

Stromsparendes IoT-System

LoRaWAN-Experimentierplattform LoRIS-Base LoRIS-BM-TRX1

Mit unserer Experimentierplattform LoRIS-Base steigen wir in die Welt von LoRaWAN ein – einer faszinierenden Kombination aus Funk- und Netzwerktechnologie, die mit wenig Energie Daten über weite Strecken in einem lizenzfreien Frequenzband übertragen kann. Die möglichen Anwendungen sind beeindruckend vielfältig: vom Temperatursensor auf dem freien Feld über den Tracker, der jede Art von Gegenstand per angeschlossenen GPS verfolgbar macht, bis zum Parkplatzsensor, der zukünftig ein intelligentes Parkraum-Management in Städten ermöglicht. Die übermittelten Daten sind dabei Ende-zu-Ende verschlüsselt. Die LoRIS-Base stellt mit dem im Funkmodul enthaltenen System-on-Chip (SoC) von STMicroelectronics die Basis-Funktionalität zur Verfügung – die Funk-Anbindung an LoRaWAN. Das SoC beherbergt aber auch die Anwendung, die bei unserer LoRIS-Base bereits eine Firmware enthält, die den Anschluss von zwei Tastern und einem Output ermöglicht, später aber an andere Anwendungen und weitere aufsteckbare Module angepasst werden kann.

Mit einem Klick
direkt zum Bausatz



LoRIS-BM-TRX1

Artikel-Nr.
156514

Bausatz-
beschreibung
und Preis:



www.elv.com



Infos zum Bausatz LoRIS-BM-TRX1



Schwierigkeitsgrad:
leicht



Besondere Werkzeuge:
keine



Lötterfahrung:
nein



Programmierkenntnisse:
nein



Elektrische Fachkraft:
nein

Basis für Experimente mit LoRaWAN

Die LoRIS-Base (LoRIS-BM-TRX1, Bild 1) stellt die Basis-Funktionalität in dem System für Experimente mit LoRaWAN zur Verfügung. Ähnlich wie bei einem Arduino oder ESP8266/32 ist die LoRIS-Base die Plattform, an die zusätzliche Peripherie angeschlossen werden kann, um entsprechende Funktionen, Schaltungen und Experimente zu ermöglichen. In unserem LoRIS-System aber auf Basis der Funktechnologie LoRa/LoRaWAN.

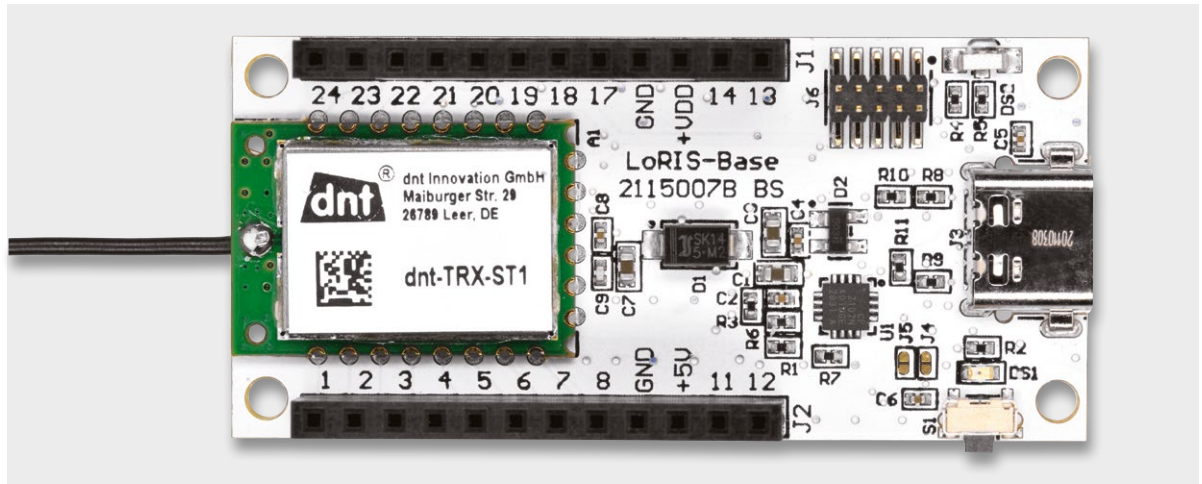
CE-zertifiziertes Einsteigersystem mit Plug-and-Play-Funktionalität

Die LoRIS-Base und auch das LoRIS-System mit übereinander steckbaren Anwendungsmodulen (Bild 2) z. B. zur Spannungsversorgung oder für Sensoren sollen vor allem dem Einsteiger eine einfache Möglichkeit geben, in das Thema LoRaWAN zu starten.

Zu dem jeweiligen Anwendungsmodul gibt es die entsprechend passende Firmware. So muss kein eigener Programm-Code erstellt werden, um beispielsweise ein Anwendungsmodul mit Temperatur-/Luftfeuchtesensor, GPS-Modul oder ein Taster-Anwendungsmodul, die als Bausätze in



Bild 1: LoRIS-Base



den folgenden Ausgaben des ELVjournals vorgestellt werden, im LoRaWAN zu betreiben. Verschiedene Powermodule ermöglichen zudem unterschiedliche Arten der Spannungsversorgung.

Das auf der LoRIS-Base verwendete Funkmodul ist CE-zertifiziert und kann somit im europäischen Wirtschaftsraum betrieben werden. Mit der CE-Kennzeichnung erklärt der Inverkehrbringer, dass das Produkt den geltenden Anforderungen genügt. Somit wird gewährleistet, dass das Gerät sich nicht von anderen Quellen stören lässt und auch andere Geräte nicht in ihrer Funktion stört.

Programmierkenntnisse sind für erste Tests mit der LoRIS-Base zunächst nicht notwendig, eine Basis-Firmware wird von uns zur Verfügung gestellt und ist bereits auf der LoRIS-Base gespeichert.

Alle Hex-Files (Firmware), Decoder-Beispiele (s. u.), USB-Seriell-Treiber und auch Code-Templates für die über den Debug-Adapter zusätzlich mögliche Programmierung mit eigenem Code (z. B. über die STM32CubeIDE) stellen wir auf der Webseite der LoRIS-Base [1] zur Verfügung. Die Hardware der LoRIS-Base ist bereits fertig aufgebaut – es müssen keine Bauteile gelötet werden.

LoRaWAN gewinnt immer mehr an Popularität

Das von dem amerikanischen Unternehmen Semtech patentierte Übertragungsverfahren LoRa bzw. das Netzwerkprotokoll LoRaWAN (Long Range Wide Area Network), dessen Spezifikationen von der LoRa Alliance festgelegt werden, haben in den letzten Jahren stark an Popularität hinzugewonnen. Sie bilden die Basis für ein sogenanntes Low Power Wide Area Network (LPWAN [2]). Damit sind hauptsächlich Sensorknoten gemeint, die energiesparend oder sogar energieautark meistens geringe Datenmengen (bei LoRaWAN 51 bis maximal 222 Bytes/Sendung) per Funk an eine entsprechende Netzwerk-Infrastruktur weiterleiten. Einen solchen Sensorknoten stellt beispielsweise unsere LoRIS-Base dar.

Was bedeutet das konkret und wie kann ich das praktisch anwenden?

Wir starten mit einem einfachen Beispiel. Stellen Sie sich vor, Sie möchten den Temperaturverlauf in Ihrem Garten überwachen und in Intervallen von beispiels-

weise zehn Minuten die Temperatur an eine Gegenstelle übermitteln. Ein vorhandenes WLAN-Netzwerk im Haus wird von der Reichweite her meistens nicht ausreichen, um einen Funkempfang zu garantieren.

Doch selbst wenn die WLAN-Netzwerkabdeckung vorhanden ist, ist die benötigte Energie für die Funkkommunikation recht hoch, sodass ein per Akku versorgter Sensor im Garten nicht lange betrieben werden kann, ohne ihn aufladen zu müssen oder zu wechseln. Andere Funktechnologien wie Bluetooth oder ZigBee sind zwar deutlich stromsparender als WLAN, von den Reichweiten her aber ebenfalls eher begrenzt. Hier kommt LoRaWAN ins Spiel, bei dem die Reichweiten je nach Umgebungsbedingungen im Bereich einige Hundert Meter bis einige Kilometer bis zu einer Funk-Gegenstelle (s. Gateway) liegen. Nicht nur den Garten, sondern ganze Landwirtschaftsbetriebe oder mit der entsprechenden Infrastruktur sogar

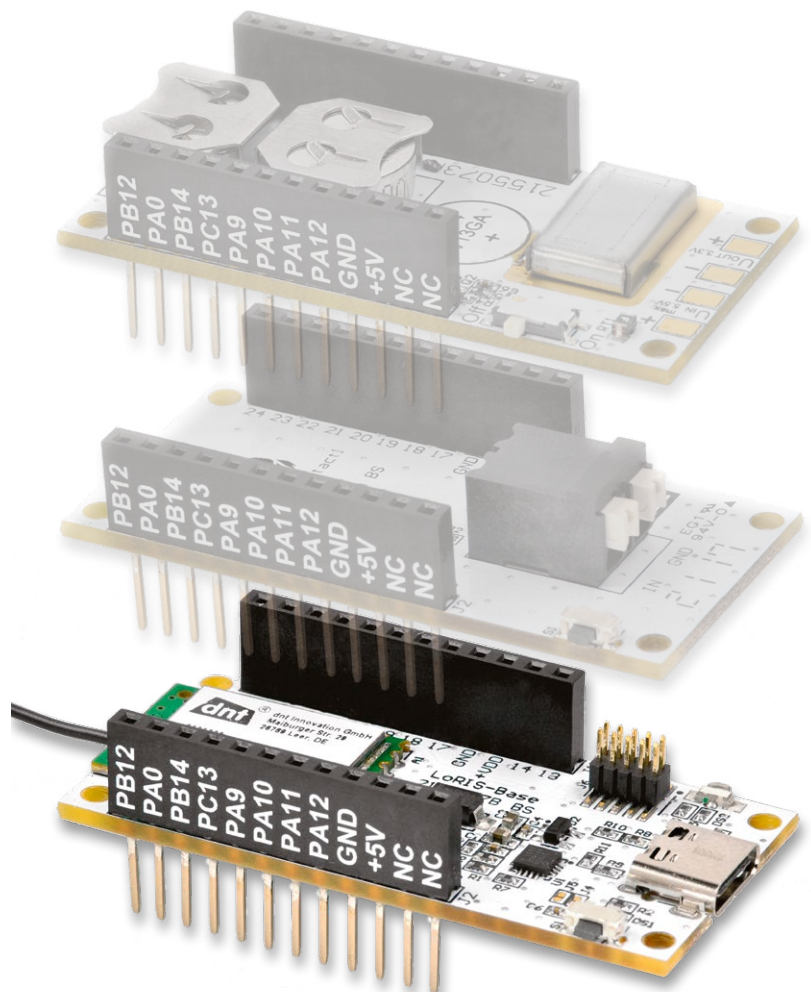


Bild 2: Die LoRIS-Base (unten) dient als Basis für zukünftige Anwendungs- und Powermodule.

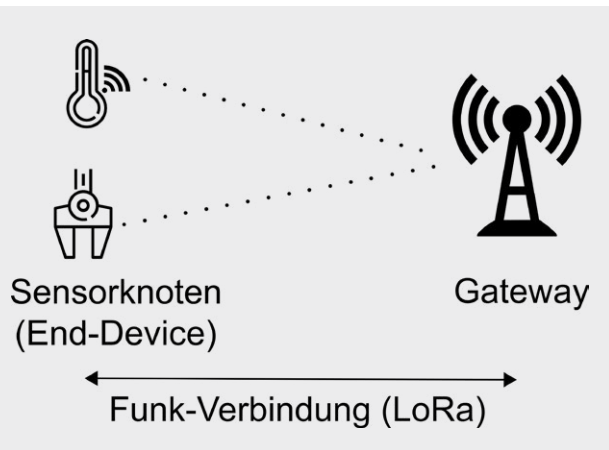


Bild 3: Der Sensorknoten benötigt zum Senden und Empfang der Daten per Funk ein LoRaWAN-Gateway als Gegenstelle – Quelle (verändert): Von Elutz – Eigenes Werk, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=85705192>

Städte oder Stadtteile kann man so per Funk abdecken. LoRa ist dabei die physikalische Schicht, die für die eigentliche Funkkommunikation zuständig ist. Sie stellt ein sehr energiesparendes Send- und Empfangsverfahren zur Verfügung. Im vergangenen ELVjournal haben wir uns die Technologie dazu genau angeschaut [3]. Das LoRIS-Base-Experimentierboard, genauer gesagt das auf dem Breakout-Board verwendete Funkmodul mit dem System-on-Chip (SoC) von STMicroelectronics, beherrscht diese Funktechnologie von Hause aus.

So ist eine besonders energiesparende, genau auf diese Technologie zugeschnittene Anwendung möglich. Im Sleep-Modus verbraucht die LoRIS-Base gerade einmal $2,5 \mu\text{A}$ bei 3 Volt (s. Technische Daten). Je nach Anwendung erreicht man damit Batterielaufzeiten von mehreren Jahren. Mithilfe von Energy Harvesting kann man sogar energieautarke Sensorknoten realisieren. Der Batteriewechsel wird dann überflüssig.

Gateway

Smartphones oder Laptops, die per Funk (WLAN) an das heimische Netzwerk angebunden sind, haben als Funk-Gegenstelle in der Regel einen WLAN Access Point (AP), über den sie in das Internet gelangen. Eine ähnliche Infrastruktur gibt es für LoRaWAN-Sensorknoten. Ihre Funk-Gegenstelle heißt in der LoRaWAN-Terminologie Gateway.

Das Gateway dient zum Empfang der Daten per Funk vom Sensorknoten (Uplink) bzw. in selteneren Fällen zum Senden von Daten (Downlink) an den LoRaWAN-Sensorknoten, der auch als End-Device bezeichnet wird (Bild 3).

Man benötigt also zwingend ein Gateway für die Funk-Kommunikation mit dem End-Device. Bei LoRaWAN gibt es zudem drei Klassen von End-Devices (A, B, C), die sich in ihren Send- und Empfangseigenschaften unterscheiden. Die energiesparendste Variante ist der Class-A-Sensorknoten. Die LoRIS-Base stellt genau einen solchen Class-A-Sensorknoten dar.

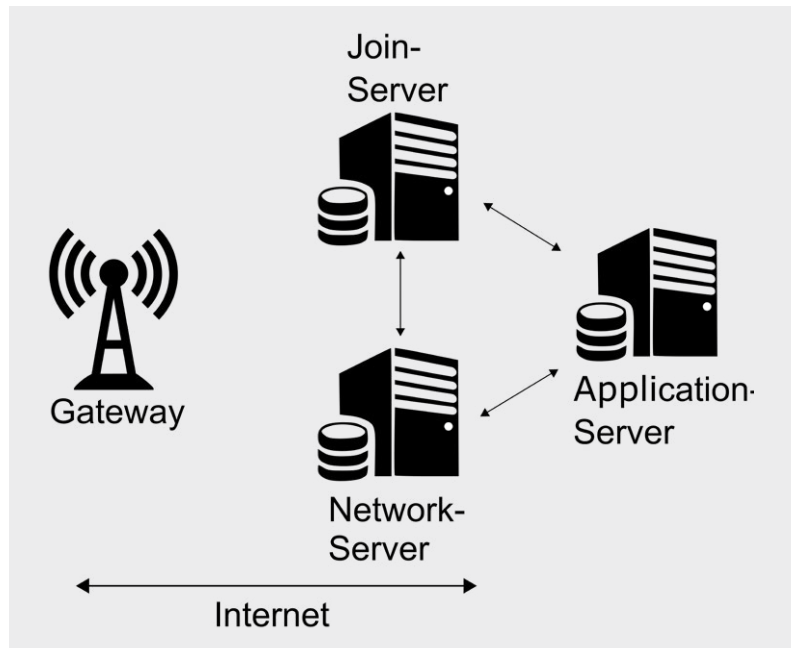


Bild 4: Netzwerkinfrastruktur bei LoRaWAN – Quelle (verändert): Von Elutz – Eigenes Werk, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=85705192>

Funk zum Netzwerk

Unser Temperatursensor im Garten aus dem obigen Beispiel kann damit die Daten sehr stromsparend per Funk (LoRa) an ein Gateway senden. Ähnlich wie beim oben erwähnten WLAN Access Point, der die per Funk vom Smartphone oder Laptop empfangenen Daten zunächst an den heimischen Router und von dort in das Internet sendet, geschieht dies bei LoRaWAN.

Hierzu kommt die zweite Schicht, die auf der Funkschicht (LoRa) aufsetzt, ins Spiel. Mit LoRaWAN gibt es ein Netzwerkprotokoll, das sich um die Organisation der Sensorknoten und deren Daten kümmert. Das Gateway ist dazu mit dem Internet verbunden, über die es eine LoRaWAN-Netzwerkinfrastruktur (Bild 4) ansprechen kann.

Join-Server, Network-Server und Application-Server

Im Wesentlichen besteht die Netzwerkinfrastruktur aus drei Server-Typen mit unterschiedlichen Aufgaben, die für die Authentifizierung, Verwaltung, Weiterleitung und Verarbeitung des Sensorknotens und die von ihm gesendeten Daten verantwortlich sind. Im Folgenden sind die Aufgaben der jeweiligen Server kurz beschrieben.

Join-Server

Der Join-Server behandelt den vertraulichen Teil der Kommunikation, der zur Aktivierung eines Sensorknotens im LoRaWAN notwendig ist. Er authentifiziert außerdem die Netzwerk- und Application-Server, speichert die grundlegenden Schlüssel und generiert benötigte Sitzungsschlüssel.

Network-Server

Der Netzwerk-Server ist für die Verwaltung des Netzwerks zuständig. Er überprüft beispielsweise die Adresse des Sensorknotens, quittiert Nachrichten oder passt Datenraten an und leitet Join-request- und Join-accept-Nachrichten (s. u.) zwischen den Geräten und dem Join-Server weiter.

Er steuert damit dynamisch die Netzwerkparameter und stellt sichere 128-Bit-AES-Verbindungen her. Außerdem überprüft er die Authentizität jedes End-Devices im Netzwerk und die Integrität jeder Nachricht.

Der Netzwerk-Server ist auch für die Weiterleitung der vom Gateway empfangenen Daten eines End-Devices an den Anwendungs-Server zuständig bzw. für die Weiterleitung von Downlink-Daten von einem Anwendungs-Server zurück zum Sensorknoten.



Application-Server

Der Anwendungs-Server ist für die sichere Handhabung, Verwaltung und Verarbeitung der Daten eines End-Devices zuständig. Von ihm werden die Daten über sogenannte „Integrations“ an Endpunkte (Server, Webseite etc.) weitergeleitet.

Anmeldeinformationen

Wir nehmen wieder unser Beispiel mit dem LoRaWAN-Sensorknoten zur Übermittlung der Temperatur aus dem Garten auf, um den Weg der Daten (Payload) vom Sensorknoten bis zum Endpunkt, an dem wir die Daten empfangen wollen, darzustellen.

Damit der Sensorknoten im LoRaWAN authentifiziert werden kann, die Daten sicher übertragen und zur Bearbeitung später an die richtige Stelle weitergeleitet werden, benötigt er zwingend zwei Identifikationsnummern und einen Schlüssel:

- **DevEUI** – eine eindeutige 64-Bit-Geräte-Adresse. Die Adressen werden von der IEEE (Institute of Electrical and Electronics Engineers, [4]) vergeben
- **JoinEUI** – dies ist die 64-Bit-Adresse, die dem Applikations-Server zugeordnet ist. Die JoinEUI ist ebenfalls eindeutig und wird von der IEEE bezogen. Sie identifiziert das Ziel der Nachrichten, die von den Sensorknoten gesendet werden. Jeder Applikations-Server bzw. eine Application hat seine eigene JoinEUI
- **Application Key (AppKey)** – Der AppKey ist 128 Bit lang und stellt den Verschlüsselungsschlüssel zwischen der Quelle der Nachricht (der DevEUI zugehörig) und dem Ziel der Nachricht (der JoinEUI zugehörig) dar. Dieser Schlüssel muss für jedes Gerät eindeutig sein

Für jede LoRaWAN-Base sind diese drei Werte bereits werkseitig festgelegt und einprogrammiert. DevEUI und JoinEUI werden von der IEEE bezogen, der AppKey dynamisch generiert.

Auf der LoRaWAN-Base ist die DevEUI auf dem Silkscreen der Leiterplattenrückseite eingepreßt, zudem werden die drei Werte der LoRaWAN-Base in Schriftform beigelegt. Mit dem LoRaWAN-Flasher-Tool für die LoRaWAN-Base können diese Werte außerdem später ausgelesen werden.

Auch wenn sich der Umgang mit der Netzwerkinfrastruktur, den IDs und dem Key zunächst vielleicht etwas kompliziert anhört, ein ähnlicher Vorgang geschieht ständig mit unseren Smartphones oder Laptops, wenn wir uns in WLAN-Netzwerke einloggen, ohne dass wir (heute) noch groß darüber nachdenken müssen.

Denn auch hier ist jedem Gerät eine eindeutige MAC-Adresse zugeordnet (vergleichbar beim LoRaWAN: DevEUI) und der Name des verwendeten WLAN-Netzwerks (SSID) sowie der Netzwerkschlüssel (ähnlich beim LoRaWAN: JoinEUI, AppKey), der zu dem Netzwerk gehört, müssen eingegeben werden. Die Daten werden ebenfalls mit Schlüsseln auf den weiteren Ebenen codiert.

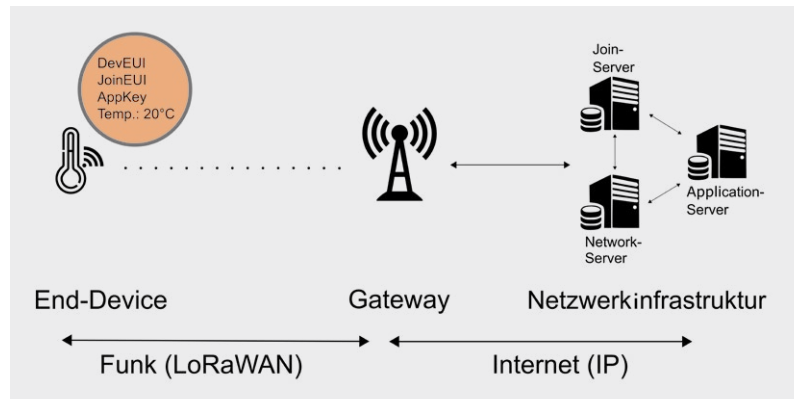


Bild 5: Lauf der Daten durch die LoRaWAN-Netzteilnehmer – Quelle (verändert): Von Elutz – Eigenes Werk, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=85705192>

Eigentlich ist das WLAN-System sogar komplizierter, denn wir müssen uns am Flughafen, im Hotel oder in anderen Gastnetzwerken bei bisher unbekanntem WLAN-Netzwerken neu anmelden. Dies ist bei LoRaWAN in Bezug auf die Gateways nicht der Fall, weil die Netzwerkinfrastruktur diese Aufgabe übernimmt.

Der Weg ist das Ziel

Senden wir nun mit den Anmeldeinformationen (DevEUI, JoinEUI, AppKey) von unserer LoRaWAN-Base oder allgemein unserem Sensorknoten Daten per LoRaWAN, dann empfängt prinzipiell jedes Gateway, das als LoRaWAN-Gegenstelle fungiert, diese Daten (Bild 5). Dazu müssen wir diese Gateways nicht speziell auswählen oder jedes Mal ein Passwort neu eingeben. Theoretisch könnte man sich also in einem großen räumlichen Bereich bewegen, ohne Anmeldeinformationen zu verändern.

Hier kommt leider ein kleiner Wermutstropfen dieser faszinierenden Funktechnologie: Mittlerweile gibt es zwar – vor allem in größeren Städten in Deutschland – eine teilweise Netzabdeckung mit LoRaWAN-Gateways bestimmter Betreiber von Netzwerkinfrastruktur. Allerdings ist diese gegenüber Ländern wie den Niederlanden oder der Schweiz, wo durch Telekomprovidern eine weitgehende Netzabdeckung schon realisiert wurde, noch recht dünn (s. a. FAQ/Material – Gateways).

Jedes LoRaWAN-Gateway wird intern einer LoRaWAN-Netzwerkinfrastruktur zugeordnet. Das kann ein freies, Community-basiertes System von Gateways, verbunden mit einem Betreiber von Netzwerkinfrastruktur wie das von TheThingsNetwork (TTN) [4] sein. Man kann aber auch eine eigene (private) Netzwerkinfrastruktur betreiben, bspw. für ein privates LoRaWAN zu Hause oder ein Unternehmen/eine Kommune etc.

In unserem Praxisbeispiel weiter unten nutzen wir die kostenlose und frei zugängliche Netzwerkinfrastruktur von TTN bzw. die des Nachfolgers TheThingsStack (TTS), da diese in Deutschland bisher am weitesten verbreitet ist und hier die meisten Gateways zur Verfügung stehen.

Beitritt zum Netzwerk

Hat ein LoRaWAN-Gateway nun unsere Daten bestehend aus den Anmeldeinformationen (DevEUI, JoinEUI) empfangen und eine Anforderung zum Beitritt (Join-request) von dem Sensorknoten erhalten, sucht es seinerseits bei den oben beschriebenen LoRaWAN-Servern, mit dem dieses Gateway verknüpft ist, nach einem entsprechenden Eintrag. Hierzu gibt es weiter unten im Praxisbeispiel eine genaue Anleitung anhand des Netzwerkinfrastrukturbetreibers TTN, wo und wie diese Daten einzutragen sind. Ist der Sensorknoten in dem entsprechenden LoRaWAN (z. B. TTN/TTS) bekannt, so wird der Join-request positiv beantwortet (Join-accept, s. Bild 6).

Das Gateway sieht dabei zwar bestimmte Metadaten, aber nicht die eigentliche Nachricht, wenn es von Sensorknoten zur Weiterleitung der Daten genutzt wird, denn die ist schon vom End-Device verschlüsselt worden (Bild 7).



Type	Time	Gateway ID	Message Type	Dev Addr	Freq (MHz)	Modulation	Bandwidth	Datarate	Coderate	CRC Status	RSSI (dB)	SNR (dB)
Rx	May/14/2021 14:17:53	3133300736083400	Confirmed Data Up	E5 C7 F7 05	868.300	LoRa	250 kHz	SF 7	4/6	Error	-110.00	-11.75
Rx	May/14/2021 14:17:35	3133300736083400	Confirmed Data Up	F2 6A 52 2A	868.300	LoRa	125 kHz	SF 7	4/7	Error	-110.00	-11.25
Ix	May/14/2021 14:17:12	3133300736083400	Unconfirmed Data Down	26 0B 03 86	868.100	LoRa	125 kHz	SF 12	4/5			
Rx	May/14/2021 14:17:12	3133300736083400	Unconfirmed Data Up	26 0B 03 86	868.100	LoRa	125 kHz	SF 12	4/5	Ok	-57.00	11.50
Tx	May/14/2021 14:17:00	3133300736083400	Join-accept	BC 93 84 91	868.100	LoRa	125 kHz	SF 12	4/5			
Rx	May/14/2021 14:16:58	3133300736083400	Join-request		868.100	LoRa	125 kHz	SF 12	4/5	Ok	-65.00	10.50

Bild 6: So sieht eine Kontaktaufnahme (Join-request) aus der Sicht des Gateways aus: Zunächst sendet der Sensorknoten eine Anfrage zum Beitritt in das Netzwerk (Join-request). Ist er im Netzwerk, mit dem das Gateway verbunden ist, bekannt, wird diese akzeptiert (Join-accept) und er erhält eine (verkürzte) 32-Bit-Session-Adresse. In den Einträgen darüber sieht man ein- und ausgehende Daten von anderen LoRaWAN-Knoten.

Durch die JoinEUI wird dem Sensorknoten der entsprechende Anwendungs-Server zugeordnet. Hier kommt (nach dem Join-Vorgang) unsere eigentliche Payload – die Temperatur – die durch das End-Device übermittelt wird, an. Zudem findet die Entschlüsselung der mit dem Verschlüsselungsverfahren AES (128 Bit) auf dem bisherigen Weg codierten Nachricht statt.

Type	Rx
Time	May/14/2021 14:16:58
Gateway ID	3133300736083400
Message Type	Join-request
Major Version	LoRaWAN R1
Join EUI	70 B3 00 00 00 00 00 00
Dev EUI	00 24 1F 00 00 00 00 00
Dev Nonce	39252
Freq (MHz)	868.100
Modulation	LoRa
Bandwidth	125 kHz
Datarate	SF 12
Coderate	4/5
IF Chain	0
CRC Status	Ok
Counter (us)	1093842174
RF Chain	1
RSSI (dB)	-65.00
SNR (dB)	10.50
SNR Min (dB)	7.75
SNR Max (dB)	13.25
CRC	8901
Size	23
Payload	ADkZBNB+1bNwudUGdaUXJABUme74ULM=

Bild 7: Detailsicht des Join-Requests. Das Gateway sieht nur die für die Einordnung in das LoRaWAN notwendigen DevEUI und JoinEUI. Die Payload ist verschlüsselt und kann vom Gateway nicht gelesen werden.

Verarbeitung und Weiterleitung der Daten

Auf dem Anwendungs-Server haben wir nun verschiedene Optionen, wie wir mit den empfangenen (Temperatur-)Daten umgehen. Beispielsweise kann man in der Netzwerkinfrastruktur von TTN/TTS die gewünschten Daten aus der Payload zerlegen (parsen), um an den Wert für die Temperatur zu kommen. Mit unserer bereits auf der LoRa-Base vorhandenen Firmware und dem dazugehörigen JavaScript-Beispielcode für den Anwendungs-Server können wir für einen ersten Einstieg in LoRaWAN zwei Taster ansteuern (Uplink) und außerdem einen Output schalten (Downlink). Dazu nutzen wir im Praxisbeispiel JavaScript auf dem Anwendungs-Server, um die Daten zu zerlegen. Analog könnte man dies später mit Daten anderer Art (Temperatur, GPS-Navigationsdaten ...) machen. Für unsere Anwendungsmodule, die wir in den kommenden Ausgaben des ELVjournals vorstellen, werden wir diese sogenannte Decoder-Software jeweils bereitstellen.

Ist man in die Netzwerkstruktur z. B. bei TTS eingeloggt, kann man dort die Daten aus der Payload bereits sehen, also beispielsweise den zuletzt gesendeten Temperaturwert. Das ist natürlich nicht besonders komfortabel, da man die Daten möglicherweise in ein eigenes System z. B. auf einem Raspberry Pi integrieren und an eine Anwendung (z. B. Node-RED) oder Webseite weiterleiten möchte.

Datenverarbeitung

Für die Weiterleitung der Daten an Endpunkte (Server/Raspberry Pi, Webseiten ...) gibt es bei TTN/TTS sogenannte „Integrations“.

Zurzeit stehen hier mit:

- MQTT
- Webhooks
- Storage Integration
- AWS IoT
- LoRa Cloud

verschiedene Möglichkeiten bereit, Daten vom Anwendungs-Server an einen Endpunkt weiterzusenden.

Wir benutzen weiter unten in unserem Beispiel MQTT – ein universelles Protokoll für den Datenaustausch –, dem wir in diesem ELVjournal ab Seite 26 einen separaten Einsteigerbeitrag mit Praxisbeispiel widmen. TTN/TTS stellt einen eigenen MQTT-Server bereit, von dem wir die Daten über das Internet mit einem Client abrufen können.

Zum Testen kann man die Daten mit einer Software wie in dem Beitrag zu MQTT in diesem ELVjournal beschrieben vom Anwendungs-Server abholen.

Schaltungsbeschreibung

Bevor wir in unser Praxisbeispiel einsteigen, schauen wir uns zunächst die Schaltung auf der LoRa-Base genauer an (Bild 8).



Spannungsversorgung

Die Spannungsversorgung der LoRIS-Base kann wahlweise über USB (3.1 Typ C/5V) oder die Buchsenleiste J1_15 +VDD (3,0–3,3 V) erfolgen.

Auf dem Silkscreen ist +VDD dabei auf der Oberseite und auch auf der Buchsenleiste entsprechend mit +VDD gekennzeichnet und auf der Unterseite der Platine zusätzlich mit der Pin-Nummerierung (15:+VDD) versehen.

Die Schottky-Diode D1 am Ausgang des internen Spannungsreglers des USB-UART-Converters (U1) dient der Entkopplung der Betriebsspannung. Die Kondensatoren C3 und C4 blockieren Störimpulse und puffern die Versorgungsspannung.

Am SoC (A1) erfüllen die Kondensatoren C7, C8 und C9 und am USB-UART-Converter am Eingang VREGION C1 und C2 die gleichen Aufgaben.

Hinweis: Achten Sie darauf, nicht gleichzeitig Versorgungsspannung über USB und +VDD zuzuführen.

USB 3.1 Typ C

Am USB-3.1-Typ-C-Eingang (J3) ermöglichen R9 und R11 an CC1 und R8 und R10 an CC2 die korrekte USB-Geräte-Erkennung.

Die Suppressordiode (Transient Voltage Suppressor, TVS) D2 dient in der Schaltung zur Unterdrückung von ESD (electrostatic discharge) und anderen transienten Überspannungsereignissen.

Auch C5 ist diesem Bereich (EMV/ESD, Störaussendung) zuzuordnen.

USB-UART-Converter

Am Eingang VBUS des USB-UART-Converters (U1) wird über den Spannungsteiler R3 und R6 das Anliegen der Versorgungsspannung über USB detektiert.

R1 dient als Pull-up-Widerstand von RST gegen VDD – RST wird während des Einschaltens und bei einem Stromausfall zurückgesetzt (Active Low).

GPIO.2 und GPIO.3 sind über die Lötbrücken J4 an J2_2, PA0 und S1 sowie J5 an NRST von U1 und an J6_10 geführt.

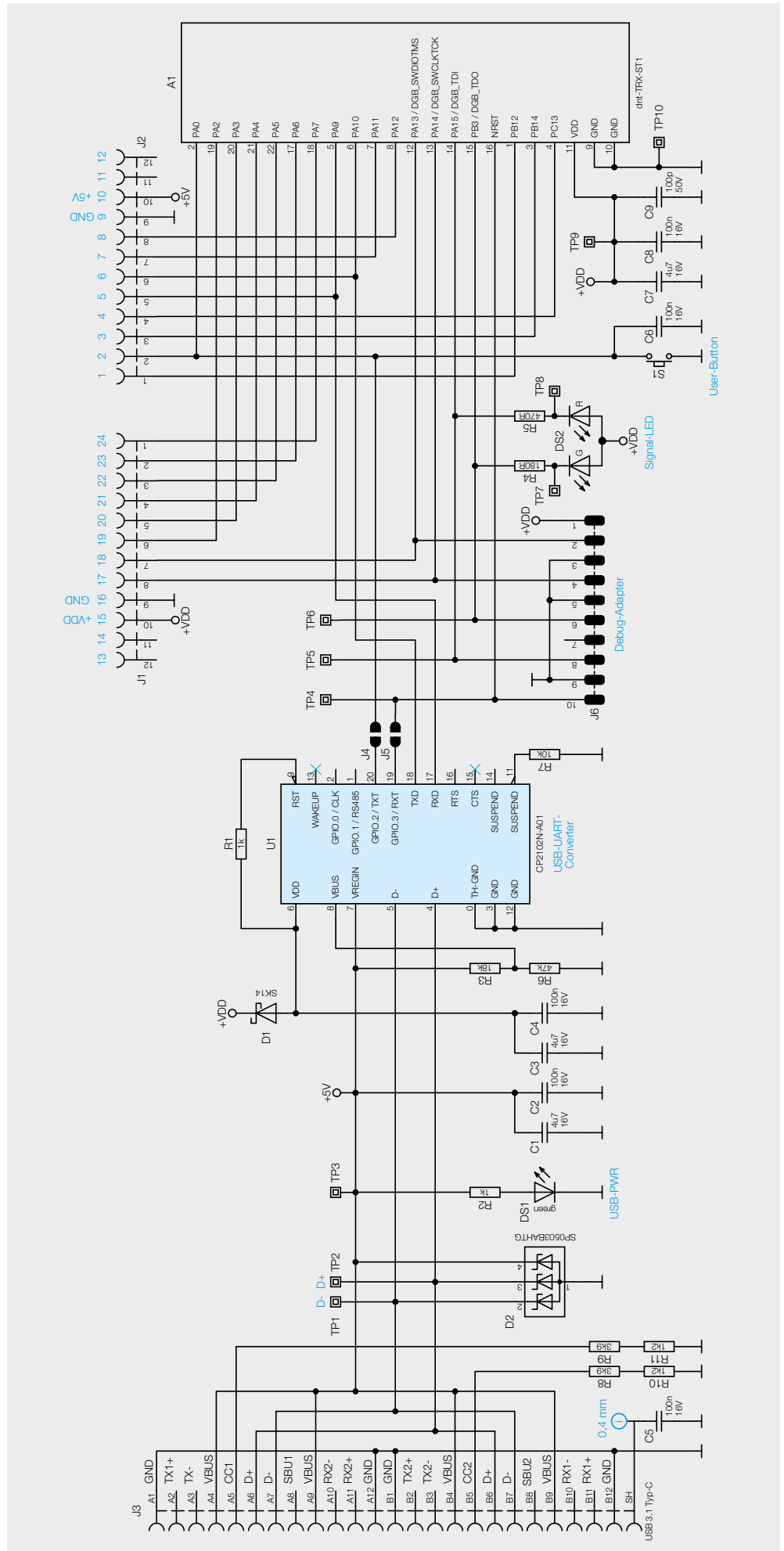


Bild 8: Schaltbild der LoRIS-Base LoRIS-BM-TRX1

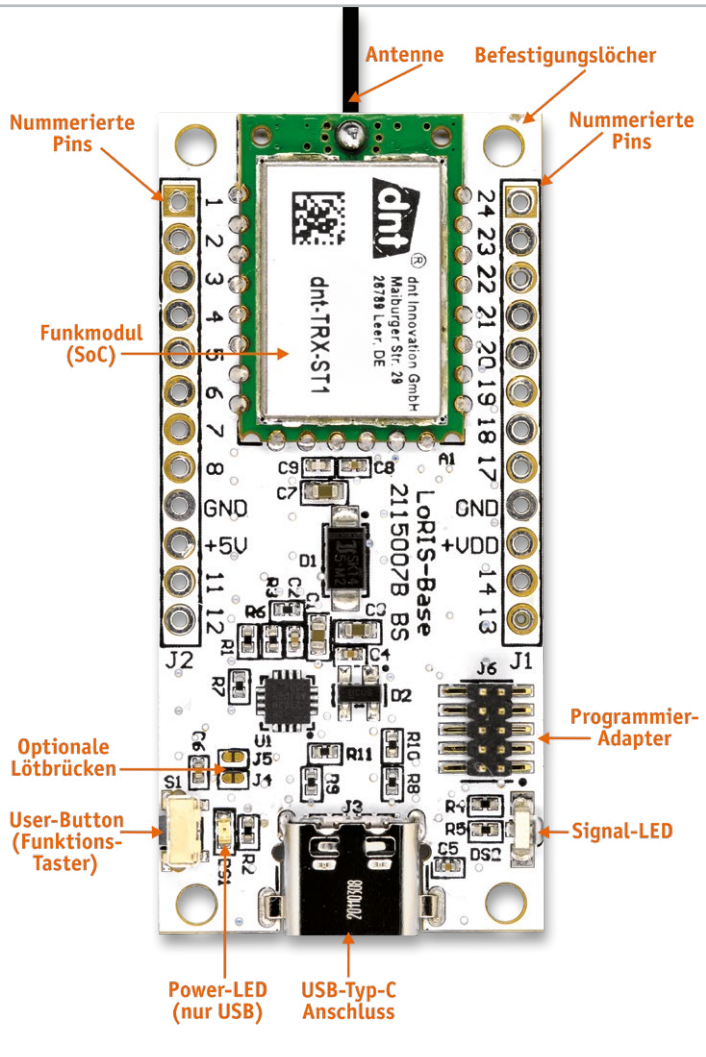


Bild 9: Platinenoberseite der LoRIS-Base (zur besseren Übersichtlichkeit ohne Buchsenleisten, die beim Bausatz bereits eingelötet sind)

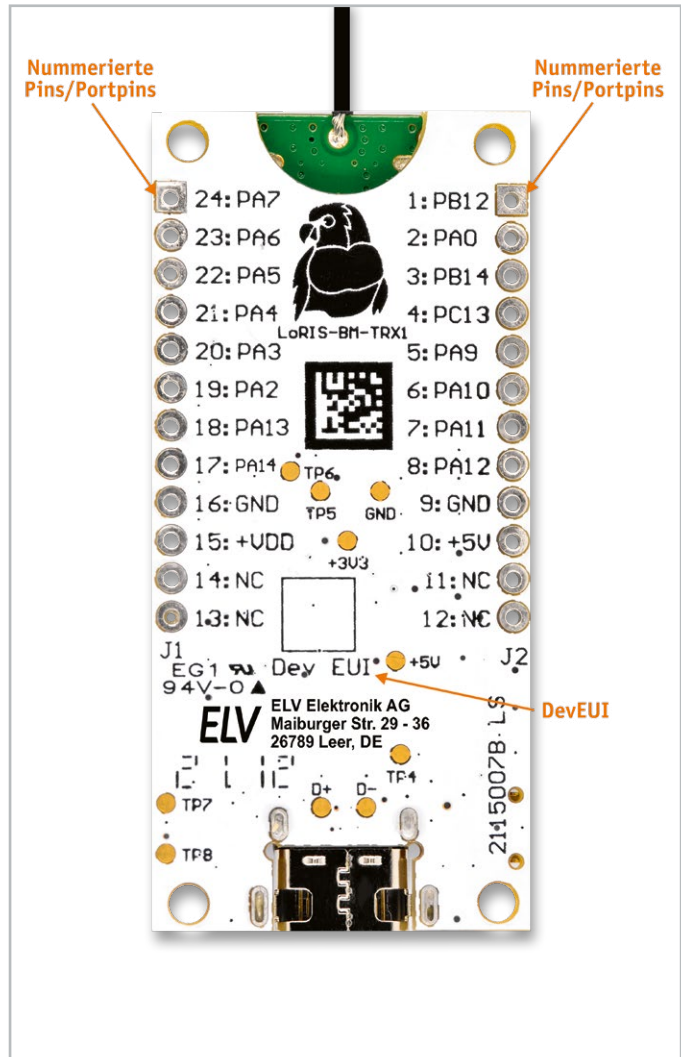


Bild 10: Platinenunterseite der LoRIS-Base (zur besseren Übersichtlichkeit ohne Buchsenleisten, die beim Bausatz bereits eingelötet sind)

Debug-Adapter

Der Debug-Adapter J6 ermöglicht die Programmierung des SoCs (A1) mit eigener Firmware unter Zuhilfenahme von externen Tools (bspw. mit einem Programmierer wie ST-LINK/V2 und eine entsprechende Entwicklungsumgebung wie STM32CubeIDE).

Signal-LED

Die Duo-LED DS2 dient der Visualisierung der verschiedenen Betriebszustände (Join-Vorgang, senden von Daten ...). Dabei liegt die grüne LED an PB3 und die rote LED an PA15 von A1. R4 und R5 dienen der Strombegrenzung der Duo-LED.

User-Button (S1)

Der Taster S1 ist über PA0 an dem SoC angeschlossen und kann zu User-Aktionen (u.a. manuelles Auslösen des Join-Vorgangs, Versetzen in den Programmiermodus über USB) verwendet werden. C6 dient zum Entprellen von S1.

Funkmodul dnt-TRX-ST1

Das Funkmodul dnt-TRX-ST1 (SoC, A1) stellt die LoRaWAN-Funktionalität zur Verfügung und hält im Flash-Speicher den entsprechenden Applikations-Code vor. Im Code-Beispiel für die LoRIS-Base (s. Praxisbeispiel) sind in diesem Fall zwei Inputs über PC13 (INPUT_1) und PA4 (INPUT_2) per internem Pull-up beschaltet und erkennen einen Interrupt über eine fallende Flanke. An PA5 (Default LOW) ist im Code-Beispiel ein Output (OUTPUT_1)

vorgesehen. Mit dieser Applikation können maximal 5 mA an Strom geliefert werden, um beispielsweise eine Low-Current-LED zu versorgen.

Eigenschaften von LoRIS-Base

Schauen wir nun noch anhand der Platine die Eigenschaften der LoRIS-Base (Bild 9 und Bild 10) an. Zur besseren Übersichtlichkeit zeigen wir die Platinenseiten mit ausgelöteten Buchsenleisten, die bei dem Bausatz im Lieferzustand schon bestückt sind. Es muss also nichts mehr gelötet werden.

Im oberen Bereich der LoRIS-Base befindet sich mit dem Funkmodul (SoC) das Herzstück des Experimentierboards. Hier liegen sowohl der Bootloader als auch die Anwendung (Firmware) im Speicher des STM32-SoCs. Am oberen Rand des Funkmoduls befindet sich die Antenne.

Links und rechts befinden sich die auf der Oberseite des Silkscreens durchnummerierten Pins. Diese sind auf den Buchsenleisten außen zusätzlich mit der Bezeichnung der Portpins versehen. Auf der Unterseite der Platine findet man zudem eine Kombination aus Pin-Nummer und Portpin auf dem Silkscreen.

Links unten befindet sich der User-Button, der verschiedene Funktionen hat. Man kann beispielsweise damit den Join-Vorgang manuell auslösen oder beim Programmieren mit dem LoRIS-Flasher-Tool den Programmiermodus beim Aufspielen der Firmware über USB auslösen.



Die Power-LED leuchtet, sobald die Versorgungsspannung über USB (Typ C) zugeführt wird. Bei Spannungsversorgung über +VDD bleibt sie ausgeschaltet, um einen besonders energiesparenden Modus zu ermöglichen.

Die Signal-LED zeigt die Betriebszustände der LoRIS-Base an. So signalisiert sie z. B. den Join-Vorgang (oranges Blinken), erfolgreiches (einmaliges grünes Leuchten) oder nicht erfolgreiches (einmaliges rotes Leuchten) Beitreten in ein LoRaWAN.

Der 10-polige Programmieradapter bietet die Möglichkeit, per Adapter und Programmer und einer geeigneten Entwicklungsumgebung (z. B. STM32CubeIDE) einen eigenen Code aufzuspielen. Optionale Lötbrücken sind für eine zukünftige Ansteuerung u. a. für einen automatischen Reset und Bootmodus vorgesehen. Zur Befestigung gibt es an allen vier Ecken Bohrlöcher mit einem Durchmesser von 2,5 mm.

Auf der Unterseite finden sich die nummerierten Pins bzw. die Funktionsbeschreibung und die entsprechenden Portpins. Außerdem ist das LoRIS-System am dem Logo mit dem LoRIS-Papagei erkennbar.

Die DevEUI ist im unteren Teil auf dem Silkscreen abgedruckt. DevEUI, JoinEUI und AppKey liegen dem Bausatz in Schriftform bei und lassen sich zusätzlich mit dem LoRIS-Flasher-Tool auslesen.

Praxisbeispiel

Wir haben bisher viel über den eigentlichen LoRa-Funk, Gateways, die Netzwerkstruktur und den Bausatz LoRIS-Base erfahren. Das Ganze wirkt auf den ersten Blick recht komplex. Für einen Einstieg in diese neue Funktechnologie muss man sich aber nur einmal die Zeit nehmen, die Eigenschaften eines Sensorknotens zu ergründen und die Integration in eine Netzwerkstruktur komplett durchzuspielen. Man merkt dann recht schnell, dass sich viele Vorgänge wiederholen und sich das vermeintliche Zahlenwirrwarr aus EUIs und Keys recht schnell entflechtet. Zudem ist unser LoRIS-System darauf ausgelegt, dank vorgegebener Firmware und Code-Templates die Anwendung der LoRIS-Base und der Anwendungsmodule so einfach wie möglich zu machen.

Nun zu unserer Schritt-für-Schritt-Anleitung zur Inbetriebnahme der LoRIS-Base und der Integration in eine Netzwerkstruktur.

Schritt 1

Hardware: LoRIS-Base und Gateway

Als Hardware benötigen wir für erste Experimente mit LoRaWAN zunächst die LoRIS-Base als Sensorknoten und ein in der Nähe liegendes LoRaWAN-Gateway, das mit dem TheThingsNetwork (TTN) oder TheThingStack (TTS) verbunden ist. Da TTN voraussichtlich Ende 2021 abgeschaltet wird, nutzen wir für unser Praxisbeispiel gleich TTS.

Ob ein Gateway in der Nähe ist, das mit TTN/TTS verbunden ist, erfährt man unter

<https://www.thethingsnetwork.org> (auf der Webseite etwas nach unten scrollen). Ist kein Gateway in der Nähe, empfehlen wir unter Material Gateways (Indoor/Outdoor), die wir getestet haben.

Schritt 2

Konto bei TTN erstellen

Neben der Hardware benötigen wir noch einen Betreiber für die Netzwerkinfrastruktur. Wir nutzen dazu TheThingsStack (TTS [6], s. a. FAQ). Dazu gehen wir mit einem Browser zunächst auf:

<https://account.thethingsnetwork.org/register> um ein Konto anzulegen (Bild 11). Das neu angelegte Konto muss danach in der Registrierungsmail bestätigt und damit das Konto aktiviert werden.

Wichtiger Hinweis:

Da die Migration der Netzwerkinfrastruktur von TheThingsNetwork zu TheThingsStack zurzeit (Stand: Juli 2021) durchgeführt wird, kann der im Praxisbeispiel beschriebene Vorgang zu einem späteren Zeitpunkt unter Umständen abweichen. Auch andere Details können sich mit der Zeit ändern. Eine aktuelle Version der Schritt-für-Schritt-Anleitung verlinken wir auf der Artikel-Detailseite zur LoRIS-Base im ELVshop.

Bild 11: Anlegen eines TTN-Accounts

Schritt 3

Anmelden bei TTS

Wir melden uns mit diesem Konto allerdings nicht im TheThingsNetwork (TTN) an, da diese Netzwerkinfrastruktur noch in diesem Jahr (2021) abgestellt wird, und nutzen den Nachfolger TheThingsStack, den wir unter:

<https://eu1.cloud.thethings.network/console/> erreichen. Hier loggen wir uns mit unseren TTN-Anmeldedaten unter „Login with The Things ID ein“ ein (Bild 12).

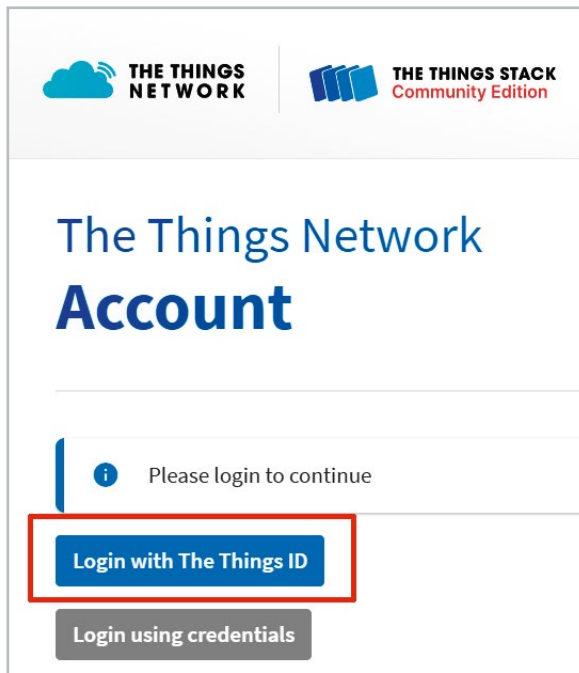


Bild 12: Anmelden bei TheThingsStack

Wir geben im folgenden Fenster unsere Anmelde-daten ein und gelangen dann auf die Übersichtsseite, auf der wir im unteren Bereich den Status der Netzwerkserver sehen. Hier finden wir u. a. die oben beschriebenen Join-, Network- und Application-Server wieder (Bild 13).

Schritt 4

Anwendung erstellen

Wir gehen in unserem Praxisbeispiel davon aus, dass wir ein in der Nähe erreichbares Gateway haben, ein eigenes (s. Material) könnten wir hier sonst konfigurieren. In einer der folgenden Ausgaben des ELVjournals werden wir noch detailliert auf Gateways, die Netzwerkinfrastruktur und deren Nutzung eingehen.

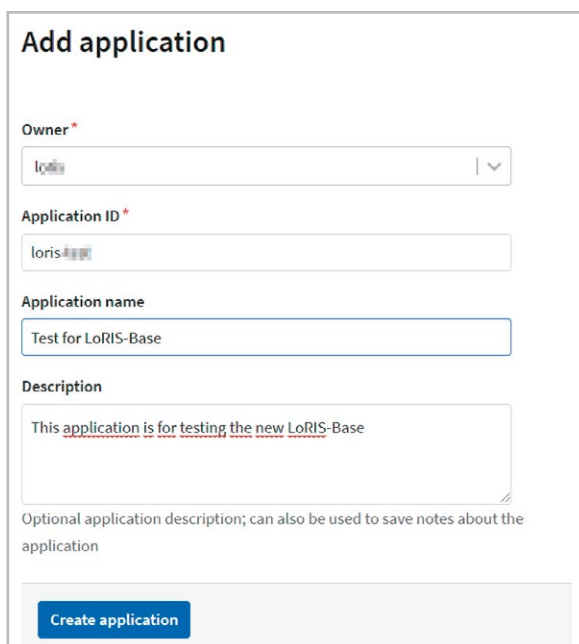


Bild 14: Eintragen der Anwendungsdaten

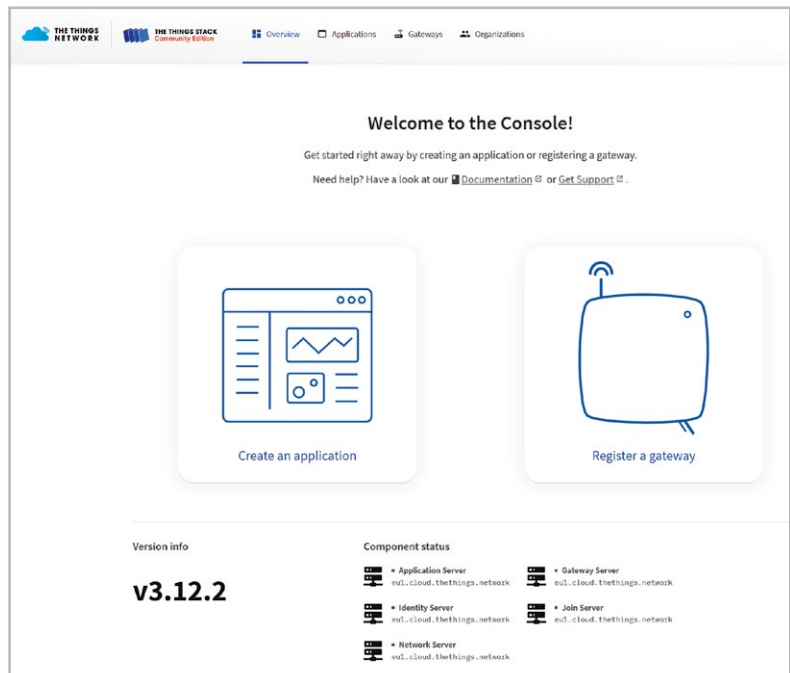


Bild 13: Übersichtsseite von TheThingsStack

Unter „Create an application“ können wir eine Anwendung für unseren Sensorknoten einrichten. Wir klicken auf den großen Button „Create an application“ und kommen zu einem weiteren Fenster, in dem wir die Anwendung zunächst definieren müssen (Bild 14).

Der Eintrag von Owner und Application ID ist dabei verpflichtend und die Application ID muss zudem eindeutig sein. Nach dem Eintragen der Daten klicken wir auf „Create application“.

Schritt 5

End-Device registrieren

Im folgenden Fenster gelangen wir auf die Übersichtsseite der Anwendung, auf der wir nun auch unser End-Device – die LoRIS-Base – eintragen können (Bild 15).

Wir klicken auf „+Add end device“ und wählen auf der folgenden Seite „try manual device registration“ (Bild 16).

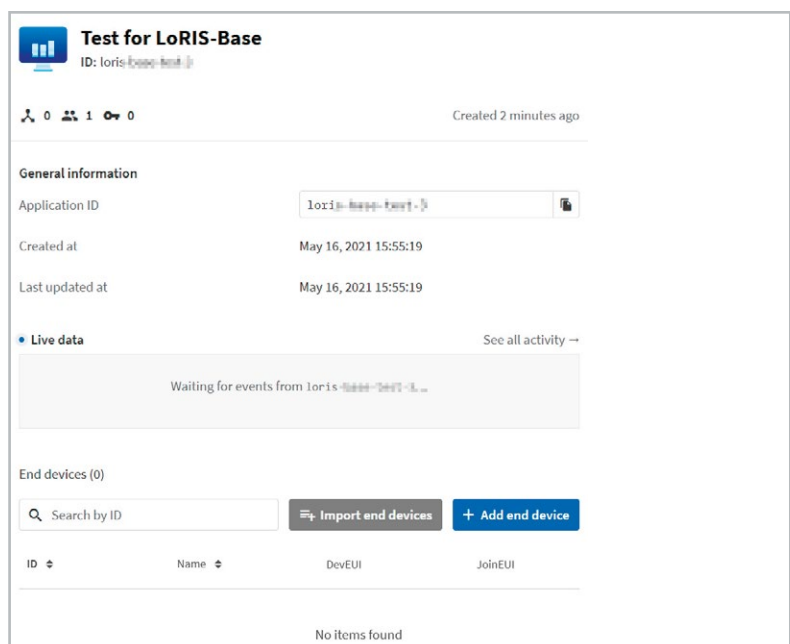


Bild 15: Übersichtsseite der Anwendung (Applications)



Bild 16: Manuelle Registrierung des Sensorknotens

Im folgenden Fenster wählen wir „Over the air activation (OTAA)“ und LoRaWAN version MAC V1.0.3 und lassen die anderen Einstellungen unverändert. Anschließend klicken wir auf Start (Bild 17).

Im folgenden Fenster müssen wir nun einen Namen für die End-Device-ID sowie die DevEUI und die JoinEUI eingeben (Bild 18). Früher hieß die JoinEUI AppEUI, beides ist aber gleichbedeutend, somit tragen wir bei AppEUI im TTS die uns vorliegende JoinEUI der LoRIS-Base ein.

Wir klicken als Nächstes auf „Network layer settings“ und müssen hier den Frequency Plan einstellen. Wir wählen „Europe 863–870 MHz (SF9 for RX2 – recommended)“, lassen den Rest unverändert und klicken auf „Join Settings“ (Bild 19).

Als letzten Wert tragen wir hier den AppKey ein, der der (initialen) Verschlüsselung im LoRaWAN dient (Bild 20).

Mit „Add end device“ schließen wir die Registrierung des Sensorknotens ab. Wir gelangen automatisch auf die Übersichtsseite des Sensorknotens (Bild 21).

Damit haben wir die Erstellung einer Anwendung sowie die Registrierung unseres Sensorknotens (LoRIS-Base) im TTS abgeschlossen. Wir lassen das Übersichtsfenster des End-Devices geöffnet, weil wir dort bei der Inbetriebnahme der LoRIS-Base im nächsten Schritt den Datenverkehr mit ansehen können.

Schritt 6

Vorbereitung der Hardware

Im Folgenden ein Beispiel für eine mögliche Schaltung der Hardware. Wir betreiben dabei die LoRIS-Base über die Spannungsversorgung per USB.

Als Erstes installieren wir den USB-Treiber, den wir im Downloadbereich des Bausatzes LoRIS-Base im ELVshop finden [1].

Die Firmware, die bereits auf der LoRIS-Base aufgespielt ist, ermöglicht die Betätigung von zwei Tastern, die einen Signal-Status (Active/Inactive) senden (Uplink zum Gateway). Zusätzlich wird die Versorgungsspannung und der Grund für die Übertragung übermittelt. Im Downlink (vom Gateway zur LoRIS-Base) lässt sich außerdem über einen Output beispielsweise eine externe LED an- und ausschalten. Allerdings findet die Übertragung von Nachrichten im Downlink bei Class-A-End-Devices erst nach einem Uplink statt. Es muss also in unserem speziellen Fall erst ein Taster gedrückt werden, um das Kommando zum An- oder Ausschalten der LED zu erhalten.

Bild 17: Eintragen des Aktivierungsmodus und der LoRaWAN-Version des End-Devices

Bild 18: Eingabe der ID für das End-Device, der DevEUI und JoinEUI (AppEUI)



Im Breadboardaufbau sieht die Schaltung folgendermaßen aus (Bild 22).

Wir haben für den Aufbau unsere Prototypenadapter-Module gewählt (s. Material), da so eine bessere Übersichtlichkeit der Schaltung auf dem Breadboard ermöglicht wird. Der Aufbau kann natürlich genauso gut mit konventionellen Bauteilen durchgeführt werden.

Die LoRIS-Base wird über den USB-Typ-C-Anschluss mit Spannung versorgt. GND wird von der LoRIS-Base an die Spannungsschiene geführt. Die LED wird direkt von einem GPIO (OUT1) des LoRaWAN Funkmoduls angesteuert. Des Weiteren wird die Kathode der LED mit der Spannungsschiene (GND) des Breadboard verbunden. Bei Spannungsversorgung über USB liefert der im USB-Seriell-Converter integrierte Spannungsregler

bis zu 100 mA Strom. Pin 21 bzw. PA4 wird an den einen Anschluss vom Taster, in unserem Fall IN2, angeschlossen. Der andere Anschluss dieses Tasters wird mit GND verbunden. Pin 4 bzw. PC13 wird mit dem zweiten Taster (IN1) an einem Anschluss verbunden und an dem anderen Anschluss der Taster ebenfalls mit GND verbunden.

Die LED mit entsprechendem Vorwiderstand wird an der Anode (+) mit Pin 22 bzw. PA5 verbunden und an der Kathode mit GND.

Register end device

From The LoRaWAN Device Repository **Manually**

1 Basic settings
End device ID's, Name and Description

2 **Network layer settings**
Frequency plan, regional parameters, end device class and session keys.

3 Join settings
Root keys, NetID and kek labels.

Frequency plan *

Europe 863-870 MHz (SF9 for RX2 - recommended)

LoRaWAN version *

MAC V1.0.3

Regional Parameters version *

PHY V1.0.3 REV A

LoRaWAN class capabilities ☹

Supports class B

Supports class C

Advanced settings ▾

< Basic settings **Join**

Bild 19: Wahl des Frequency Plans

Register end device

From The LoRaWAN Device Repository **Manually**

1 Basic settings
End device ID's, Name and Description

2 Network layer settings
Frequency plan, regional parameters, end device class and session keys.

3 **Join settings**
Root keys, NetID and kek labels.

Root keys

AppKey *

3C 61 64

Advanced settings ▾

< Network layer settings **Add end device**

Bild 20: Eingabe des AppKeys

loris-test-button
ID: loris-test-button

Last seen info unavailable ↑ n/a ↓ n/a Created 1 minute ago

Overview Live data Messaging Location Payload formatters

General information

End device ID: loris-test-button

Description: LoRIS base end device with button functionality

Created at: May 17, 2021 11:55:56

Activation information

AppEUI: 70 B3

DevEUI: 00 D3

Root key ID: n/a

AppKey:

NwkKey: n/a

Session information

No data available

• Live data See all activity →

Waiting for events from loris-test-button...

Bild 21: Übersichtsseite des End-Devices

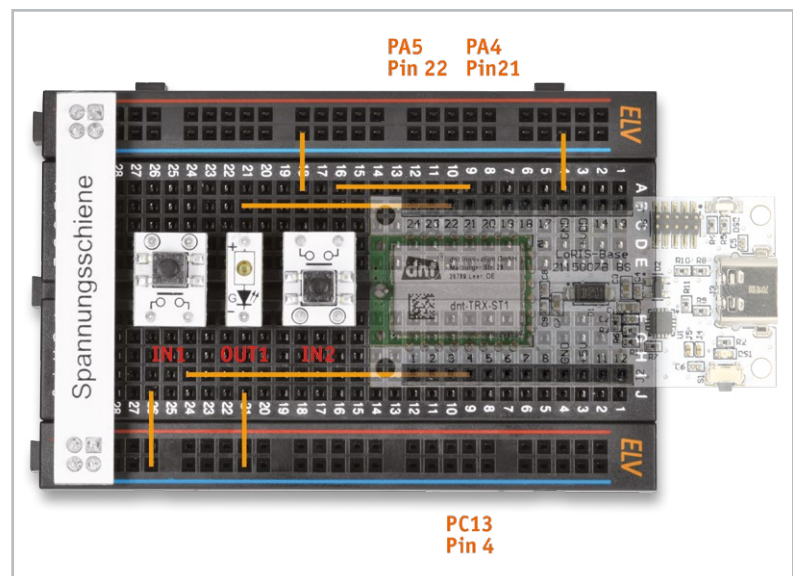


Bild 22: Schaltung der LoRIS-Base mit Tastern und LED



Schritt 7

Inbetriebnahme

Nachdem man die Schaltung noch mal hinsichtlich des Aufbaus kontrolliert hat, kann nun über USB Spannung zugeführt werden. Es sollte sofort die grüne Power-LED auf der LoRIS-Base aufleuchten (Bild 23). Ist dies nicht der Fall, bitte die Spannungsversorgung sofort abtrennen und die Schaltung erneut kontrollieren.

Join-Vorgang mit dem Gateway/TTS

Nach der Stromzufuhr sollte die Signal-LED orange blinken (Bild 24). Die Signal-LED zeigt den Verbindungsstatus mit dem LoRaWAN-Netzwerk (bzw. einem verfügbaren Gateway) an. Man kann den Join-Vorgang alternativ auch mit einem kurzen Druck auf den User-Button manuell auslösen.

Oranges Blinken bedeutet, dass versucht wird, eine Verbindung aufzubauen. Sobald die LoRIS-Base dem Netzwerk beigetreten ist (Join-accept), leuchtet die Status-LED kurz grün auf (Bild 25). Nach erfolgreichem Beitritt erlischt die Signal-LED. Ist der Join-Vorgang nicht erfolgreich, leuchtet die Signal-LED rot auf. Es gibt insgesamt drei Join-Versuche beim „Power on (Reset)“. Wir können diesen Vorgang im Übersichtsfenster des End-Devices bei TTS im Bereich Live Data mitverfolgen (Bild 26).

Hinweis: Im Betrieb kann man durch Druck auf den Taster unterhalb der Power-LED (< 5 Sekunden) manuell einen Join-Vorgang bzw. ein Status-Update auslösen. Ein Druck > 5 und < 8 Sekunden löst einen Soft-Reset aus (alles über 8 Sekunden wird ignoriert).

Hinweis: Um einen späteren stromsparenden Betrieb über +VDD und GND beispielsweise als abgesetzter Sensor zu ermöglichen, ist in dieser Konfiguration die Power-LED deaktiviert.

Datenverkehr im TTS

Wir testen nun einmal durch Drücken eines Tasters, welche Daten wir im Fenster „Live Data“ dabei erhalten. Wir gucken uns dabei unter „Receive Data Message“ den Raw Payload an (Bild 27). Da das Backend von TTS zur Zeit noch in der aktiven Entwicklung ist, kann sich die Ansicht zum Zeitpunkt des Erscheinens des ELVjournals geändert haben. Wichtig ist hier, eine ankommende Nachricht nach Druck auf einen Taster zu sehen.

Hinweis: Aktuelle Informationen zur LoRIS-Base, den Code-Beispielen und der Schritt-für-Schritt-Anleitung finden Sie auf der LoRIS-Base-Artikel-Detailseite im ELVshop.

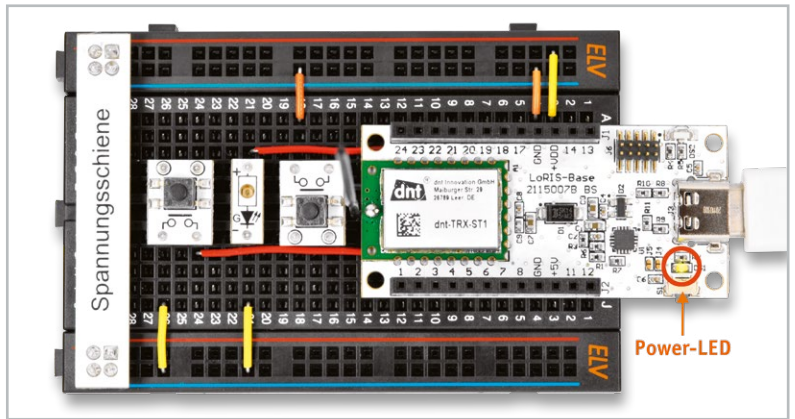


Bild 23: Nach Anschluss per USB sollte die Power-LED (grün) dauerhaft leuchten.

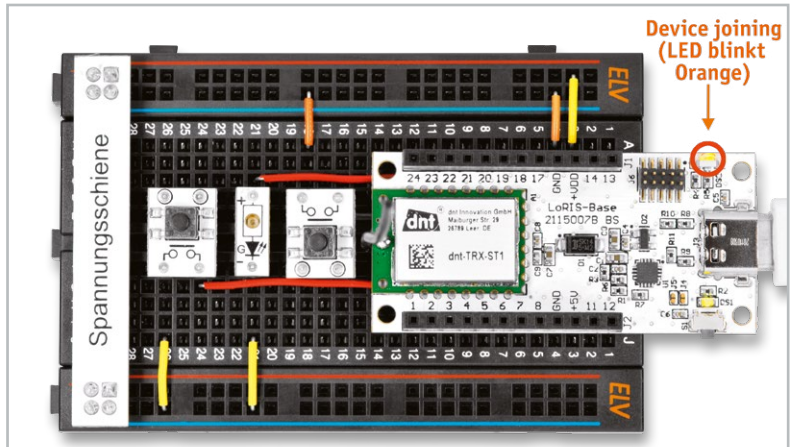


Bild 24: Oranges Blinken signalisiert den Versuch, einem LoRaWAN beizutreten.

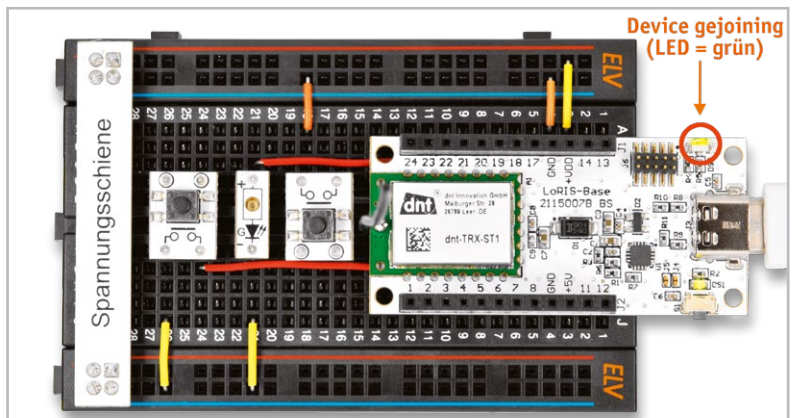


Bild 25: Ist der Beitritt in ein LoRaWAN erfolgreich, leuchtet die Signal-LED kurz grün auf.

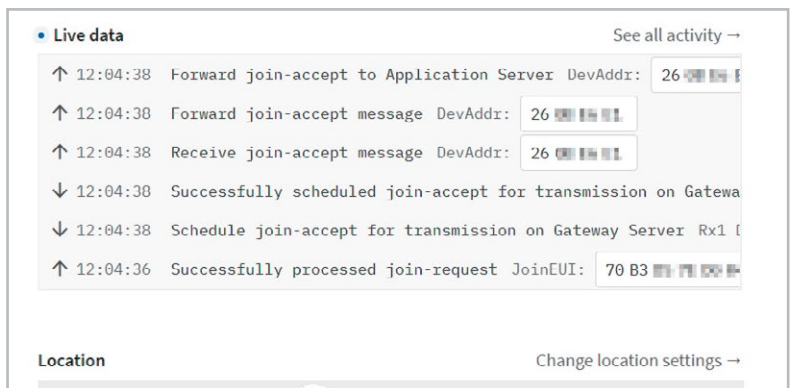


Bild 26: Join-Vorgang der LoRIS-Base im TTS

```
Bandwidth: 125000 SNR: 10.25 RSSI: -81 Raw payload: 40 E1 E6 0B 26 80 03 00 14 35 E3 03 65 80 7A C5 C6 67 01 05 7E 48 2A
```

Bild 27: Rohdaten der LoRIS-Base

**Schritt 8****Entschlüsselung der Rohdaten**

Mit diesen Rohdaten sehen wir nur, dass etwas gesendet wurde. Um die Payload zu analysieren und zu zerlegen, gibt es im TTS sogenannte Payload Formatters (Hilfsprogramme, die die Payload bearbeiten). Wir nutzen für unsere LoRIS-Base den folgenden Code, um die gewünschten Daten zu erhalten.

```
var tx_reason = [ "Timer_Event", "User_Button", "Input_Event" ];

function Decoder( bytes, port )
{
    var decoded = {};

    if( port === 10 )
    {
        if( bytes.length >= 5 )
        {
            decoded.TX_Reason = tx_reason[( bytes[0] & 0x0F )];
            decoded.Supply_Voltage = ( bytes[3] << 8 ) | bytes[4];

            if( bytes[5] === 0x00 )
            {
                if( bytes[6] & 0x01 )
                {
                    decoded.Input_1 = "Active";
                }
                else
                {
                    decoded.Input_1 = "Inactive";
                }
            }
            if( bytes[6] & 0x04 )
            {
                decoded.Input_2 = "Active";
            }
            else
            {
                decoded.Input_2 = "Inactive";
            }
        }
        if( bytes[7] === 0x01 )
        {
            if( bytes[8] & 0x01 )
            {
                decoded.Output_1 = "Active";
            }
            else
            {
                decoded.Output_1 = "Inactive";
            }
        }
    }
}

return decoded;
}
```

Wir gehen auf Payload formatters → Uplink, wählen als Formatter Type JavaScript und fügen unter „Formatter Parameter“ den obigen Code ein. Der Code ist auch im Download-Bereich des Bausatzes LoRIS-Base im ELVshop zu finden, um ihn einfach zu kopieren. Wir schließen den Vorgang mit „Save changes“ ab (Bild 28).

Jetzt gehen wir wieder auf Applications bzw. direkt in unser End-Device und schauen uns an, was im Fenster Live Data mit der Payload passiert, wenn wir erneut eine Taste drücken. In der Nachricht sollte dann die mithilfe des Payload Formatters entschlüsselte Payload zu sehen sein (Bild 29).

Wir sehen nun, dass der Input_1 (Taster) gedrückt wurde und die Status des anderen angeschlossenen Tasters bzw. des Outputs. Zudem werden die aktuelle Versorgungsspannung und der Grund für die Übertragung gesendet.

```
E1 Payload: { Input_1: "Active", Input_2: "Inactive", Output_1: "Inactive", Supply_Voltage: 3200, TX_Reason: "Input_Event" }
```

Bild 29: Entschlüsselte Payload

Sonderfall Downlink (Output, bspw. LED)

Die Funktion des Downlinks vom TTS zur LoRIS-Base kann man testen, indem man im End-Device → Messaging → Downlink FPort 10 auswählt und eine 01 02 (OUTPUT LOW, bspw. LED AUS) oder 01 03 (OUTPUT HIGH, bspw. LED AN) als Payload eingibt und auf „Schedule downlink“ klickt.

Die Nachricht wird dann in eine Warteschlange geschrieben, da Class-A-Nodes ein Empfangsfenster erst nach der Absendung von Daten (Uplink an ein Gateway/TTS) öffnen (Bild 30). Der Output (bspw. LED) wird also erst gesetzt, nachdem ein Taster gedrückt und damit ein Empfangsfenster geöffnet wurde.

Schritt 9**Integrationen**

Jetzt kommen wir zum letzten Schritt, bei dem wir die erhaltenen Daten an einen Endpunkt weiterleiten, um sie beispielsweise auf einem Server oder einer Webseite ansehen oder weiterverarbeiten zu können. Bei TTS heißt diese Form der Datenweiterleitung Integrations.

Wir nutzen als Integration MQTT – ein universelles Protokoll für den Datenaustausch. TTS stellt dazu einen MQTT-Server bereit. So müssen wir keinen eigenen MQTT-Server aufsetzen und können die Daten von überall aus dem Internet abrufen.

Wir gehen dazu in unserer Application auf Integrations → MQTT. Hier sind alle grundlegenden Verbindungsdaten zu dem TTS-MQTT-Server bereits eingetragen. Zur Absicherung der Verbindung müssen wir zunächst aber noch einen API Key generieren und klicken dazu auf „Generate new API Key“ (Bild 31). Unmittelbar darauf erscheint der Key, den man gleich kopieren und sicher verwahren sollte.

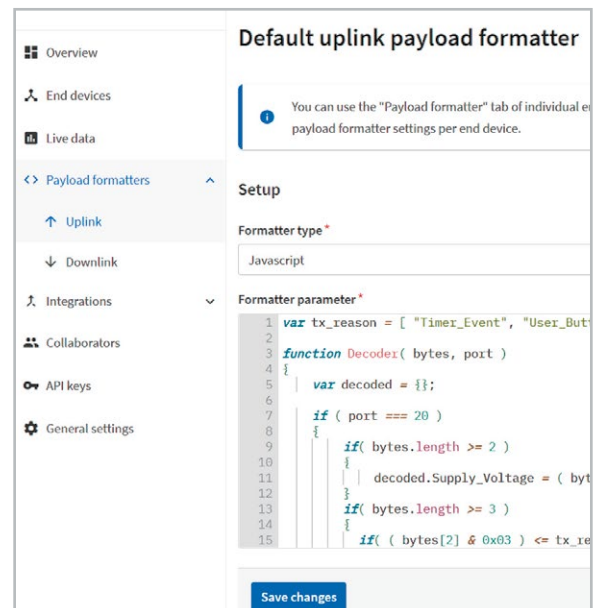


Bild 28: Einfügen des Payload-Formatter-Codes



Wie kann man nun auf diese Daten zugreifen? MQTT ist fast immer in sogenannter Broker-Software als Schnittstelle vorhanden, z. B. in Node-RED, mit dem man ein ganzes System aus Sensoren zusammenfassen und über ein Dashboard anzeigen kann. Zu Node-RED haben wir bereits zahlreiche Beiträge veröffentlicht [7] und auch die Integration von MQTT beschrieben. Im Download-Bereich des Bausatzes LoRIS-Base stellen wir ein Node-RED-Beispiel (Flow) zur Verfügung, um das MQTT-Beispiel auch in Node-RED zu nutzen.

Um den Abruf der Daten zu demonstrieren, benutzen wir den auch im MQTT-Beitrag in diesem ELVjournal ab Seite 26 verwendeten Client MQTT.fx. Details zur Handhabung der Software finden Sie dort.

Zunächst muss eine Connection mit folgenden Daten erstellt werden (Bild 32):

- Broker Address: eu1.cloud.thethings.network
- Broker Port: 1883
- User Name: xxx@ttn (das ist der Username aus den MQTT Credentials bei TTS, Bild 31)
- Password: xxxx (das ist der API Key aus den MQTT Credentials bei TTS)

Wie bei MQTT üblich muss man sich noch auf einen Topic subscriben. Dazu verbinden wir uns zunächst per MQTT.fx mit dem MQTT-Server von TTS, klicken auf Subscribe und tragen dort:

v3/username@ttn/devices/#

ein, wobei der Username wieder derjenige aus den MQTT-Credentials bei TTS ist.

Anschließend klicken wir auf Subscribe und Betätigen einen Taster, der an unserer LoRIS-Base angeschlossen ist. Nun sollten wir die Daten im rechten Fenster erhalten (Bild 33). Um die Ausgabe etwas lesbarer zu gestalten, können wir unten rechts noch „JSON Pretty Format Decoder“ auswählen.

In den kommenden Ausgaben des ELVjournals werden wir weitere Integrationen vorstellen und die Möglichkeiten der Datenweiterleitung im Detail beschreiben. Das Praxisbeispiel mit MQTT soll nur als erste Anschauung und Startpunkt für eigene Experimente dienen, allerdings lässt sich dieses universelle Protokoll sehr gut in andere Software-Plattformen integrieren.

Ausblick und Möglichkeiten des LoRIS-Systems

Auch wenn es zunächst viele Informationen gibt, die im Zusammenhang mit LoRaWAN erforscht und verarbeitet werden müssen, so lohnt es sich doch, diese spannende Funktechnologie und ihre vielfältigen Anwendungsmöglichkeiten zu erforschen. Mit der Zeit wird die Community weiter anwachsen und es wird viele Beispiele im Internet für Projekte geben.

Hinsichtlich der LoRIS-Base und den kommenden Anwendungs- und Powermodulen werden wir diesen offenen Ansatz unterstützen und für das jeweilige Anwendungsmodul die entsprechende Firmware als Hex-File zur Verfügung stellen. Diese lässt sich mit unserem LoRIS-Flasher-

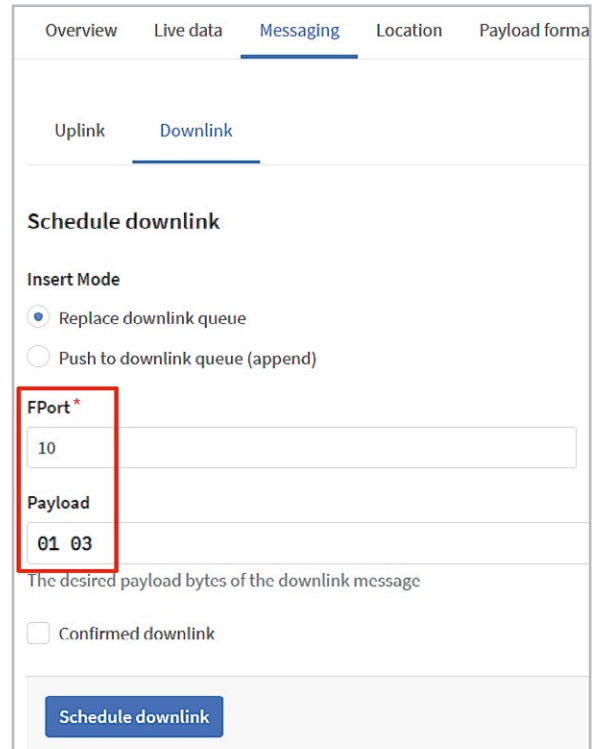


Bild 30: Test des Downlinks – dem Übertragen von Daten an die LoRIS-Base aus dem TTS heraus

Tool einfach per USB auf die LoRIS-Base aufspielen. Auch den jeweiligen Decoder für TTS (Payload Formatter) werden wir im Download-Bereich des Bausatzes LoRIS-Base im ELVshop bereitstellen.

Wer schon Programmiererfahrung hat und mit Programmier- und Entwicklungsumgebung umgehen kann, dem bietet sich die Chance zu eigenen Entwicklungen, die über die Programmierschnittstelle der LoRIS-Base auf das Board gespielt werden können. Dazu stellen wir ein Code-Template bereit, das einen Startpunkt für eigene Programme bietet.

Das ELVjournal wird das faszinierende Thema von LoRaWAN weiterverfolgen und schon in der nächsten Ausgabe ein Anwendungsmodul (universeller Tasterzugang) und ein Powermodul (LR44-Knopfzellen) veröffentlichen. Mit diesen drei Modulen kann dann schon ein autarker Sensorknoten im LoRaWAN realisiert werden. Außerdem werden wir die technischen Details der LoRIS-Base weiter beleuchten und uns Gateways genauer anschauen. **ELV**

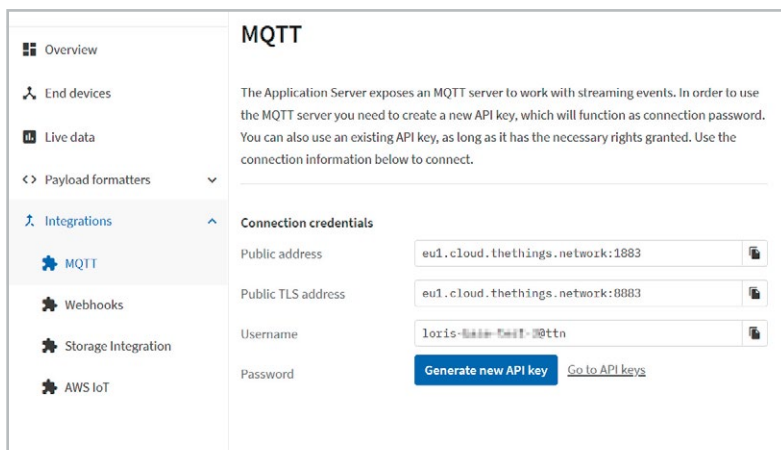


Bild 31: MQTT-Verbindungsdaten

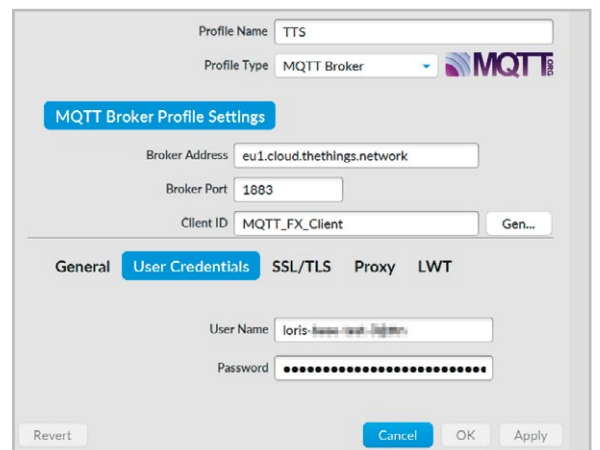


Bild 32: Erstellen einer Verbindung in MQTT.fx



The screenshot shows the MQTT.fx web interface. At the top, there are menu items 'File', 'Extras', and 'Help'. Below that, a search bar contains 'TTS' and buttons for 'Connect' and 'Disconnect'. The main interface has tabs for 'Publish', 'Subscribe', 'Scripts', 'Broker Status', and 'Log'. The 'Subscribe' tab is active, showing a subscription to 'v3/loris-...@ttn/devices/#'. A message is received from 'v3/loris-...@ttn/devices/loris-.../up' at '17-05-2021 15:01:52.54112925'. The message content is a JSON object with the following structure:

```

{
  "end_device_ids": {
    "device_id": "loris-...",
    "application_ids": {
      "application_id": "loris-..."
    }
  },
  "dev_eui": "00D3F0C4B0000000",
  "join_eui": "70B3000000000000",
  "dev_addr": "26...",
  "correlation_ids": [ "as:up:01F5X6P7...", "ns:uplink:01F5X...", "pba:uplink:01F5...", "rpc:/ttn.lorawan.v3.GsNs/HandleUplink:01F5X...", "rpc:/ttn.lorawan.v3.NsAs/HandleUplink:01F5X..." ],
  "received_at": "2021-05-17T13:01:51.165226309Z",
  "uplink_message": {
    "session_key_id": "AX15...",
    "f_port": 20,
    "f_cnt": 28,
    "frm_payload": "DIYGAAAAAAAAA==",
    "decoded_payload": {
      "Input_1": "Active",
      "Input_2": "Inactive",
      "Output_1": "Inactive",
      "Supply_Voltage": 3206,
      "TX_Reason": "Input_Event"
    }
  }
}

```

At the bottom right, it says 'Payload decoded by JSON Pretty Fomat Decoder'.

Bild 33: Ausgabe der Daten des Sensorknotens in MQTT.fx

Material

Gateway – Indoor: Dragino Indoor LoRaWAN Gateway LPS8-868: Artikel-Nr. 252153

Gateways – Outdoor: MikroTik LoRaWAN – wAP LoRa8 kit – https://mikrotik.com/product/wap_lr8_kit

Antenne: Antenna kit for LoRa – https://mikrotik.com/product/lor_a_antenna_kit

Passendes Zubehör zum Prototyping:

- ELV Steckplatine/Breadboard mit 830 Kontakten, schwarze ELV Version – Artikel-Nr. 250986
- ELV Steckplatine/Breadboard mit 400 Kontakten, schwarze ELV Version – Artikel-Nr. 251467
- ELV Bausatz Prototypenadapter PADx für Steckboards:
 - PAD1: Artikel-Nr. 153761
 - PAD2: Artikel-Nr. 154712
 - PAD3: Artikel-Nr. 154743
 - PAD4: Artikel-Nr. 155107
 - PAD6: Artikel-Nr. 155858
- ELV Bausatz Lochrasterplatine PAD5 für Steckboards mit Spannungsreglern: Artikel-Nr. 155289
- ELV Steckbrücken-Set für Steckplatinen und Breadboards, 350-teilig – Artikel-Nr. 058831
- Hochwertige ELV Stechkabel-Sets: Artikel-Nr. 251220 bis 251225
- Shiverpeaks USB-3.1-Kabel Gen. 2, USB 3.1 USB-Stecker (Typ C) auf USB 3.0 USB-Stecker (Typ A) – Artikel-Nr. 126336

Aktuelles Zubehör zu LoRaWAN finden Sie im ELVshop unter dem Stichwort „LoRaWAN“

**Widerstände:**

180 Ω/SMD/0402	R4
470 Ω/SMD/0402	R5
1 kΩ/SMD/0402	R1, R2
1,2 kΩ/SMD/0402	R10, R11
3,9 kΩ/SMD/0402	R8, R9
10 kΩ/SMD/0402	R7
18 kΩ/SMD/0402	R3
47 kΩ/SMD/0402	R6

Kondensatoren:

100 pF/50 V/SMD/0402	C9
100 nF/16 V/SMD/0402	C2, C4–C6, C8
4,7 μF/16 V/SMD/0603	C1, C3, C7

Halbleiter:

ELV211746/SMD	U1
SK14/SMD	D1
SP0503BAHTG/SMD	D2
LED/green/SMD/0603	DS1
Duo-LED/rot-grün/SMD	DS2

Sonstiges:

Funkmodul dnt-TRX-ST1	A1
Taster mit 1,2-mm-Tastknopf, 1x ein, SMD, 1,8 mm	S1
Buchsenleiste, 1x 12-polig, 10 mm Pinlänge, gerade, bedruckt, J1, für LoRIS-BM-TRX1	J1
Buchsenleiste, 1x 12-polig, 10 mm Pinlänge, gerade, bedruckt, J2, für LoRIS-BM-TRX1	J2
USB-Buchse, Typ C, SMD	J3
Stiftleiste, 2x 5-polig, 5,97 mm, gerade, RM = 1,27 mm, SMD	J6
Aufkleber mit DevEUI-Adresse, Matrix-Code	

Geräte-Kurzbezeichnung:	LoRIS-BM-TRX1
Versorgungsspannung:	3,0–3,3 V _{dc}
Stromaufnahme:	min. 2,5 μA @ 3,0 V (Sleep) max. 38 mA @ 3,0 V (TX+Signal-LED) min. 3,7 mA @ 5 V (USB-PWR LED) (Sleep) max. 44 mA @ 5 V (USB) (TX+Signal-LED)
Umgebungstemperatur:	-10 bis +55 °C
Frequenzband:	L-Band 865,0–868,0 MHz M-Band 868,0–868,6 MHz O-Band 869,4–869,65 MHz
Duty-Cycle:	L-Band < 1 % pro h M-Band < 1 % pro h O-Band < 10 % pro h
Typ. Funk-Sendeleistung:	+ 10 dBm
Empfängerkategorie:	SRD category 2
Typ. Funk-Freifeldreichweite:	> 500 m
Leitungslängen an J1 und J2:	10 cm max.
Abmessungen:	26 x 55 x 19 mm (ohne Antenne)
Gewicht:	12 g

**Weitere Infos:**

- [1] Hex-Files (Firmware), USB-Seriell-Treiber, Decoder-Beispiel und Node-RED Flow-Beispiele sind im Download-Bereich des Bausatzes unter der Artikel-Nr. 156514 zu finden
- [2] ELVjournal 1/2021, Energiesparer – LPWAN – Low Power Wide Area Networks: Artikel-Nr 251814
- [3] ELVjournal 3/2021, Technik LoRaWAN: Artikel-Nr. 252094
- [4] Institute of Electrical and Electronics Engineers:
<https://www.ieee.org/>
- [5] TheThingsNetwork: <https://www.thethingsnetwork.org>
- [6] TheThingsStack: <https://www.thethingsindustries.com/docs/>
- [7] ELVjournal-Fachbeitrag Node-RED Einführung (3 Teile):
Artikel-Nr. 251410, 251516 und 251603

Alle Links auch online unter: de.elv.com/elvjournal-links

ELV[®] Unsere Social-Media-Kanäle

Den Kontakt zum ELVjournal finden Sie nicht nur über unsere E-Mail-Adresse redaktion@elvjournal.com oder die Postanschrift ELV Elektronik AG · Redaktion ELVjournal · Maiburger Straße 29–36 · 26789 Leer, sondern auch über unsere Social Media-Kanäle.

Hier erfahren Sie außerdem Neuheiten zu Bausätzen, Angeboten oder neuen Produkten im Sortiment von ELV. Auch interessante Nachrichten werden hier veröffentlicht. Besuchen Sie uns dort gerne.



Youtube



www.youtube.com/user/elvelektronik/



Facebook



www.facebook.com/elvelektronik



Twitter



www.twitter.com/elvelektronik



Fragen und Antworten zu LoRaWAN



Was kann ich mit LoRaWAN machen?

Die Möglichkeiten der Anwendungen mit LoRaWAN sind äußerst vielfältig. Im Grunde eignet sich die Technologie für alle Sensoren, die stromsparend Daten übermitteln sollen, die außerhalb der Reichweite normaler Netzwerke wie beispielsweise WLAN liegen (s. a. „Was brauche ich an Hardware?“). Ob es der im Beitrag erwähnte Temperatursensor im Garten, ein Taster oder Sensor am entfernt gelegenen Gartenhaus oder Wohnmobil, das Asset-Management eines Unternehmens, das Gegenstände lokalisieren oder managen will, oder die Smart City handelt, die mit Parkplatzsensoren das Parkraum-Management ermöglichen will: Die LoRIS-Base bietet eine günstige und spannende Möglichkeit, die Chancen dieser Technologie auszutesten und mit ihr zu experimentieren.

Was sind die Vor- und Nachteile?

Es gibt bereits zahlreiche Verfahren, um per Funk Daten von Sensoren zu übermitteln. LoRaWAN stellt eine besonders stromsparende Möglichkeit dar, Daten per Funk über weite Entfernungen an die nächste Gegenstelle (Gateway) in einem lizenzfreien Frequenzband zu senden. Allerdings mit Einschränkungen: Die Datenmengen bewegen sich bei 51–222 Bytes/Sendung und der Duty-Cycle in dem genutzten Frequenzbereich (863–870 MHz) muss eingehalten werden.

Was brauche ich an Hardware, um mit LoRaWAN loszulegen?

Für erste Experimente reicht zunächst die LoRIS-Base und ein in der Nähe befindliches Gateway (s. Material). Auf der LoRIS-Base ist bereits ein Beispielcode aufgespielt, mit dem man zwei Taster und einen Output (bspw. LED) ansteuern kann.

Im Praxisbeispiel wird beschrieben, wie man die Taster anschließt und Daten an TheThingsNetwork (TTN/TTS) weiterleitet. Bereits in der nächsten Ausgabe vom ELVjournal erscheinen zwei LoRIS-Module, mit denen eine Spannungsversorgung mit Knopfzellen und die Ansteuerung eines externen Tasters ermöglicht wird. Weitere Anwendungs- und Powermodule (Temperatur, GPS, Energy-Harvesting...) erscheinen in den darauffolgenden Ausgaben des ELVjournals.

Außerdem wird ein Gateway in der Nähe (~500 m bis einige Kilometer Entfernung) benötigt. In Deutschland sind zurzeit Gateways mit Anbindung an TTN/TTS am weitesten verbreitet. Hierauf basiert auch unser Praxisbeispiel. Ob ein Gateway in der Nähe ist, das mit TTN/TTS verbunden ist, erfährt man unter <https://www.thethingsnetwork.org> (auf der Webseite etwas nach unten scrollen).

Ist kein Gateway in der Nähe, dann seien Sie einer der Ersten, der diese spannende Technologie fördert und die Community zu dieser Funktechnologie ausbaut. Unter „Material“ empfehlen wir günstige Gateways (Indoor/Outdoor).

Muss ich programmieren können?

Nein. Für jedes Anwendungsmodul gibt es für die angebotene Funktionalität die passende Firmware, die per USB und unserem LoRIS-Flasher-

Tool auf die LoRIS-Base aufgespielt werden kann. Für die Integration in die Netzwerkinfrastruktur (z. B. TTN/TTS) bieten wir ebenfalls einen Beispielcode an. Fortgeschrittene Nutzer können per Programmieradapter (STLINK/V2, Olimex 10-12 JTAG-Adapter – auf richtigen Anschluss achten!) und Entwicklungsumgebung (z. B. STM32CubeIDE) einen eigenen Code schreiben. Hierzu bieten wir Code-Templates an.

Auf der LoRIS-Base ist bereits ein Beispielcode aufgespielt, der zwei Taster und einen Output (bspw. LED) ansteuern kann (s. Praxisbeispiel).

Muss ich löten können?

Was brauche ich an Zubehör?

Die LoRIS-Base ist bereits fertig aufgebaut und es muss nichts mehr gelötet werden. Für Experimente mit dem Beispielcode (Taster/LED) findet man unter „Material“ geeignetes Zubehör wie Breadboards, Steckbrücken oder Prototypenadapter. Das meiste des benötigten Materials sollte sich allerdings bereits in der Bastelkiste befinden.

Wie versorge ich die LoRIS-Base mit Spannung?

Dazu gibt es zwei Möglichkeiten: Zum einen vor allem für Experimente auf dem Breadboard die komfortable Spannungsversorgung per USB-Typ-C, zum anderen über +VDD mit 3 bis 3,3 Volt. Es dürfen nicht beide Spannungsversorgungen gleichzeitig angeschlossen sein.

TheThingsIndustries (TTI), TheThingsNetwork (TTN) und TheThingsStack (TTS)

Für die Anbindung an eine Netzwerkinfrastruktur, über die wir unseren Sensorknoten betreiben können, benötigen wir einen entsprechenden Anbieter. Mit TheThingsIndustries [5] gibt es einen Anbieter, der in der Vergangenheit unter dem Namen TheThingsNetwork (TTN) eine kostenfreie, Netzwerkinfrastruktur zur Verfügung gestellt hat. Mit TheThingsStack gibt es mittlerweile einen Nachfolger dazu. Diese Netzwerkanbindung nutzen wir auch für unser Praxisbeispiel.

Aktuelle Informationen zur LoRIS-Base und den Anwendungs- und Powermodulen finden Sie auf den jeweiligen Artikel-Detailseiten im ELVshop oder direkt auf unserer LoRaWAN-/LoRIS-Landing-Page.