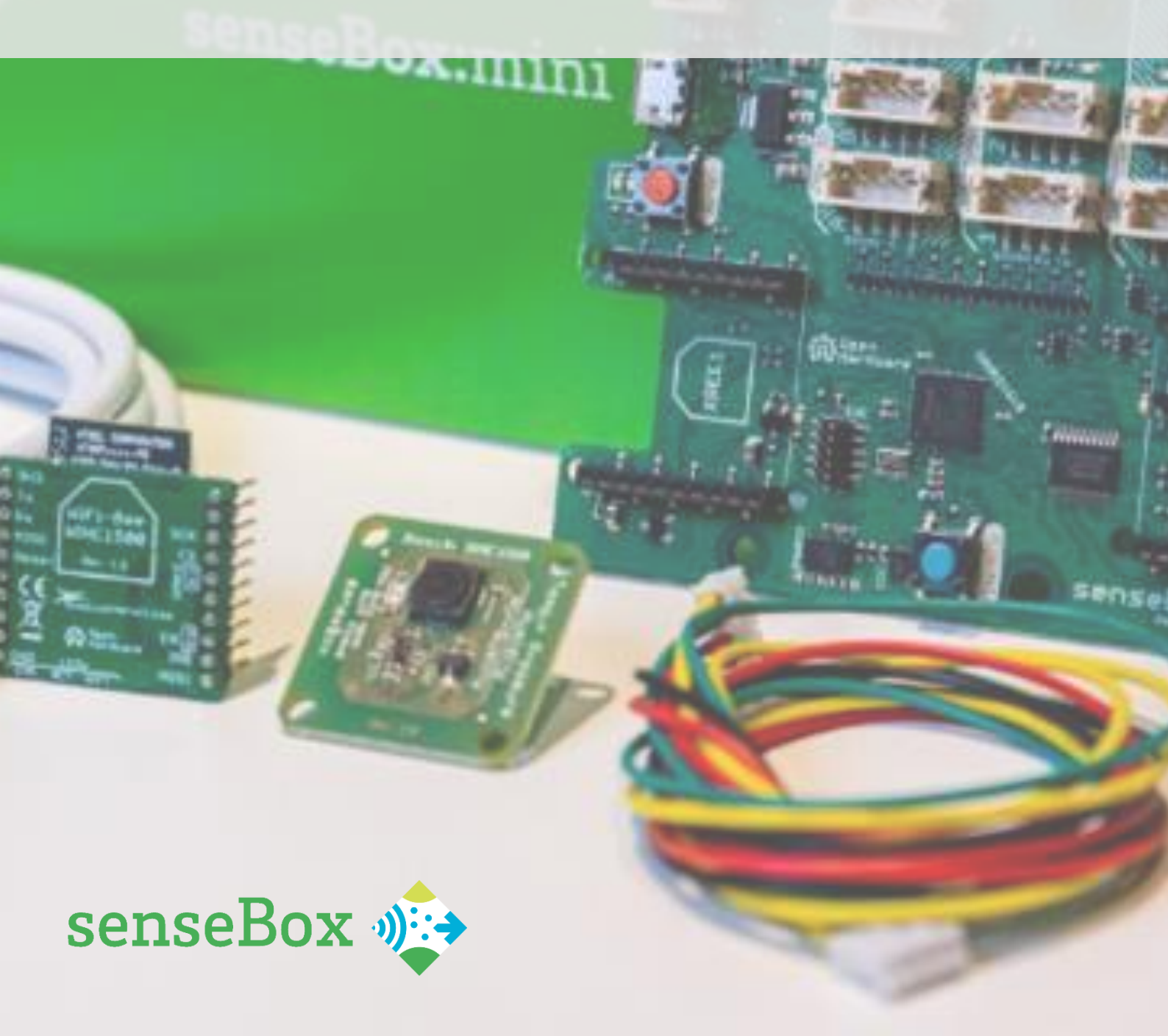


**Moderner Informatikunterricht mit der  
senseBox:  
LoRaWAN als Technologie der Zukunft?**



# Ein Unterrichtskonzept zum Einsatz von LoRaWAN in Verbindung mit The Things Network in der Schule

Theorie, Praxisbeispiele und eine Einbindung in die Bildungsstandards Informatik

## Motivation

Die Schule hat die Aufgabe, die Lernenden zu einer verantwortlichen Teilhabe am Leben zu befähigen und sollte dabei vor allem gesellschaftliche und wirtschaftliche Veränderungsprozesse berücksichtigen. Diese lassen sich in der heutigen Zeit besonders in dem Bereich der Digitalisierung einordnen, weshalb die Bedeutung der digitalen Bildung in der Schule sowie des Faches Informatik stets an Zuwachs gewinnt. Die Bildungsstandards für das Fach Informatik der Gesellschaft für Informatik e.V. berücksichtigen die durch digitale Technologien durchdrungene Welt ebenfalls und betonen, dass die Schülerinnen und Schüler am Ende ihrer Schulzeit die Kompetenzen besitzen sollten, Zukunftsaufgaben zu bewältigen und mitzugestalten. Eine dieser Zukunftsaufgaben stellt ein sicherer, verlässlicher und schneller Weg der Informations- und Datenübertragung dar. Obwohl diese bereits durch bekannte Möglichkeiten, wie WLAN oder Bluetooth, Umsetzung findet, sollten sich die Lernenden darüber bewusst werden, welche Potenziale es in Zukunft auszuschöpfen gibt und wo Gefahren und Grenzen bisheriger Vorgehensweisen liegen. Beispielsweise stellt LoRa in diesem Gebiet einen entscheidenden Faktor der zukünftigen Entwicklung dar und könnte als Grundlage dienen, um den Schülerinnen und Schülern nachhaltige theoretische und konzeptuelle Hintergründe zu den Informations- und Kommunikationstechnologien näher zu bringen. Statt ausschließlich die Theorie zu einem solchen Thema zu vermitteln, bietet ein Einsatz von Mikrocontrollern wie der senseBox im Unterricht die Möglichkeit, praktisch an verschiedenen Projekten zu arbeiten und somit einen Einblick in die ‚Blackbox‘ der digitalen Welt zu erlangen.

Im Folgenden sind nun einige Unterrichtsideen zum Themenkomplex ‚LoRa und LoRaWAN‘ zu finden, die jeweils an die Bildungsstandards Informatik der Sekundarstufe II angebunden sind und deren Umsetzung durch ausführliche Projektbeschreibungen keine Hürde im Schulalltag darstellen sollte. Die Projekte und Materialien eignen sich zum Nachbau mit der senseBox:edu und dem dazu passenden ‚LoRa-Bee‘ und können mit der visuellen Programmieroberfläche Blockly für die senseBox programmiert werden. Zur Datenübertragung wird das The Things Network (TTN) genutzt, welches offen und kostenlos nutzbar ist.

## Einbindung in die Bildungsstandards Informatik

### 1. LoRa als Technologie moderner Vernetzung

Der *Inhaltsbereich ‚Informatiksysteme‘* sieht vor, dass Schülerinnen und Schüler Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken beschreiben und erläutern können. An dieser Stelle kann LoRa in Verbindung mit The Things Network als Beispiel der Informations- und Datenübertragung angeführt werden. Vorteil davon ist die Unabhängigkeit von kommerziellen Anbietern und die Faszination der noch oft unbekannt hohen Reichweite. Dies ermöglicht einen freien und damit einfachen Zugang zum Netzwerk und ermöglicht SuS auch über den Schulunterricht hinaus sich mit der Technologie zu beschäftigen und sich dafür zu begeistern.

Die Interaktion zwischen den verschiedenen Systemen und ihr Beitrag zum Internet of Things wird für die SuS deutlicher, wenn sie konkrete Netzwerkstrukturen vermittelt bekommen und ihren Mehrwert erkennen. Eine Unterrichtsstunde, die sich mit allgemeinen Informationen zur Funktionsweise der Technologie beschäftigt, kann als Einstieg in eine Unterrichtseinheit dienen und anschließend durch die kritische Betrachtung verschiedener Methoden der Datenübertragung vertieft werden. Die Bildungsstandards erwähnen nämlich auch die ökonomischen und ökologischen Folgen von Informatiksystemen als Bestandteil des Unterrichts und betonen, dass an dieser Stelle das Thema über die technischen Aspekte hinaus betrachtet werden sollte.

→ Material 1: Allgemeine Informationen zu LoRa und TTN (S. 4)

→ Projekt 1: TTN Mapper (S. 7)



Quelle: [thethingsnetwork.org](http://thethingsnetwork.org)

## 2. Codierung und Decodierung von Daten

Im *Inhaltsbereich 'Informationen und Daten'* steht die systematische Darstellung und automatische Verarbeitung von Daten als Träger von Informationen im Vordergrund. Es wird gefordert, dass die Schülerinnen und Schüler Information als Daten mit Datentypen und in Datenstrukturen abbilden sowie Operationen auf statischen und dynamischen Datenstrukturen verwenden, modellieren und implementieren können. Zu zeigen, wie Daten in geeigneter Weise codiert werden, um sie mit dem Computer verarbeiten zu können, ist somit ein Teilbereich, den die Schülerinnen und Schülern im Gebiet der Datenverarbeitung beherrschen sollten. Auch an dieser Stelle empfiehlt sich eine praxisnahe Anwendung, sodass am Beispiel von LoRa und TTN erarbeitet werden kann, wie eine Codierung bzw. Decodierung erfolgt, welche Rolle dabei Bytes und Bits spielen sowie welche Möglichkeiten der Cayenne Payload liefert. Letzterer empfiehlt sich vor allem für die Lernenden, die sich zuvor noch nicht mit dem Thema befasst haben, da er einfach in Blockly zu integrieren ist und die Codierung nicht aktiv selbstständig durchgeführt werden muss.

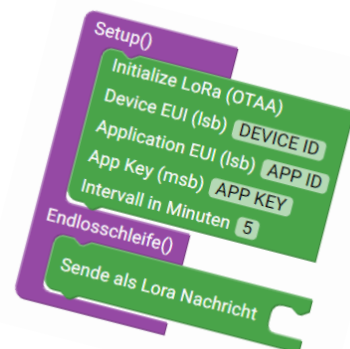
→ Material 2: Informationen zu Bytes/ Bits und Cayenne (S. 14)

→ Projekt 2: Datenübertragung mit dem Cayenne Payload (S. 16)

Befinden sich die Schülerinnen und Schüler bereits auf einem höheren Niveau und sind mit der Thematik des Codierens bzw. Decodierens vertraut, eignet sich eine tiefgreifendere Behandlung des Themas. Somit können sie statt des Cayenne Payloads eine eigene Codierung bzw. Decodierung der Daten durchführen und diese in den Zusammenhang von LoRa und The Things Network einbetten.

→ Projekt 3: LoRa Message, Decodierung der Daten und Integration an die openSenseMap (S. 23)

Da die Programmierung der senseBox mithilfe der graphischen Programmieroberfläche Blockly vollständig durchführbar ist, sind keine umfassenden Kenntnisse im Bereich der Programmiersprachen notwendig. Trotz dessen wird mit den zuvor genannten Projekten durchgehend das algorithmische und logische Denken der Schülerinnen und Schüler gefördert, wie es im Inhaltsbereich 'Algorithmen' der Bildungsstandards gewünscht ist.



## Material 1

### Allgemeine Informationen zu LoRa/ LoRaWAN und The Things Network

#### Was ist LoRa?

LoRa („*Long Range*“) beschreibt eine Art der Informations- und Datenübertragung, die sich durch eine hohe Reichweite und Energieeffizienz auszeichnet. Der Funkstandard wurde für kleine Geräte entwickelt, um von diesen Daten über Gateways in das Internet zu senden. Besteht ein Netzwerk von Sendern, Empfängern und Servern, so ermöglicht LoRa die Interaktion zwischen den verschiedenen Systemen und Techniken, wodurch es sich für die Vernetzung von Objekten zu einem Internet der Dinge (*Internet of Things – IoT*) eignet.

#### LoRa vs. LoraWAN

Damit diese Vernetzung erfolgreich abläuft, wird die Funktechnik durch LoRaWAN („*Long Range Wide Area Network*“) unterstützt. Es beschreibt eine Spezifikation für drahtlose batteriebetriebene Systeme in einem regionalen, nationalen oder auch globalen Netzwerk. Dadurch wird die Kommunikation der einzelnen Komponenten untereinander ohne die Notwendigkeit einer komplexen Installation ermöglicht.

Der Unterschied zwischen LoRa und LoRaWAN wird mit dem folgenden Vergleich etwas deutlicher:

Die Menschen nutzen zur Kommunikation ihre Stimmbänder und ihr Gehör, um Laute zu erzeugen und vom Gegenüber gehört zu werden – was in unserem Fall LoRa entsprechen würde. Allerdings bedarf es einer gemeinsamen Sprache und Grammatik, die die Verständigung untereinander ermöglicht – womit LoRaWAN zum Tragen kommen würde.

#### Aber wie genau funktioniert der Langstreckenfunk?

1. Beispielsweise sind an öffentlichen Gebäuden oder Parkplätzen verschiedene Sensoren und Sender angebracht. Diese sind miteinander verknüpft, ermitteln Daten und senden sie zum LoRaWAN Gateway.
2. Das LoRaWAN-Gateway ist der Empfänger der Daten. Es nimmt die verschlüsselten Daten auf.
3. Anschließend erfolgt die Weiterleitung der Daten zur gewünschten Anwendungsschnittstelle (API). Dies geschieht mittels einer sicheren LoRaWAN-Protokoll-Verbindung.

4. Letztlich nimmt der LoRaWAN Netzwerks server die Daten an der API auf und stellt sie zur Weiterverarbeitung zur Verfügung. Es können beispielsweise Datenanalysen durchgeführt werden.



#### LoRa Sensoren

Senden Messwerte via LoRa an Gateway.



#### LoRa Gateway

Sendet Daten via LTE/LAN an Server.



#### LoRa Server

Verarbeitet Daten, verwaltet Devices, etc.

Quelle: LineMetrics GmbH 2020

### Vor- und Nachteile von LoRa

LoRa zeichnet sich vor allem durch eine Datenübertragung aus, die trotz eines geringen Leistungsverbrauchs hohe Reichweiten erreichen kann. Während gängige Sensorlösungen, die auf WLAN, Bluetooth o.Ä. basieren, unter optimalen Bedingungen eine maximale Reichweite von ca. 100 Metern haben, überwindet LoRa bis zu 50 km und verbraucht dabei nur ein Drittel des Stroms, den ein WLAN-Modul benötigen würde. Folglich entstehen nur niedrige Infrastrukturkosten und LoRa bietet die Möglichkeit einer ökologischen und ökonomischen effizienten Vernetzung. Ein weiterer Vorteil stellt die Nutzung freier Frequenzbänder durch LoRaWAN dar. So fallen keine Lizenzgebühren und Mobilfunkkosten an, sodass die Technologie bundesweit kostenlos zum Einsatz kommen kann. Hinzu kommt die Sicherheit der Technologie, die sich durch die Ende-zu-Ende Verschlüsselung auszeichnet. Dabei wird sowohl für das Netzwerk als auch für die Anwendung jeweils eine Sicherheitsebene verwendet.

Als nachteilig erweist sich hingegen die Übertragungsrate der Daten im LoRaWAN. Diese ist auf 50 Kilobit pro Sekunde begrenzt, wodurch keine großen Datenmengen übermittelt und Echtzeitanalysen aufgestellt werden können. Trotz dessen stellt LoRa einen entscheidenden Faktor in der zukünftigen Entwicklung dar und bietet ein enormes Potenzial, um die Entwicklung der Smart Cities voranzutreiben.



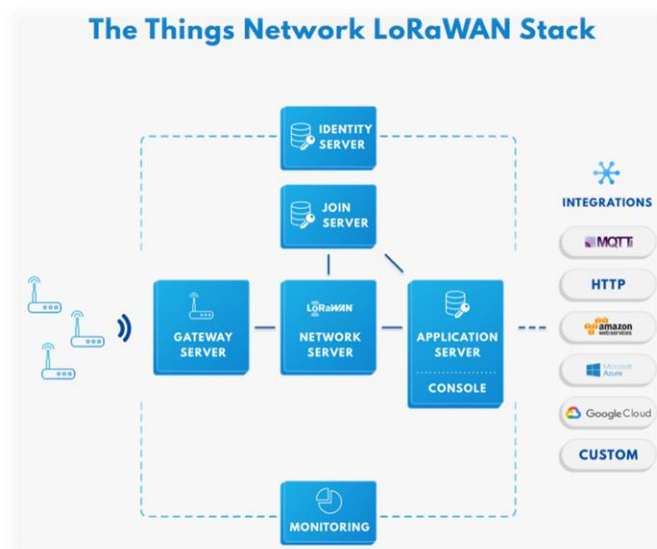
## The Things Network

Wie zuvor bereits beschrieben, ist ein Netzwerkservers notwendig. Dort werden die übermittelten Daten an die LoRaWAN-Gateways nämlich temporär gespeichert, damit ein Anwendungsprogramm sie weiterverarbeiten kann. *The Things Network* bietet als Infrastruktur solch einen Server und beinhaltet zudem eine communitybasierte Initiative, die die Errichtung eines energiesparenden Weitbereichs-Netzwerks für das *Internet of Things (IoT)* verfolgt. Sie besteht seit dem Jahr 2015 und wurde durch zwei Niederländer gegründet, die nun zusammen mit Freiwilligen die Bereitstellung, Errichtung und Betreuung von Gateways übernehmen. Dabei handelt es sich jeweils um lokale Gruppen, die den Aufbau des Netzwerks unterstützen und dabei helfen, die Technologie bekannt zu machen.

*The Things Network* ist somit ein Open Source-Projekt, das ein offenes, globales, dezentrales und freies IoT schaffen möchte. Davon profitieren sowohl private Nutzer\*innen, als auch Unternehmen, denn wenn ein flächendeckendes LoRa-Netz erst einmal geschaffen wurde, so eröffnen sich vielfältige Anwendungsbereiche von Smart Home bis zur Verkehrsflusskontrolle. Da durch die frei nutzbaren Gateways ein großer Kostenfaktor wegfällt, können Pilotprojekte günstig und schnell umgesetzt werden sowie zum IoT beitragen.

Mithilfe des *TTN Mappers* kann überprüft werden, wie die tatsächliche Abdeckung des Netzwerks durch Gateways ist. Dafür messen mitwirkende die Leistung der Gateways in ihrer Umgebung und laden die Informationen anschließend auf die frei zugängliche TTN Mapper-Website hoch.

Die nebenstehende Abbildung gibt einen Überblick über die Rolle von *The Things Network* in der Datenübertragung mit LoRa.



Quelle: The Things Industries 2020

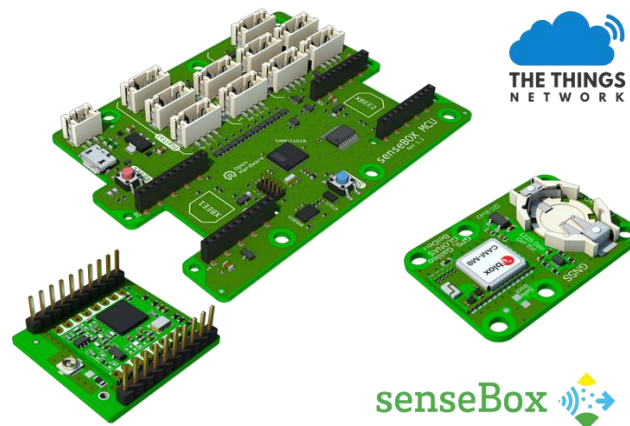
## Projekt 1

### TTN Mapper

Erfasse die Abdeckung und das Potential von LoRa

Das Ziel ist die Entwicklung eines TTN Mappers mit Hilfe des LoRa-Bees und des GPS-Moduls. Die Daten werden über das freie TheThingsNetwork zum TTN Mapper übertragen. Der TTN Mapper ist eine Karte, auf welcher man Orte mit TTN Empfang kartieren kann. Mithilfe der senseBox kannst du dich an der Plattform beteiligen.

#### Build your TTN-Mapper



Für das Projekt benötigst du folgende Komponenten:

- 1x senseBox MCU
- 1x LoRa-Bee
- 1x GPS-Modul
- (evtl. Powerbank)

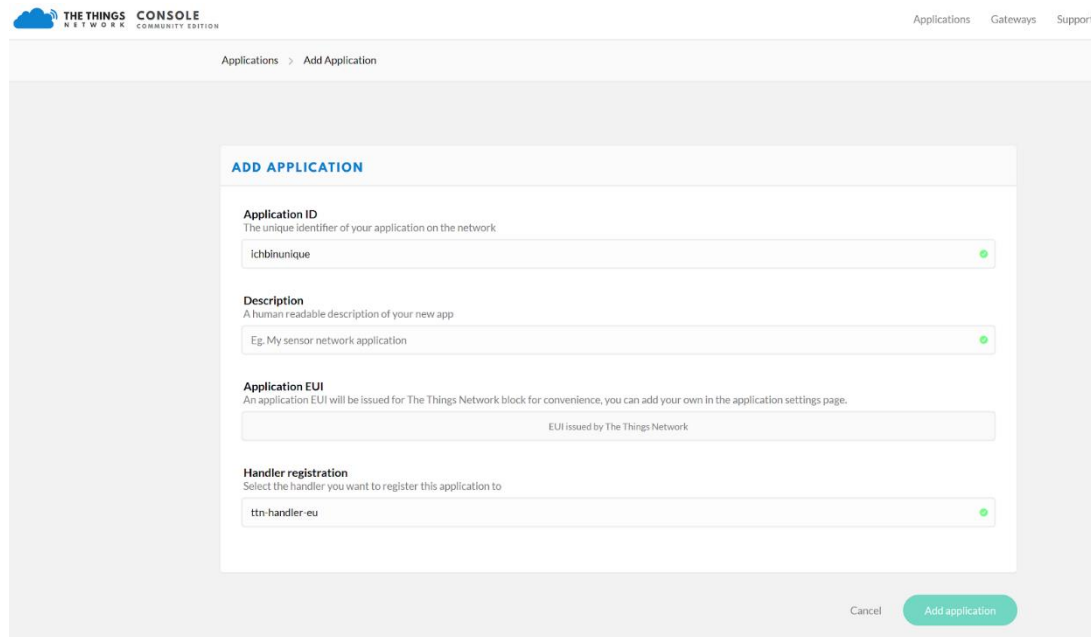
#### Aufbau

Verbinde das LoRa-Bee mit XBEE1 deiner senseBox MCU. Verbinde außerdem das GPS Modul mit einem I2C Steckplatz. Zusätzlich brauchst du eine möglichst mobile Stromversorgung. Am einfachsten ist eine USB Powerbank oder ein Akku mit passendem Anschluss für die senseBox MCU.



## Registrierung auf TheThingsNetwork und Anlegen einer Application

Besuche die Website [thethingsnetwork.org](http://thethingsnetwork.org) und erstelle dir einen Account. Wenn du eingeloggt bist, gehe auf die TTN Console. Dort erstellst du unter "Applications" eine neue Application. Gib dort eine "Application ID" an, welche einzigartig ist. Alle anderen Auswahlmöglichkeiten lässt du unverändert.

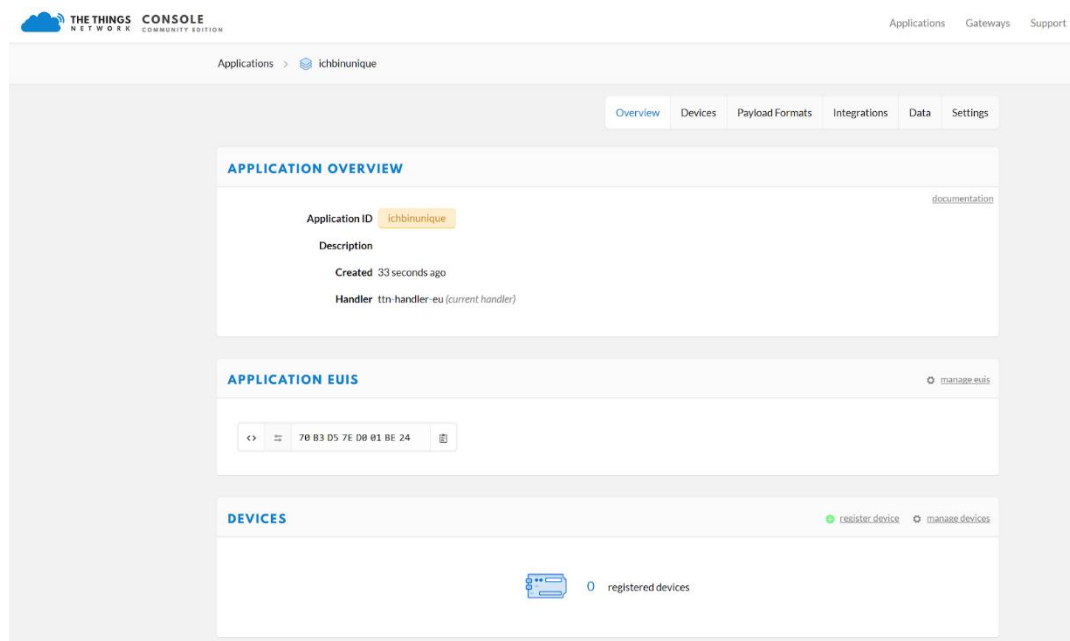


The screenshot shows the 'ADD APPLICATION' form in the TTN Console. The form has four main sections:

- Application ID:** A text input field containing 'ichbinunique' with a green checkmark on the right.
- Description:** A text input field containing 'Eg. My sensor network application' with a green checkmark on the right.
- Application EUI:** A text input field containing 'EUI issued by The Things Network'.
- Handler registration:** A dropdown menu with 'ttn-handler-eu' selected and a green checkmark on the right.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Add application' (highlighted in green).

Danach erhältst du eine Übersicht mit der gewählten "Application ID" und den "Application EUIS".



The screenshot shows the 'APPLICATION OVERVIEW' page for the application 'ichbinunique'. The page has a navigation bar with tabs: 'Overview', 'Devices', 'Payload Formats', 'Integrations', 'Data', and 'Settings'. The 'Overview' tab is active.

The 'APPLICATION OVERVIEW' section displays the following information:

- Application ID:** ichbinunique (with a 'documentation' link)
- Description:** (empty)
- Created:** 33 seconds ago
- Handler:** ttn-handler-eu (current handler)

The 'APPLICATION EUIS' section shows a list of EUIs with a 'manage euis' link. One EUI is visible: 70 B3 D5 7E D8 01 BE 24.

The 'DEVICES' section shows a 'register device' button (with a green checkmark) and a 'manage devices' link. Below this, it shows '0 registered devices' with a device icon.

Nun musst du in deiner neuen Application noch einen Device hinzufügen. Gehe dafür unter der Rubrik "Devices" auf "register device". Gib deinem Device eine "Device ID". Durch Klicken auf die geschwungenen Pfeile ("generate") bei "DeviceEUI" wird dir automatisch einen DeviceEUI generiert.

The screenshot shows the 'REGISTER DEVICE' form in The Things Console. The form is titled 'REGISTER DEVICE' and has a 'bulk import devices' link. It contains the following fields:

- Device ID:** This is the unique identifier for the device in this app. The device ID will be immutable. The value 'uniquedevice' is entered.
- Device EUI:** This is the unique identifier for this device on the network. You can change the EUI later. A generate button is present.
- App Key:** This App Key will be used to secure the communication between you device and the network. A generate button is present.
- App EUI:** The value '78 B3 D5 7E D0 01 BE 24' is displayed.

At the bottom right, there are 'Cancel' and 'Register' buttons.

Klicke dann auf "Register" und du erhältst folgende Übersicht mit der "Application ID" und der "Device ID".

The screenshot shows the 'DEVICE OVERVIEW' page in The Things Console. The page displays the following information:

- Application ID:** ichbinunique
- Device ID:** uniquedevice
- Activation Method:** OTAA
- Device EUI:** 00 34 B9 CF E2 7F C0 10
- Application EUI:** 78 B3 D5 7E D0 01 BE 24
- App Key:** [Redacted]
- Status:** never seen
- Frames up:** 0 (with a link to reset frame counters)
- Frames down:** 0

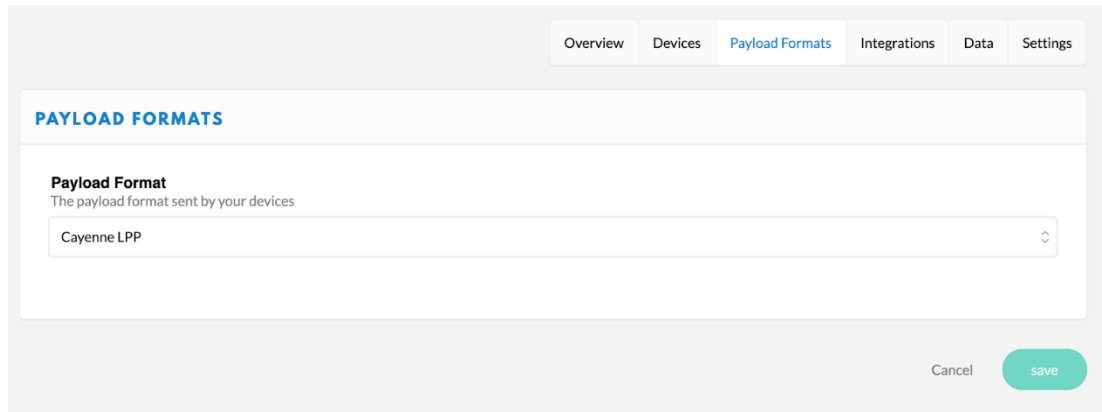
Da wir die Daten später an den TTN Mapper senden möchten, musst Du außerdem eine "Integration" hinzufügen. Diese findest du oben in deinem Menü. Klicke darauf und klicke dann gleich auf den Button mit der Aufschrift "Add Integration". Dort hast du dann die Möglichkeit, verschiedenste Integrations hinzuzufügen, um die Daten von TTN an andere Services weiterzuleiten. Wähle dort den "TTN Mapper" aus. Gib nun eine "Process ID" an, die du frei wählen kannst. Gib deine E-Mail Adresse an und gebe einen Experimente Namen an. Du könntest den Experimente Namen auch auslassen, bist dann aber nicht mehr in der Lage deine eigenen Messungen auf dem TTN Mapper zu identifizieren. Klicke dann auf "Add Integration".

The screenshot shows a web interface for adding a new integration. At the top, there are navigation tabs: Overview, Devices, Payload Formats, Integrations (selected), Data, and Settings. Below the tabs is a header 'ADD INTEGRATION'. The main content area features a card for 'TTN Mapper (v2.7.1)' by 'JP Meijers'. The card includes a location pin icon, a description 'Map your devices and coverage with TTN Mapper.', and a link to 'documentation'. Below the card are four form fields, each with a title, a description, and a text input field with a green checkmark on the right:

- Process ID**: The unique identifier of the new integration process. Input: 'blockly'.
- E-mail address**: The e-mail address to be used for queries about the quality of data. Input: (empty).
- Port filter**: Optional: only send messages sent on this LoRaWAN FPort to TTN Mapper. Input: (empty).
- Experiment name**: Optional: the name of the experiment. Input: 'blockly'.

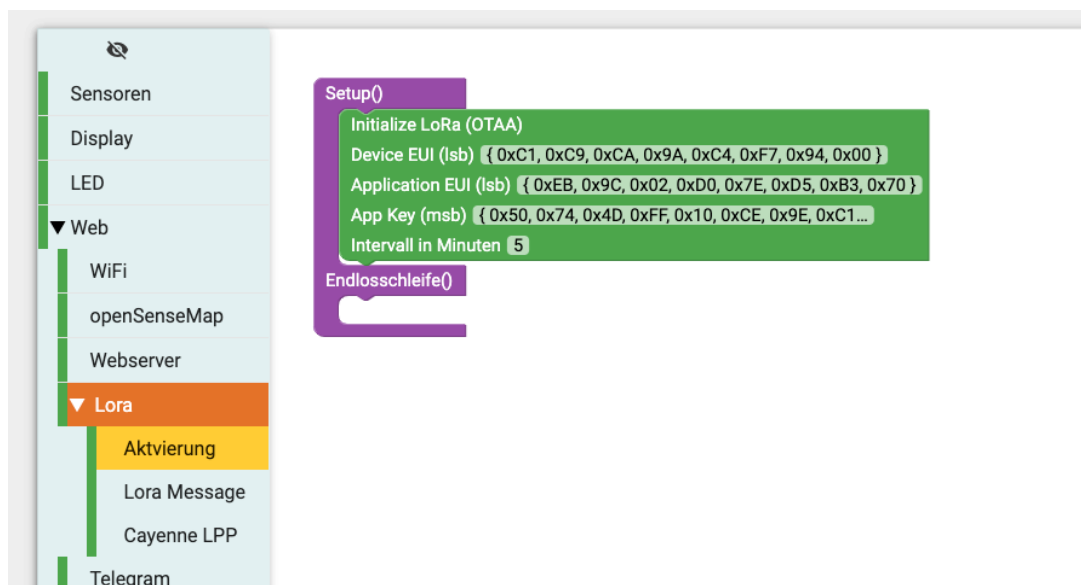
At the bottom right of the form, there are two buttons: 'Cancel' and 'Add integration'.

Zuletzt musst du das Payload Format bei TTN auf Cayenne LPP umstellen. Dazu klickst du auf den Reiter Payload Format, wählst dann Cayenne LPP aus und klickst auf "save". Nun entschlüsselt TTN die Nachrichten selbstständig nach der Cayenne LPP Spezifikation.



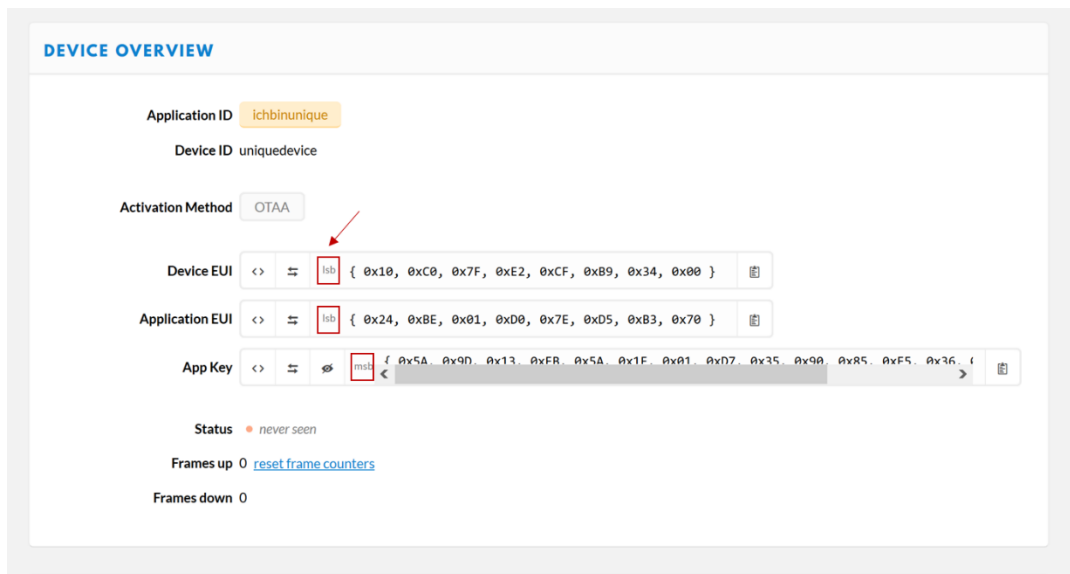
## Blockly

Öffne Blockly und beginne mit dem Code für deine senseBox MCU. Um die TTN Infrastruktur zu nutzen, müssen wir zunächst eine Aktivierung starten. Dazu nutzen wir die OTAA Aktivierung. Je nach Anwendungsfall kannst du das Übertragungsintervall anpassen. Bitte denke daran, dass TTN unter einer Fair Use Policy läuft. Das bedeutet, dass man seine Übertragungsraten möglichst gering halten sollte.

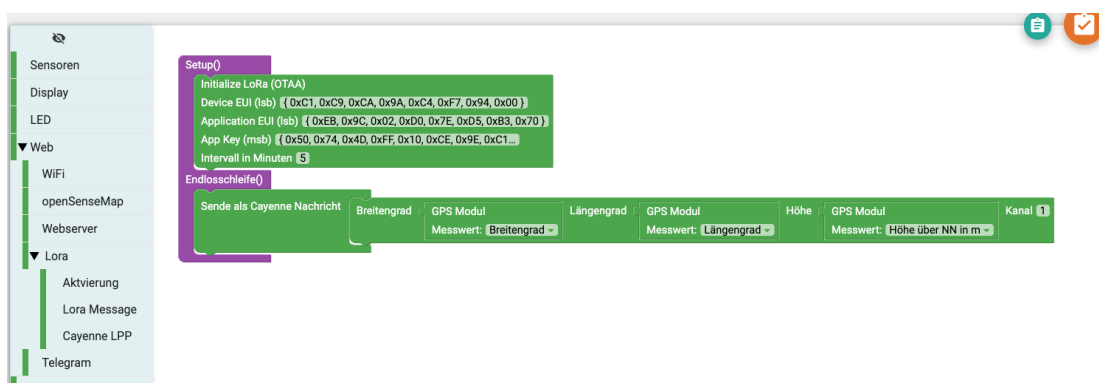


Füge nun deine TTN EUIs ein. Achte darauf, dass Du die Keys im richtigen Format einfügst. “Device EUI” und “Application EUI” müssen im lsb Format genutzt werden. Der “AppKey” im msb Format. Beim Kopieren der Keys musst du deshalb bei TTN in der Device Overview die

Einstellungen wie hier gezeigt vornehmen. Du kannst das Format ändern, indem du auf die Icons am Anfang (<> und ⇌) klickst.



Es gibt verschiedene Wege LoRa Daten zu versenden. In diesem Fall nutzen wir das Cayenne Low Power Payload (LPP), da es sehr einfach ist und bereits ein fertiger Code für diesen Anwendungsfall existiert. Wähle den 'Sende als Cayenne Nachricht' Block und sende eine Koordinatenmessung. Füge die entsprechenden GPS-Blöcke ein. Den Channel brauchst Du nicht zu ändern.



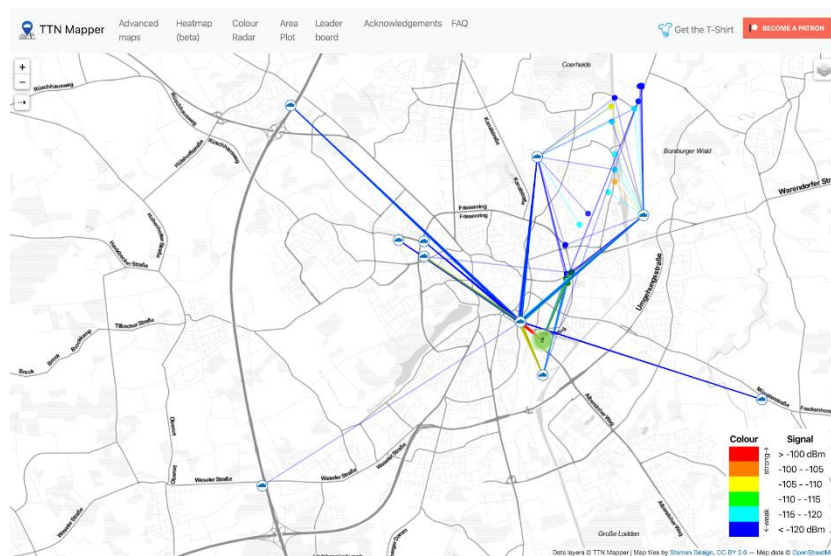
Kompiliere nun den Sketch und lade ihn auf die senseBox MCU. Sie sollte nun auf GPS Daten warten und diese zum TTN übertragen. Nun kannst du einen Akku anschließen und den LoRa Empfang in deiner Umgebung messen.

## TTN Mapper

Deinen persönlichen TTN Mapper findest du unter dem Namen deines Experiments. Öffne dazu:

[http://ttnmapper.org/experiments/?experiment=EXPERIMENT\\_NAME](http://ttnmapper.org/experiments/?experiment=EXPERIMENT_NAME)

und ersetze EXPERIMENT\_NAME mit dem Namen deines Experiments. Es dauert ein bisschen bis die senseBox GPS-Daten empfängt. Zwischendurch kannst Du die Seite neu laden, um die neuesten Messungen zu sehen.



### Tipps

Du kannst in der TTN Konsole nachschauen, ob LoRa-Daten bei TTN ankommen. Klicke in deiner Application auf den Data Tab und die neusten Nachrichten sollten nach kurzer Zeit auftauchen. Falls keine Nachrichten ankommen, schau nochmal über deine EUIs und dessen Format. Ansonsten könnte es sein, dass in deiner Umgebung kein LoRa Gateway in Reichweite ist.

Es kann eine Weile dauern, bis das GPS Modul Daten empfängt. Manchmal dauert es bei der ersten Nutzung mehrere Stunden. Wurden aber einmal Daten empfangen, sollte es beim nächsten Mal schneller funktionieren. Die Nutzung einer kleinen Knopfzelle verbessert den GPS-Empfang.

## Material 2

### Informationen zu Bytes/ Bits und Cayenne

Wie du schon weißt, ist *The Things Network* ein öffentliches Gemeinschaftsnetz, das frei zugänglich ist. Es verwendet eine Politik des fairen Zugangs, wodurch der Datentransfer über LoRa jedoch begrenzt ist. Daher sollten die vorhandenen Kapazitäten möglichst sinnvoll ausgeschöpft und effizient genutzt werden. Dies ist mit einer Codierung der Daten als Bytes möglich, wodurch keine großen Datenmengen, sondern untergliederte Informationen weitergeleitet werden.

#### Was sind Bytes und Bits?

Bytes beschreiben eine Maßeinheit zur Speicherung von Daten, die in der Informatik und Digitaltechnik verwendet wird. Sie beruft sich auf das Konzept der Binärzahlen, bei denen es immer zwei mögliche Zustände gibt (0 oder 1). Ein Bit (*binary digit*) ist die kleinste elektronische Speichereinheit des Computers, die auch nur diese zwei Zustände annehmen kann. Acht Bits ergeben zusammen ein Byte, weshalb man hier zwischen  $2^8 = 256$  verschiedenen Werten unterscheiden kann. Zwei Bytes umfassen hingegen schon 16 Bits und ermöglichen folglich eine Entscheidung zwischen  $2^{16} = 65536$  Werten.

Diese Theorie wird etwas deutlicher, wenn du folgendes Beispiel betrachtest: Misst du mit einem Sensor die Luftfeuchtigkeit deiner Umgebung, die Werte zwischen 0% und 100% annehmen kann und möchtest diese Daten per LoRa übermitteln, so reicht 1 Byte aus, falls es sich nur um ganzzahlige Werte handelt. Möchtest du deine Messung allerdings präzisieren und auch Dezimalzahlen erfassen, so benötigst du 2 Bytes, um die Werte korrekt zu senden.

#### Welche Rolle spielt dabei Cayenne?

Damit die gemessenen Werte effizient übermittelt, aber anschließend trotzdem korrekt dargestellt werden können, bedarf es einer Codierung und Decodierung. Wird beispielsweise eine Dezimalzahl in Prozent erfasst, so wird sie zuerst mit 100 multipliziert, um sie mit zwei Bytes codieren und an TTN schicken zu können (z.B. 75,03%  $\rightarrow$  75,03\*100 = 7503). Dort wird sie dann wieder durch 100 dividiert und somit decodiert, womit der zu Beginn erfasste Werte geliefert wird (7503/100 = 75,03).



Den Prozess der Decodierung brauchst du in diesem Projekt allerdings nicht selbst vornehmen, sondern überlässt ihn dem *Cayenne Low Power Payload (LPP)*. Er berücksichtigt die Beschränkung des Datentransfers für TTN-Nutzer und enthält einen Decoder, wodurch die Daten automatisch wieder in ein lesbares Format transformiert werden. Somit ist es eine bequeme und einfache Möglichkeit, um Daten über Netzwerke wie LoRaWAN zu senden.

## Projekt 2

### Datenübertragung mit dem Cayenne Payload

Ziel dieses Projektes ist es, ein Umweltphänomen deiner Umgebung mithilfe der senseBox zu messen und die Daten per LoRa zu messen und im *The Things Network* abzurufen (sowie an die openSenseMap weiterzuleiten).

Dafür benötigst du folgende Komponenten:

- 1x senseBox MCU
- 1x LoRa-Bee
- 1x Temperatur- und Luftfeuchtigkeitssensor (HDC1080)
- 1x JST-Kabel
- (evtl. PowerBank)

#### Aufbau

Schließe den Temperatur- und Luftfeuchtigkeitssensor mithilfe des JST-Kabels an einem der I2C/Wire-Anschlüsse an deine senseBox MCU an. Platziere zudem das LoRa-Bee auf dem XBEE-Steckplatz 1. Fall du die senseBox nicht dauerhaft über deinen Laptop mit Strom versorgen möchtest, brauchst du eine zusätzliche mobile Stromquelle, z.B. einen Akku oder eine PowerBank.

#### The Things Network

Um die Vorteile von TTN nutzen zu können, registriere dich auf der Seite *thethingsnetwork.org*. Wähle anschließend im Dropdown-Menü „Console“ aus und füge eine neue Anwendung hinzu, indem du zuerst „Application“ und anschließend „Add Application“ auswählst. Hier gibst du ausschließlich eine individuelle Application ID an und bestätigst dann deine Eingabe.

Hast du die Application hinzugefügt, wird dir darauffolgend eine Übersicht über die von dir gewählte Application ID und die Application EUI5 geliefert.

**THE THINGS NETWORK CONSOLE** COMMUNITY EDITION

Applications Gateways Support

Applications > Add Application

**Application ID**  
The unique identifier of your application on the network  
project\_lora

**Description**  
A human readable description of your new app  
Eg. My sensor network application

**Application EUI**  
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.  
EUI issued by The Things Network

**Handler registration**  
Select the handler you want to register this application to  
ttn-handler-eu

Cancel Add application

Füge nun zu deiner neuen Application noch einen Device hinzu. Gehe dafür unter der Rubrik „Devices“ auf „register device“. Gib deinem Device eine Device ID. Durch einen Klick auf die geschwungenen Pfeile („generate“) bei der DeviceEUI, erfolgt die Generierung der DeviceEUI automatisch.

**THE THINGS NETWORK CONSOLE** COMMUNITY EDITION

Applications Gateways Support

Applications > project\_lora\_test > Devices

**REGISTER DEVICE** bulk import devices

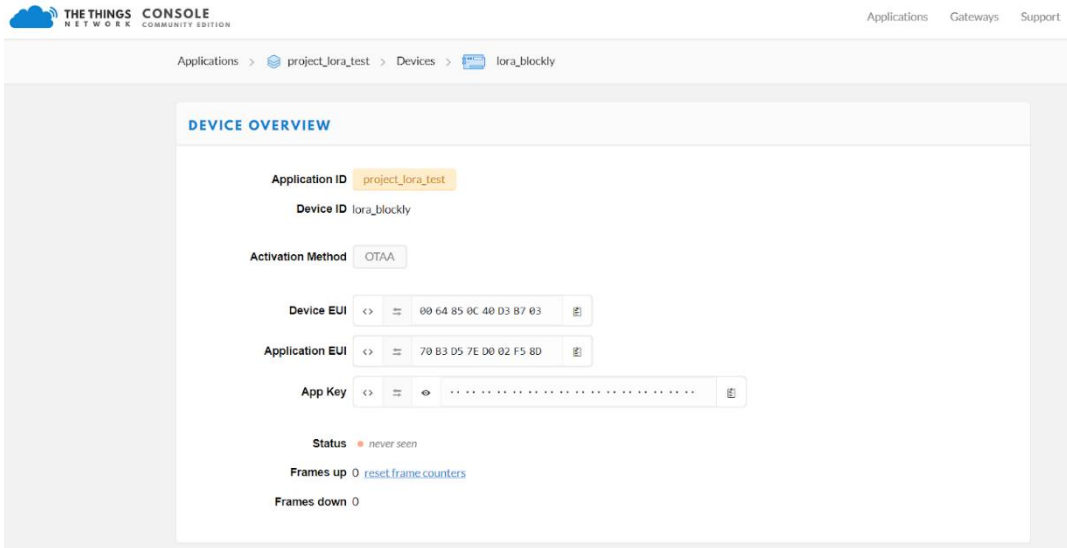
**Device ID**  
This is the unique identifier for the device in this app. The device ID will be immutable.  
[Empty field]

**Device EUI**  
The device EUI is the unique identifier for this device on the network. You can change the EUI later.  
[generate icon] this field will be generated

**App Key**  
The App Key will be used to secure the communication between you device and the network.  
[generate icon] this field will be generated

**App EUI**  
70 B3 D5 7E D0 02 F5 8D

Durch einen Klick auf „Register“ speicherst du deine Eingabe und *The Things Network* liefert dir eine Übersicht mit der Application ID und der Device ID.




Nun hast du bereits erfolgreich eine Application und einen Device hinzugefügt. Möchtest du die Daten nicht nur bei The Things Network, sondern auf einer anderen Plattform abrufen können, so kannst du über dieses Projekt hinaus eine Integration anlegen. Diese Möglichkeit wird dir in der Menüleiste unter 'Integration' geboten.

### openSenseMap (BETA)

*Die Weiterleitung der Daten mit dem Cayenne-Payload ist auch an die openSenseMap möglich. Wie du die passenden Einstellungen dafür triffst, wird nun erläutert. Allerdings handelt es sich hierbei um eine BETA-Version, weshalb es zu kleinen Problemen kommen kann. Wählst du also eine andere oder keine Integration, so kannst du dieses Kapitel zur openSenseMap überspringen und mit dem nachfolgenden Kapitel ‚Cayenne LPP‘ weitermachen.*

Die Möglichkeit des Hinzufügens der *openSenseMap* als Integration wird dir in der Menüleiste unter „Integration“ geboten. Klicke auf „Add Integration“ und wähle anschließend den Button „HTTP Integration“ aus. Bestimme auch hier wieder eine individuelle ID und wähle anschließend „default key“ als Access Key aus, „POST“ als Method und füge *https://ttn.opensensemap.org/v1.1* als URL ein, sodass die Daten an die *openSenseMap* weitergeleitet werden.

Applications > project\_lora\_test > Integrations



**HTTP Integration** (v2.6.0)  
The Things Industries B.V.  
Sends uplink data to an endpoint and receives downlink data over HTTP.  
[documentation](#)

**Process ID**  
The unique identifier of the new integration process

**Access Key**  
The access key used for downlink

devices
messages

**URL**  
The URL of the endpoint

**Method**  
The HTTP method to use

Da die erhobenen Daten auf der *openSenseMap* als Endpunkt angezeigt werden sollen, muss hier zunächst einmal eine neue senseBox registriert werden. Melde dich dazu auf *opensensemap.org* an und füge eine neue senseBox hinzu. Wähle dabei das Modell senseBox:home V2 mit der Verbindungsart LoRa aus klicke auf den passenden Sensor und das Umweltphänomen, dass du messen möchtest. Klicke anschließend auf „The Things Network“ und fülle die Felder aus. Verwende als Dekodierungs-Profil ‚Cayenne LPP (beta)‘ sowie die Application ID und die Device ID, die du zuvor benannt hast, denn nur so kann eine genaue Zuordnung zwischen der senseBox, der *openSenseMap* und *The Things Network* stattfinden.

Bestätige anschließend deine Eingaben und schon hast du die Registrierung der neuen senseBox auf der *openSenseMap* abgeschlossen.

Erweitert

MQTT >

TheThingsNetwork - TTN >

Die openSenseMap bietet eine Integration mit TheThingsNetwork an. Für eine Erklärung der Parameter siehe hier

TheThingsNetwork

**Dekodierungs-Profil**

**TTN Application-ID**

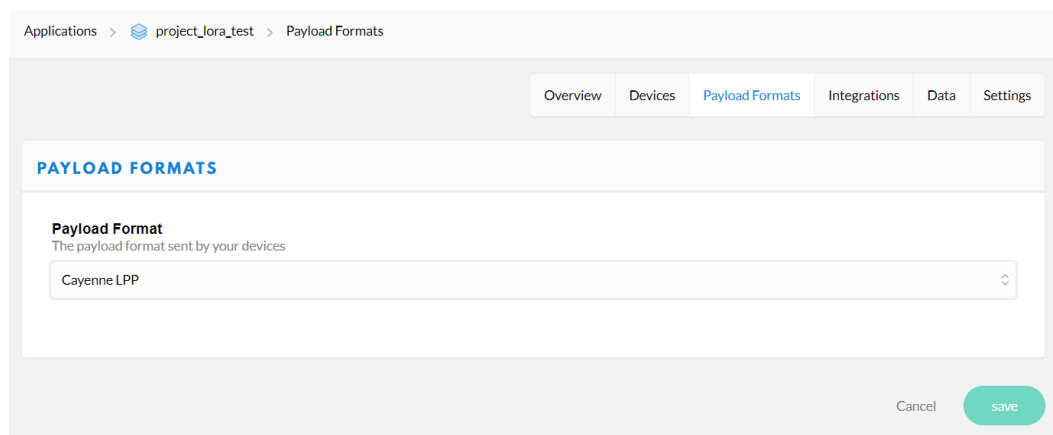
**TTN Device-ID**

**Dekodierungsoptionen**

Phänomen	Typ	Cayenne LPP Phänomen	Cayenne LPP Channel
<input type="text" value="rel. Luftfeuchte"/>	<input type="text" value="HDC1080"/>	<input type="text" value="Luftfeuchtigkeit"/>	<input type="text" value="1"/>

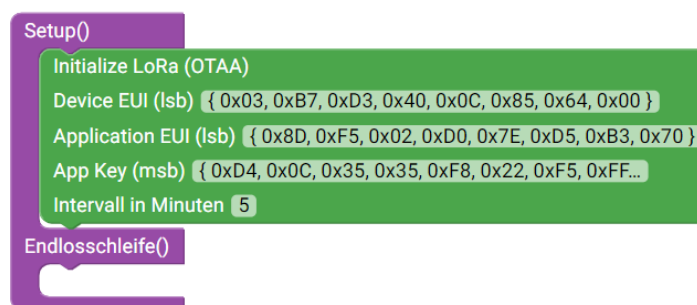
## Cayenne LPP

Jetzt musst du nur noch vorgeben, wie die Daten decodiert werden sollen. Da wir uns in diesem Projekt für den Cayenne LPP entschieden haben, wähle dieses Format unter dem Menüpunkt „Payload Formats“ aus. Damit hast du alle nötigen Einstellungen bei *The Things Network* getroffen.

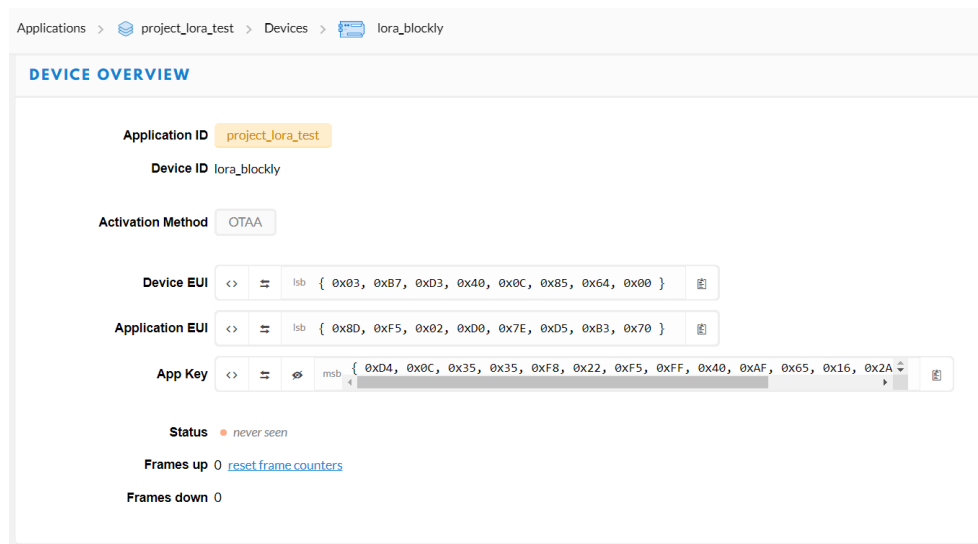


## Blockly

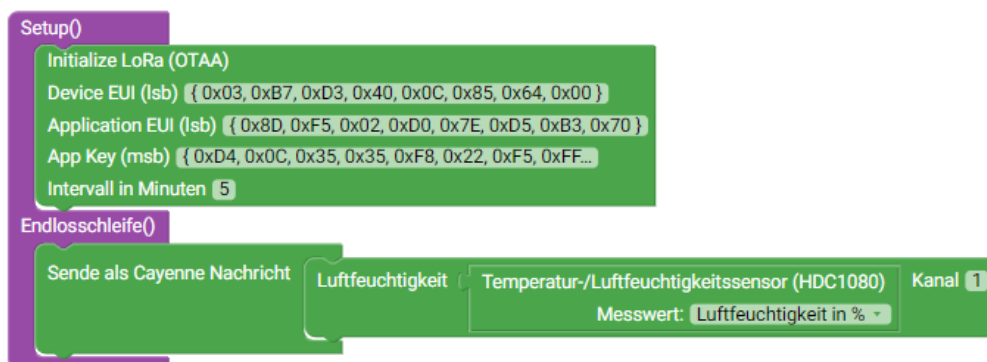
Zuletzt muss noch deine bereits aufgebaute senseBox in das bisherige System eingebunden werden. Dazu bietet sich am besten *Blockly* an, womit du die senseBox in wenigen Schritten programmieren kannst. Rufe die Seite [blockly.sensebox.de](http://blockly.sensebox.de) auf und wähle die senseBox MCU aus. Damit du die TTN Infrastruktur nutzen kannst, muss diese zuerst im 'Setup' initialisiert werden. Nutze dazu den Block 'Initialize Lora (OTAA)', den du unter der Kategorie 'Web/Lora' findest. Das Übertragungsintervall kannst du anpassen, aber denke daran, dass die übertragbare Datenmenge begrenzt ist.



Füge nun deine TTN EUIs ein und achte dabei auf das richtige Format. Device EUI und Application EUI müssen im Isb-Format genutzt werden. Der AppKey im msb-Format. In der Übersicht bei *The Things Network* kannst du durch einen Klick auf die Pfeile die Veränderung zum richtigen Format vornehmen und anschließend die passenden Werte kopieren und in *Blockly* einfügen.



Füge anschließend in der 'Endlosschleife' den Block 'Sende als Cayenne Nachricht' ein und wähle das Umweltphänomen und den passenden Sensor aus der Kategorie 'Sensoren' aus. Die Werte werden so automatisch decodiert. Möchtest du mehrere Umweltphänomene gleichzeitig erfassen, so solltest du darauf achten, dass für jedes Phänomen ein neuer Kanal verwendet wird.





Kompiliere nun den Sketch und übertrage ihn auf deine senseBox. Damit hast du das Projekt abgeschlossen und kannst nun eigenständig Umweltdaten erheben und sie effizient per *LoRa* verschicken. Der Abruf erfolgt am schnellsten über die TTN Konsole. Klicke dazu in deiner Application auf den Data Tab und die neusten Nachrichten sollten nach kurzer Zeit auftauchen. Falls keine Nachrichten ankommen, schau nochmal über deine EUIs und dessen Format. Ansonsten könnte es sein, dass in deiner Umgebung kein LoRa Gateway in Reichweite ist.

## Projekt 3

### LoRa Message, Decodierung der Daten und Integration an die openSenseMap

Ziel dieses Projektes ist es, eine die Daten, die mit der senseBox über LoRa versendet werden, eigenständig zu decodieren und in die openSenseMap zu integrieren.

Dafür benötigst du folgende Komponenten:

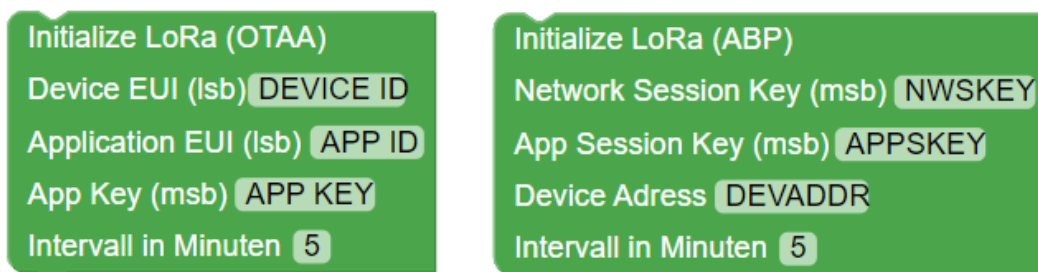
- 1x senseBox MCU
- 1x LoRa-Bee
- 1x Temperatur- und Luftfeuchtigkeitssensor (HDC1080)
- 1x JST-Kabel
- (evtl. PowerBank)

#### Aufbau

Schließe den Temperatur- und Luftfeuchtigkeitssensor mithilfe des JST-Kabels an einem der I2C/Wire-Anschlüsse an deine senseBox MCU an. Platziere zudem das LoRa-Bee auf dem XBEE-Steckplatz 1. Fall du die senseBox nicht dauerhaft über deinen Laptop mit Strom versorgen möchtest, brauchst du eine zusätzliche mobile Stromquelle, z.B. einen Akku oder eine PowerBank.

#### Blockly

##### 1. Aktivierung



Mit diesen Blöcken kannst du das Lora-Bee initialisieren und dich mit dem The Things Network verbinden. Je nachdem, welche Aktivierungsmethode du verwendest (OTAA oder ABP)

musst du die entsprechenden ID's in dem Block eintragen. Zusätzlich kann das Übertragungsintervall in Minuten eingestellt werden.

## 2. LoRa Nachricht



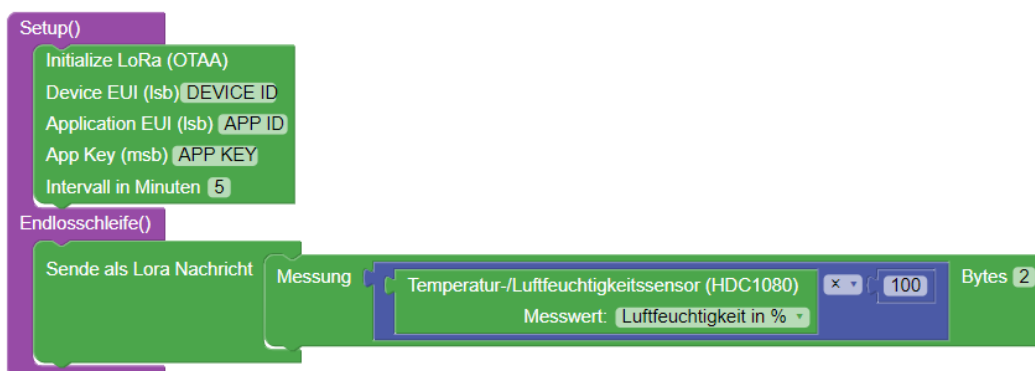
Verwende diese Blöcke, um eine Nachricht über das LoRa Netzwerk zu versenden. Hierbei wird kein spezielles Format für die Nachricht (oft auch Payload genannt) verwendet. Die Daten werden hierbei als Byte codiert, sodass diese schnell und effizient über das LoRa Netz übertragen werden können. Anschließend müssen die Daten in der TTN Console wieder über einen speziellen Decoder in ein lesbares Format umgewandelt werden.

### Decoder erstellen

Um das Prinzip der Decodierung von Daten zu verstehen, solltest du dich im Bereich der Bytes und Bits auskennen. Informationen dazu erhältst du in Material 2.

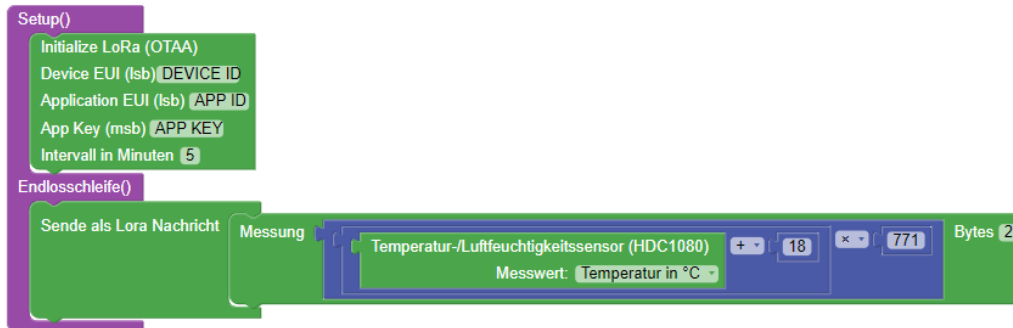
Bist du mit der Thematik vertraut, kannst du mit dem TTN Decoding nun schnelle Lösungen erstellen, um auf deine, von der senseBox gemessenen Daten, zuzugreifen.

In Blockly solltest du dafür folgende Blöcke verwenden:



Die Dokumentation zeigt eine Funktion, um einen Wert mit 16 Bit zu versenden. Diese Funktion kann genutzt werden, um die Werte für die Luftfeuchtigkeit zu versenden. Der Wert für die Luftfeuchtigkeit liegt zwischen 0% und 100%. Der Messwert wird mit 100 multipliziert, um die zwei Nachkommastellen beizubehalten.

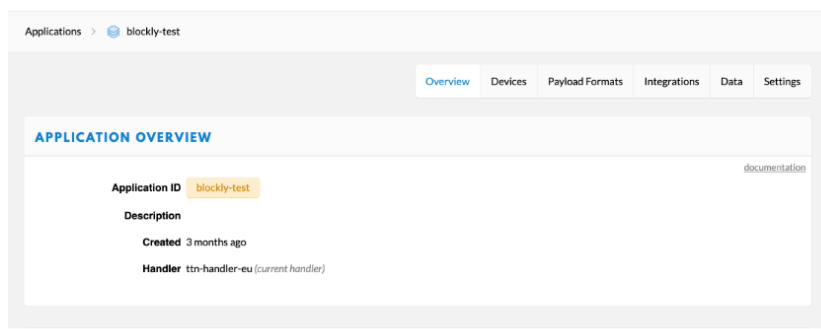
Da der Wert für die Temperatur auch in den negativen Bereich fallen kann, muss zunächst der Wert in einem Positiven konvertiert und anschließend mit einem Wert multipliziert werden, um die Nachkommastellen zu behalten.



In diesem Beispiel wird ein Temperatursensor verwendet, der Werte ab -18 Grad Celsius gemessen hat und die Temperatur mit einer Genauigkeit von 0.0013 Grad Celsius rausgibt.

## The Things Network

In der TTN Console kommen nun zwei verschiedene Werte von der senseBox an. Diese Werte müssen nun mithilfe eines Decoders wieder von Bytes zu den ursprünglichen Werten decodiert werden. Den Decoder kannst du unter Payloads definieren:



```
/**
 * Convert the array of bytes to an unsigned integer, LSB.
 *
 * BEWARE: This is only safe up to 0xFFFFFFFF, so: 6 bytes.
 */
function uint(bytes) {

  return bytes.reduceRight(function(acc, b) {
    // We expect an unsigned value, so to support more than 3 bytes
    // don't use any bitwise operators, which would always yield a
    // signed 32 bits integer instead.
    return acc * 0x100 + b;
  }, 0);
}
```

Hier kann nun der Decoder definiert werden

```
function Decoder(bytes) {  
  
    var i = 0;  
    var decoded = {};  
    // The index in the bytes array that needs to be handled next;  
    // use along with "i++" which returns the current value, and  
    // then increments it for the next usage  
  
    // Relative humidity with 0.01 precision, 0.00 through 100.00, LSB  
    var humidity = uint(bytes.slice(i, i+=2)) / 100;  
    // Temperature with 0.0013 precision, -18.000 through 67.000, LSB  
    var temperature = uint(bytes.slice(i, i+=2)) / 771 - 18;  
    // Unary plus operator to cast string result of toFixed to number  
    temperature = +temperature.toFixed(3);  
  
    decoded['TEMPERATURE_SENSOR_ID'] = temperature  
    decoded['HUMIDITY_SENSOR_ID'] = humidity  
  
    return decoded;  
}
```

Die Funktion nimmt die ersten zwei Bytes und konvertiert sie zum Messwert für Luftfeuchtigkeit. Die nächsten zwei Bytes werden zu einem Messwert für die Temperatur konvertiert. Wenn du die openSenseMap als Endpunkt (<https://ttn.opensensemap.org/v1.1>) in der TTN HTTP Integration verwendest, werden die Messwerte mit deinen Sensor IDs verknüpft und die Messwerte auf der openSenseMap angezeigt.

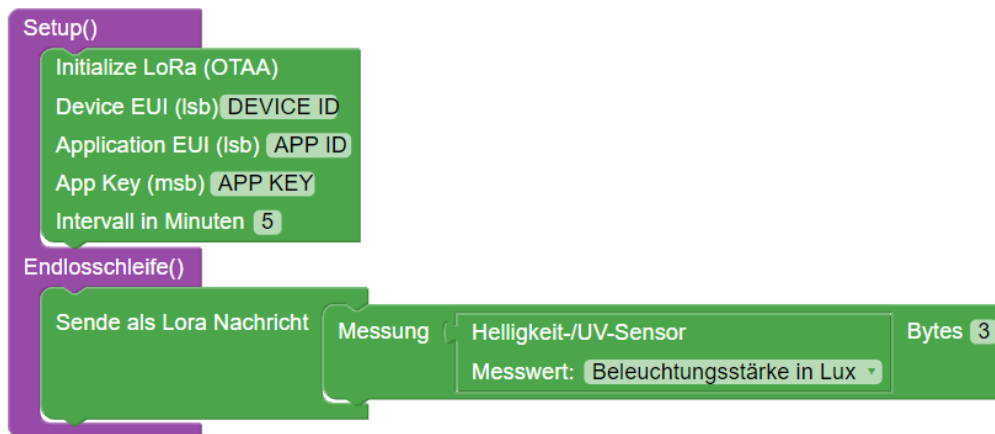
Anschließend solltest du dich noch auf [opensensemap.org](https://opensensemap.org) anmelden und eine neue senseBox hinzufügen. Wähle den passenden Sensor und das Umweltphänomen, dass du messen möchtest, aus. Klicke anschließend auf 'The Things Network' und fülle die Felder aus. Verwende dafür die Application ID und die Device ID, die du zuvor benannt hast, denn nur so kann eine genaue Zuordnung zwischen der senseBox, der openSenseMap und The Things Network stattfinden.

### Einige wichtige Punkte sind zu beachten:

Die Dekodierung schlägt fehl/ gibt falsche Werte zurück, wenn du nicht alles sendest, was dekodiert werden muss. Wenn der Decoder z.B. einen Feuchte- und einen Temperaturwert akzeptiert, die senseBox aber nur einen Feuchtwert sendet, erhältst du keine korrekten Werte.

Beim Senden von Werten, die größer als 3 Bytes sind, ist es ein wenig anders. Werte für die Helligkeit können im Maximum größer sein als ein 2 Byte Integer. Daher werden 3 Bytes benötigt.

Du könntest das Ganze wie folgt senden:



Die Decodierung müsste dann folgendermaßen angepasst werden:

```
var lux = bytes[i] | bytes[i+1]<<8 | bytes[i+2]<<16;  
i = i + 3 // increment counter afterwards
```