



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra telekomunikační techniky

## Low power wireless sensor network

**Tomáš Hyhlík**

Vedoucí: Ing. Bc. Marek Neruda, Ph.D

Školitel–specialista: Ing. Bc. Lukáš Vojtěch, Ph.D

Obor: Elektronika a komunikace

Studijní program: Elektronika

Říjen 2019

**Poděkování**

**Prohlášení**

## Abstrakt

**Klíčová slova:**

**Vedoucí:** Ing. Bc. Marek Neruda, Ph.D

## Abstract

**Keywords:**

**Title translation:** Low power wireless sensor network

## Obsah

0.1 Seznam zkratk	1	3.2 Výběr komponent	7
<b>1 Low power wireless network technologies</b>	<b>2</b>	3.2.1 Microcontroller	7
1.1 IQRf	2	3.2.2 LoRa transceiver	8
1.2 Wireless M-bus	2	3.2.3 RS485 transceiver	8
1.3 Zigbee	3	3.3 Implementace LoRaWAN sítě	9
1.4 Bluetooth	3	3.4 Implementace komunikačního protokolu v síti IMA_RS485	9
1.5 LoRa	3	3.4.1 Syntaxe příkazů	9
1.6 Sigfox	3	3.4.2 Statusy	10
1.7 Z-Wave	4	3.4.3 ACK neboli potvrzení	10
1.8 Thread	4	3.4.4 Přidávání LoRaWAN zařízení do systému	10
1.9 RPMA	4	3.4.5 Odesílání dat ze senzoru	11
<b>2 Stanovení požadavků systému</b>	<b>5</b>	3.4.6 Dotaz na příznaky	12
<b>3 Realizace gatewaye</b>	<b>6</b>	3.5 Paměť	12
3.1 Výběr přenosové technologie	6	3.6 Komunikace přes USB	12
3.1.1 Zabezpečení protokolu LoRaWAN	6	3.6.1 Log aplikace	12
		3.6.2 Konfigurace systému	13
		3.7 Koncová zařízení	17

3.7.1 Zpracování dat jednotlivých typů koncových zařízení . . . . .	17
3.7.2 Přidávání koncových zařízení ze serveru K4 . . . . .	18
3.8 Zapojení . . . . .	20
3.9 Instalace . . . . .	20
3.10 Zdrojové soubory projektu . . . .	21
<b>Literatura</b>	<b>22</b>

## Obrázky

1.1 LoRa spread factor options [?] ...	3
2.1 Blokový diagram navrženého systému .....	5
3.1 Vývojový kit NUCLEO-L073RZ [1]	7
3.2 LoRa transceiver RFM95w [3] ...	8
3.3 RS485 transceiver [7] .....	8

## Tabulky

3.1 Syntaxe příkazu pro komunikaci se serverem přes RS485 .....	9
3.2 Příklad sekvence příkazů pro přidávání LoRaWAN zařízení do systému .....	11
3.3 Nastavení USB terminálu .....	12
3.4 Defaultní konfigurace systému ..	16
3.5 Typy koncových zařízení .....	17
3.6 Pinout připojení externích periférií k procesoru .....	20

## ■ 0.1 Seznam zkratek

<b>AppSKey</b>	.....	Application Session Key
<b>CPU</b>	.....	Central Processing Unit
<b>LPWAN</b>	.....	Low Power Wide Area Network
<b>LAN</b>	.....	Local Area Network
<b>NwkSKey</b>	.....	Network Session Key
<b>RF</b>	.....	Radio Frequency

# Kapitola 1

## Low power wireless network technologies

### 1.1 IQRF

This technology aims to make it easy to implement wireless solutions. It enables peer-to-peer, star and mesh network communication modes. The IQRF alliance provide IQRF transceivers for \$15-20 with a few serial interfaces such as SPI, I2C, UART etc. and they also provide open source SDK which makes it very easy to use IQRF modules. The SDK is based on Java so it's compatible with various platforms such as Linux and Windows [?] [?] [?] [?].

### 1.2 Wireless M-bus

*"Wireless Meter Bus has its origins within the Meter-Bus standards. This is a field bus standard aimed at applications for collecting meter data for gas, electricity, water, etc."* [?] It supports a few application modes for differing applications.

- S1 Unidirectional, data are transmitted only a several times a day.
- S2 Bidirectional version of S1.
- T1 Unidirectional transmission of data with a period of a few seconds of minutes.
- T2 Bidirectional version of T1.
- C1 Unidirectional transmission of bigger amount of data.
- C2 Bidirectional version of C1.

Usually one M-bus device support only a few of these application modes [?] [?] [?] [?].



## 1.3 Zigbee

Zigbee, developed by zigbee alliance is usually used for mesh sensor networks because of its short range. This technology is standardized since 2003, so there is many available nodes at the market by now [?] [?] [?].

## 1.4 Bluetooth

Bluetooth has the big advantage, taht it's built in almost every mobile phone, tablet or laptop so there are more options to control the network. The Bluetooth 4.0+ also called BLE (Bluetooth Low Energy) aims to low power wireless sensor networks. It can be used for point-to-point, broadcast or mesh network topology [?] [?] [?] [?].

## 1.5 LoRa

The name LoRa stands for "Long Range" wireless communication with low data rate and power consumption. The protocol enables to modify SF which affects the communication range and data rate. The 1.1 shows this dependence.

Spreading Factor (125kHz Lora)	Bit rate(bps)	Range (varies on propagation conditions)	Time on air(ms) (10 bytes payload)	0.1% of Time on air waiting time	1% of Time on air waiting time
SF7	5470	2 km	56 ms	1 min	6s
SF8	3125	4 km	100 ms	1 min 40s	10s
SF9	1760	6 km	200 ms	3 min 20s	20s
SF10	980	8 km	370 ms	6 min 10s	37s
SF11	440	14 km	740 ms	12 min 20s	1 min 14s
SF12	290	20 km	1400 ms	23 min 20s	2min 20s

**Obrázek 1.1:** LoRa spread factor options [?]

This technology is very attractive for its long range capability and easy to connect nodes. It's complicated to build a full-capacity gateway which is capable of receiving packets at all frequency channels and SF in parallel. The transceiver for this application costs about \$130. Although it's also possible to build single-channel gateway which is way too cheaper, but it can receive packets at only one frequency channel and SF at once [?] [?] [?] [?] [?] [?] [?] [?].

## 1.6 Sigfox

This technology focuses on short message and long range communication applications [?] [?].

## ■ 1.7 Z-Wave

Z-Wave is intended for wireless connectivity for all possible smart home products, controlled by PC, phone, voice, etc. It's based on mesh network topology so every non-battery powered device works as a router to enhance the network range so the more devices are connected in one network, the stronger the network is [?] [?].

## ■ 1.8 Thread

This technology based on IPv6 was developed for home network controlled by smartphone, tablet or PC [?] [?] [?].

## ■ 1.9 RPMA

The "Random Phase Multiple Access" developed by Ingenu designed for M2M and IoT applications [?] [?] [?]. *"RPMA has been deployed for the Machine Network, but can also be rolled out as a private network installation. It is highly suitable for regions, where the rollout of 3GPP LPWA technologies is lagging, where cellular coverage is generally weak, or where users would like to exert full control over their network deployments."*[?]

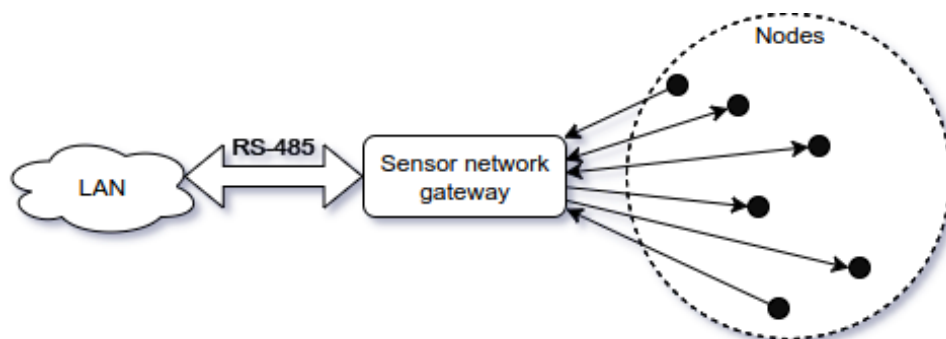
## Kapitola 2

### Stanovení požadavků systému

Cílem tohoto projektu je návrh, realizace a otestování gatewaye, která shromažďuje data z bezdrátových koncových zařízení a přeposílá je přes RS485 LAN na PC master, který je dále přeposílá na IMA K4 server, kde jsou data zpracovávány.

Předpokládá se, že koncová zařízení jsou senzory nebo aktuátory napájeny z baterie, tudíž pro jejich dlouhodobou životnost je kladen důraz na nízkou spotřebu vybrané bezdrátové technologie.

Drátová síť RS485, přes kterou gateway komunikuje se PC masterem používá síťový protokol původně navržený pro přístupové systémy.



Obrázek 2.1: Blokový diagram navrženého systému

## Kapitola 3

### Realizace gatewaye

#### 3.1 Výběr přenosové technologie

Pro tento systém je použita RF technologie LoRa se standardizovaným síťovým protokolem LoRaWAN, ale s požitím pouze jednoho kanálu. Jedenokanálové řešení bylo zvoleno z toho důvodu, že plnohodnotný LoRa transceiver, který přijímá na všech osmi kanálech je příliš drahý (přibližně desetinásobná cena) a složitý k implementaci, zatímco v tomto projektu je kladen důraz na cenu a jednoduchost řešení.

Vybraná technologie používá topologii typu hvězda, tedy koncová zařízení komunikují přímo s gatewayí, zbytek času mohou být ve stavu nízké spotřeby, což má za důsledek delší životnost baterie. Koncová zařízení od různých výrobců jsou plně kompatibilní s cizí gatewayí, tudíž není problém je implementovat do tohoto systému. Je pouze třeba je překonfigurovat pro vysílání na jednom použitém kanále.

##### 3.1.1 Zabezpečení protokolu LoRaWAN

Protokol LoRaWAN používá AES-128 na 2 způsoby, pro síťové a aplikační zabezpečení. Jsou zde tedy 2 šifrovací klíče, NwkSKey a AppSKey.

##### Síťové zabezpečení

Síťové zabezpečení je zde aby bylo hackerům zabráněno odesílání duplikovaných packetů nebo vytváření a vysílání packetů s nasimulovanými daty. Poslední 4 byty packetu obsahují MIC (Message Integrity Code), který je získán zašifrováním dat síťovým klíčem NwkSKey obsahujících mimo jiné celý payload packetu (včetně packet counter). Toto umožňuje odhalit jakoukoliv manipulaci s daty v packetu. LoRaWAN packet také obsahuje counter počítající od nuly od doby kdy bylo LoRaWAN zařízení spuštěno.

Toto umožňuje odhalit duplikování packetů.

### ■ Aplikační zabezpečení

Aplikační klíč AppSKey (Application Session Key) je použit pro zašifrování dat aplikační zprávy (App message) [8] [9].

## ■ 3.2 Výběr komponent

### ■ 3.2.1 Microcontroller

Pro toto zařízení je zvolen mikrocontroller STM32L073RZ se zaměřením na nízkou spotřebu, jelikož je levný, má dostačující vlastnosti a je dostupný ve formě vývojového kitu NUCLEO-L073RZ který byl použit pro vývoj zařízení. Mezi hlavní vlastnosti patří [1]:

- Architektura ARM Cortex-M0+ 32-bit RISC
- Interní Flash paměť 192 KB
- Interní SRAM paměť 20 KB
- Interní EEPROM paměť 6 KB
- Až 32 MHz CPU
- 2x SPI, 3x I2C, 4x USART, LIN, ADC

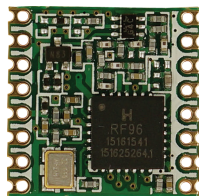
Pořizovací cena kitu přímo na stránce výrobce [www.st.com](http://www.st.com) je \$13 [1] [2].



**Obrázek 3.1:** Vývojový kit NUCLEO-L073RZ [1]

### 3.2.2 LoRa transceiver

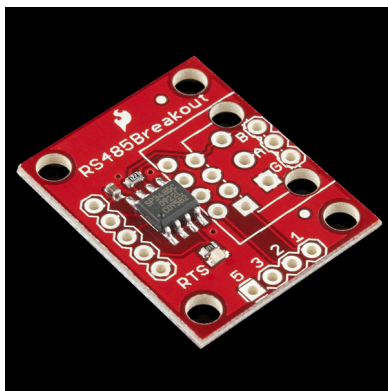
LoRa transceiver čip doposud vyrábí pouze Semtech, pro použití v Evropském pásmu je určen typ SX1276. V tomto návrhu je použita deska RFM95w od firmy HopeRF s integrovaným čipem SX1276 [3]. Pro vývoj zařízení byl využit tento transceiver v tzv. Dragino LoRa Shield [4], který má stejně jako použitý vývojový kit, pinout kompatibilní s Arduino UNO. Pořizovací cena samotného transceiveru RFM95w je okolo \$7, cena Dragino Shieldu se pohybuje okolo \$22 na ebay.



Obrázek 3.2: LoRa transceiver RFM95w [3]

### 3.2.3 RS485 transceiver

SparkFun Transceiver Breakout - RS485 převádí rozhraní UART na RS485, při vstupním napětí 3.3 V. A je dostupný za cenu okolo \$10 [7].



Obrázek 3.3: RS485 transceiver [7]

### 3.3 Implementace LoRaWAN sítě

Jednokanálové použití technologie LoRa umožňuje použít transceiver navržený pro koncová zařízení, kterým jsou pakety kontinuálně odposlouchávány na jednom nastaveném kanále a SF. Tyto dva parametry jsou nakonfigurovány na všech koncových zařízeních v síti.

Použitý protokol IMA\_RS485, původně navržen pro přístupové systémy, umožňuje posílat data pouze pomocí příkazu "průchod", který má kapacitu pro data z koncového zařízení pouze 6 B. Systém je proto navržen neobvyklým způsobem. Přijme-li gateway LoRaWAN packet, nejprve zkontroluje zda zná adresu zařízení, pokud ano, přečte z paměti i typ zařízení, packet dešifruje a dekoduje payload, čímž získá konečné hodnoty senzorů, které pak dále pošle přes RS485 rozhraní na server. LoRaWAN device address a typ každého zařízení v síti je uložena v EEPROM (non-volatile) paměti gatewaye a jsou nastavována K4 serverem. Všechna zařízení mají nastavené stejné šifrovací klíče a gateway je má uložené v EEPROM.

Pro tento projekt byla vyvinuta knihovna pro dekódování payloadu na základě dokumentů [8] [9].

### 3.4 Implementace komunikačního protokolu v síti IMA\_RS485

Každé zařízení na této sběrnici má svoji adresu, která mu je nastavena externě. Server má adresu 0xFF, adresa pro všechny (broadcast) je 0x00 a zařízení v této síti mohou mít adresu libovolnou (krom těchto dvou). V tomto projektu je tento protokol naprogramován v souborech rs485\_protocol.h a rs485\_protocol.c.

#### 3.4.1 Syntaxe příkazů

popis	adresa příjemce	adresa odesílatele	typ příkazu	délka dat	data	crc
počet bytů	1	1	1	2	délka dat	1

**Tabulka 3.1:** Syntaxe příkazu pro komunikaci se serverem přes RS485

Veškeré typy příkazu jsou zdefinovány konstanty s předponou CKP\_CMD\_ v souboru ./Inc/rs485\_protocol.h.

Příkazy odeslané ze serveru obsahují navíc synchronizační byte na začátku 0xAA. crc je pro kontrolu XOR přes všechny předchozí byty v celém příkazu

kromě synchronizačního bytu.

### ■ 3.4.2 Statusy

Zařízení má dva možné statusy, offline a online. Status zařízení je odesílán periodicky s typem příkazu 0x10 a jedním bajtem dat označujícím status. Server na tento příkaz neodpovídá.

#### ■ Offline

Je-li zařízení zapnuto, je ve stavu offline. Nemá povoleno odesílat data ze senzorů, pouze odpovídá na příkazy serveru a následně dostane příkaz od serveru pro přechod do stavu online. Odesílání příkazu signalizující tento stav je s periodou 30 s a obsahuje bajt 0xEE.

#### ■ Online

V tomto stavu je povoleno odesílání dat ze senzorů. Odesílání příkazu signalizující tento stav je s periodou 45 s a obsahuje bajt 0x00.

### ■ 3.4.3 ACK neboli potvrzení

Zařízení odpovídá na každý platný příkaz od serveru ACK. Typ příkazu ACK je 0x06 a v datech je jeden byte, což je typ příkazu na který právě odpovídá potvrzením.

Server odpovídá ACK se stejným typem příkazu 0x06, ale s žádnými daty v příkazu. Délka dat je tedy 0.

### ■ 3.4.4 Přidávání LoRaWAN zařízení do systému

Pokud server dostane od zařízení příkaz oznamující že je ve stavu offline, nejprve tomu zařízení v RS485 síti pošle seznam adres LoRaWAN modulů v síti a pak ho přepne do stavu online.

Přijímání seznamu adres je tvořeno sekvencí příkazů typu 0x8F.

První Byte dat je counter packetu začínající od nuly, který označuje číslo odeslaného packetu v sekvenci. Na každý tento packet v sekvenci zařízení odpoví ACK příkaz, který se liší od obvyčejného ACK příkazu tím, že v datech packetu navíc obsahuje counter packety v sekvenci.

První příkaz této sekvence má délku dat 2 byty, které mají hodnotu 0x00



přičemž první je counter. Další příkazy hned za counter bytem obsahují několik osmibytových adres, jejichž počet je různý.

LoRaWAN používá 4-bytové devAddr. První 4 byty jsou devAddr, pátý byte je typ zařízení a ostatní zatím nejsou použity. Momentálně je zaveden pouze jeden typ zařízení a to je RH1S001, pro nějž hodnota tohoto bytu je 0.

Příkaz ukončující tuto sekvenci příkazů má délku 4 byty, což je tedy counter, 0xFF a 2 byty crc přes všechny odeslané adresy (nepodstatné, tudíž ho nepoužívám).

V tabulce 3.2 je pro názornost příklad sekvence příkazů. Jak již bylo řečeno, příkazy od serveru lze jednoduše odlišit tím, že vždy začínají bytem 0xAA.

příkaz	data
start	AA 10 FF 8F 02 00 00 00 62
ACK	FF 10 06 02 00 8F 00 64
data	AA 10 FF 8F 21 00 01 B1 C4 12 00 00 00 00 00 B2 C4 12 00 00 00 00 00 B3 C4 12 00 00 00 00 00 B4 C4 12 00 00 00 00 00 44
ACK	FF 10 06 02 00 8F 01 65
data	AA 10 FF 8F 19 00 02 B5 C4 12 00 00 00 00 00 B6 C4 12 00 00 00 00 00 F6 1F 01 26 00 00 00 00 B6
ACK	FF 10 06 02 00 8F 02 66
konec	AA 10 FF 8F 04 00 03 FF 2A 57 E5
ACK	FF 10 06 02 00 8F 03 67

**Tabulka 3.2:** Příklad sekvence příkazů pro přidávání LoRaWAN zařízení do systému

### 3.4.5 Odesílání dat ze senzoru

Jak již bylo zmíněno, protokol byl navrhnut pro přístupové systémy a nedošlo k žádné úpravě pro tuto odlišnou aplikaci. Data ze senzorů se tedy posílají s použitím stávajícího příkazu "průchod". Typ tohoto příkazu je 0x10. První byte dat označuje typ průchodu, byl zvolen konstantní byte 0xD0. Dále následuje LoRaWAN adresa modulu od kterého byl přijat packet. Dále následují 4 byty dat ze senzoru. Dále 2 byty oznamující čas průchodu, což v tomto projektu není použito a tyto dva byty mají tedy hodnotu 0xFF. A nakonec jsou další 2 byty obsahující data ze senzoru. Celý tento příkaz průchod tedy obsahuje pouze 6 bytů pro data ze senzoru.

Server na tento příkaz odpovídá ACK a má čas na odpověď standardně 3 sekundy, ale tento parametr je nastavitelný. Pokud v tomto timeoutu neodpoví,

zařízení příkaz zopakuje a změní typ příkazu na 0x20. Pokud server ani na třetí opakování neodpoví ACK, zařízení se přepne do stavu offline a vymaže frontu příkazů k odeslání.

### 3.4.6 Dotaz na příznaky

Server se může zeptat s jak dlouhými adresami zařízení pracují. Je to typ příkazu 0x49 a délka dat je 0. Zařízení na to odpoví ACK, ale navíc je v datech příkazu byte 0x04 a server pak počítá s tím, že pracují se 64-bit adresami (ve skutečnosti ale používám 32-bitové a zbylé 4 byty jsou pro data).

## 3.5 Paměť

Konfigurace systému, DevAddr a typ všech zařízení v síti jsou uložena v non-volatile paměti procesoru EEPROM, která je o velikosti 6144 B. Paměť je tedy rozdělená tak, že od adresy 0 až po 6080 je prostor pro ukládání LoRaWAN zařízení a od 6080 až po 6144 je prostor pro ukládání konfigurace systému.

Každé LoRaWAN zařízení v síti má v paměti uložené device address (4 byty), typ zařízení (1 byte) a další 3 byty jsou rezervovány. Jedno zařízení zabírá tedy 8 B, takže jich pro jednu síť může být až 760.

## 3.6 Komunikace přes USB

Komunikace přes USB s PC má za účel konfiguraci systému a Log aplikace. Lze k tomu použít libovolný terminál s nastavením viz tabulka 3.2.

Baud rate	115200
Data bits	8
Parity	none
Stop bits	1
Flow control	none

**Tabulka 3.3:** Nastavení USB terminálu

Při komunikaci jsou data oddělována bytem 0x0D, ale je akceptována i sekvence 0x0D 0x0A.

### 3.6.1 Log aplikace

Gateway odesílá informace přes USB o tom, co právě provádí. Níže je příklad výpisu dat pro případ, že gateway přijme packet. Nejprve jsou vypsána

data týkající se LoRaWAN protokolu, DevAddr bylo rozpoznáno, payload byl dešifrován a na základě typu senzoru byl dekodován payload a vyčteny hodnoty senzorů LoRaWAN zařízení. Poslední řádek obsahuje data odeslané přes RS485 na server.

---

```

-----
Rx -> LoRaWAN, pktCnt: 406
RSSI: -29, SNR: 9, length: 22
Packet rawData: 40 F6 1F 01 26 C0 A1 30 08 D4 5D 93 F0 F0 F6 60 C0
                04 BC BE 4B 24
devAddr: F6 1F 01 26
MHDR: 40
FCtrl: c0
FCnt: 12449
FPort: 08
MIC: BC BE 4B 24
adaptive data rate: 1
ack: 0
direction: UP
message (encrypted): D4 5D 93 F0 F0 F6 60 C0 04
message (decrypted): 01 44 6C 83 05 00 FF FF 71

Sensor type: RHF1S001
temperature: 27.46 C, humidity: 58 %
period: 10 s, RSSI: -29 dBm, SNR: 9 dB, battery voltage: 2.6 V
Tx -> RS-485: " FF 10 10 0D 00 D0 F6 1F 01 26 BA 0A 3A E3 FF FF 09
               1A 96"

```

---

### 3.6.2 Konfigurace systému

Pro vstup do configuration setup menu musí být odeslán příkaz "config". Z menu je možné vystoupit kdykoliv příkazem "quit"(bez uložení). Při vstupu do stavu konfigurace systému je pozastavena činnost gatewaye (přijímání LoRaWAN packetů, komunikace se serverem,...). V terminálu je nejprve vypsána kompletní aktuální konfigurace systému a následně menu konfigurace. Uživatel pak vybírá zadáním čísla. Níže je zobrazen příklad výpisu po vstupu do konfigurace.

---

```

-----_Entering configuration setup_-----

System configuration:

*** LoRa channel:
channel: 0 (868.1 Mhz)
SF7

```

```

*** RS485 channel:
my address: 10
server address: FF
timeout: 3 s

*** LoRaWAN keys:
NwSKey: FD 90 0D 8C 70 9F 19 24 18 EC FD D4 28 0C AC 47
AppSKey: 68 9F D0 AC 7A 0F 95 58 B1 19 A0 16 17 F4 16 33

Config menu:
1 -> Config LoRa channel
2 -> Config RS485 channel
3 -> Config LoRaWAN protocol
4 -> Print all LoRaWAN devices
5 -> Erase all LoRaWAN devices
6 -> Restore to default configuration
7 -> Exit without save
8 -> Save and exit

```

---

Jsou zde tedy 3 stavy konfigurace, přičemž je možné vždy jednotlivá nastavení přeskokovat odesláním "prázdného příkazu" 0x0D (v terminálu obvykle stačí stisknout Enter). Systém při konfiguraci vždy vypíše jaká data mají být zadána v jakém tvaru a zároveň současnou hodnotu měněného parametru. Zadaná data uživatelem jsou vždy zkontrolována zda splňují požadovaný tvar. Pokud ne, uživatel je o tom informován a vyzván k dalšímu pokusu. Po provedení konfigurace následuje vždy návrat do menu. Pro uložení nové konfigurace je potřeba v menu vybrat "Save and exit", systém pak následně vypíše které parametry byly změněny.

### ■ Config LoRa channel

Konfigurace LoRa RF kanálu zahrnuje nastavení SF a frekvenční pásmo. Níže je příklad konfigurace.

---

```

LoRa channel configuration:
Enter SF number (7-12)
(current: 7)
8
SF8 set.

Enter LoRa channel number (0-7)
ch0 is 868.1 Mhz
ch1 is 868.3 Mhz
ch2 is 868.5 Mhz
ch3 is 867.1 Mhz
ch4 is 867.3 Mhz
ch5 is 867.5 Mhz
ch6 is 867.7 Mhz

```

```
ch7 is 869.0 Mhz
(current: 0)
1
channel 1 set.
```

---

### ■ Config RS485 channel

Konfigurace RS485 kanálu pro komunikaci se serverem zahrnuje nastavení adresy tohoto zařízení, adresa serveru a timeout, což je doba čekání na zprávu ACK od serveru po odeslání příkazu obsahujícího data ze senzorů. Níže je příklad konfigurace.

---

```
RS485 channel configuration:
Enter address of this device, FF and 00 are reserved.
(current: 10)
11
Address of this device is set to: 11

Enter server address:
(current: FF)
FE
Server address is set to: FE

Enter timeout (seconds)
(current: 3)
5
timeout set to: 5 s
```

---

### ■ Config LoRaWAN protocol

Konfigurace LoRaWAN zahrnuje nastavení šifrovacích klíčů pro LoRaWAN protokol. Níže je příklad konfigurace.

---

```
LoRaWAN protocol configuration:
Enter NwSKey (16 bytes in HEX)
(current: FD 90 0D 8C 70 9F 19 24 18 EC FD D4 28 0C AC 47)
11111111222222223333333344444444
NwSKey set to: 11 11 11 11 22 22 22 22 33 33 33 33 44 44 44 44

Enter AppSKey (16 bytes in HEX)
(current: 68 9F D0 AC 7A 0F 95 58 B1 19 A0 16 17 F4 16 33)
11111111222222223333333344444444
AppSKey set to: 11 11 11 11 22 22 22 22 33 33 33 33 44 44 44 44
```

---

### ■ Print all LoRaWAN devices

Vypíše všechna LoRaWAN zařízení uložená v paměti. Níže je příklad.

---

```

number.....0:
Device Address: B1 C4 12 00
Device Type: RH1S001
number.....1:
Device Address: B2 C4 12 00
Device Type: RH1S001
number.....2:
Device Address: B3 C4 12 00
Device Type: RH1S001
number.....3:
Device Address: B4 C4 12 00
Device Type: IMA_tempPress
number.....4:
Device Address: B5 C4 12 00
Device Type: IMA_tempPress

```

---

### ■ Restore to default configuration

Po zvolení této možnosti je načtena defaultní konfigurace systému, která obsahuje hodnoty viz tabulka 3.4.

popis	hodnota
RS485 myAddr	0x10
RS485 serverAddr	0xFF
RS485 timeout	3
LoRa SF	SF7
LoRa channel	0 (868.1 Mhz)
NwSKey	FD 90 0D 8C 70 9F 19 24 18 EC FD D4 28 0C AC 47
AppSKey	68 9F D0 AC 7A 0F 95 58 B1 19 A0 16 17 F4 16 33

**Tabulka 3.4:** Defaultní konfigurace systému

## 3.7 Koncová zařízení

Jelikož používaný protokol ke komunikaci se serverem je datově omezen na 6B na jeden paket, payload senzoru je dekodován v gatewayi a v packetu odeslaném na server jsou pouze nejdůležitější data, která se vejdu do této velikosti. Na základě typu koncového zařízení jsou data v packetu zakódována, tudíž typ koncového zařízení musí být v gatewayi implementován. Typ koncového zařízení je definován jedním bytem a je uložen v gatewayi i na serveru společně s DevAddr u každého koncového zařízení.

Momentálně jsou podporovány dva typy koncových zařízení, dle potřeby není problém rozšířit FW gatewaye o další typy koncových zařízení.

Typ zařízení	Hodnota
RHF1S001	0x00
IMA_tempPress	0x01

**Tabulka 3.5:** Typy koncových zařízení

### 3.7.1 Zpracování dat jednotlivých typů koncových zařízení

Níže je popsáno pro jednotlivá koncová zařízení jak jsou data uložena ve struktuře, jak jsou data z této struktury zpracována a zobrazena a nakonec jak vybraná data jsou zapsána do výsledného bufferu o délce 6B, který je odeslán na K4 server

#### RHF1S001

Senzor od firmy RisingHF měří teplotu a vlhkost.

```

1  /* RHF1S001 data structure */
2  typedef struct {
3      int16_t temperature;
4      uint8_t humidity;
5      uint16_t period;
6      int8_t rssi;
7      int8_t snr;
8      uint8_t battery;
9  } RHF1S001_data_t;
10
11  /* Print the data from the structure */
12  printf("temperature: %d.%d C, ", RHF1S001_data.temperature / 100,
13        RHF1S001_data.temperature % 100);
14  printf("humidity: %d %%\n", RHF1S001_data.humidity);
15  printf("period: %d s, ", (int)RHF1S001_data.period);
16  printf("RSSI: %d dBm, ", RHF1S001_data.rssi);
17  printf("SNR: %d dB, ", RHF1S001_data.snr);
18  printf("battery voltage: %d.%d V\r\n", RHF1S001_data.battery/10,
19        RHF1S001_data.battery % 10);

```

```

18
19  /* Put the data into 6B long buffer, that is transmitted to the K4 server */
20  buffer[0] = RHF1S001_data.temperature & 0xFF;
21  buffer[1] = RHF1S001_data.temperature >> 8;
22  buffer[2] = RHF1S001_data.humidity;
23  buffer[3] = RHF1S001_data.rssi;
24  buffer[4] = RHF1S001_data.snr;
25  buffer[5] = RHF1S001_data.battery;

```

## IMA\_tempPress

Senzor vytvořený ve firmě IMA, měřící teplotu a tlak.

```

1  /* IMA_tempPress data structure */
2  typedef struct {
3      int16_t temperature;
4      uint16_t pressure;
5      int8_t rssi;
6      int8_t snr;
7  } IMA_tempPress_data_t;
8
9  /* print the data from the structure */
10 printf("temperature: %d.%d C, ", IMA_tempPress_data.temperature / 100,
11       IMA_tempPress_data.temperature % 100);
12 printf("pressure: %d.%d Pa\r\n", IMA_tempPress_data.pressure/10,
13       IMA_tempPress_data.pressure % 10);
14 printf("RSSI: %d dBm, SNR: %d dB\r\n", IMA_tempPress_data.rssi,
15       IMA_tempPress_data.snr);
16
17 /* Put the data into 6B long buffer, that is transmitted to the K4 server */
18 buffer[0] = IMA_tempPress_data.temperature & 0xFF;
19 buffer[1] = IMA_tempPress_data.temperature >> 8;
20 buffer[2] = IMA_tempPress_data.pressure & 0xFF;
21 buffer[3] = IMA_tempPress_data.pressure >> 8;
22 buffer[4] = IMA_tempPress_data.rssi;
23 buffer[5] = IMA_tempPress_data.snr;

```

## 3.7.2 Přidávání koncových zařízení ze serveru K4

Koncová zařízení sítě se nastavují ze serveru K4 v podobě seznamu offline karet s délkou UID 8B. LoRaWAN device address je dlouhá 4B, jeden byte je navíc použit pro typ koncového zařízení, zbylé 3 byty jsou nuly. Jelikož typ zařízení je uložen v gateway i na serveru. Při odesílání zprávy průchod se tedy už typ zařízení neposílá z důvodu datového omezení v této zprávě (6B). Na serveru K4 se UID nastavuje jako dekadické číslo. Níže je příklad vytvoření výsledného čísla obsahující DevAddr a typ zařízení, které se zadává do K4 serveru.



#### ■ Příklad

Pro případ, kde typ zařízení je 01 a DevAddr AABBCDD (little endian) výsledné číslo v hexadecimální podobě je 01DDCCBBAA. Následně se překládá do decimalní podoby, výsledné číslo k zadání do K4 serveru je tedy 8016149418.

## 3.8 Zapojení

LoRa shield [4] je nasazen přímo na vývojový kit. Nukleo kit neobsahuje ISCP konektor, který je součástí pinoutu Arduina a LoRa shield má SPI piny MISO a MOSI přivedeny právě na tento konektor. Musí být tedy propojeny externě viz obrázek ???. Jumperry na shieldu musí také být stejně jako v obrázku.

Pro komunikaci s LoRa transceiverem je tedy použito SPI1, pro komunikaci přes USB je použito USART2 a pro komunikaci přes RS485 je použito UART1.

Periférie	Název pinu	Pin procesoru
RS485 transceiver	RX	PC1
	TX	PC0
	RTS	PB1
LoRa transceiver	CS	PB6
	CLK	PA5
	MISO	PA6
	MOSI	PA7
	RST	PC7
	DIO0	PA10

**Tabulka 3.6:** Pinout připojení externích periférií k procesoru

## 3.9 Instalace

K nahrání zkompilevaného programu do procesoru z PC není potřeba instalovat žádný SW. Kit je potřeba připojit k PC přes USB. V PC se to zobrazí jako flash disk. Zkompilevaný program s koncovkou .binary stačí překopírovat na toto zařízení. Po dobu kopírování souboru bliká na kitu LED1 červená/zelená. Jakmile kopírování skončí, program se spustí. Kit je také možné restartovat černým tlačítkem reset.

Pro uvedení Gatewaye do provozu je nutné se připojit k zařízení přes USB a nastavit všechny parametry viz sekce v tomto dokumentu: "Konfigurace systému".

## 3.10 Zdrojové soubory projektu

Projekt byl naprogramován v AtolicTrueSTUDIO, což je IDE založené na Eclipse. K programování procesoru byly použity standartní HAL knihovny od firmy ST Microelectronics. Pro šifrování LoRaWAN packetu byla použita knihovna AES-128, dostupná na githubu [5] a knihovna OpenPANA také dostupná z githubu [6]. Níže je seznam zdrojových souborů.

Drivers .....	STM32 Drivers
Inc .....	Headers
aes.h .....	AES-128 library for LoRaWAN packet encryption
cmac.h .....	library for CMAC calculation in LoRaWAN protocol
LinkedList_ByteArray.h ..	Byte array linked list library for stacks
LoRaWAN_packet.h .....	LoRaWAN library for packet data decoding
stm32l0xx_hal_conf.h .....	STM32 HAL library
ByteArray.h .....	Library for Byte array operations
LoRa.h .....	Library for interfacing LoRa transceiver
main.h .....	Main file
stm32l0xx_it.h .....	STM32 HAL library
eeprom.h .....	Library for eeprom operations
LoRa_sensors.h .....	Library for decoding data from payload
rs485_protocol.h .....	Library for RS485 IMA protocol
usb.h ...	Library for USB communication and system configuration
Src .....	Sources
aes.c .....	source file to the aes.h
cmac.c .....	source file to the cmac.h
LinkedList_ByteArray.c	source file to the LinkedList_ByteArray.h
LoRaWAN_packet.c .....	source file to the LoRaWAN_packet.h
stm32l0xx_hal_msp.c .....	HAL source file
ByteArray.c .....	source file to the ByteArray.h
LoRa.c .....	source file to the LoRa.h
main.c .....	main source file
stm32l0xx_it.c .....	HAL source file
eeprom.c .....	source file to the eeprom.h
LoRa_sensors.c .....	source file to the LoRa_sensors.h
rs485_protocol.c .....	source file to the rs485_protocol.h
system_stm32l0xx.c .....	HAL source file
usb.h .....	source file to the usb.h



## Literatura

- [1] *NUCLEO-L073RZ*. ST Microelectronics [Online]. Available: <https://www.st.com/en/evaluation-tools/nucleo-l073rz.html> [Accessed: 19-Sep-2018].
- [2] *NUCLEO-L073RZ* ARM Mbed. [Online]. Available: <https://os.mbed.com/platforms/ST-Nucleo-L073RZ/> [Accessed: 19-Sep-2018].
- [3] *RFM95/96/97/98(W) - Low Power Long Range Transceiver Module*. HopeRF electronic. V1.0. [Online]. Available: [http://wiki.dragino.com/index.php?title=Lora\\_Shield](http://wiki.dragino.com/index.php?title=Lora_Shield) [Accessed: 19-Sep-2018].
- [4] *Lora Shield*. Dragino. [Online]. Available: [http://wiki.dragino.com/index.php?title=Lora\\_Shield](http://wiki.dragino.com/index.php?title=Lora_Shield) [Accessed: 19-Sep-2018].
- [5] *tiny-AES128-C* bitdust. [Online]. Available: <https://github.com/bitdust/tiny-AES128-C> [Accessed: 19-Sep-2018].
- [6] *openpana*. OpenPANA. [Online]. Available: <https://github.com/OpenPANA/openpana> [Accessed: 19-Sep-2018].
- [7] *SparkFun Transceiver Breakout - RS-485* Sparkfun. [Online]. Available: <https://www.sparkfun.com/products/10124> [Accessed: 20-Sep-2018].
- [8] *LoRaWAN Specification*. LoRa Alliance. v1.1. Sparkfun. [Online]. Available: <https://loro-alliance.org/resource-hub/lorawantm-specification-v11> [Accessed: 20-Sep-2018].
- [9] Robert Miller. *LoRa Security Building a Secure LoRa Solution*. MWR Labs Whitepaper. [Online]. Available: <https://labs.mwrinfosecurity.com/assets/BlogFiles/mwri-LoRa-security-guide-1.2-2016-03-22.pdf> [Accessed: 20-Sep-2018].