

# Budowa lokalizatora GPS z interfejsem LoRaWAN



Pracownia, Bydgoszcz 2019

# Plan warsztatów

- Technologia LoRa/LoRaWAN
- The Things Network
  - zakładanie konta i realizacja komunikacji w sieci TTN
- Obserwacja pasma - spektrogram
- Lokalizator
  - realizacja komunikacji z modułem GNSS
- Sposoby integracji z TTN
  - przesyłanie danych GPS przez sieć TTN
- Platforma Signomix
  - zakładanie konta i realizacja wizualizacji danych
- Ciekawostki
- Pytania i odpowiedzi

# LoRa i LoRaWAN



# Sieci LPWAN (*Low Power Wide Area Network*)

Podobne do sieci komórkowych lecz zoptymalizowane dla IoT

- sieć stacji bazowych (struktura gwiazdy)
- duży zasięg stacji (kilometry)
- niska moc sygnału nadawanego
- urządzenia końcowe zasilane baterijnie
- zwykłe wykorzystywane pasma otwarte (< 1GHz)

Ograniczenia związane z optymalizacją dla IoT

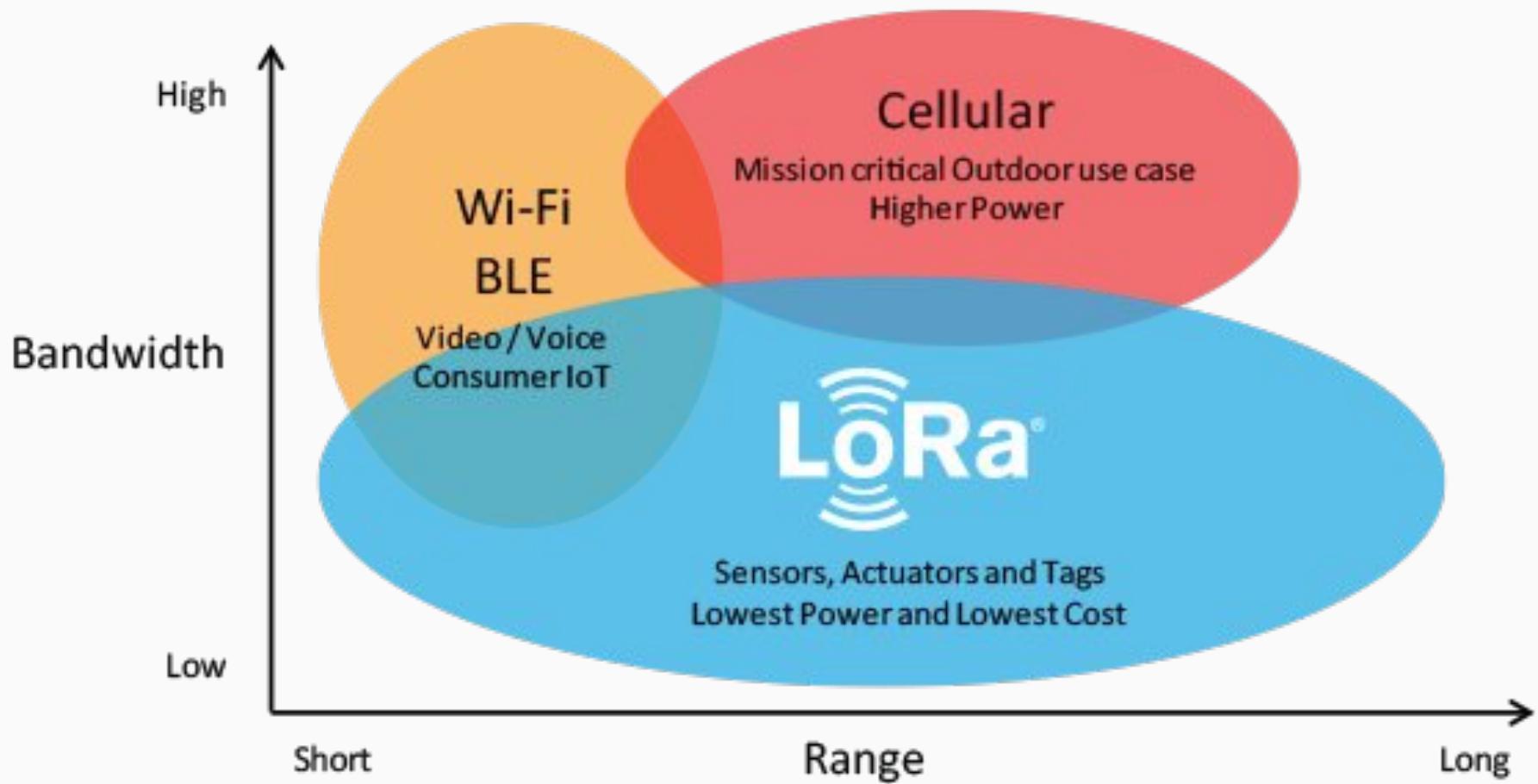
- surowe zasady dotyczące zajętości łącza
- bardzo zwięzłe i rzadko wysyłane wiadomości
- mała szybkość transmisji

LoRaWAN, Sigfox, NB-IoT (odpowiedź 3GPP)



**NB-IoT**

# Technologie LPWAN



źródło: <https://www.semtech.com/lora>

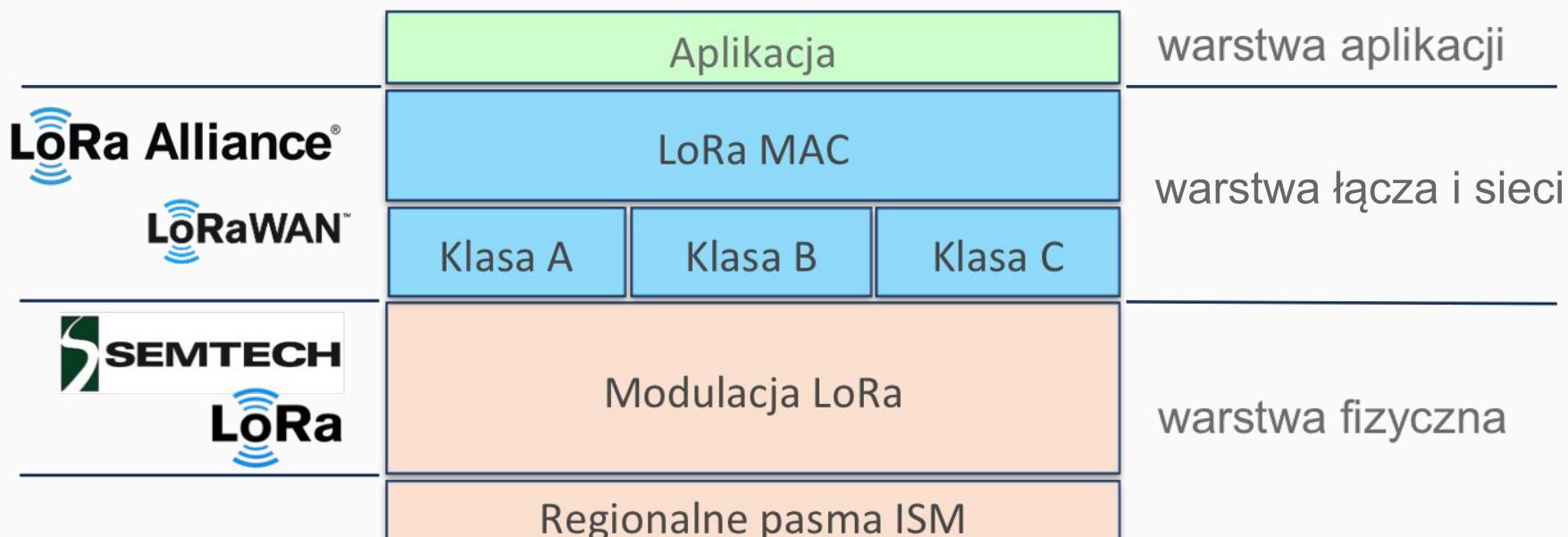
# LoRa i LoRaWAN

## LoRa

- warstwa fizyczna dla komunikacji bezprzewodowej dalekiego zasięgu

## LoRaWAN

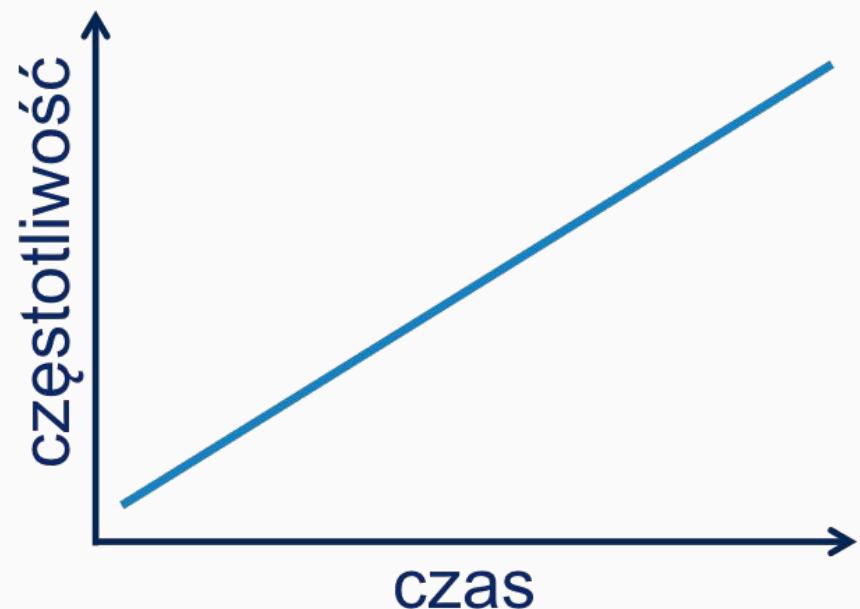
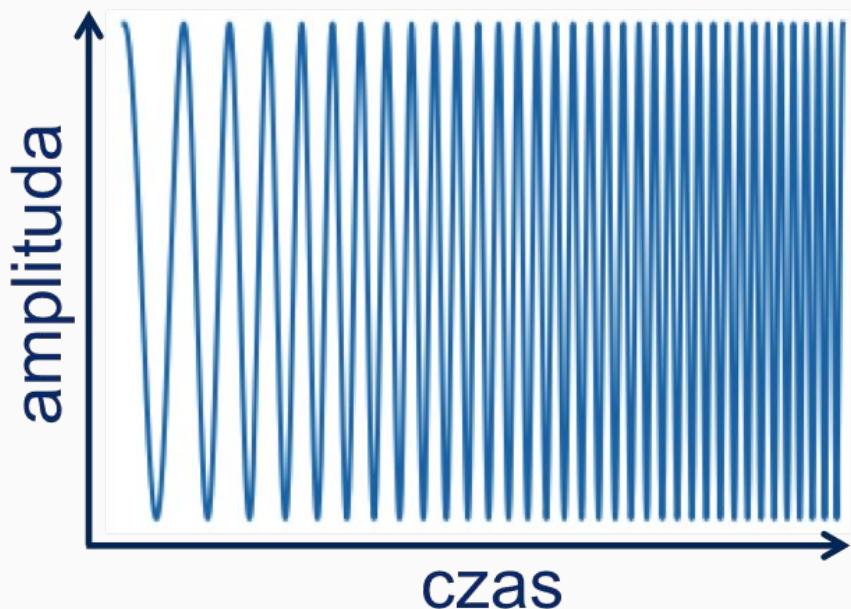
- definiuje architekturę sieci i protokół kontroli dostępu do nośnika (MAC) wykorzystujące warstwę fizyczną LoRa i FSK



## LoRa - modulacja

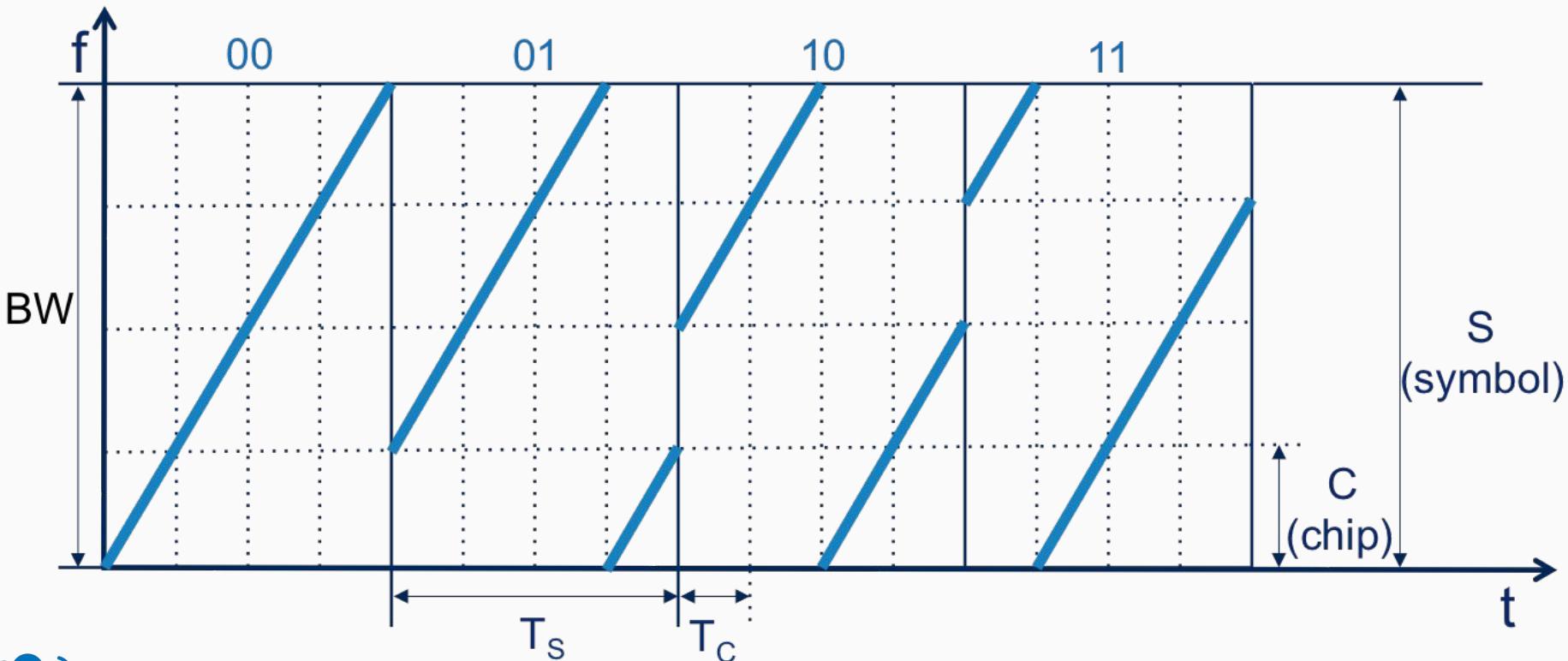
Oparta na technologii rozpraszania widma CSS (*Chirp Spread Spectrum*)

*Chirp* - sygnał o rosnącej lub malejącej częstotliwości

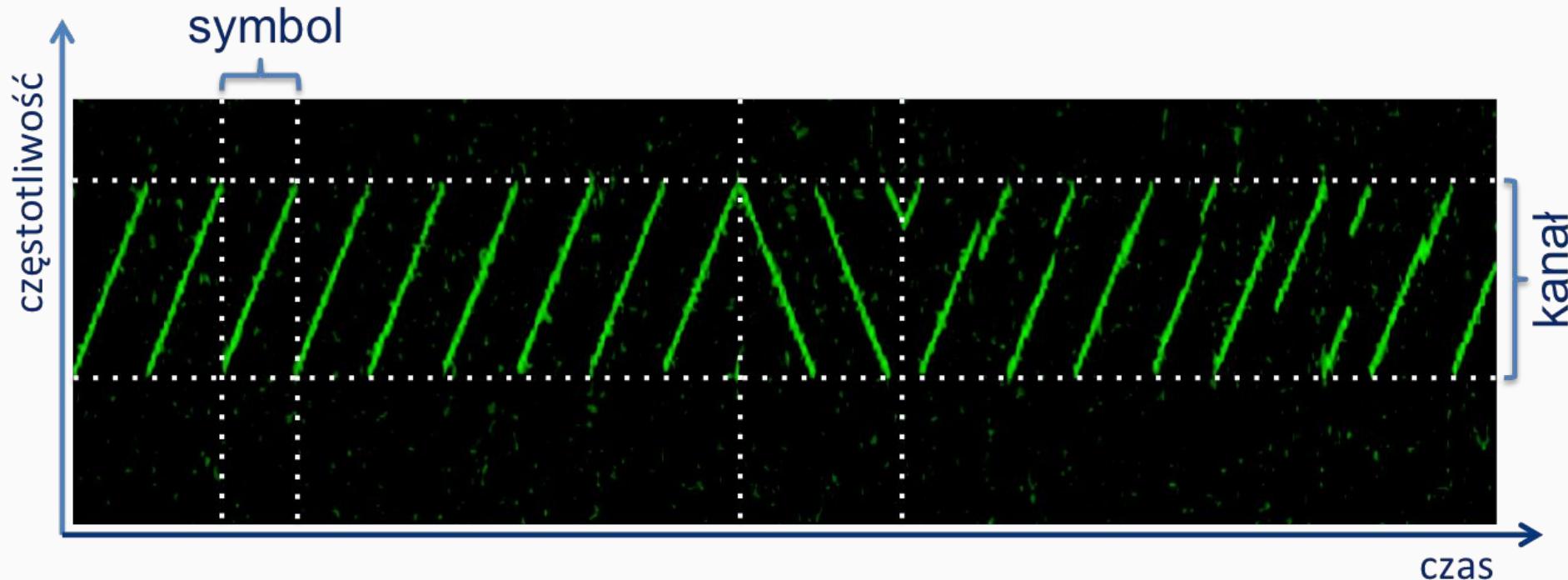


## Przykład

- SF2 - 2 bity/symbol
- symbol - sekwencja 4 chipów ( $2^{SF}$ )

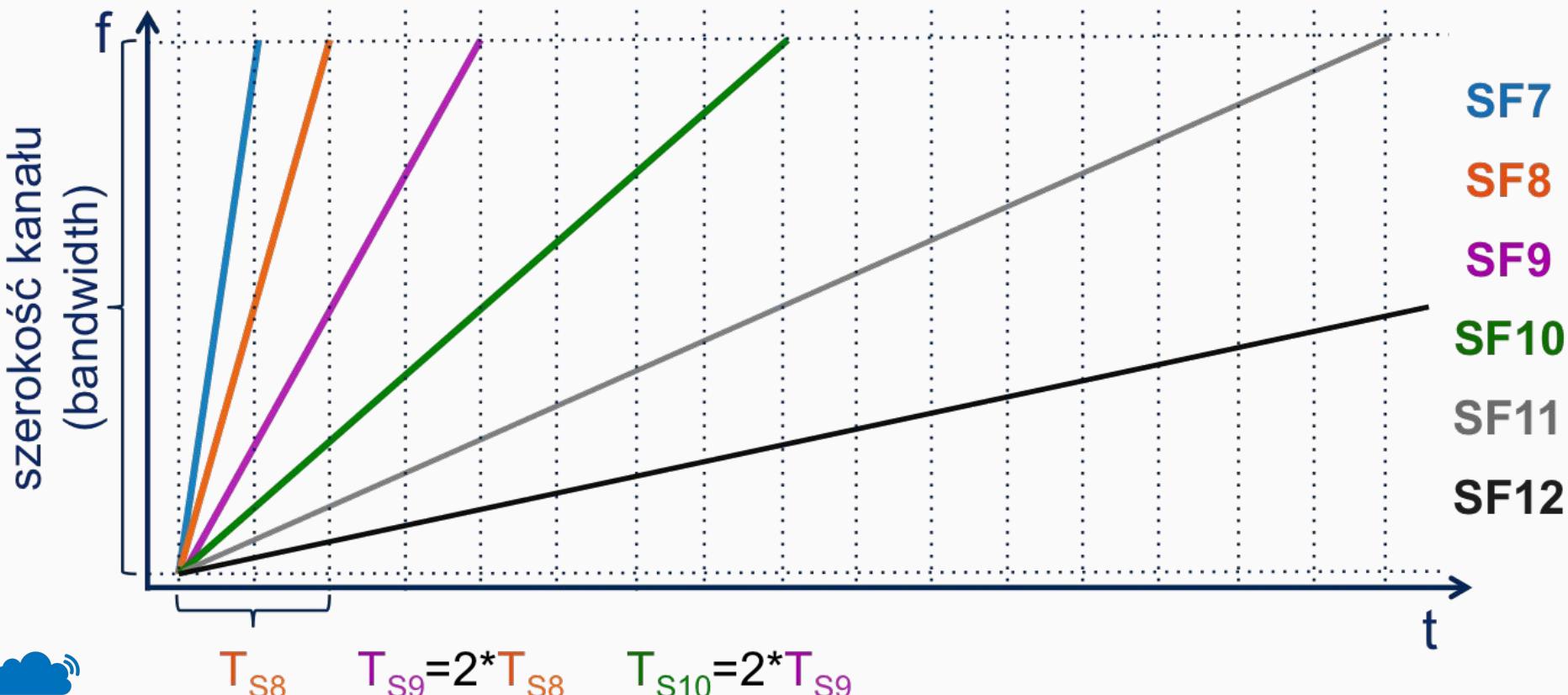


## LoRa - modulacja - struktura pakietu



## SF (*Spreading Factor*)

- współczynnik rozprzestrzeniania/rozproszenia
- liczba bitów przypadająca na symbol



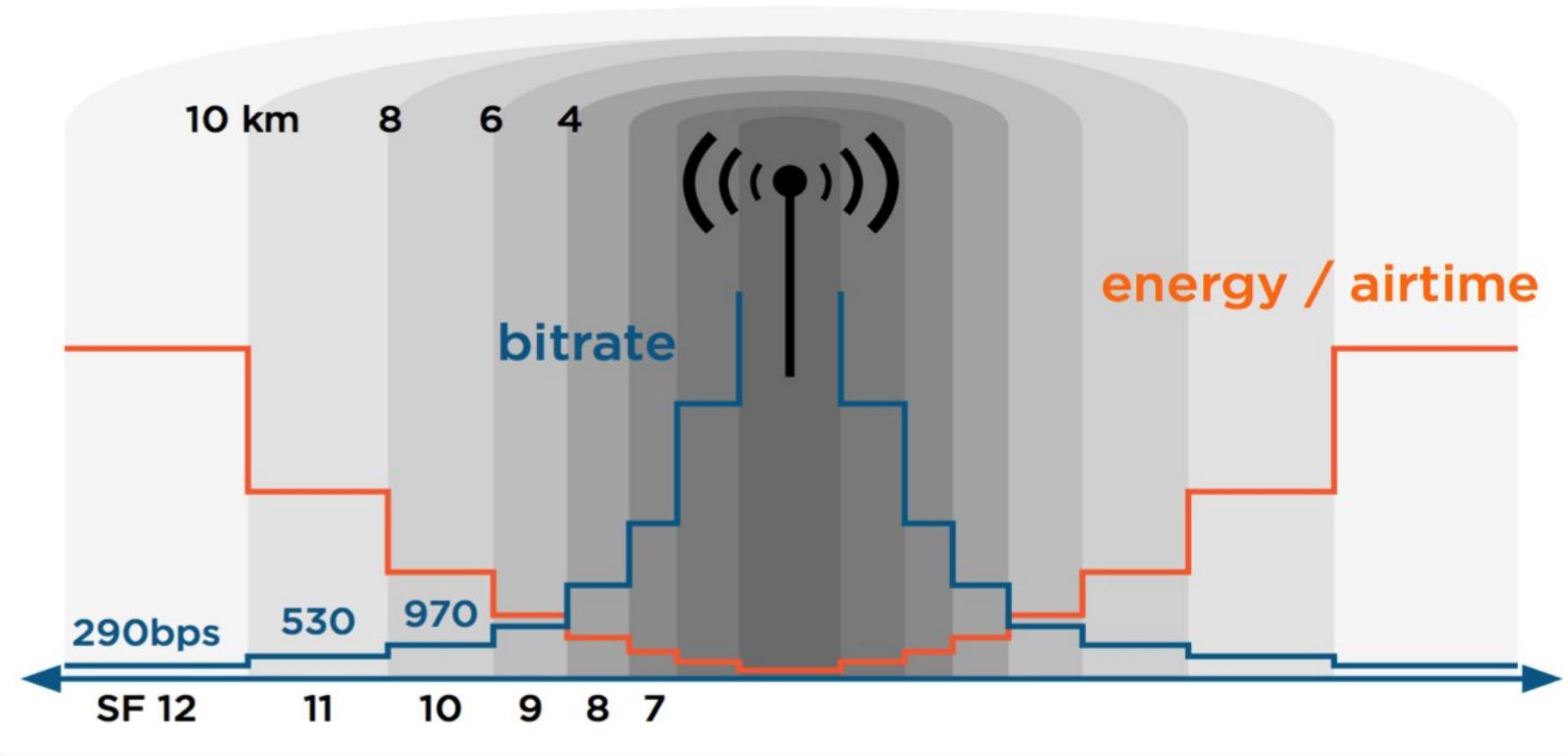
## LoRa - transmisja c.d.

BW = 125 kHz

CR = 4/5

SF	chip/symbol	DR (bit/s)	Czułość odbiornika (dBm)	Czas transmisji (10 bajtów)
7	128	5468	-123	56 ms
8	256	3125	-126	103 ms
9	512	1757	-129	205 ms
10	1024	976	-132	371 ms
11	2048	537	-135	741 ms
12	4096	292	-137	1483 ms

## LoRa - transmisja

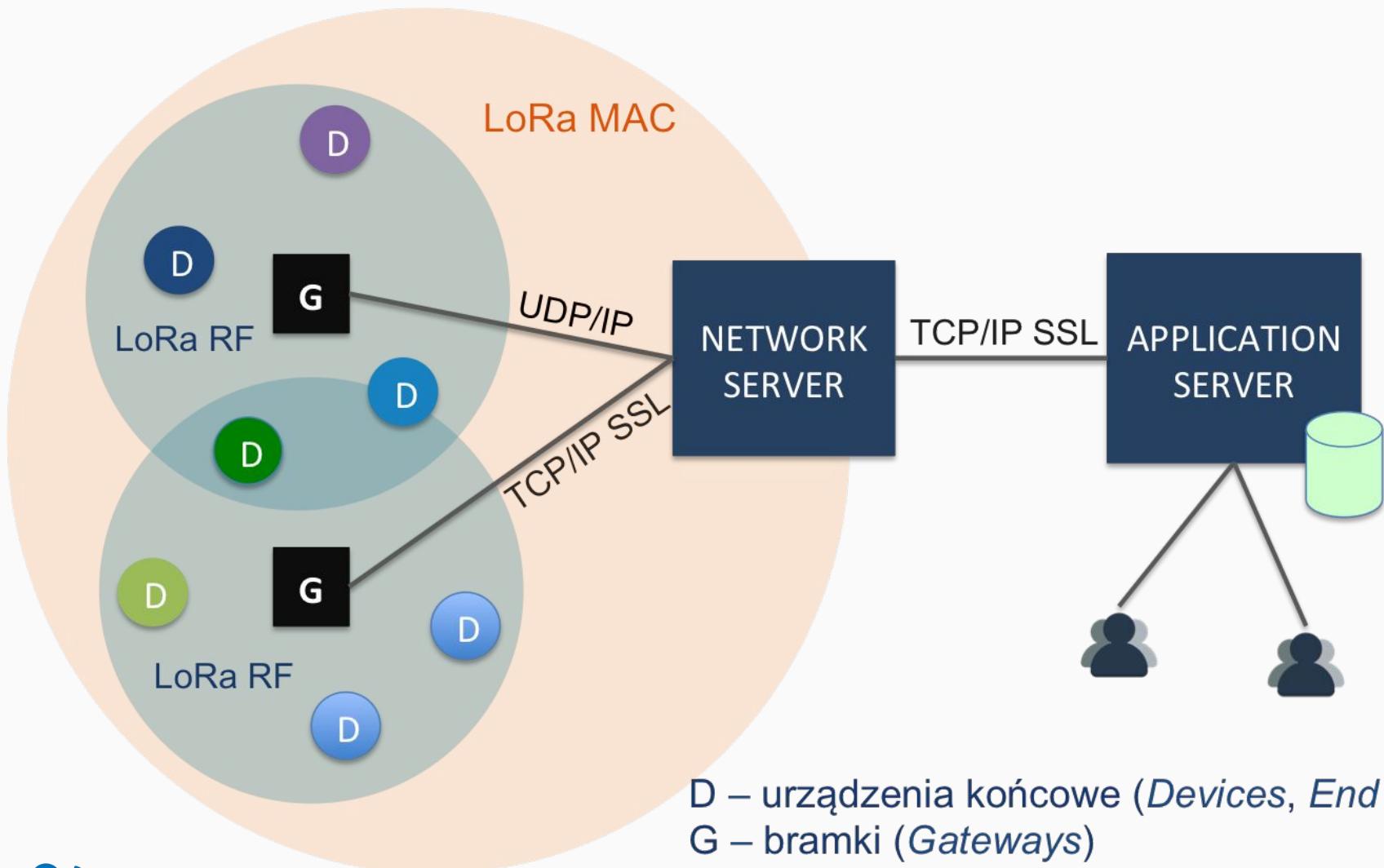


### Układy Semtech SX1272, SX1276, SX1278

- różne zakresy częstotliwości (860...1000 MHz, 137...1020 MHz, 137...525 MHz)
- maksymalny budżet łącza (do 157 dBm lub 168 dBm)
- wysoka czułość (do -137 dBm lub -148 dBm)
- obsługa innych rodzajów modulacji (np. FSK, OOK)
- moc nadajnika do +20 dBm (100 mW)

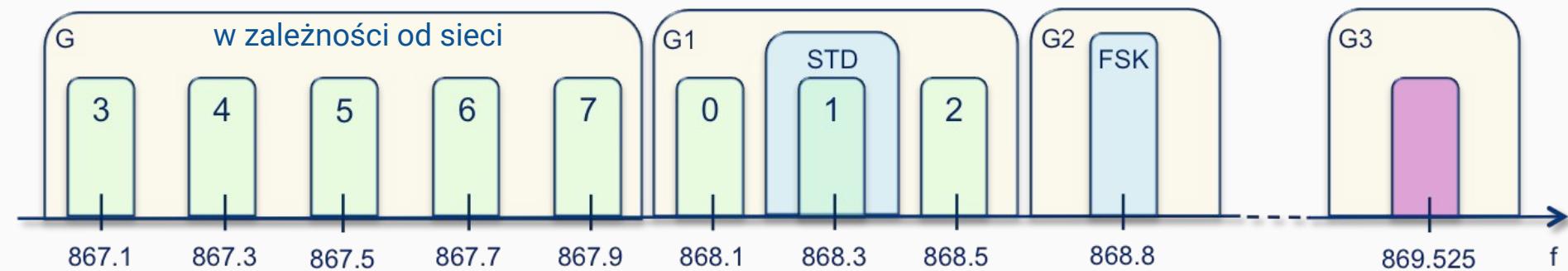


# LoRaWAN - uproszczona struktura sieci



## LoRaWAN - podział pasma dla EU (przykład)

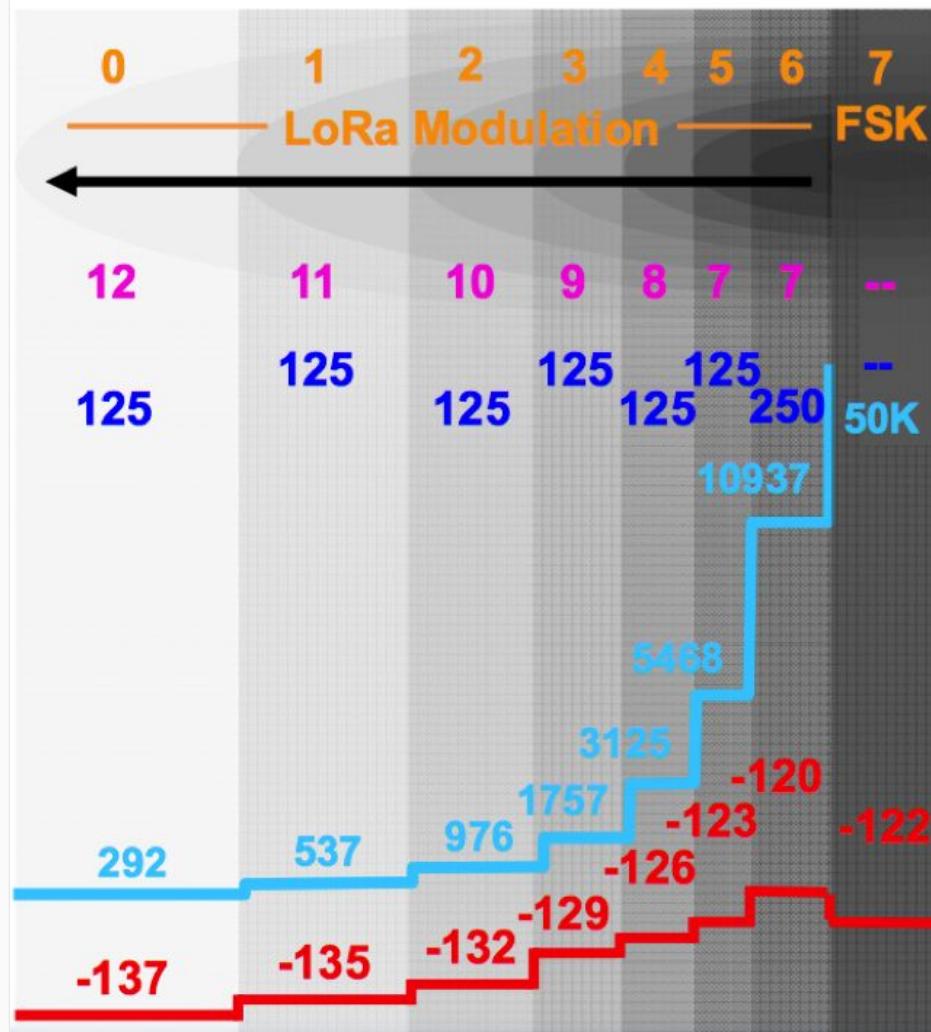
Nazwa	Zakres (MHz)	Moc (dBm)	Zajętość (%)
G	867.0 - 868.0	14	1
G1	868.0 - 868.6	14	1
G2	868.7 - 869.2	14	0.1
G3	869.4 - 869.65	27	10
G4	869.7 - 870.0	14	1



## LoRaWAN - DR (*Data Rate*)

DR	Modulacja	SF	BW (kHz)	bit/s	Maks. liczba bajtów
0	LoRa	12	125	250	51
1	LoRa	11	125	440	51
2	LoRa	10	125	980	51
3	LoRa	9	125	1760	115
4	LoRa	8	125	3125	242
5	LoRa	7	125	5470	242
6	LoRa	7	250	11000	242
7	FSK			50000	

# LoRa - transmisja



**Data Rate (DR)**

**Range**

**Spreading Factor (SF)**

**Bandwidth (BW) (kHz)**

**Bitrate (BR) (bps)**

**Receive Sensitivity (dBm)**

# LoRaWAN - klasy urządzeń końcowych

czas pracy na baterii ↑

A

## CZUJNIKI ZASILANE BATERYJNIE

- odbiór możliwy tylko po nadaniu wiadomości (dwa krótkie okna odbiorcze)
- klasa obowiązkowa dla wszystkich urządzeń

B

## URZĄDZENIA WYK. ZASILANE BATERYJNIE

- komunikacja szczelinowa z synchronizacją (dodatkowe okna odbiorcze w określonych odstępach czasu)

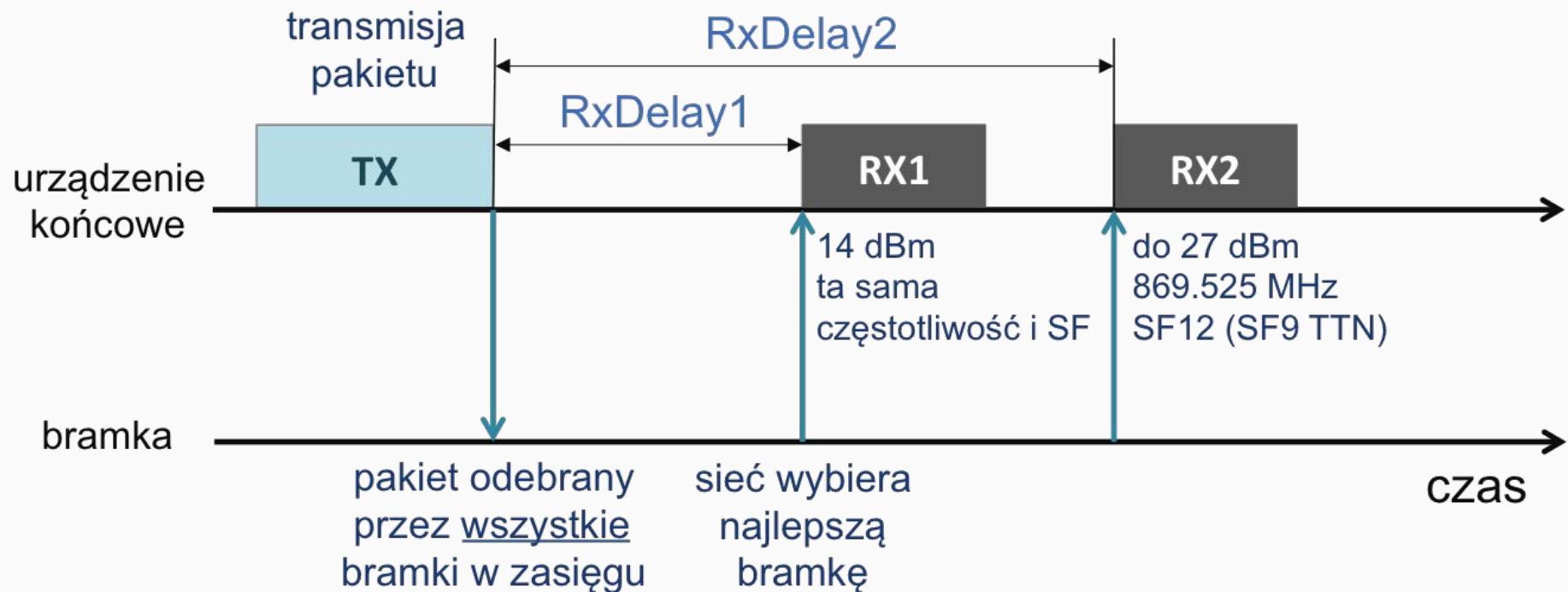
C

## URZĄDZENIA WYK. ZASILANE Z SIECI

- duże zużycie energii
- urządzenia praktycznie w ciągłym nasłuchu
- brak opóźnień w komunikacji do urządzenia

opóźnienie w przesyłaniu informacji do urządzenia →

# LoRaWAN - klasa A - nadawanie i odbiór



## Sposoby realizacji:

- nadajnik-odbiornik z wbudowanym stosem LoRaWAN
  - przykład - Microchip RN2483, Murata CMWX1ZZABZ
  - można skupić się tylko na docelowej funkcjonalności
  - możliwa implementacja własnego oprogramowania na wbudowanym mikrokontrolerze z wykorzystaniem udostępnionego stosu LoRaWAN
  - precertyfikacja stosu LoRaWAN
- nadajnik-odbiornik bez stosu LoRaWAN (tylko modulacja)
  - przykład - RFM95W
  - mikrokontroler zewnętrzny musi obsłużyć komunikację LoRaWAN

## LoRaWAN - bramki

- Wielokanałowość i multi-SF
- Semtech SX1301 z SX125X
- Minimalne oprogramowanie - tylko do przesyłania pakietów (*Packet Forwarder*)
- Możliwość instalacji dodatkowego oprogramowania



## LoRaWAN - pojemność bramki

### Długość przesyłanych informacji

- wpływ na zajętość kanału (ograniczenie w specyfikacji LoRaWAN)

### Częstotliwość wysyłania informacji

- wpływ na zajętość kanału (ograniczenie w zaleceniach ETSI)

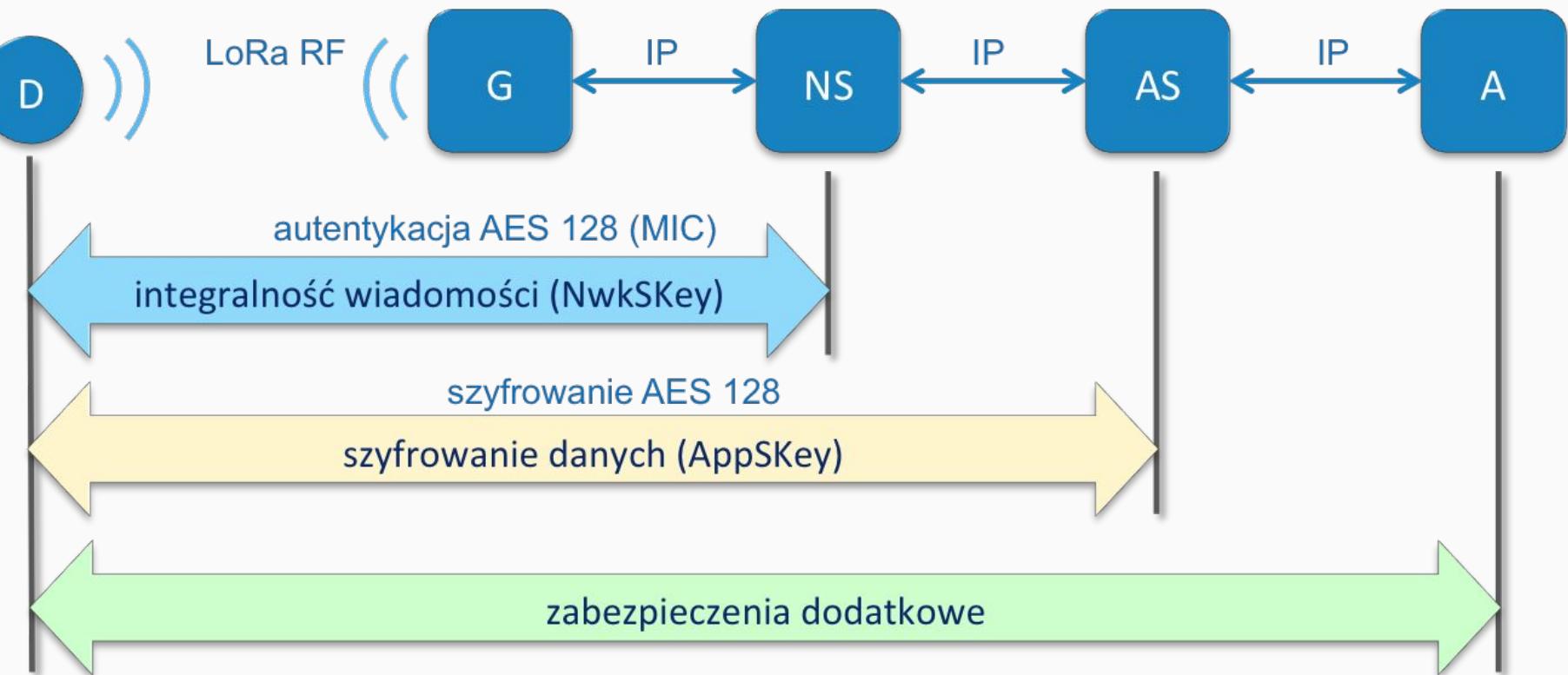
### Odległość urządzeń końcowych od bramki

- wpływ na zajętość kanału (zmiana SF → czas transmisji)
- moc nadajnika ograniczona w zaleceniach ETSI

### Liczba dostępnych kanałów

- zwiększenie liczby kanałów (ograniczenie w zaleceniach ETSI)
- praca z różnym SF (zmiana SF → czas transmisji)

# LoRaWAN - bezpieczeństwo



# LoRaWAN - bezpieczeństwo

## Dwie warstwy zabezpieczeń

- sieci (*Network Session Key*)
  - uwierzytelnia węzeł w sieci poprzez sprawdzanie integralności wiadomości
- aplikacji (*Application Session Key*)
  - szyfrowanie danych aplikacji
  - warstwa sieci nie ma dostępu do danych aplikacji

## Stosowana technologia

- AES 128 (klucze 128 bitów - unikalne dla każdej sesji)
- dane mogą być opcjonalnie zabezpieczone silniejszym algorytmem (*end-to-end*)

## Adres urządzenia (*Device Address*)

- unikalny w obrębie sieci - umożliwia wybór prawidłowych kluczy

## Sposoby ustalania informacji szyfrujących

- prekonfiguracja urządzenia
- OTA (*Over-The-Air*)

## ABP (*Activation By Personalization*)

- *DevAddr*, *NwkSKey* i *AppSKey* są zapisywane w pamięci urządzenia na etapie produkcji/programowania
- urządzenie jest przypisane do jednej sieci i gotowe do komunikacji

## OTAA (*Over-the-Air Activation*)

- adres urządzenia i klucze sesji są generowane podczas procedury JOIN
- oparta na globalnie unikalnych 64-bitowych identyfikatorach urządzenia (*DevEUI*) i aplikacji (*AppEUI*) oraz klucza uwierzytelniającego (*AppKey*)
- komunikacja nie jest możliwa bez aktywacji
- umożliwia zmianę sieci (*roaming*)

### Sieć prywatna

- indywidualnie zarządzana
- dedykowane wdrożenia

### Scentralizowana sieć publiczna

- zwykle zarządzana przez operatora telekomunikacyjnego
- duży zasięg geograficzny

### Rozproszona/wspólna sieć publiczna

- brak jednego właściciela i punktu kontrolnego
- model internetowy
- najlepszy przykład - TTN (*The Things Network*)

## Czym jest TTN?

- Projekt zainicjowany w 2015 w Amsterdamie
- Inicjatywa, której celem jest budowa globalnej sieci dla **Internetu Rzeczy** w oparciu o technologię **LoRaWAN**
- Otwarta, bezpłatna, zdecentralizowana sieć tworzona i utrzymywana przez użytkowników



**THE THINGS  
N E T W O R K**

**90944**

**MEMBERS**

**9558**

**GATEWAYS**

**147**

**COUNTRIES**

## Technologia

- budowa i utrzymanie sieci LoRaWAN w mieście i okolicach
- generowanie propozycji rozwiązań wykorzystujących sieć
- tworzenie i aktualizacja mapy zasięgu sieci

## Organizacja i edukacja

- pozyskiwanie partnerów i współpraca z nimi
- organizacja spotkań i warsztatów związanych z technologią

## Reklama

- prowadzenie blogów i kanałów informacyjnych

## Poland

Bydgoszcz ✓

Gdansk

Gdynia

Gliwice

Lodz

Ostrów Wielkopolski

Poznan

Siechnice

Torun ✓

Warsaw

Wrocław ✓

# Ograniczenia sieci LoRaWAN

## 1 - zalecenia ETSI (EN300.220)

- zajętość pasma 1%, moc nadajnika 25 mW (14 dBm)

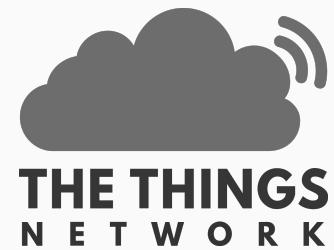


## 2 - liczba kanałów obsługiwanych przez bramki

- zwykle 8 kanałów (częstotliwości)

## 3 - zasady korzystania z otwartej sieci TTN (*Fair Access Policy*)

- czas nadawania przez urządzenie: do 30 sekund na dobę
- przy długości danych 10 bajtów dla urządzenia mamy
  - 20 wiadomości na dobę przy SF12
  - 500 wiadomości na dobę przy SF7
  - maksymalnie 10 wiadomości zwrotnych (*downlink*) na urządzenie na dobę
  - nie można wysyłać wszystkich wiadomości z potwierdzeniem (*confirmed uplink*)



# Realizacja komunikacji w sieci TTN



**THE THINGS  
N E T W O R K**

# Konto użytkownika TTN

## Konsola The Things Network

- dodawanie aplikacji i urządzeń
- definiowanie sposobu dekodowania danych
- podgląd danych

<https://account.thethingsnetwork.org/register>

Można później dołączyć do wybranej lokalnej społeczności TTN. Zachęcamy!



### CREATE AN ACCOUNT

Create an account for The Things Network and start exploring the world of Internet of Things with us.

**USERNAME**  
This will be your username — pick a good one because you will not be able to change it.

**EMAIL ADDRESS**  
You will receive a confirmation email, as well as occasional account related emails. If this email address is managed by a third party (such as for corporate email addresses), this third party might block emails coming from The Things Network. This email address is not public.

**PASSWORD**  
Use at least 6 characters.

[Create account](#)

By registering an account you agree to our [Terms and Conditions](#) and [Privacy Policy](#).

# Tworzenie aplikacji TTN

**ADD APPLICATION**

**Application ID**  
The unique identifier of your application on the network  
**unikalna\_nazwa\_aplikacji**

**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

**1**

Cancel **Add application**

**Applications > unikalna\_nazwa\_aplikacji**

**2**

Overview Devices **Payload Formats** Integrations Data Settings

### APPLICATION OVERVIEW

Application ID **unikalna\_nazwa\_aplikacji**  
Description  
Created 23 minutes ago  
Handler ttn-handler-eu (current handler)

**Applications > unikalna\_nazwa\_aplikacji > Payload Formats**

**3**

Overview Devices **Payload Formats** Integrations Data Settings

### PAYOUT FORMATS

**Payout Format**  
The payload format sent by your devices  
**Cayenne LPP**

Cancel **save**

# Rejestrowanie urządzenia pod aplikacją TTN

1

DEVICES

register device

manage devices

0 registered devices

2

REGISTER DEVICE

Device ID  
This is the unique identifier for the device in this app. The device ID will be immutable.  
unikalna\_nazwa\_urzadzenia

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

App Key  
The App Key will be used to secure the communication between your device and the network.  
this field will be generated

App EUI  
70 B3 D5 7E D0 01 E6 D4

Cancel Register

3

Applications > unikalna\_nazwa\_aplikacji > Devices > unikalna\_nazwa\_urzadzenia

Overview Data Settings

## DEVICE OVERVIEW

Application ID  
unikalna\_nazwa\_aplikacji

Device ID  
unikalna\_nazwa\_urzadzenia

Activation Method  
OTAA

Device EUI  
00 6A A9 19 88 41 77 AA

Application EUI  
70 B3 D5 7E D0 01 E6 D4

App Key  
... (redacted)

Status  
never seen

Frames up  
0 reset frame counters

Frames down  
0

# Rejestrowanie urządzenia pod aplikacją TTN c.d.

4

Applications > unikalna\_nazwa\_aplikacji > Devices > unikalna\_nazwa\_urzadzenia > Settings

DEVICE SETTINGS	SETTINGS
General	Description A human-readable description of the device
Location	
	<b>Device EUI</b> The serial number of your radio module, similar to a MAC address 00 6A A9 19 88 41 77 AA
	<b>Application EUI</b> 70 B3 D5 7E D0 01 E6 D4
	<b>Activation Method</b> OTAABP ABP
	<b>Device Address</b> The device address will be assigned by the network server
	<b>Network Session Key</b> Network Session Key will be generated
	<b>App Session Key</b> App Session Key will be generated
	<b>Frame Counter Width</b> 16 bit 32 bit
	<input type="checkbox"/> <b>Frame Counter Checks</b> Disabling frame counter checks drastically reduces security and should only be used for development purposes

Delete Device Cancel Save

5

DEVICE OVERVIEW

**Application ID** unikalna\_nazwa\_aplikacji  
**Device ID** unikalna\_nazwa\_urzadzenia

**Activation Method** ABP

**Device EUI** 00 6A A9 19 88 41 77 AA

**Application EUI** 70 B3 D5 7E D0 01 E6 D4

**Device Address** 26 01 13 9C

**Network Session Key** ...

**App Session Key** ...

**Status** never seen

**Frames up** 0 [reset frame counters](#)

**Frames down** 0

# Platforma do budowy lokalizatora

## Sprzęt - moduł LoRa32u4 II

- Atmega32U4 @ 8MHz, 3.3V, 32K FLASH, 2K RAM
- natywna obsługa USB
- zasilanie 3.3..5V, układ ładowania akumulatora Li-Po
- układ Semtech SX1276
- antena 868 MHz ze złączem U.FL

## Oprogramowanie:

- Arduino IDE
- plik z definicjami płytki → Arduino/hardware
  - LoRa32u4II z BSFrance  
[http://bsfrance.fr/documentation/11355\\_LORA32U4II/BSFrance.zip](http://bsfrance.fr/documentation/11355_LORA32U4II/BSFrance.zip)
- biblioteka zewnętrzna → Arduino/libraries
  - Arduino-LMIC Library  
<https://github.com/matthijskooijman/arduino-lmic>

# Programowanie urządzenia - komunikacja z SX1276

Device Address: 26 01 13 9C

Network Session Key: msb { 0xD8, 0xF4, 0x1F, 0xF5, 0x06, 0x5E, 0x3A, 0x3A, 0x01, 0x78, 0x65, 0x23, 0x9C, { } }  
1  
2  
3

App Session Key: msb { 0x6F, 0x2A, 0xCA, 0xCA, 0x9D, 0xDB, 0xFE, 0x3D, 0xDD, 0x89, 0x0B, 0xF1, 0xF6, { } }

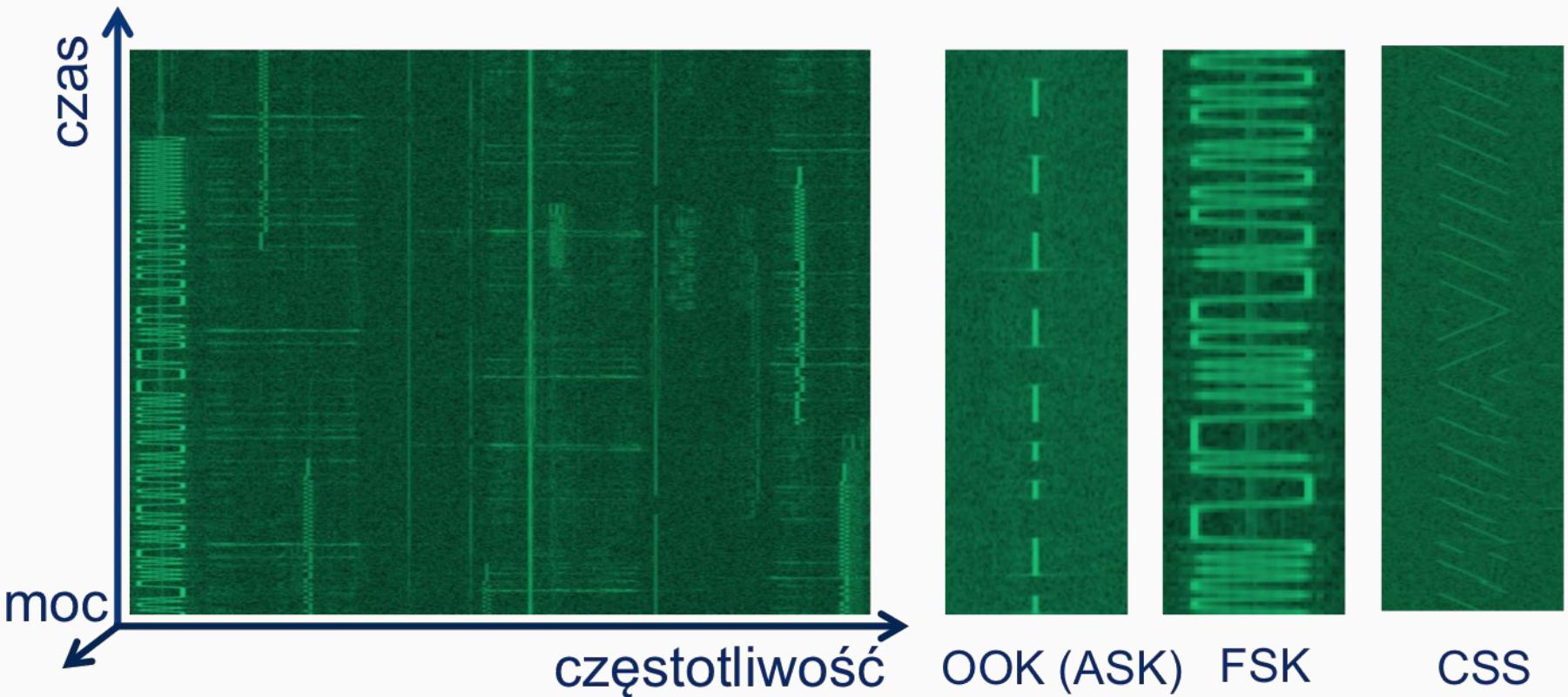
```
static const u1_t PROGMEM NWKSKEY[16] = { 0xD8, 0xF4, 0x1F, 0xF5, 0x06, ... };  
static const u1_t PROGMEM APPSKEY[16] = { 0x6F, 0x2A, 0xCA, 0xCA, 0x9D, ... };  
static const u4_t DEVADDR = 0x2601139C;
```

```
const lmic_pinmap lmic_pins = {  
    .nss = 8,  
    .rxtx = LMIC_UNUSED_PIN,  
    .rst = 4,  
    .dio = { 7, 6, LMIC_UNUSED_PIN },  
};
```

# Obserwacja pasma



# Spektrogram

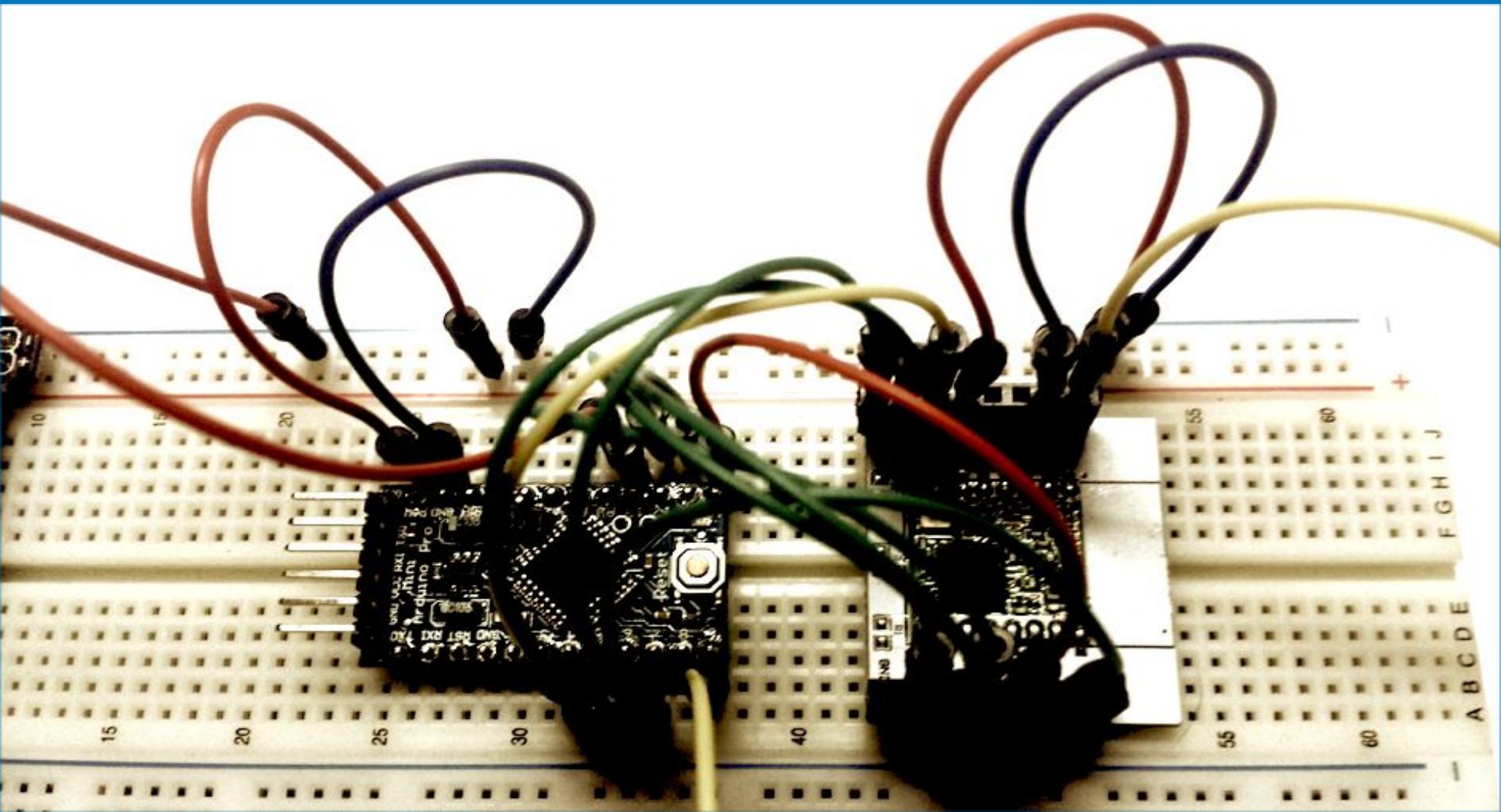


## Przygotowanie urządzenia i obserwacja transmisji

- wykorzystanie kanałów
- procedura JOIN
- zmiana współczynnika rozproszenia
- zmiana mocy nadawanego sygnału

```
...
const unsigned TX_INTERVAL = 60;
...
LMIC_setDrTxpow(DR_SF7, 14);
...
```

# Budujemy lokalizator



## Budowa lokalizatora - założenia

- Wysyłanie bieżącej lokalizacji (współrzędne geograficzne i wysokość nad poziomem morza) ze stałym interwałem
- Komunikacja LoRaWAN
- Zasilanie z USB i/lub baterijne



Dodatkowo wysyłane będą informacje określające jakość pomiaru pozycji:

- HDOP (*Horizontal Dilution Of Precision*)
- liczba śledzonych satelitów

# Sposoby komunikacji z modułami GNSS (*Global Navigation Satellite System*)

Stosowane interfejsy:

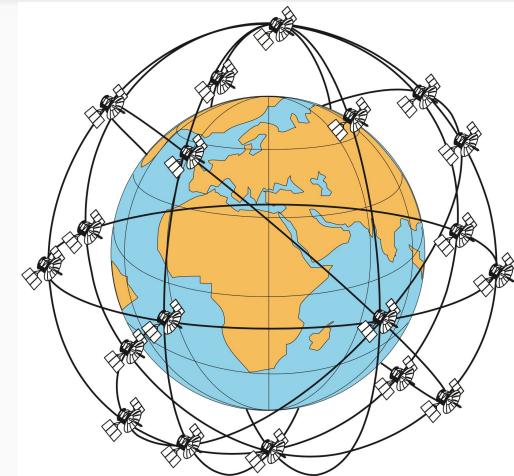
- **port szeregowy (UART)**
- I2C

Protokoły

- **tekstowy - protokół NMEA-0183**
  - opracowany przez *National Marine Electronics Association*
  - zaimplementowany praktycznie we wszystkich odbiornikach GNSS
  - przykład - ramka GGA

**§GPGGA,181129.00,5307.18494,N,01759.44889,E,1,08,1.09,98.7,M,35.9,M,,\*65**

- binarny
  - zależny od producenta układu, na którym bazuje odbiornik GNSS
- możliwość parametryzacji sposobu komunikacji



# Budowa lokalizatora - komponenty

## Sprzęt:

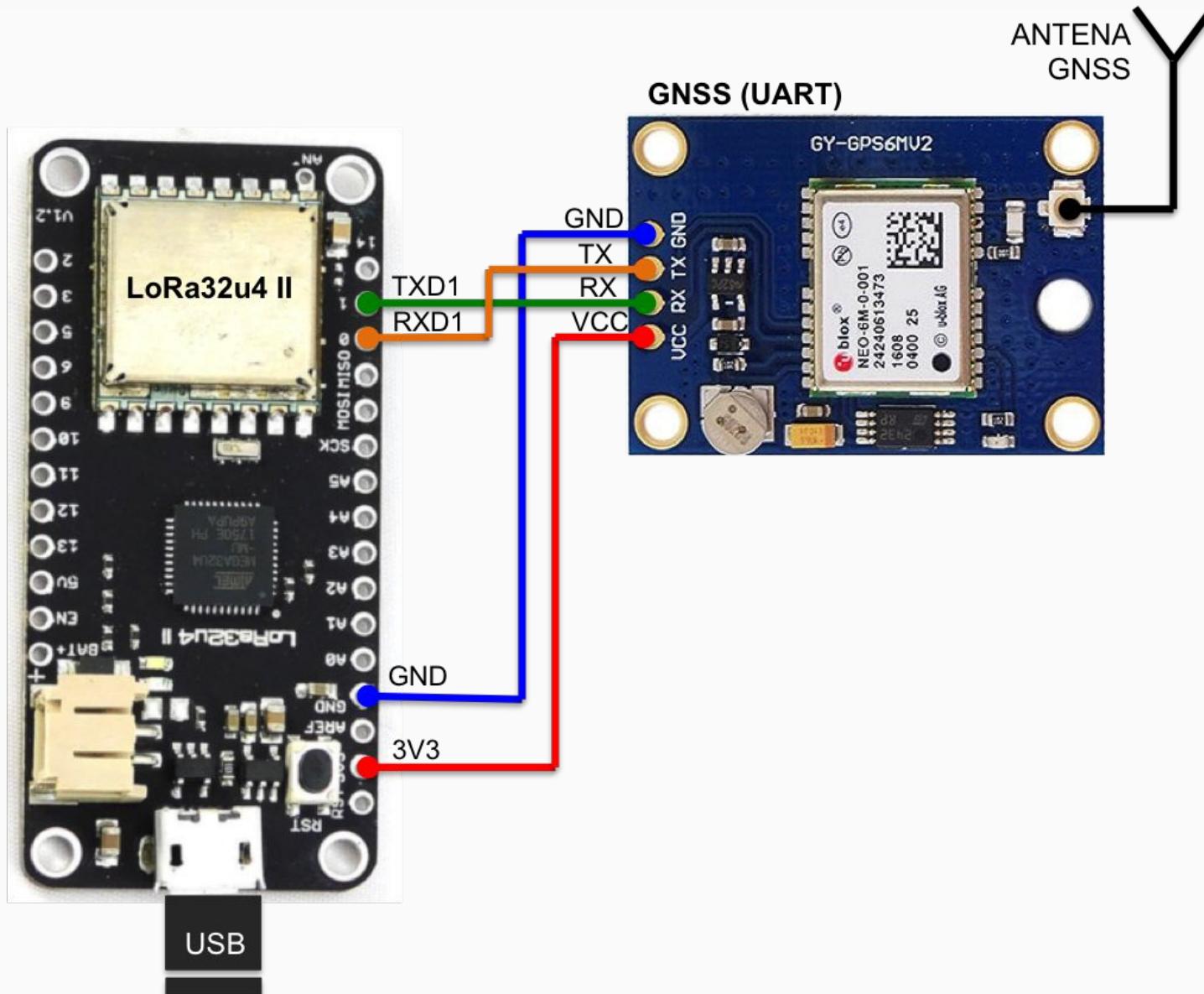
- moduł LoRa32u4 II
- **odbiornik GNSS (GPS) z interfejsem UART i anteną**

## Dodatkowe oprogramowanie:

- biblioteka zewnętrzna → Arduino/libraries
  - TinyGPSPlus - NMEA Parsing Library

<https://github.com/mikalhart/TinyGPSPlus>

# Sposób połączenia komponentów



## Sprzęt:

- moduł LoRa32u4 II
- odbiornik GNSS (GPS) z interfejsem UART i anteną
- **akumulator Li-Po 3.7V**

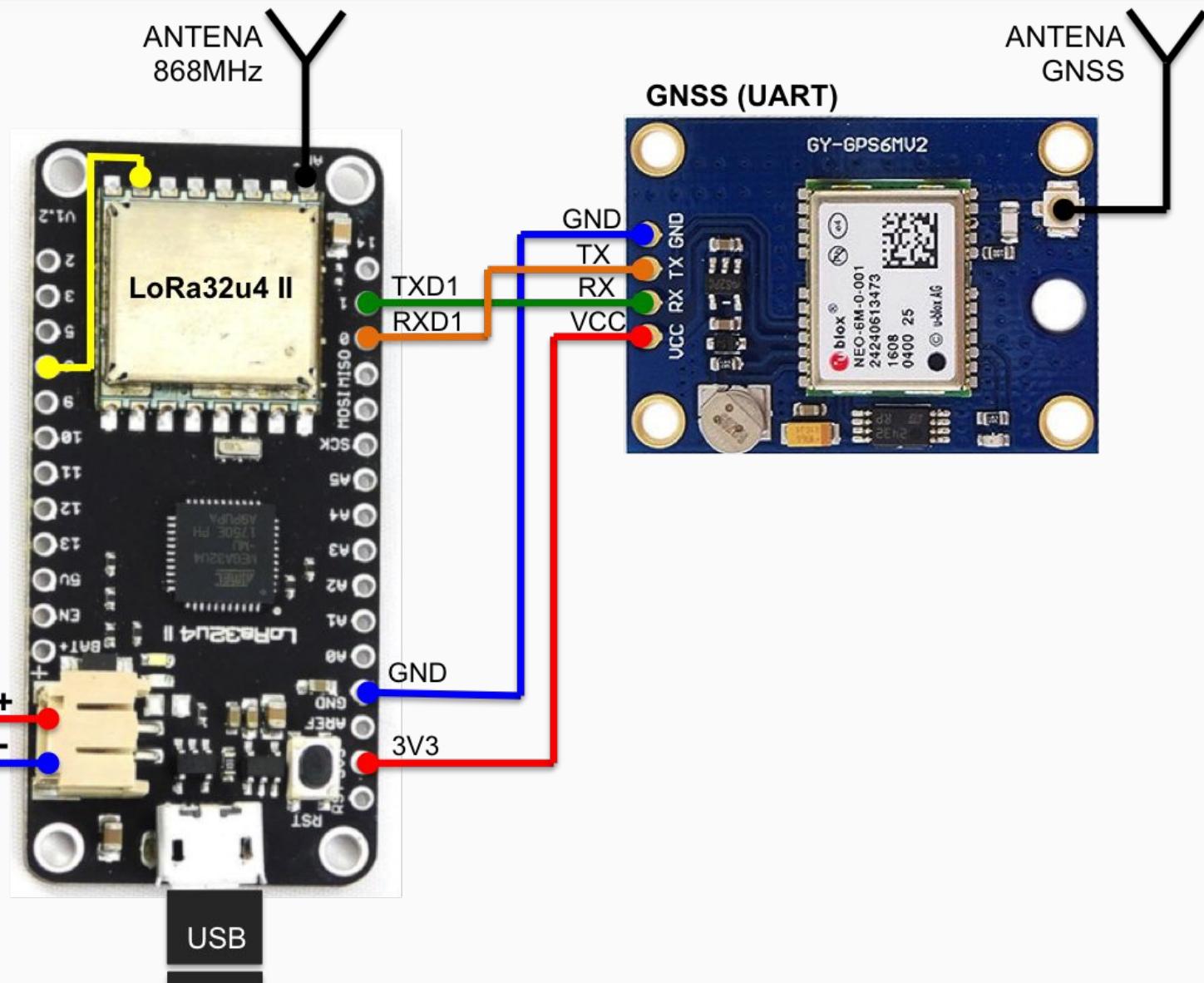
## Dodatkowe oprogramowanie:

- biblioteka zewnętrzna → Arduino/libraries
  - The Things Network Arduino Library (tylko Cayenne LPP)  
<https://github.com/TheThingsNetwork/arduino-device-lib>

# Sposób połączenia komponentów



AKUMULATOR  
Li-Pol  
3.7V



# Programowanie urządzenia - wykorzystanie Cayenne LPP

```
#include <CayenneLPP.h>
CayenneLPP lpp(51);

...
lpp.reset();
lpp.addGPS(1, gps.location.lat(), gps.location.lng(), gps.altitude.meters());
lpp.addAnalogInput(2, (float)gps.satellites.value());
lpp.addAnalogInput(3, gps.hdop.hdop());
...
LMIC_setTxData2(2, lpp.getBuffer(), lpp.getSize(), 0);
```

## PAYOUT FORMATS

### Payload Format

The payload format sent by your devices

Cayenne LPP

Custom

Cayenne LPP

- Bardzo łatwy sposób przygotowania i przesyłania danych
- Odbierane dane są dekodowane automatycznie po stronie aplikacji

```
{
  "analog_in_2": 11,
  "analog_in_3": 0.88,
  "gps_1": {
    "altitude": 76.19,
    "latitude": 53.1581,
    "longitude": 18.2246
  }
}
```

# Programowanie urządzenia - zrzucanie zbędnych kilobajtów

Error compiling for board LoRa32u4II (868-915MHz).

[Copy error messages](#)

Sketch uses 32426 bytes (113%) of program storage space. Maximum is 28672 bytes.

Global variables use 1489 bytes of dynamic memory.

Sketch too big; see <http://www.arduino.cc/en/Guide/Troubleshooting#size> for tips on reducing it.

Error compiling for board LoRa32u4II (868-915MHz).

14

LoRa32u4II (868-915MHz) on /dev/cu.usbmodem1411

arduino-lmic/src/lmic/config.h

...

```
// Uncomment this to disable all code related to joining
#defineDISABLE_JOIN
// Uncomment this to disable all code related to ping
#defineDISABLE_PING
// Uncomment this to disable all code related to beacon tracking.
// Requires ping to be disabled too
#defineDISABLE_BEACONS
...
```

Done compiling.

Sketch uses 27458 bytes (95%) of program storage space. Maximum is 28672 bytes.  
Global variables use 1438 bytes of dynamic memory.

18

LoRa32u4II (868-915MHz) on /dev/cu.usbmodem1411

# Programowanie urządzenia - własny sposób kodowania

```
...
uint8_t data[10];

...
while (Serial1.available() > 0)
    gps.encode(Serial1.read());

...
uint32_t latitude = ((gps.location.lat() + 90) / 180.0) * 16777215;
uint32_t longitude = ((gps.location.lng() + 180) / 360.0) * 16777215;
uint16_t altitude = gps.altitude.meters();
uint8_t hdev = gps.hdop.hdop() * 10;
data[0] = latitude >> 16;
data[1] = latitude >> 8;
data[2] = latitude;
data[3] = longitude >> 16;
data[4] = longitude >> 8;
data[5] = longitude;
data[6] = altitude >> 8;
data[7] = altitude;
data[8] = hdev;
data[9] = gps.satellites.value();

...
```

# Programowanie urządzenia - własny dekoder

## PAYLOAD FORMATS

### Payload Format

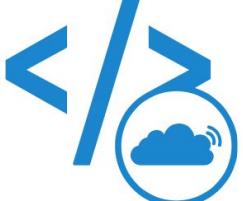
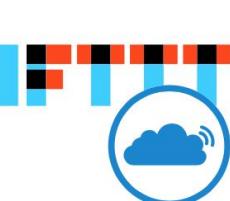
The payload format sent by your devices

Custom

```
function Decoder(bytes, port) {
    var decoded = {};
    decoded.lat = ((bytes[0]<<16)>>>0) + ((bytes[1]<<8)>>>0) + bytes[2];
    decoded.lat = (decoded.lat / 16777215.0 * 180) - 90;
    decoded.lon = ((bytes[3]<<16)>>>0) + ((bytes[4]<<8)>>>0) + bytes[5];
    decoded.lon = (decoded.lon / 16777215.0 * 360) - 180;
    var alt = ((bytes[6]<<8)>>>0) + bytes[7];
    var sign = bytes[6] & (1 << 7);
    if(sign) {
        decoded.altitude = 0xFFFF0000 | alt;
    } else {
        decoded.altitude = alt;
    }
    decoded.hdop = bytes[8] / 10.0;
    decoded.sats = bytes[9];
    return decoded;
}
```

```
{
    "alt": 73,
    "hdop": 0.8,
    "lat": 53.15814037073494,
    "lon": 18.224677933733346,
    "sats": 11
}
```

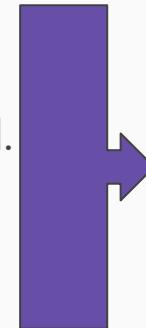
# Przegląd sposobów integracji z TTN

					
AllThingsTalk Maker v2.6.0 AllThingsTalk	Collos v2.7.10 Semtech Corporation	Data Storage v2.0.1 The Things Industries B.V.	MyDevices v2.6.0 myDevices	OpenSensors v2.6.0 The Things Industries B.V.	TTN Mapper v2.7.1 JP Meijers
					
EVRYTHNG v2.6.0 EVRYTHNG	HTTP Integration v2.6.0 The Things Industries B.V.	IFTTT Maker v2.6.0 The Things Industries B.V.	TagoIO v2.7.5 TagoIO	ThingSpeak v2.7.14 MathWorks®	Ubidots v2.7.10 Ubudots

# Interfejsy wymiany danych z TTN

## Interfejsy programistyczne:

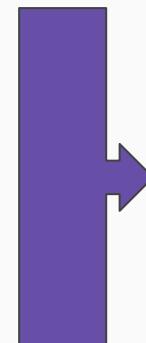
- ApplicationManager API, Data API
- MQTT
  - MQTT client library for Java, JavaScript, Python, itd.
  - MQTT Dash for Android, MQTTTool for iOS
- DataStorage (REST API danych i urządzeń)
- HTTP Integration (webhooks)
- TTN SDK for: GO, Java, Node-RED, Node.js, Python



Jeżeli chcesz zbudować własne rozwiązanie w oparciu o dane

## Integracje z serwisami usługowymi:

- AWS IoT - infrastruktura oraz serwisy do budowy własnych rozwiązań
- AllThingsTalk - platforma developerska (od SDK dla urządzeń do API usług platformy)
- Collos - servisy lokalizacyjne
- OpenSensors - 'Twitter for Sensors'
- IFTTT Maker

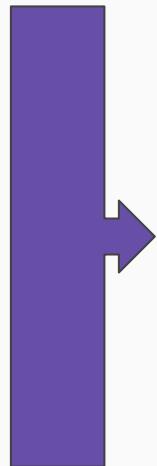


Potrzebujesz wybranych serwisów bezpośrednio lub do wykorzystania we własnym rozwiązaniu

# Przykłady platform zintegrowanych z TTN

## Integracje z platformami IoT:

- myDevices Cayenne
- Tago
- ThingSpeak (anityka, integracja z MATLAB)
- EVRYTHNG (ukierunkowane na e-Commerce)
- **TTN Mapper**
- Platformy korzystające z HTTP Integration
  - Ubidots
  - thethings.io
  - **Signomix**



Otrzymujesz funkcjonalność udostępnianą przez daną platformę

# TTN Mapper

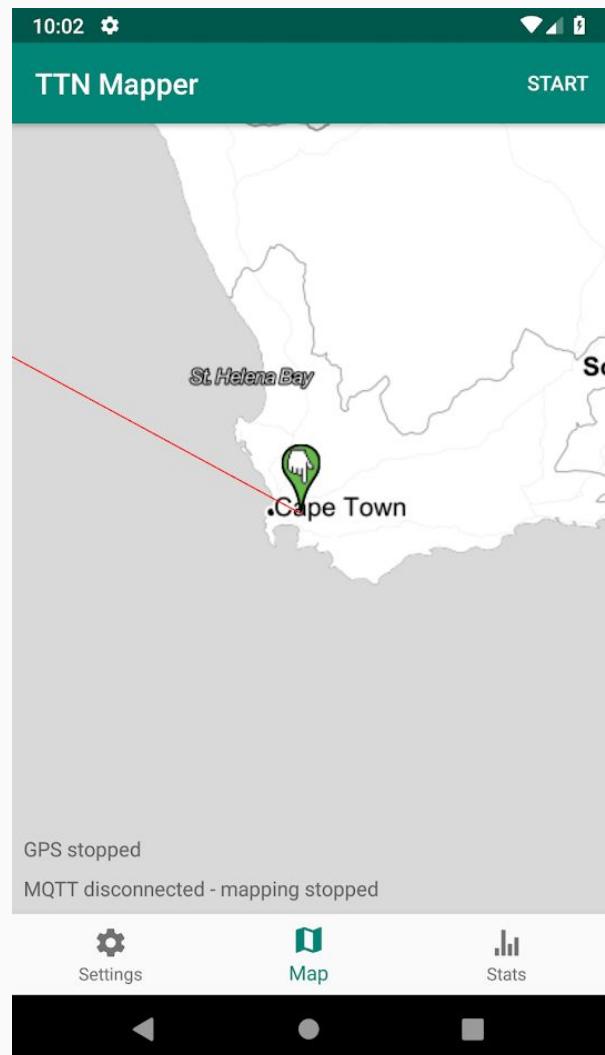
**TTN Mapper** jest systemem zbudowanym w celu tworzenia i udostępniania aktualnych map zasięgu bramek TTN.

Oprogramowanie jest dostępne jako Open Source, a dzięki zaangażowaniu społeczności TTN mamy dostęp do map zasięgu na całym świecie.

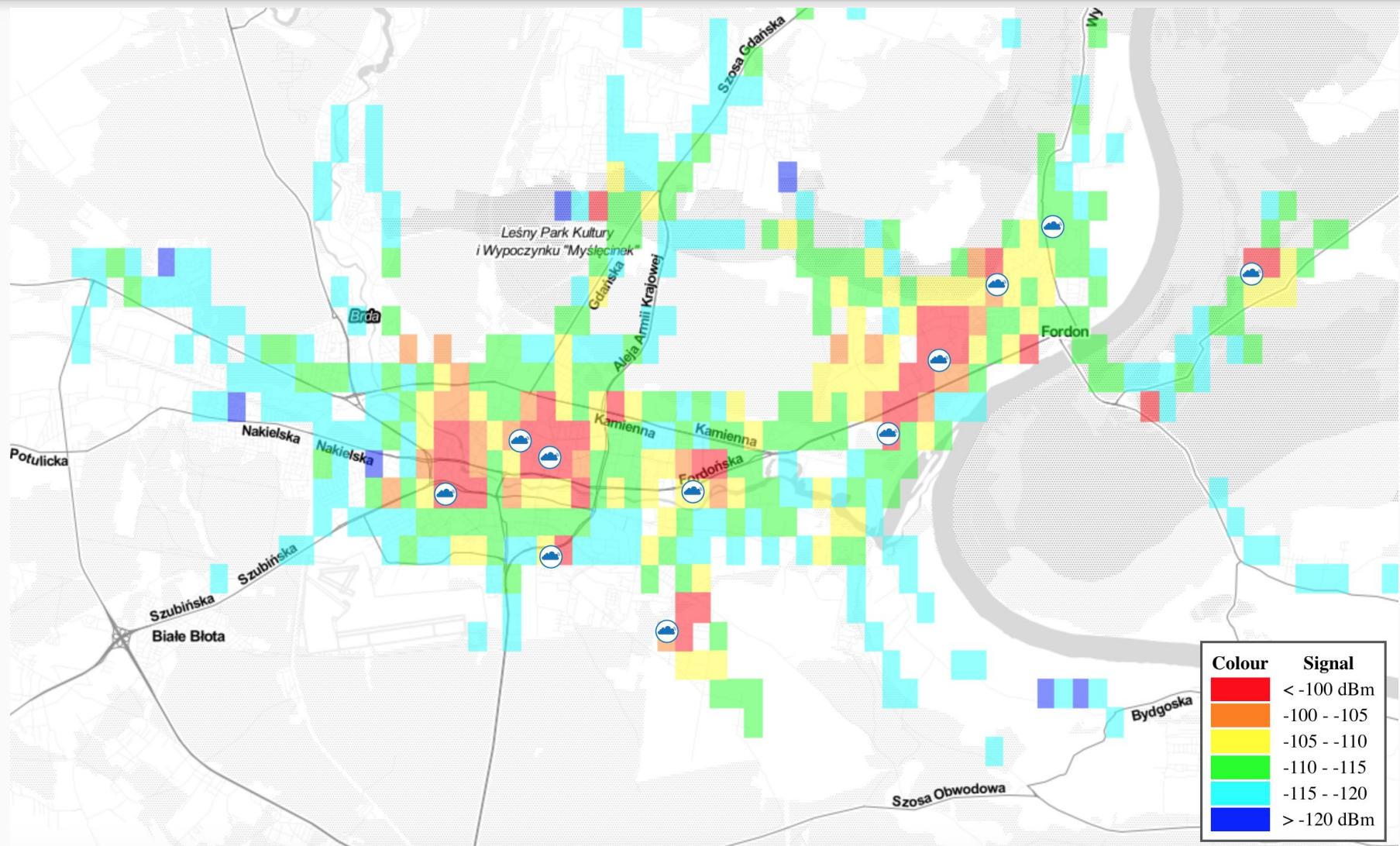
<https://ttnmapper.org/>

Dwa sposoby dostarczania danych do TTN Mapper:

- poprzez użycie urządzenia z modułem GPS, wysyłającego dane bezpośrednio do TTN Mapper (dedykowany interfejs integracyjny)
- poprzez użycie dowolnego urządzenia LoRaWAN transmitującego dane do sieci TTN oraz smartfona z zainstalowaną aplikacją TTN Mapper (Android lub iOS).



# TTN Mapper - tworzenie mapy zasięgu sieci



# Wizualizacja danych na platformie



# Konto użytkownika platformy Signomix

Otwarta platforma do rejestracji i wizualizacji danych

Proces zakładania konta:



- wypełnienie formularza rejestracyjnego
- odebranie wiadomości z potwierdzeniem i kliknięcie na podany odnośnik
- podane kanały notyfikacji służą do przesyłania informacji o zdarzeniach

<https://signomix.com/app/#!register>

The image contains two side-by-side screenshots of the Signomix web interface. The left screenshot shows the 'Tworzenie konta użytkownika' (Create User Account) form. It has fields for 'Login (nazwa użytkownika)', 'E-mail', 'Hasło', and 'Powtóż hasło'. Below these is a note about password complexity and a checkbox for accepting terms and conditions. At the bottom are 'Zarejestruj się' and 'Porzuć' buttons. The right screenshot shows the 'moje konto' (My Account) page. It lists account details: 'Login' (tester1), 'Adres e-mail' (tester1@myserver.com), and a table of notification channels: 'Kanal Notyfikacji GENERAL' (SIGNOMIX), 'Kanal Notyfikacji INFO' (SMTP:tester1@myserver.com), 'Kanal Notyfikacji WARNING' (SMTP:tester1@myserver.com), and 'Kanal Notyfikacji ALERT' (SMTP:tester1@myserver.com). It also shows a 'Kod Potwierdzenia' (Verification Code) field with '6022140857'. At the bottom right is a red button labeled 'Chcę usunąć moje konto' (I want to delete my account).

# Signomix - rejestracja urządzenia

1

SIGNOMIX

moje urządzenia

EUI	NAZWA	TYP	STATUS	DZIAŁANIE
-----	-------	-----	--------	-----------

2

SIGNOMIX

## Nowe urządzenie

Typ urządzenia

\* Nazwa

\* EUI

Urządzenie IoT zarejestrowane w infrastrukturze The Things Network

Urządzenie IoT zarejestrowane w infrastrukturze The Things Network

DEVICE OVERVIEW

Application ID

Device ID

Activation Method

Device EUI

Application EUI

Nowe urządzenie

Pomiary

EUI aplikacji

ID aplikacji

Zapisz

# Signomix - integracja z aplikacją użytkownika

- Signomix korzysta z *HTTP Integration*
- Dane odebrane z urządzenia przez serwer aplikacyjny są po zdekodowaniu wysyłane w formacie JSON na wskazany adres URL

**SIGNOMIX** Komunikaty Pulpity Urządzenia Konto Dokumentacja Wyloguj się

## Modyfikacja urządzenia

Typ urządzenia  
TTN

Urządzenie IoT zarejestrowane w infrastrukturze  
The Things Network

\* Nazwa  
test  
Użyj wygodnej dla Ciebie nazwy

\* EUI  
006AA919884177AA  
Unikalny identyfikator tego urządzenia podany przez producenta. Jeśli pozostawisz to pole puste, zostanie wygenerowany automatycznie.

Klucz  
NDEwNjI0NTc5MDA2NDIxNzKxOA

Poufny klucz wykorzystywany do autoryzowania przesyłanych danych. Jeśli pozostawisz to pole puste, zostanie wygenerowany automatycznie.

### ADD INTEGRATION



#### 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

signomix

##### Access Key

The access key used for downlink

default key  devices  messages

##### URL

The URL of the endpoint

<http://signomix.com/api/ttn>

##### Method

The HTTP method to use

POST

##### Authorization

The value of the Authorization header

NDEwNjI0NTc5MDA2NDIxNzKxOA

##### Custom Header Name

An optional custom HTTP header that you would like to add to the request

##### Custom Header Value

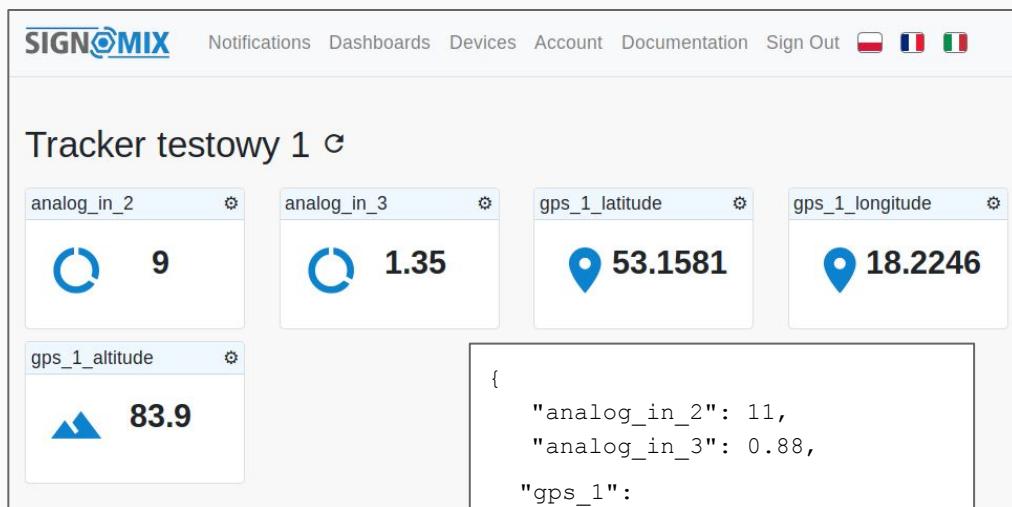
The value of the custom Header

Cancel

Add integration

# Signomix - domyślny pulpit urządzenia

- Po zarejestrowaniu urządzenia automatycznie tworzony jest dla niego domyślny pulpit (*dashboard*) o nazwie takiej jak EUI urządzenia
- Dane są natychmiast dostępne do przeglądania
- Jeśli dla urządzenia nie został podany skrypt procesora danych, to nazwy zmiennych zgodne z danymi przesyłanymi z TTN w formacie JSON (z wyjątkiem danych GPS)



## Zmiana definicji pulpitu

\* Nazwa

00D327B4EDD4D6A4

Wybierz unikalną nazwę pulpitu (alfanumeryczną, bez spacji)

Pulpit może być współdzielony

\* Tytuł

Tracker testowy 1

Wybierz tytuł dla pulpitu

Zespół

,

Loginy użytkowników oddzielone przecinkami

## Kontrolki +

NAZWA	TYP	OPERACJA
analog_in_2	Etykieta	↓ ↑ 🖊️ 🗑️
analog_in_3	Etykieta	↓ ↑ 🖊️ 🗑️
gps_1_latitude	Etykieta	↓ ↑ 🖊️ 🗑️
gps_1_longitude	Etykieta	↓ ↑ 🖊️ 🗑️
gps_1_altitude	Etykieta	↓ ↑ 🖊️ 🗑️

Zapisz

Porzuć

# Signomix - modyfikacje pulpitu

Modyfikacja domyślnego pulpitu lub utworzenie nowego pozwala na:

- zmianę sposobu prezentacji danych
- wizualizację przekroczeń progów
- dodanie elementów tekstowych
- prezentację danych z różnych urządzeń
- publikowanie pulpitów
- osadzenie raportów grupowych



SIGNOMIX Komunikaty Pulpity Urządzenia Konto Dokumentacja Wyloguj się

## Smogometr 2 C ↗

Test czujnika smogu przygotowanego przez społeczność [The Things Network](#) oraz stowarzyszenie [Otwarta Sieć Rzeczy](#).

Średnia dobowa PM 2.5

Średnia dobowa PM 10

Położenie czujnika

PM 2.5 37.1  $\mu\text{g}/\text{m}^3$

PM 10 41.5  $\mu\text{g}/\text{m}^3$

Temperatura 26.1 °C

Wilgotność 66.6 %

Zmiany PM 2.5

Zmiany PM 10

Zmiany temperatury

Zmiany wilgotności

Mapa

Leaflet | © OpenStreetMap contributors

The dashboard displays real-time and historical data for air quality, temperature, and humidity. It includes a map showing the location of the sensor in Łódź, Poland. The charts show fluctuations over a 24-hour period from June 25 to June 27, 2019.

# Signomix - rejestracja urządzenia c.d.

SIGNOMIX

Komunikaty Pulpity Urządzenia Konto Dokumentacja Wyloguj się

## Modyfikacja urządzenia

Typ urządzenia  
TTN

Urządzenie IoT zarejestrowane w infrastrukturze  
The Things Network

\* Nazwa  
test

Użyj wygodnej dla Ciebie nazwy

\* EUI  
006AA919884177AA

Unikalny identyfikator tego urządzenia podany przez producenta. Jeśli pozostawisz to pole puste, zostanie wygenerowany automatycznie.

Klucz  
NDEwNjI0NTc5MDA2NDIxNzKxOA

Poufny klucz wykorzystywany do autoryzowania przesyłanych danych. Jeśli pozostawisz to pole puste, zostanie wygenerowany automatycznie.

Pomiary  
 Nazwy pomiarów oddzielone przecinkami

Zespół

Loginy innych użytkowników mających dostęp do danych z tego urządzenia. Oddzielone przecinkami.

Opis

Skrypt dekodera danych

Pozostaw puste jeśli nie wiesz o co chodzi.

Skrypt procesora danych.

```
// zmieniamy nazwy danych na bardziej przyjazne
putData('temperature',getValueOf('temperature_1'))
putData('humidity',getValuerInRelative_humidity_2')
```

- Zmiana nazewnictwa danych
- Przetwarzanie danych przy użyciu skryptu
  - zmiany nazw odbieranych danych
  - korekcja wartości
  - tworzenie danych wyliczeniowych
  - powiadomienia
  - polecenia dla urządzeń
  - zapis danych do urządzeń wirtualnych

```
// skryptu możemy użyć skorygowania odebranych wartości danych
// lub dostarczenia danych wyliczanych
var time=sgx.getTimestamp('analog_in_2')
var sat=sgx.getValue('analog_in_2')
var hdop=sgx.getValue('analog_in_3')
sgx.put('satelites',sat,time)
sgx.put('hdop',hdop,time)
// jeśli zastosujemy skrypt do utworzenia lub zmodyfikowania danych,
// to musimy pozostałe dane zaakceptować
sgx.accept('gps_1_latitude')
sgx.accept('gps_1_longitude')
sgx.accept('gps_1_altitude')
```

# Signomix - powiadomienia

- Wizualizacja ostrzeżeń na dedykowanym pulpicie
- Automatyczne powiadomienia o niedostępności urządzeń
  - Signomix monitoruje aktualność danych dla każdego urządzenia co 15 minut
  - jeśli dane są starsze niż zadeklarowana długość cyklu ( $>0$ ), to informacja jest przesyłana kanałem GENERIC do właściciela urządzenia oraz wszystkich kont w przypisanym zespole
- Generowanie powiadomień w oparciu o analizę danych (definicja urządzenia: skrypt procesora danych), np:
  - "elektroniczne ogrodzenie"
  - monitorowanie stanu baterii

```
// położenie centrum obszaru monitorowanego
var homeLon = 51.104878
var homeLat = 17.012639
// aktualne położenie czujnika
var lon = sgx.getValue('gps_1_longitude')
var lat = sgx.getValue('gps_1_latitude')
// jeśli czujnik oddali się ponad 100 metrów
if(sgx.getDistance(homeLat,homeLon,lat,lon) > 100){
    sgx.addNotification("alert","obiekt poza ogrodzeniem")
}
```



## Definicja kontrolki

### Typ

Etykieta

### EUI urządzenia

AA6637AC0B234D38

EUI urządzenia, do którego masz uprawnienia

### Nazwa danej

pm100

Nazwa danej rejestrowanej przez urządzenie (np. temperatura)

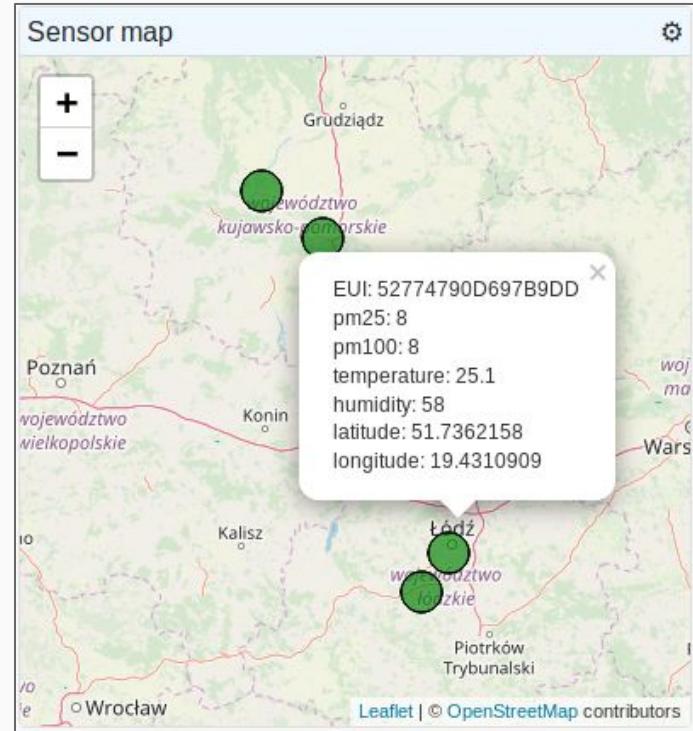
### Zakresy ostrzeżeń

>100:>50

Zakresy ostrzeżeń w formacie {alertCondition}[:{warningCondition}] (np.  
<-10>40:<0>30 )

# Signomix - mapy dla grup urządzeń

- Przypisanie urządzenia do grupy umożliwia przedstawienie wysyłanych przez nie danych na raporcie zbiorczym lub mapie. Raport zawiera odnośniki do domyślnych pulpitów wymienionych urządzeń (EUI pulpitu jest takie samo jak EUI urządzenia).
- Warunki
  - grupa urządzeń jest zdefiniowana przez administratora platformy Signomix
  - parametr *Zespół (Team)* każdego urządzenia musi zawierać ID autora raportu grupowego
  - parametr *Pomiary (Measurements)* urządzenia musi zawierać nazwy pomiarów podane w definicji grupy
  - jeśli raport grupowy jest udostępniany publicznie, to podlinkowane raporty szczegółowe będą dostępne jeśli również mają ustawiony parametr "współdzielony" (*shared*)



**SIGNOMIX** Alerts Dashboards Devices Account Documentation Sign Out

## Raport smogowy c

Raport						
DEVICE EUI	PM25	PM100	TEMPERATURE	HUMIDITY	DATE	DETAILS
009FB2FF8EBB8E00	4.19	6.9	29.4	39.5	2019-06-26 01:47:01	<a href="#">show ...</a>
AA6637AC0B234D38	30.8	32.3	28.7	59.8	2019-06-26 10:01:49	<a href="#">show ...</a>
52774790D697B9DD	22	22	30.1	99.9	2019-06-26 10:04:21	<a href="#">show ...</a>

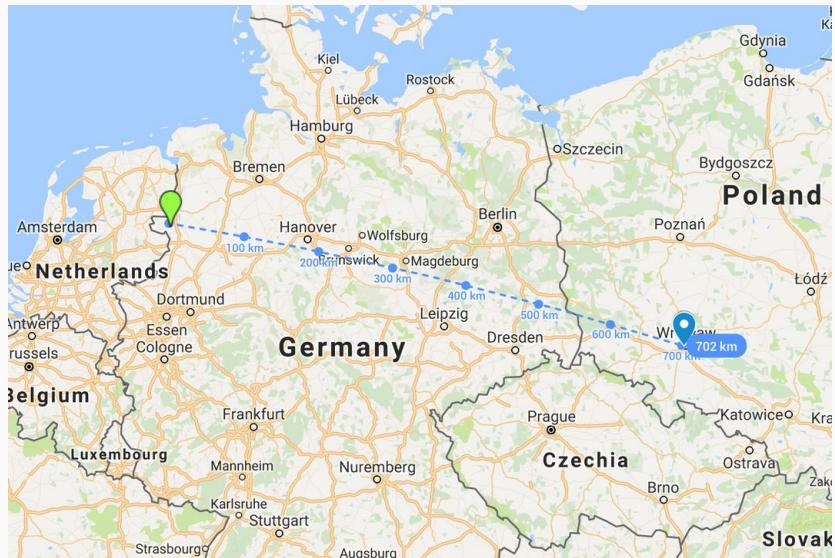
# Ciekawostki



# Zasięg transmisji LoRaWAN

## Rekordy odległości transmisji LoRaWAN

- teoretyczna maksymalna odległość - 800 km
- sierpień 2017 - 702 km
- lipiec 2019 - 741 i 766 km



<https://www.thethingsnetwork.org/article/lorawan-distance-world-record>

# Lokalizacja za pomocą technologii LoRaWAN

## Metadane

- czas odebrania danych przez bramkę
- poziom odbieranego sygnału (RSSI - *Received Signal Strength Indicator*)
- stosunek sygnału do szumu (SNR - *Signal to Noise Ratio*)
- współrzędne geograficzne bramki

## Wyznaczanie pozycji nadajnika za pomocą lateracji

- odległość na podstawie czasu propagacji sygnału
- odległość na podstawie tłumienia/zanikania sygnału

## Przykłady usług związanych z lokalizacją LoRaWAN

- Collos - Collaborative Location Service
- Kerlink - Wanesy Geolocation Solution

```
{
  "time": "2019-10-03T17:21:31.146577222Z",
  "frequency": 867.9,
  "modulation": "LORA",
  "data_rate": "SF7BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "nmg-gateway",
      "gtw_trusted": true,
      "timestamp": 3832111796,
      "time": "2019-10-03T17:34:14Z",
      "channel": 7,
      "rssi": -120,
      "snr": -4.75,
      "latitude": 53.130524,
      "longitude": 18.11909,
      "altitude": 63,
      "location_source": "registry"
    },
    {
      "gtw_id": "utp-gateway",
      "gtw_trusted": true,
      "timestamp": 3136174628,
      "time": "2019-10-03T17:21:30.113248226Z",
      "channel": 7,
      "rssi": -116,
      "snr": -2,
      "latitude": 53.143066,
      "longitude": 18.13399,
      "altitude": 80
    },
    {
      "gtw_id": "eui-b827ebffffd72eaf",
      "timestamp": 1053208004,
      "time": "2019-10-03T17:21:31.114451Z",
      "channel": 7,
      "rssi": -69,
      "snr": 11,
      "latitude": 53.15817,
      "longitude": 18.22477,
      "altitude": 86
    }
  ]
}
```

# Śledzenie balonów stratosferycznych - SR0FLY-TTN

Przykład lokalizatora LoRaWAN przygotowanego do misji stratosferycznych

- układ RN2483
- moduł GNSS u-BloxMAX-M8C
- czujnik temperatury i ciśnienia BME280
- czujnik promieniowania UV VEML6070
- analogowy czujnik temperatury TMP36
- bateria 3.6V typu AA



**COPERNICUS  
PROJECT**



## Nadajnik-odbiornik LoRaWAN w satelicie

- przelatujące okresowo na niskiej orbicie satelity odbierają wiadomości wysyłane przez urządzenia
- satelity przechowują tymczasowo odebrane wiadomości i retransmitują je do stacji naziemnej, gdy znajdzie się w zasięgu
- stacja naziemna przekazuje wiadomości do sieci



?

Dziękujemy za udział  
w warsztatach



ttn.bydgoszcz@gmail.com

<https://thethingsnetwork.org/community/bydgoszcz>

[https://twitter.com/ttn\\_bydgoszcz](https://twitter.com/ttn_bydgoszcz)