



LoRa-Workshop 2 – LoRa und TheThingsNetwork Alexander Knaak, Benno Hölz SFZ Südwürttemberg

## The Things Network (TTN) Ein Server für unsere Messwerte



### TheThingsNetwork (TTN) Ein Server für unsere Messwerte

# Application

- Sammelpunkt aller Geräte und Daten
- Ausgangspunkt für die Verarbeitung der Daten

# Device

- Sensorknoten
- Senden die Daten über das Gateway an die Application

### TheThingsNetwork (TTN) Ein Server für unsere Messwerte

- 1. Konto erstellen: https://www.thethingsnetwork.org/
- 2. Application erstellen (Console)
- 3. Device zur Application hinzufügen

### Im TheThingsNetwork die Console öffnen.



## Auf das Symbol Applications klicken



#### Auf add application klicken, um eine neue Applikation zu erstellen.

Applications	
APPLICATIONS	add application

# Es öffnet sich ein Fenster in dem Details zur Applikation eingetragen werden müssen.

Application ID		
The unique identifier of you	r application on the network	
alexanders_test_application	20	
Description A human readable description	on of your new app	
Testapplikation für den Lo	RaWAN Workshop	
Application EUI	na martin i rennange	
Application EUI An application EUI will be is:	sued for The Things Network block for convenience, you can add your own in the application settings page. EUI issued by The Things Network	
Application EUI An application EUI will be is Handler registration	sued for The Things Network block for convenience, you can add your own in the application settings page. EUI issued by The Things Network	
Application EUI An application EUI will be is: Handler registration Select the handler you want	sued for The Things Network block for convenience, you can add your own in the application settings page. EUI issued by The Things Network	
Application EUI An application EUI will be is Handler registration Select the handler you want ttn-handler-eu	sued for The Things Network block for convenience, you can add your own in the application settings page. EUI issued by The Things Network	

Im Reiter *Payload Formats* wählen wir *Custom* und fügen folgenden Code unter *decoder* ein (mehr dazu später):

```
function Decoder(bytes, port) {
1
    var decoded = {};
2
3
    if (port === 1)
4
5
    Ł
       decoded.epoch = ((bytes[4] << 32) + (bytes[3] << 24)
6
           + (bytes[2] << 16) + (bytes[1] << 8) + (bytes[0])
           );
       decoded.temperature = (bytes[9] << 8) + (bytes[8]);</pre>
7
8
    }
9
    return decoded;
10 }
```

→ Bestätigen mit save payload functions!

In der erstellten Applikation können nun die einzelnen Sensorknoten (devices) angelegt werden.

Dazu klickt man im Abschnitt DEVICES auf register device

APPLICATION OVERVIEW	
Application ID         alexanders_test_application           Description         Testapplikation für den LARAWAN Workshop           Created         36 seconds ago           Handler         tim-handler-eu (current handler)	documentation
APPLICATION EUIS           • = 7083057ED00367E2	O manage suis
DEVICES 0 registered devices	• resister device • manage devices

# Es öffnet sich ein neues Fenster in dem die *Device ID* des Sensors festgelegt werden muss.

Device ID This is the un esp32_01	ique identifier for the device in this app. The device ID will be immutable.	
esp32_01		
		0
Device EU The device E	] UI is the unique identifier for this device on the network. You can change the EUI later.	
/	this field will be generated	
The App Key	will be used to secure the communication between you device and the network.	
'	tris neio wii be generateo	
App EUI		
70 B3 D5 7	E D0 03 67 E2	0

In den *Settings* des Gerätes muss in unserem Fall die *Activation Method* von OTAA auf ABP geändert werden.

of the device			
			0
o module, similar to a M	AC address		
D 4A			🥑 8 bytes
			0
	of the device	or the device	rof the device

In der *Device Overview* können nun die wichtigsten Parameter zur Einbindung des Sensors in die LoRaWAN Applikation ermittelt werden.

DEVICE OVERVIEW							
Application I	D alexanders_test_application						
Device II	D esp32_01						
Activation Metho	ABP						
Device EU	UI 💠 🌣 00 A7 23 82 31 7A 2D 4A 🗈						
Application EL	UI 💠 😄 70 B3 D5 7E D0 03 67 E2 👔						
Device Addres	ss ↔ ≒ 26011558 @						
Network Session Ke	ey 🗘 😄 🚿 msb_{		0	0	0	0	0 8
App Session Ke	ey 🗢 🏛 🕫 msb [	1	1		1		0 E

#### ESP und LoRaWAN Werte ins Netzwerk schicken

Benötigt auf dem ESP:

- boot.py (Hauptprogramm)
- config.py
- encryption\_aes.py
- sx127x.py
- ttn
  - ttn\_eu.py

Die config.py enthält folgende Informationen:

- Pinbelegung des LoRa Chips f
  ür verschiedene Microcontroller
- Das Sendeintervall der LoRaWAN-Botschaft
- Die LoRa-Parameter (Frequenz, Kanal, Bandbreite, ...)
- Die TTN-Parameter der Applikation

Da hier der ESP32 genutzt wird, muss die entsprechende Definition einprogrammiert werden. Die anderen Gerätekonfigurationen können entweder auskommentiert oder gelöscht werden.

```
ES32 TTG0 v1.0
  #
1
2
  device_config = {
       'miso':19,
3
4
       'mosi':27,
       'ss':18,
5
6
       'sck':5.
7
      'dio_0':26,
8
       'reset':14.
9
       'led':2,
10 }
```

In den *lora\_parameters* wird die Einstellung des LoRa Chips vorgenommen und verwaltet. Hier wird die Frequenz, der Sendekanal und der Aufbau der LoRa Botschaft definiert.

```
lora_parameters = {
1
      'frequency': 868E6, #LoRa Frequenz Europa
2
3
       'channel': 0, #Der verwendete Sendekanal
       'tx_power_level': 2, #Sendeleistung
4
5
      'signal_bandwidth': 'SF7BW125', #Bandbreite
       'spreading_factor': 7, #Spreizfaktor
6
7
      'coding_rate': 5,
8
      'preamble_length': 8,
      'implicit_header': False,
9
      'sync_word': 0x12,
10
      'enable_CRC': True, #muss auf True gesetzt werden!
11
       'invert IQ': False.
12
13 }
```

Damit die Daten des ESP auch in der erstellten Applikation ankommen, müssen die benötigten Adressen und Schlüssel aus der Application-Overview noch eingefügt werden.

(An die eigenen Parameter anpassen!)

Die Datei boot.py enthält ein kleines Beispielprogramm, das zufällige Temperaturwerte generiert und diese zusammen mit einem Zeitstempel verschickt.

Die Werte werden in der TTN-Console unter Data angezeigt:

Applications	> 🥥 la	ora32_test_b	enno > De	vices >	nelte	ec32 > E	Data					
										Overview	Data	Settings
APPLIC	ATION	DATA									II pause	e 🗑 <u>clear</u>
Filters	uplink	downlink	activation	ack	error							
21	time •43•41	counter	port 1		avload: 61 (	1 00 00 0	0,00,00,000	D00 enoch: 35	3 temperature: 13	1		
	.40.41			P	ayioau. OI (	51 00 00 0	00000000	epoch. 33	o temperature. xo			

 $\rightarrow$  Payload wird mithilfe des Decoders dekodiert!

# Aufgabenstellung:

- Eingang der Alarmanlage ist ein digitales Signal (ESP Eingang Pin0 wird mit GND verbunden)
- Wenn die Alarmanlage auslöst soll eine LoRaWAN-Botschaft gesendet werden
- Jedes mal wenn die Alarmanlage auslöst soll ein Zähler hochgezählt werden

Normalerweise verschwinden die übermittelten Werte nach kurzer Zeit wieder aus der TTN-Console. Um die Werte zu sichern, verwenden wir die **Storage Integration**, mit deren Hilfe sich Werte bis zu 7 Tage speichern und auch abrufen lassen.

# Storage Integration Integration einbinden

#### Unter Integrations auf add intregration klicken:

Applications > <a>&gt;</a> lora32_test_benno > Integrations						
	Overview	Devices	Payload Formats	Integrations	Data	Settings
INTEGRATIONS					• add	Lintegration

# Anschließend die *Data Storage Integration* auswählen und auf *Add integration* klicken:



# Storage Integration Daten abrufen

### In der Integration-Overview auf go to platform klicken:

		Overview	Devices	Payload Formats	Integrations	Data	Settings
INTEGRATION OVERV	TEW						
Status Integration info Platform Author Description	Running     DataStorage (v2.0.1)     The Things Industries BV. Stores data and makes it available through an APL	Your data is sto	red for seven	ı days.			

Auf *Authorize* klicken und den Access Key aus der Application Overview eingeben. Anschließend hat man verschiedene Möglichkeiten, die Daten abzurufen.

# Storage Integration Daten abrufen

# Über einen Klick auf *Try it out* können Datensätze abgerufen werden, der ensprechende Curl-Befehl für die Kommandozeile kann kopiert werden (mit " statt ')



#### Storage Integration Daten abrufen mit Python

```
#
    Imports...
1
2
3 context = ssl.create_default_context()
  context.check hostname = False
4
5
6 url = 'https://lora32_test_benno.data.thethingsnetwork.
      org/api/v2/query'
7
  args = '?last=7d'
8
9
10 access_key = '<access-key>'
  headers = {
11
           'Accept': 'application/json',
12
           'Authorization': 'key ' + access_key
13
14
15 req = urllib.request.Request(url + args, headers=headers)
16
17 # Datenverarbeitung...
```