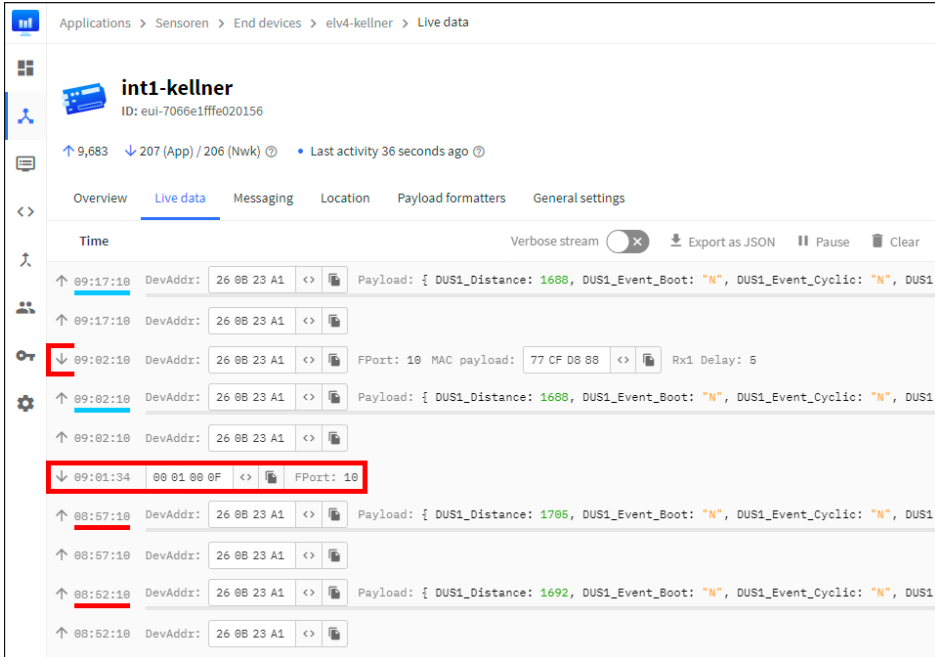


Abbildung 8.30: Das Protokoll von Up- und Downlink-Daten des ELV-LW-INT1

Das Konfigurieren von Parametern kann je nach Gerät sehr komplex werden. Arbeiten Sie sorgfältig beim Ermitteln der Downlink-Bytes und schauen Sie in den Anleitungen oder im Internet nach Zahlenbeispielen. Meist passen die Default-Einstellungen, aber gerade die Sendintervalle sind manchmal zu klein gewählt und für eine längere Lebensdauer der Batterien sind größere Abstände für das Übermitteln von Werten sinnvoll. Es hängt im Einzelfall aber immer von der jeweiligen Anwendung und dem Sensorwert ab, den Sie ermitteln wollen: Bei Temperatur oder Luftdruck genügen vielleicht Werte alle fünfzehn Minuten – bei der Windgeschwindigkeit sollte man jedoch kurzfristiger reagieren können, um z. B. eine Markise rechtzeitig einzufahren, bevor ein aufkommender Sturm sie beschädigt.

Inhalt

Kapitel 1 Einführung – Wer ist LoRa? 9

1.1 Ein Protokoll erobert die Welt	9
1.2 Ein Exkurs in die Technik – keine Angst vor Fachbegriffen	12
1.3 Der Einstieg für alle	18

Teil I Die Grundlagen – Hardware und Software

Kapitel 2 Ein passendes Gateway 21

2.1 DRAGINO LPS8	23
2.2 MikroTik LR8	28
2.3 Seeed Technology SenseCAP M2	36
2.4 The Things Industries TBMH100 Minihub	39
2.5 CapiBridge Gateway	41

Kapitel 3 Module und Knoten 45

3.1 IoT-Module und kein Überblick?	45
3.2 Ein System für fast alles: die Papageien von ELV	46
3.3 Das Basis-Modul als zentrales Element	48
3.4 Die Stromversorgung der Sensoren	51
3.5 Offen für eigene Entwicklungen und Erweiterungen	56
3.6 Gut verpackt: Welches Gehäuse für die Anwendung?	63

Kapitel 4 Sensoren und Aktoren 71

4.1 DRAGINO LWL02 – Wasserleck-Sensor	71
4.2 DAGINO LDS02 – Tür- und Fenstersensor	73
4.3 DRAGINO LHT65 – Temperatur- und Luftfeuchtesensor	74
4.4 Seeed Technology – Generic Node	76
4.5 kuando – eine Busylight-Anzeige für LoRaWAN	78

4.6	DRAGINO LT-22222-L – IO-Controller	79
4.7	DRAGINO LA66 – LoRaWAN-Shield	81
4.8	Stemedu LoRa-Node	82
4.9	Heltec ESP32-LoRa-Modul SX1276	83

Kapitel 5 Das ELV-Modulsystem 85

5.1	Viele Anwendungen – das Komplettsystem für LoRaWAN	85
5.1.1	Kontakte für alle – das Kontakt-Interface stellt sich vor	86
5.1.2	Temperatur und Luftfeuchtigkeit – sehr interessant für viele	88
5.1.3	Temperatur mal zwei – Differenz oder synchron	89
5.1.4	Bitte auf Abstand – die Distanzüberwachung	90
5.1.5	Schön hell hier – ein einfaches Luxmeter	91
5.1.6	Gut bestrahlt – die Erfassung weiterer optischer Messwerte	93
5.1.7	Bewegung mit Sicherheit – ein einfacher Bewegungs- melder	93
5.1.8	Nur nicht zu schnell – ein Beschleunigungsmodul	95
5.1.9	Wie ist die Lage? – Der Lagesensor zeigt es an	95
5.1.10	Immer dieser Druck – der Luftdrucksensor kennt die Höhe	97
5.1.11	Dicke Luft? – ein Schnüffler für Drinnen	98
5.1.12	Die Luftgüte-Ampel für jedermann	99
5.1.13	Spannend unter Strom – das Strom- und Spannungsmodul	101
5.1.14	Wo bin ich? – Ein GPS-Modul zur Ortsbestimmung	102
5.1.15	Eine Anzeige für viele Module – das Display-Modul	103
5.1.16	Guter Anschluss – ein Interface zu UART und I ² C	104
5.1.17	Wetter ist immer – eine komplette Wetterstation	106
5.1.18	Ab auf die Weide – eine Zaunüberwachung	116
5.2	Der Typ LW – die Kompletthauteile für das LoRaWAN	118
5.2.1	Alles kippt – ein Outdoor-Erschütterungssensor	118
5.2.2	Wassersensor-Bausatz für das LoRaWAN	119
5.2.3	Eine Schnittstelle für Energiezähler	121

5.2.4	Ein universelles Interface	122
5.2.5	Alles auf Abstand: Der Ultraschallsensor	123
5.2.6	Ab in den Garten: Der optimale Bodenfeuchtigkeits- sensor	125
5.2.7	GPS die zweite – ein fast fertiger GPS-Tracker	126
5.2.8	Nur keinen (Fein-)Staub aufwirbeln!	128
5.2.9	Sensoren für das Smart Home	129

Teil II Raus in die Wildnis – Daten ins Internet

Kapitel 6 The Things Stack & Co. 135

6.1	Welche Plattform soll es sein?	137
6.2	Helium und andere	137
6.3	Der Favorit: TTS – The Things Stack	139

Kapitel 7 Sensoren anmelden 145

7.1	Beginnen wir einfach: Temperatur und Luftfeuchtigkeit	146
7.2	Damit ein Licht aufgeht: die Busylight-Anzeige	153
7.3	Wetter ist immer: das Board für die Wetterstation	160
7.4	Das Interface und seine Möglichkeiten	164

Kapitel 8 Software 167

8.1	Download der Firmware	167
8.2	Daten auswerten	176
8.3	Die Bausteine konfigurieren	183
8.4	Daten am Haken: einen Webhook programmieren	195
8.5	Visualisierung mit externen Dienstleistern	219
8.6	Roter Alarm: Node-RED und MQTT	239

Kapitel 9 Anwendungsbeispiele – für alle etwas dabei 253

9.1	Türen, Fenster und Briefkästen überwachen	253
-----	---	-----

9.2	Ein Klassiker: die LoRaWAN-Wetterstation	259
9.3	Ist der Parkplatz noch frei?	261
9.4	Oje – Wasser im Keller	265
9.5	Pegelmessung an einem Bachlauf	269
9.6	LoWTrack – GPS-Positionsmeldung in einer App	272
	Index	282