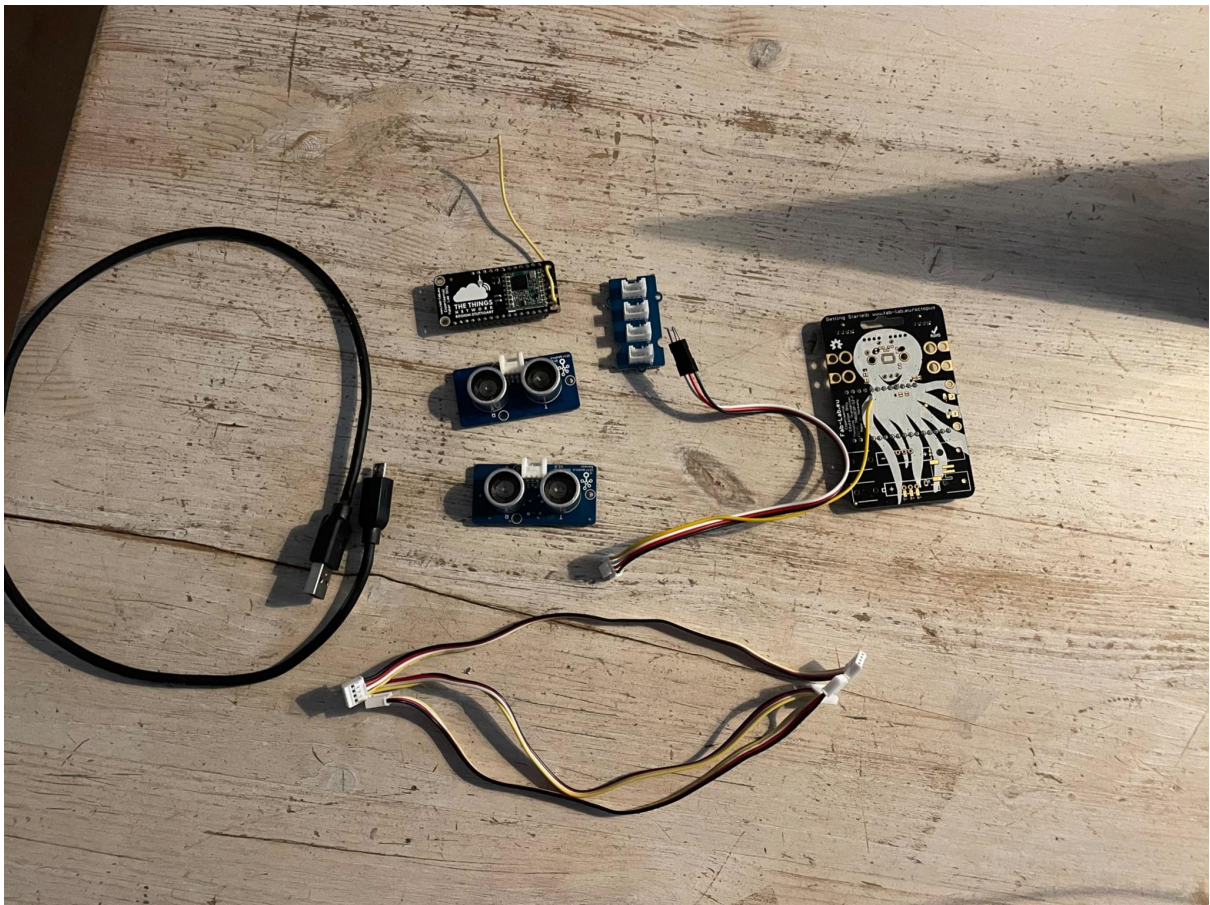


## Dokumentation Personenzähler

### Was wird benötigt?

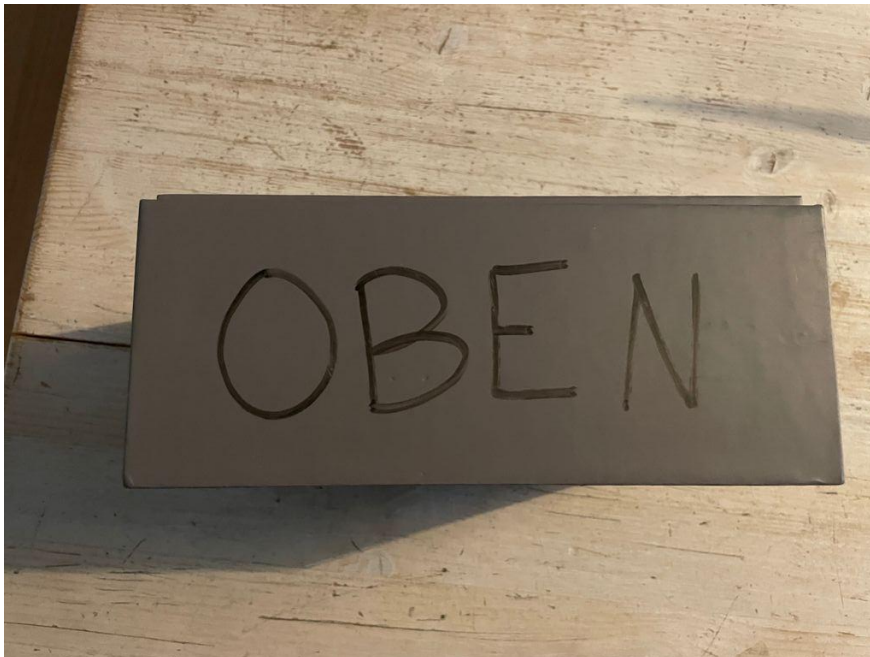
- 1 X IoT Octopus
- 1 X LoRa-Wings
- 4 X Jumper Kabel (digital)
- 2 x Ultrasonic Ranger Ultraschallsensor
- 1 X Mini USB Kabel
- 1 x HUB
- 1 x löten des gelben Kabels (für die Datenübertragung an den Port 0 des IoT Octopus)
- 1 X LoRaWAN Gateway - siehe Anleitung Dokumentation zur Verwendung von LoRaWAN



## Allgemeine Hinweise und Funktionsweise der Schrank

Nachfolgend wird beschrieben wie die Hardware vorbereitet und der Code übertragen werden kann. Hierfür werden zunächst die Schritte erläutert und nachfolgend auch durch Bilder/Screenshots dargestellt.

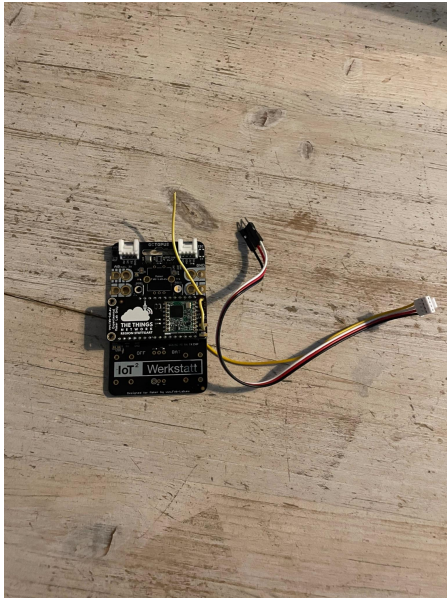
Die Schranke misst die Anzahl der Menschen die sich in einem Raum befinden. Bei Anschluss der Schranke werden die Abstände, die die beiden Ultraschallsensoren messen als Grundwert genommen. Daher sollte die Schranke im Türrahmen so befestigt werden, dass die Sensoren die gegenüberliegende Türseite messen können. Verändert sich dieser Wert wird eine von zwei Möglichen Sequenzen ausgelöst. Ändert sich der gemessene Wert zuerst bei Sensor 1 und dann 2, geht eine Person in den Raum und die Zahl der gesamten Menschen erhöht sich um 1. Misst zuerst der Sensor 2 und dann 1 einen anderen Wert, verlässt jemand den Raum und die gesamte Anzahl reduziert sich um 1. Da sich nicht negativ viele Menschen in einem Raum befinden können, ist der default-Wert 0. Um die Berechnung richtig laufen lassen zu können, muss die Box daher so angebracht werden, dass das "OBEN" bei Anbringung an den Türrahmen gelesen werden kann. Ebenfalls sollte die Schranke an der rechten Türseite angebracht werden. Dabei ist die Ausgangsposition aus dem heraus.



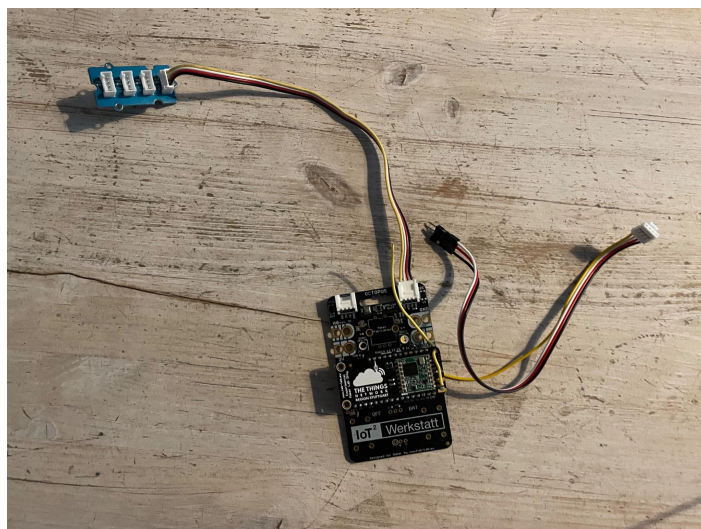
## Hardware vorbereiten

Wir beginnen damit alle Bauteile an den Octopus anzuschließen. Hierbei wie folgt vorgehen:

1. Datenkabel (gelb) des digitalen Jumpers an den Port 0 des Octopus löten
2. LoRA-Wings auf den Octopus stecken



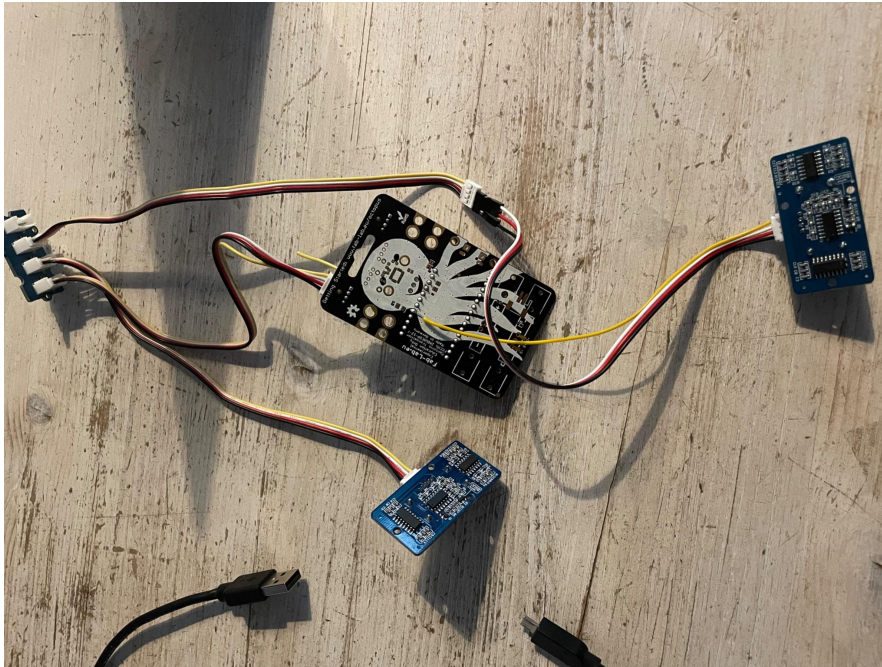
3. Den HUB mit einem digitalen Jumper-Kabel mit dem Grove I<sup>2</sup>C Anschluss (oben



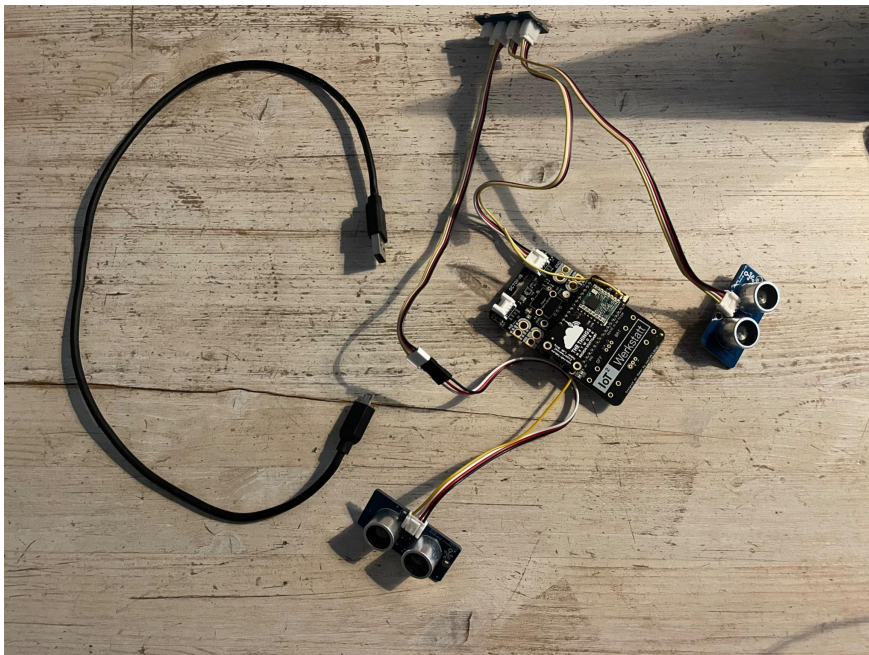
rechts) verbinden.

4. Einen der beiden Sensoren mit einem digitalen Jumper-Kabel mit dem HUB verbinden.





5. Den digitalen Jumper der mit dem gelben Kabel an den Octopus gelötet ist mit



6. Das Mini USB Kabel mit dem Octopus verbinden

### **Code kopieren, vorbereiten und übertragen**

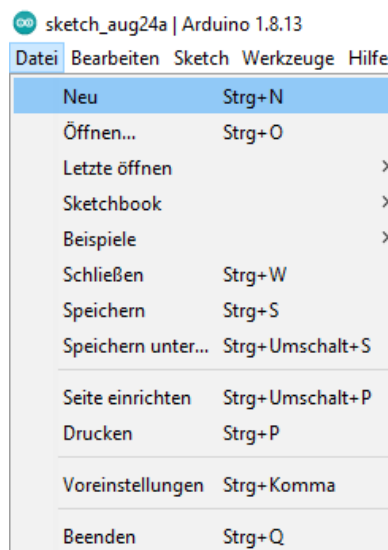
Nachdem die Hardware miteinander verbunden wurde können wir den Code auf den Octopus übertragen. Hierbei wie folgt vorgehen:

Achtung: Bevor diese Schritte durchgeführt werden, sollte die Dokumentation zur Verwendung von LoRaWAN abgeschlossen sein.

1. Code kopieren:
2. Arduino IDE öffnen und eine neue Datei erstellen.
3. Alles markieren (Strg + A) und den Code einfügen.
4. Die Datei abspeichern.
5. Falls gewünscht die Grenzwerte für die Lautstärke anpassen (green, yellow, red).
6. Auf der Seite [thethingsnetwork.org](http://thethingsnetwork.org) das gewünschte end device auswählen.
7. Im Bereich Overview die AppEUI, DevEUI und den AppKey kopieren und im Code einfügen. Achtung, die richtige Formatierung (<>) und Sortierung lsb beachten!
8. Die Datei erneut speichern.
9. In der Arduino IDE unter Werkzeuge den Upload Speed auf 115200 stellen und den richtigen Port auswählen.
10. Das LoRaWAN Gateway an den Strom anschließen
11. Den Octopus über das USB Kabel mit dem Computer verbinden und den Code auf den Octopus übertragen.
12. Wenn die Übertragung abgeschlossen ist wird in der Konsole "Hard resetting via RTS pin ..." angezeigt.
13. Auf Werkzeuge → Serieller Monitor klicken um die Ausgaben zu überwachen.
14. Auf der Seite [thethingsnetwork.org](http://thethingsnetwork.org) sollten die gesendeten Daten im Bereich Live data angezeigt werden.

Screenshots zu den einzelnen Schritten:

Schritt 1:



Schritt 2:

sketch\_aug24a | Arduino 1.8.13

Datei Bearbeiten Sketch Werkzeuge Hilfe

```
✓ ↻ 📄 ⬆ ⬇
sketch_aug24a
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

#### Schritt 4:

sketch\_aug24a | Arduino 1.8.13

Datei Bearbeiten Sketch Werkzeuge Hilfe

Neu	Strg+N	
Öffnen...	Strg+O	
Letzte öffnen		>
Sketchbook		>
Beispiele		>
Schließen	Strg+W	
Speichern	Strg+S	
Speichern unter...	Strg+Umschalt+S	
Seite einrichten	Strg+Umschalt+P	
Drucken	Strg+P	
Voreinstellungen	Strg+Komma	
Beenden	Strg+Q	

#### Schritt 5:

Ampel | Arduino 1.8.13

Datei Bearbeiten Sketch Werkzeuge Hilfe

```
✓ ↻ 📄 ⬆ ⬇
Ampel$
// Sound Sensor
#define SoundSensorPin A0
#define VREF 5.0

float voltageValue, dbValue;
int intDbValue;
int sumDbValues = 0;
int numberOfValues = 0;
int averageDbValue = 0;

// Diese Werte (in db) anpassen um die Grenzen der Ampel zu verändern
// Ein Geräuschpegel unter diesem Wert wird als grün angezeigt
float green = 70.0;
// Ein Geräuschpegel unter diesem Wert und über dem green Wert wird als gelb angezeigt
float yellow = 80.0;
// Ein Geräuschpegel über diesem Wert wird als rot angezeigt (sollte dem yellow Wert entsprechen)
float red = 80.0;
```

#### Schritt 6:

End devices (2)


Search by ID

Import end devices

Add end device

ID	Name	DevEUI	JoinEUI	Last seen
eui-70b3d57ed00446bd-ampel		70 B3 D5 7E D0 04 46 BD	00 00 00 00 00 00 00 00	3 hours ago
eui-70b3d57ed00446e7-schranke		70 B3 D5 7E D0 04 46 E7	00 00 00 00 00 00 00 00	Unknown

Schritt 7:



**eui-70b3d57ed00446bd-ampel**  
 ID: eui-70b3d57ed00446bd-ampel

Last seen 3 hours ago
 

↑ 48

↓ 22

Overview

Live data

Messaging

Location

Payload formatters

Claiming

General information

End device ID

eui-70b3d57ed00446bd-ampel

Description

This end device has no description

Created at

Aug 24, 2021 10:57:09

Activation information

AppEUI

0x00, 0x00, 0x00, 0x00, 0x00, 0x...

DevEUI

0xBD, 0x46, 0x04, 0xD0, 0x7E, 0x...

Root key ID

n/a

AppKey

0x31, 0xE3, 0x31, 0xBE, 0x63...

NwkKey

n/a

```

sketch_aug24a$

//LoRaWAN (c) 2015 Thomas Telkamp and Matthijs Kooijman
#include <lmic.h>
#include <hal/hal.h>
#include <ESP8266WiFi.h>

const lmic_pinmap lmic_pins = {
    .nss = 2, // Connected to pin D
    .rxtx = LMIC_UNUSED_PIN, // For placeholder only, Do not connected on RFM92/RFM95
    .rst = LMIC_UNUSED_PIN, // Needed on RFM92/RFM95? (probably not) D0/GPIO16
    .dio = {
        15, 15, LMIC_UNUSED_PIN
    }
};

static const u1_t PROGMEM DEVEUI[8]={
    0xBD,0x46,0x04,0xD0,0x7E,0xD5,0xB3,0x70};
void os_getDevEui (u1_t* buf) {
    memcpy_P(buf, DEVEUI, 8);
}

static const u1_t PROGMEM APPEUI[8]={
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
void os_getArtEui (u1_t* buf) {
    memcpy_P(buf, APPEUI, 8);
}

static const u1_t PROGMEM APPKEY[16]={
    0xA3,0x08,0x97,0x62,0xA4,0xB6,0x7F,0x48,0x05,0xD2,0xE5,0x63,
    0x00,0x00,0x00,0x00};
void os_getDevKey (u1_t* buf) {
    memcpy_P(buf, APPKEY, 16);
};

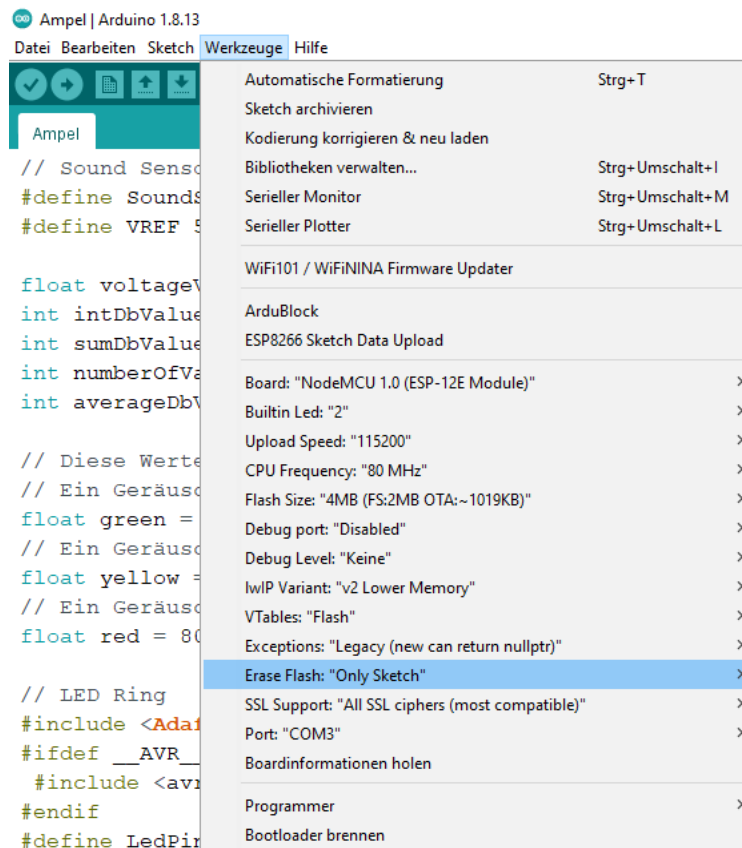
```

## Schritt 8:

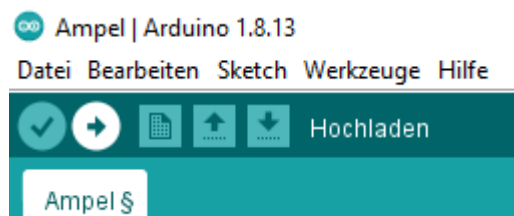
Neu	Strg+N	
Öffnen...	Strg+O	
Letzte öffnen		>
Sketchbook		>
Beispiele		>
Schließen	Strg+W	
Speichern	Strg+S	
Speichern unter...	Strg+Umschalt+S	
Seite einrichten	Strg+Umschalt+P	
Drucken	Strg+P	
Voreinstellungen	Strg+Komma	
Beenden	Strg+Q	



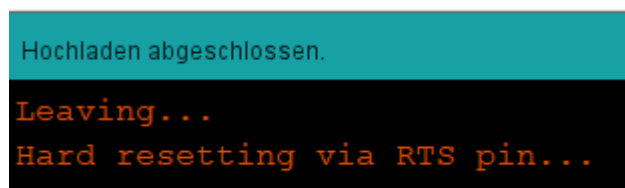
## Schritt 9:



## Schritt 10:



## Schritt 11:



Schritt 12:

Ampel | Arduino 1.8.13

Datei Bearbeiten Sketch Werkzeuge Hilfe

Ampel

// Sound Sensor  
#define SOUND\_SENSOR\_PIN 2  
#define VREF 5  
  
float voltageV;  
int intDbValue;  
int sumDbValue;  
int numberOfValues;  
int averageDbValue;  
  
// Diese Werte sind für den Sound Sensor  
// Ein Geräusch in dB  
float green = 0;  
// Ein Geräusch in dB  
float yellow = 0;  
// Ein Geräusch in dB  
float red = 0;  
  
// LED Ring  
#include <Adafruit\_NeoPixel.h>  
#ifdef \_\_AVR\_\_  
#include <avr/io.h>  
#endif  
#define LED\_PIN 13

Automatische Formatierung Strg+T

Sketch archivieren

Kodierung korrigieren & neu laden

Bibliotheken verwalten... Strg+Umschalt+I

**Serieller Monitor Strg+Umschalt+M**

Serieller Plotter Strg+Umschalt+L

WiFi101 / WiFinINA Firmware Updater

ArduBlock

ESP8266 Sketch Data Upload

Board: "NodeMCU 1.0 (ESP-12E Module)" >

Built-in LED: "2" >

Upload Speed: "115200" >

CPU Frequency: "80 MHz" >

Flash Size: "4MB (FS:2MB OTA:~1019KB)" >

Debug port: "Disabled" >

Debug Level: "Keine" >

hwlp Variant: "v2 Lower Memory" >

VTables: "Flash" >

Exceptions: "Legacy (new can return nullptr)" >

Erase Flash: "Only Sketch" >

SSL Support: "All SSL ciphers (most compatible)" >

Port: "COM3" >

Boardinformationen holen

Programmer >

Bootloader brennen

Schritt 13:

THE THINGS NETWORK

THE THINGS STACK Community Edition

Overview Applications Gateways Organizations

LoRaWAN Octopus

Overview

**End devices**

Live data

Payload formatters

Integrations

Collaborators

API keys

General settings

Applications > LoRaWAN Octopus > End devices > eui-70b3d57ed00446bd-ampel > Live data

eui-70b3d57ed00446bd-ampel

ID: eui-70b3d57ed00446bd-ampel

Last seen info unavailable 48 n/a

Overview **Live data** Messaging Location Payload formatters Claiming General settings

Time	Type	Data preview
12:37:27	Forward uplink data message	Payload: { field1: 72 } 48 FPort: 1 Data rate: SF9BW125 SNR: -13.25 RSSI: -110
12:36:06	Forward uplink data message	Payload: { field1: 70 } 46 FPort: 1 Data rate: SF9BW125 SNR: -8 RSSI: -120
12:35:26	Forward uplink data message	Payload: { field1: 88 } 58 FPort: 1 Data rate: SF9BW125 SNR: -10.5 RSSI: -121
12:32:47	Forward uplink data message	Payload: { field1: 79 } 4F FPort: 1 Data rate: SF9BW125 SNR: -10.5 RSSI: -109
12:32:06	Forward uplink data message	Payload: { field1: 80 } 58 FPort: 1 Data rate: SF9BW125 SNR: -11.5 RSSI: -111
12:30:46	Forward uplink data message	Payload: { field1: 89 } 59 FPort: 1 Data rate: SF9BW125 SNR: -6.5 RSSI: -113
12:28:06	Forward uplink data message	Payload: { field1: 87 } 57 FPort: 1 Data rate: SF9BW125 SNR: -11.25 RSSI: -113
12:26:45	Forward uplink data message	Payload: { field1: 85 } 55 FPort: 1 Data rate: SF9BW125 SNR: -7.5 RSSI: -116
12:26:06	Forward uplink data message	Payload: { field1: 93 } 60 FPort: 1 Data rate: SF9BW125 SNR: -9.5 RSSI: -114
12:24:44	Forward uplink data message	Payload: { field1: 91 } 58 FPort: 1 Data rate: SF9BW125 SNR: -9.25 RSSI: -107
12:24:03	Forward uplink data message	Payload: { field1: 74 } 4A FPort: 1 Data rate: SF9BW125 SNR: -7 RSSI: -117
12:23:24	Forward uplink data message	Payload: { field1: 91 } 58 FPort: 1 Data rate: SF9BW125 SNR: -6.5 RSSI: -109
12:21:24	Forward uplink data message	Payload: { field1: 71 } 47 FPort: 1 Data rate: SF9BW125 SNR: -6.5 RSSI: -116
12:19:22	Forward uplink data message	Payload: { field1: 75 } 4B FPort: 1 Data rate: SF9BW125 SNR: -8.25 RSSI: -113