



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV AUTOMATIZACE A INFORMATIKY**

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**KOMUNIKACE AUTONOMNÍCH  
MOBILNÍCH ROBOTŮ A NAVIGAČNÍCH  
BODŮ**

COMMUNICATION OF AUTONOMOUS MOBILE ROBOTS AND NAVIGATION  
SPOTS

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

Bc. Jakub Středa

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. Daniel Zuth, Ph.D.

BRNO 2017



# Zadání diplomové práce

Ústav: Ústav automatizace a informatiky  
Student: **Bc. Jakub Středa**  
Studijní program: Strojní inženýrství  
Studijní obor: Aplikovaná informatika a řízení  
Vedoucí práce: **Ing. Daniel Zuth, Ph.D.**  
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

## Komunikace autonomních mobilních robotů a navigačních bodů

### Stručná charakteristika problematiky úkolu:

Práce se bude zabývat realizací komunikace v situaci několika autonomních mobilních robotů a inteligentních navigačních bodů. Po rešeršní části bude vybrané schéma komunikace realizováno na vývojových kitech Nucleo a bezdrátovými moduly. Výstupem práce bude ověření funkčnosti a vytvoření ukázkových kódů v prostředí mbed.

### Cíle diplomové práce:

1. Popište možné schéma komunikace
2. Popište technologie použitelné pro tento typ úlohy
3. Vyberte a otestujte bezdrátové moduly
4. Realizujte vybrané varianty na vývojových kitech

### Seznam doporučené literatury:

MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky .. Praha: BEN - technická literatura, 2003. µC & praxe. ISBN 8073000776.

Home | mbed. [online]. Copyright © ARM Ltd. Copyright 2016 [cit. 04.11.2016]. Dostupné z: <https://www.mbed.com/en/>.

IQRF - Technology for wireless. IQRF - Technology for wireless [online]. Copyright © MICRORISC s.r.o. [cit. 04.11.2016]. Dostupné z: <http://www.iqrf.org/>

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

---

doc. Ing. Radomil Matoušek, Ph.D.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **ABSTRAKT**

Práce se zabývá návrhem a realizací komunikace mezi autonomními mobilními roboty a stacionárními lokalizačními stanicemi s využitím vývojových desek Nucleo a bezdrátových IQRF modulů. Práce je v teoretické části zaměřena na charakteristiku vhodných hardwarových prvků a komunikačních technologií. V praktické části se práce zabývá popisem možných schémat komunikace, použitého hardwaru, vypracovaného algoritmu, softwaru jednotlivých členů a diagnostických nástrojů vyvinutých pro účely práce. V poslední části je pozornost věnována testování výsledného řešení.

## **ABSTRACT**

The thesis deals with design and realization of communication among autonomous mobile robots and stationary localization stations using Nucleo development boards and a wireless IQRF modules. The theoretical part is focused on characterization of suitable hardware elements and communication technologies. In the practical part, the thesis deals with the description of possible communication schemes, used hardware, developed algorithm, software of individual members and diagnostic tools developed for thesis's purpose. In the last part, attention is paid to testing the resulting solution.

## **KLÍČOVÁ SLOVA**

IQRF technologie, bezdrátová komunikace, mbed, mikrokontrolér, vývojová deska Nucleo, SPI, I<sup>2</sup>C, UART

## **KEYWORDS**

IQRF technology, wireless communication, mbed, microcontroller, development board Nucleo, SPI, I<sup>2</sup>C, UART



## **BIBLIOGRAFICKÁ CITACE**

STŘEDA, Jakub. *Komunikace autonomních mobilních robotů a navigačních bodů*, Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky.





## **PODĚKOVÁNÍ**

Tímto děkuji svému vedoucímu práce Ing. Danielu Zuthovi, Ph.D. za jeho ochotu a čas při vytváření této práce. Za podporu bych dále rád poděkoval své rodině a blízkým.



## **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Daniela Zutha, Ph.D. a s použitím literatury uvedené v seznamu literatury.

V Brně dne 19. 5. 2017

.....

Jakub Středa



# OBSAH

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>ÚVOD .....</b>                       | <b>15</b> |
| <b>2</b> | <b>ROZBOR PROBLEMATIKY .....</b>        | <b>17</b> |
| <b>3</b> | <b>PLATFORMA .....</b>                  | <b>19</b> |
| 3.1      | MBED .....                              | 19        |
| 3.2      | ARDUINO .....                           | 21        |
| 3.3      | SINGLE BOARD COMPUTER.....              | 23        |
| <b>4</b> | <b>KOMUNIKAČNÍ TECHNOLOGIE.....</b>     | <b>25</b> |
| 4.1      | KOMUNIKAČNÍ SBĚRNICE .....              | 25        |
| 4.1.1    | SPI .....                               | 25        |
| 4.1.2    | I <sup>2</sup> C.....                   | 26        |
| 4.1.3    | UART .....                              | 26        |
| 4.1.4    | CAN .....                               | 26        |
| 4.1.5    | MODBUS .....                            | 27        |
| 4.2      | BEZDRÁTOVÉ KOMUNIKAČNÍ TECHNOLOGIE..... | 27        |
| 4.2.1    | IQRF.....                               | 27        |
| 4.2.2    | Bluetooth.....                          | 28        |
| 4.2.3    | ZigBee.....                             | 29        |
| 4.2.4    | Wi-Fi.....                              | 30        |
| 4.2.5    | IrDA .....                              | 30        |
| <b>5</b> | <b>REALIZACE .....</b>                  | <b>33</b> |
| 5.1      | ROZBOR.....                             | 33        |
| 5.1.1    | Ukázkové situace.....                   | 33        |
| 5.1.2    | Vybrané řešení .....                    | 36        |
| 5.2      | MBED ST-NUCLEO F303-K8 .....            | 37        |
| 5.3      | IQRF MODUL DCTR-72DAT .....             | 38        |
| 5.4      | PRŮBĚH KOMUNIKACE .....                 | 40        |
| 5.4.1    | IQRF.....                               | 40        |
| 5.4.1    | Master věž – robot .....                | 42        |
| 5.4.2    | Master věž – slave věž.....             | 42        |
| 5.5      | ALGORITMUS.....                         | 44        |
| 5.5.1    | IQRF modul.....                         | 44        |
| 5.5.2    | Master věž.....                         | 45        |
| 5.5.3    | Slave věž.....                          | 46        |
| 5.5.4    | Robot.....                              | 46        |
| 5.6      | STRUKTURA DAT .....                     | 47        |
| 5.7      | HLEDÁNÍ DOSTUPNÝCH ROBOTŮ .....         | 49        |
| 5.8      | PŘENOS POLOH .....                      | 49        |
| 5.9      | SOFTWARE .....                          | 50        |
| 5.9.1    | mbed .....                              | 51        |
| 5.9.2    | IQRF modul.....                         | 53        |
| 5.9.3    | Diagnostický software.....              | 53        |
| <b>6</b> | <b>EXPERIMENTY .....</b>                | <b>57</b> |
| 6.1      | POUŽITÝ HARDWARE A SOFTWARE .....       | 57        |
| 6.2      | ČAS PŘENOSU .....                       | 57        |
| 6.2      | DOSAHI IQRF MODULU.....                 | 58        |
| 6.3      | REÁLNÁ SITUACE .....                    | 58        |

|   |                                 |    |
|---|---------------------------------|----|
| 7 | ZÁVĚR .....                     | 61 |
| 8 | SEZNAM POUŽITÉ LITERATURY ..... | 63 |
| 9 | SEZNAM PŘÍLOH .....             | 67 |

# 1 ÚVOD

V oblasti automatizace, ale i v jiných průmyslových odvětvích, je nezbytné pro určité aplikace komunikovat mezi více zařízeními. Příkladem může být snímání teploty, kdy je snímací prvek připojen k vyhodnocovacímu zařízení pomocí komunikační linky. Mezi složitější situace patří například koordinace vzájemného pohybu členů jednoho celku. Moderní druhy komunikace jsou přítomné v každodenním životě již mnoho let. Ustupující pevné telefonní linky nahrazují mobilní zařízení. Ve vyspělých zemích je zavedení internetu v domácnosti již běžnou skutečností. Příchod stále nových možností uplatnění komunikačních technologií se však nezastavuje.

Do povědomí se dostává tzv. Internet věcí (IoT, Internet of Things). Tato technologie je zaměřena zejména na běžná domácí zařízení, jako například osvětlení, televize, pračka či rychlovarná konvice, a jejich propojení do jedné sítě a případně i k internetu. Záměr je ovládat vzdáleně (z jiné místnosti, zahrady nebo dokonce ze zahraničí) požadované spotřebiče. Důvodů může být několik. Například mít v přesně daný čas uvařenou vodu na čaj nebo na dovolené ovládat světla. Velice zajímavá možnost je volba automatických programů simulujících přítomnost osob v budově či bytě, a tím odradit potenciální zloděje.

Propojení více zařízení s sebou nese potřebu zajistit spolehlivou výměnu informací. Cílem této práce je navrhnout, realizovat a otestovat komunikaci pro situaci několika autonomních mobilních robotů a inteligentních navigačních bodů. Stanice slouží k určení polohy robotů ve vymezeném prostoru, avšak samotný proces lokalizace není předmětem této práce. Je nutné zajistit dostatečnou spolehlivost a rychlost výměny informací a navrhnout takový systém, aby při výpadku některého z prvků byla zajištěna funkcionality celku.

V teoretické části jsou charakterizovány hardwarové prvky vhodné pro tento úkol. V jednotlivých kapitolách jsou popsány jejich hlavní charakteristiky a vlastnosti. Dále jsou s využitím odborné literatury uvedeni zástupci z řad komunikačních technologií. V praktické části práce jsou představeny zvolené komponenty, komunikační technologie a vývoj vlastního řešení. Následuje kapitola věnující se testování realizovaného řešení a na závěr práce jsou zhodnoceny dosažené výsledky.

Výstupem práce je kompletní komunikační struktura mezi mobilními roboty a lokalizačními věžemi. Výsledné řešení by mohlo být použito v průmyslovém prostředí například ve skladovacích prostorech pro automatizaci provozu.





## 2 ROZBOR PROBLEMATIKY

Pro určení polohy a navigaci v prostoru je vyvinuto mnoho technologií. Pro pokrytí velkých ploch, případně celého zemského povrchu, jsou vhodné družicové navigační systémy GNSS (Global Navigation Satellite System). Nejrozšířenější technologií je systém GPS (Global Positioning System), který využívá americké družice na oběžné dráze Země. Jedná se o velmi spolehlivou metodu. O tom svědčí i fakt, že účelem, pro který byl tento systém zbudován, je vojenské využití. K určení polohy se využívá informace o poloze konkrétního satelitu a době přenosu signálu k objektu, jehož pozici zjišťujeme. Jsou tedy potřeba minimálně tři družice pro jednoznačné určení polohy. Dále existují alternativy jako například ruský Glonass, čínské Pej-tou či evropské Galileo, jejichž princip je totožný. Tato metoda lokalizace nemusí být vždy dostupná či vhodná z důvodu nedostatečné přesnosti, nedostatku signálu v budovách a podzemí nebo energetické náročnosti [1].

Tato práce má za úkol realizovat komunikaci pro soustavu stacionárních lokalizačních věží a autonomních robotů, jejichž poloha je určována podobným způsobem. V případě potřeby stanovení pozice je robot zastaven a pomocí senzorů jsou hledány stacionární věže, které vysílají dva druhy signálu: světelný paprsek a ultrazvukové vlny. Z časového úseku mezi přijetím obou signálů, které je způsobeno jejich různou rychlostí šíření v prostoru, a znalostí polohy dané věže je určena vzdálenost od této stanice. Je potřeba zajistit, aby věže v případě potřeby odvysílaly lokalizační sekvenci. Hlavním záměrem je navrhnout kompletní strukturu a průběh komunikace, což zahrnuje mimo jiné rozhodnutí, jakým způsobem bude určen okamžik, kdy věže vysílají lokalizační sekvenci a zda bude sekvence vysílána na žádost robota či se bude periodicky opakovat. Dále je důležité zajistit funkcionalitu i v případě, že se robot vzdálí z dosahu dané skupiny věží a následně se opět přiblíží. V případě návratu do dosahu stanic by mělo dojít k opětovnému navázání spojení. Rovněž je potřeba umožnit výměnu informací o poloze jednotlivých mobilních prvků například z důvodu plánování tras nebo přidělování úkolů dle polohy daného robota.

Celý proces by měl být navržen jako autonomní a k tomu je potřeba zvolit vhodný hardware pro zajištění výpočetního výkonu. Při výběru je možné volit z široké škály možností jako například jednodeskové počítače či různé vývojové desky. Je potřeba zvážit nejen jejich výpočetní kapacitu, ale i spotřebu energie v případě provozu na baterii a v neposlední řadě cenu, která by neměla být vzhledem k náročnosti úlohy neúměrně vysoká.

Pro komunikaci je rovněž potřeba zvolit vhodnou technologii. Přenos informací může být realizován pomocí metalických či jiných vodičů, bezdrátově, případně užitím kombinace těchto variant. Jednotlivé technologie se navzájem liší fyzikálním principem, spolehlivostí, maximálním dosahem, ale i zakomponovanými pomocnými a zabezpečovacími funkcemi.



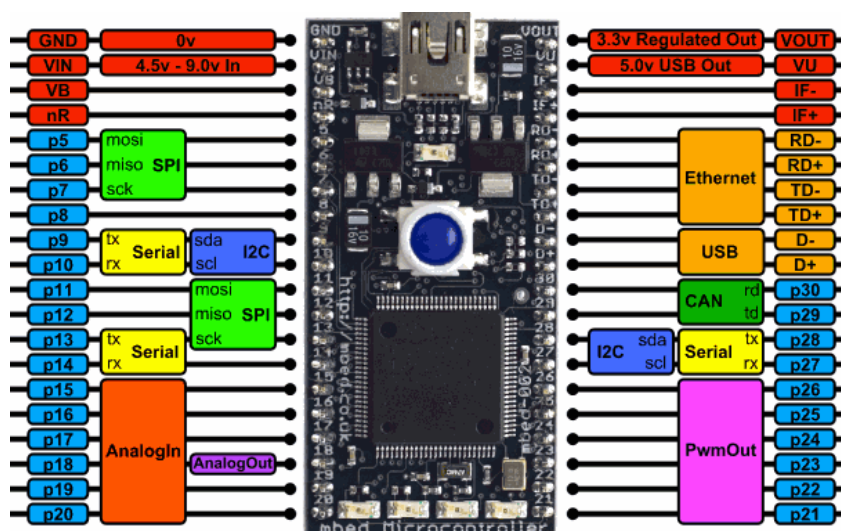
## 3 PLATFORMA

Důležitými požadavky pro použitý hardware jsou nejen výkon, ale i spotřeba energie, spolehlivost a v neposlední řadě cena. Jelikož nároky na výpočetní úkony nejsou příliš vysoké a robot je mobilní, je vhodné zvolit kompromis mezi výkonem, rozměry a spotřebou energie. Z toho důvodu je na místě se zaměřit spíše na jednodeskové počítače a vývojové desky. Na trhu se vyskytuje obrovské množství možností. Na následujících stránkách je stručná charakteristika několika z nich.

### 3.1 mbed

Platforma mbed vyvinutá společností ARM je primárně určena pro rozvíjející se obor Internet věcí. Jedná se o kolekci hardwaru a softwaru, která zajišťuje kompletní potřeby pro vývoj a aplikaci. O výkon se starají 32bitové Cortex-M procesory postavené dle RISC architektury. Na tuto platformu se soustředí i jiní výrobci a proto byl zaveden standard *mbed enabled*, který zaručuje, že hardware je plně kompatibilní.

Vývojové desky mbed, jejichž zástupce je zobrazen na obr. 1, nabízí standardní paletu vstupně-výstupních pinů, A/D a D/A převodníky, piny pro komunikaci přes sběrnice jako jsou například SPI a I<sup>2</sup>C, piny umožňující PWM modulaci a další. Některé typy desek navíc disponují dalšími konektory a obvody jako například Ethernetové rozhraní, Wi-Fi či Bluetooth modul [2].



Obr. 1: Vývojová deska mbed NPX LPC1768 s označenými vývody [3]

Výběr ze široké škály mbed enabled vývojových desek včetně jejich základních charakteristik je umístěn v tab. 1 [4].

|                 |   |
|-----------------|---|
| Nordic nRF51822 | Cortex-M0, 16 MHz<br>128 KB Flash, 16 KB RAM    |
| FRDM-KL25Z      | Cortex-M0+, 48 MHz<br>128 KB Flash, 16 KB RAM   |
| NUCLEO-F303K8   | Cortex-M4, 72 MHz<br>64 KB Flash, 16 KB SRAM    |
| mbed LPC1768    | Cortex-M3, 96 MHz<br>512 KB Flash, 32 KB RAM    |
| NUCLEO-F446RE   | Cortex-M4, 180 MHz<br>512 KB Flash, 128 KB SRAM |

Tab. 1: Výběr modelů mbed enabled vývojových desek

Vývoj samotného programu probíhá pomocí programovacího jazyka C/C++. Hlavní výhodou platformy mbed je základní knihovna mbed.h, která obsahuje mimo jiné nástroje pro práci s periferiemi. Definuje třídy, které jsou uvedeny v tab. 2 [5].

|            |                |                  |               |             |
|------------|----------------|------------------|---------------|-------------|
| AnalogIn   | Callback       | FileSystemLike   | LowPowerTimer | SerialBase  |
| AnalogOut  | CircularBuffer | I2CSlave         | PlatformMutex | Serial      |
| BusInOut   | DigitalInOut   | I2C              | PortInOut     | Stream      |
| BusIn      | DigitalIn      | InterruptIn      | PortIn        | Ticker      |
| BusOut     | DigitalOut     | InterruptManager | PortOut       | Timeout     |
| CANMessage | DirHandle      | LocalFileSystem  | RawSerial     | TimerEvent  |
| CAN        | Ethernet       | LowPowerTicker   | SPISlave      | Timer       |
| CThunk     | FileHandle     | LowPowerTimeout  | SPI           | Transaction |

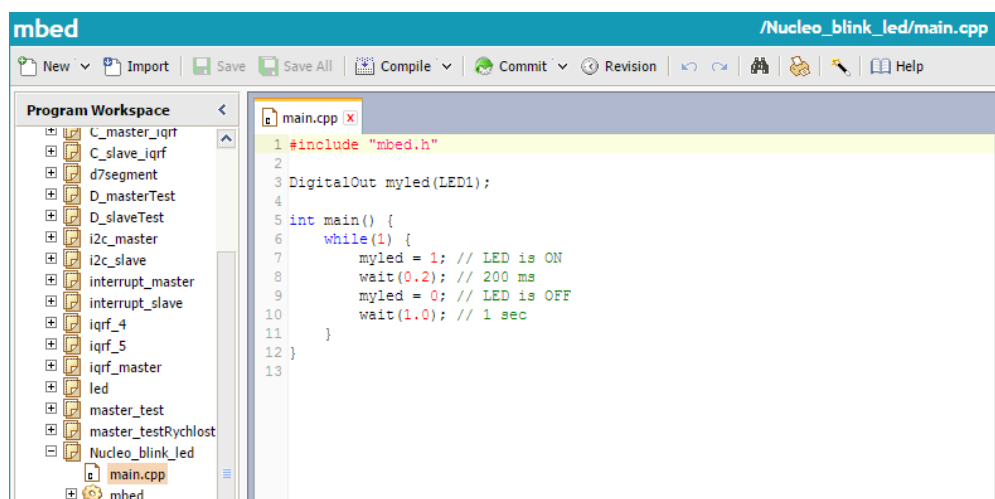
Tab. 2: Seznam tříd knihovny mbed.h

Primárně je pro vývoj aplikací pro platformu mbed určen online editor na stránkách výrobce ([5]). Zde je uživateli po registraci vytvořen profil a přidělen prostor na vzdáleném úložišti. Psaní, editace i zálohování probíhá v okně prohlížeče. Kompilace je rovněž prováděna na straně serveru a uživateli je následně umožněno stáhnout již hotový binární kód. Vývojová deska se po připojení k počítači nahlásí jako záznamové zařízení. Na toto zařízení je poté možné nahrát zmíněný binární soubor a vestavěný programátor se postará o nahrání programu do paměti mikrokontroléru [6].

Použití online prostředí (obr. 2) má své výhody i nevýhody. Značnou výhodou je možnost přístupu k projektu téměř odkudkoli a jakéhokoli počítače, jelikož není potřeba stahovat a instalovat do počítače žádný software, textový editor ani kompilátor. Vše je řešeno na straně serveru. Další výhodou je možnost publikování svého zdrojového kódu přímo na zmíněných stránkách. Této možnosti je hojně využíváno ostatními uživateli, a jelikož jsou

kódy zveřejňovány zpravidla jako open-source, je možné je dále použít či se jimi inspirovat. Rozsáhlá komunita značně zjednodušuje vývoj, jelikož řešení i diskuze daných problémů jsou soustředěny na jednom místě. Uživatel proto nemusí řešení hledat na několika místech.

Na druhou stranu je zde nutnost připojení k internetu. Dále není možné zařízení ladit (debug). Pro náhled do běžícího programu je potřeba využít jiné způsoby (rozsvícení LED při určité události, posílání údajů po komunikační sběrnici aj.). Pro vývoj je možné rovněž použít IDE třetích stran, jako jsou Keil uVision4 či LPCXpresso, které kromě zmíněných funkcí disponují i možnostmi ladění programu [2, 6].



Obr. 2: Online vývojové prostředí mbed

## 3.2 Arduino

Arduino (obr. 3) je open-source platforma, která vznikla na počátku 21. století v Itálii. Původní záměr byl vyvinout levnou technologii pro studenty technologických oborů, ale díky své univerzálnosti se rozšířila velmi rychle i mimo své původní zamýšlené pole působnosti. Jedná se o jednodeskové počítače s procesory ATmega od firmy Atmel. Dokumentace je volně dostupná, je tedy možné si Arduino postavit svépomocí [7].



Obr. 3: Arduino UNO [8]

Arduino disponuje digitálními i analogovými vstupy a výstupy, dále A/D a D/A převodníky a výstupy umožňující PWM modulaci. Pomocí těchto pinů dokáže pracovat s velkým množstvím součástek a přístrojů od LED přes gyroskopy po motorové pohony. Pro Arduino je vyvinuto mnoho tzv. shieldů, rozšiřujících modulů, které se na desku nasadí a jsou určeny například pro komunikaci přes Ethernet, Bluetooth, ovládání zobrazovacích zařízení či krokových motorů. Ve spojení s hotovými knihovnami, které jsou rovněž volně ke stažení, je práce s platformou Arduino velmi jednoduchá.

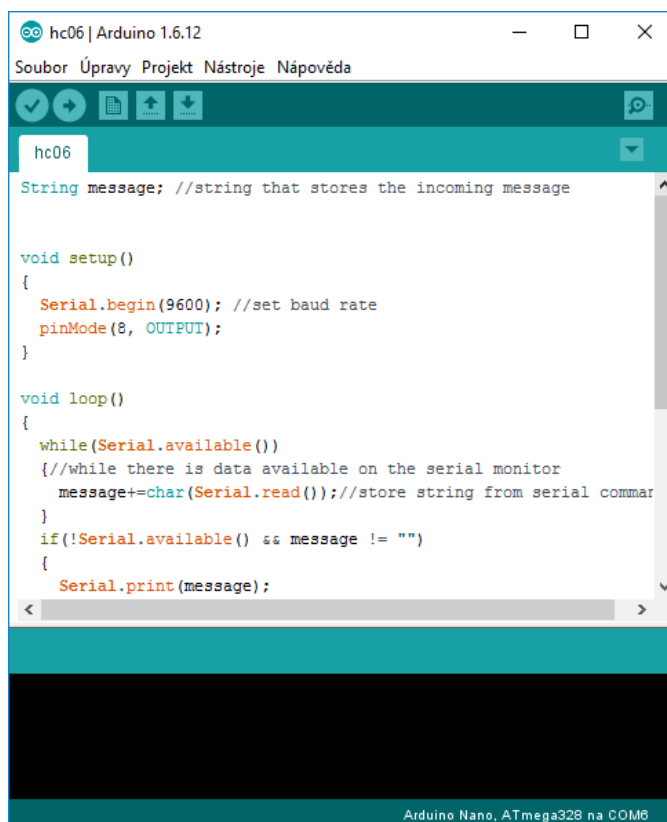
Díky faktu, že Arduino je open-source platforma, se na trhu pohybuje mnoho klonů: Seeeduno, Pinguino, Freeduino, aj. Ve většině případů jsou ve všech aspektech stejné nebo velmi podobné. Není ovšem zaručena kompatibilita se všemi periferiemi jako jsou například již zmíněné shieldy. Výběr ze základních modelů Arduino vývojových desek je v tab. 3 [9].

|                   |  |
|-------------------|--|
| Arduino Nano      | ATmega328, 16 MHz<br>32 KB Flash, 2 KB SRAM<br>22 digitálních, 8 analogových I/O pinů  |
| Arduino UNO       | ATmega328P, 16 MHz<br>32 KB Flash, 2 KB SRAM<br>14 digitálních I/O pinů, 6 analogových vstupů  |
| Arduino MEGA 2560 | ATmega2560, 16 MHz<br>256 KB Flash, 8 KB SRAM  |
| Arduino Yún       | ATmega328, 16 MHz<br>32 KB Flash, 2 KB SRAM<br>14 digitálních I/O pinů, 6 analogových vstupů<br>integrováný Ethernet modul                         |
| Arduino Robot     | ATmega32U4, 16 MHz<br>32 KB Flash, 2.5 KB SRAM<br>5 digitálních I/O pinů, z nichž lze 4 použít jako analogové vstupy                               |
| Arduino Esplora   | ATmega32U4, 16 MHz<br>32 KB Flash, 2.5 KB SRAM<br>integrováný joystick a gamepad   |
| Arduino Gemma     | ATTiny85, 8 MHz<br>8 KB Flash, 512 B SRAM<br>3 digitální I/O piny, 1 analogový vstup<br>miniaturní rozměry (2,8 cm) cílí na tzv. weareble aplikace |

Tab. 3: Výběr základních Arduino vývojových desek

Programování probíhá pomocí speciálního jazyka (Arduino language) odvozeného z jazyka Wiring. Je možné psát i v jiném jazyce (například C, C++) s pomocí externího kompilátoru [7]. Pro vývoj je určeno Arduino IDE (obr. 4) volně stažitelné ze stránek výrobce ([10]). Jedná se o plnohodnotné programovací prostředí, které umožňuje, kromě vytváření a editace programů, i následnou kompilaci a nahrání do paměti mikrokontroléru. Dále umožňuje použití sériového monitoru, který je vhodným nástrojem pro testování a ladění kódu. K vývoji je

možné použít i jiné varianty. Například pro Visual Studio od firmy Microsoft bylo vydáno rozšíření, které má všechny zmíněné vlastnosti Arduino IDE a navíc disponuje možností přímého ladění programu. Mezi novinky patří online vývojové prostředí na stránkách výrobce ([10]).



Obr. 4: Arduino IDE

### 3.3 Single Board Computer

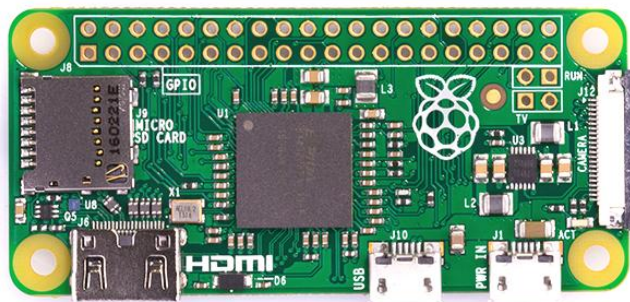
Jednodeskový počítač (SBC, Single-Board Computer) je výpočetní přístroj umístěný na jediné desce plošných spojů (DPS). Jelikož je s postupným technologickým vývojem umožněno vyrábět stále kompaktnější součástky a navyšovat hustotu plošných spojů, je možné osadit veškeré komponenty potřebné k sestavení osobního počítače na jednu DPS. V současné době je vyráběno mnoho variant jednodeskových počítačů, které se navzájem mohou lišit výpočetním výkonem, rozměry či účelem použití [11].

Jedním ze zástupců SBC je Raspberry Pi, což označuje malý jednodeskový počítač vyvinutý firmou Raspberry Pi Foundation za účelem podpoření výuky techniky ve školách. Stal se však velmi rozšířeným i mimo původní oblast díky své univerzálnosti. V současné době jsou na trhu modely Raspberry Pi 1, 2, 3 a Raspberry Pi Zero, z nichž některé se dělí ještě na více verzí. Oproti platformám Arduino či mbed se liší hlavně tím, že na něm běží operační systém Linux speciálně upravený právě pro tento účel. Na novějších modelech je možné použít i distribuci Windows 10 IoT Core. Raspberry disponuje GPIO (General-Purpose Input/Output) piny, jež slouží k nízkoúrovňové komunikaci s okolním hardwarem. Je možné



je použit k ovládání periferií a jiných součástek nebo komunikaci po sběrnicích jako například I<sup>2</sup>C, SPI, UART [12].

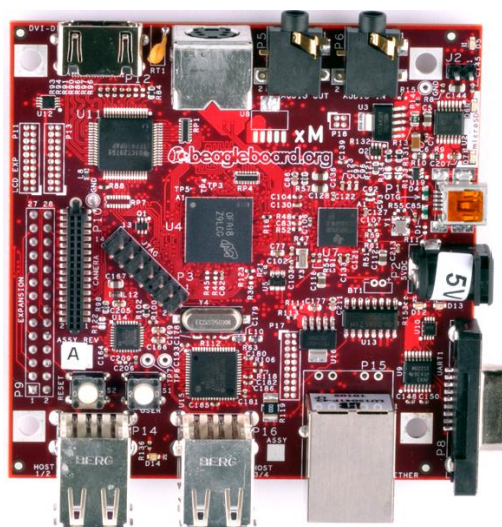
Výpočetní výkon Raspberry Pi je srovnatelný se slabším stolním počítačem. Nejlevnější model Raspberry Pi Zero (obr. 5) disponuje procesorem Broadcom BCM2835 taktovaným na 1 GHz, 512 MB operační paměti, dedikovaným grafickým čipem, micro HDMI slotem pro připojení monitoru, USB portem pro připojení například polohovacího zařízení či klávesnice a mnoha dalšími obvody a součástkami. I přes své kapesní rozměry je na něm možné konat běžnou kancelářskou práci, ale i vyvíjet složitější projekty [13].



Obr. 5: Raspberry Pi Zero [14]

Kromě kancelářské práce je možností použití mnoho. Pomocí Raspberry Pi je možné si doma vytvořit multimediální centrum. Po připojení externího úložiště a nainstalování potřebného softwaru se může Raspberry proměnit v knihovnu filmů, hudby či fotografií. Existují rovněž projekty na vytvoření jednoduchého embedded serveru. Dále je možné si vytvořit z tohoto jednodeskového počítače Wi-Fi router. Díky GPIO pinům se také hodí k měření, automatizaci či řízení [12].

Druhým zástupcem je jednodeskový počítač BeagleBoard/BeagleBone (obr. 6) od firmy Texas Instruments ve spolupráci s firmou Digi-Key. Jedná se o open-source platformu hodící se pro mnoho aplikací od kancelářských úkonů přes domácí automatizaci po složitější průmyslové použití. Jako výpočetní jednotka jsou použity ARM procesory [15].



Obr. 6: BeagleBoard-xM [15]



## 4 KOMUNIKAČNÍ TECHNOLOGIE

Vzhledem k dané úloze je zřejmé, že kromě popsané sekvence pro určení vzdálenosti mezi robotem a lokalizační věží je nutná komunikace mezi jednotlivými prvky navzájem. Je vhodné, aby tato výměna informací byla rychlá a robustní. Dalšími vlastnostmi, které je potřeba brát v úvahu, jsou spolehlivost, náročnost na výpočetní výkon a energetické nároky.

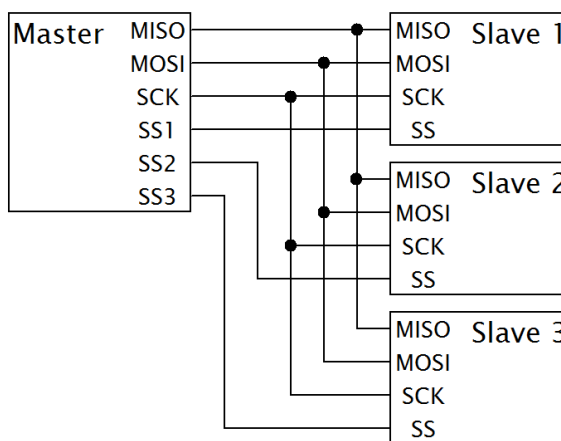
Ke komunikaci mezi věžemi a roboty lze využít mnoho existujících technologií, případně i jejich kombinací. Jedno z možných dělení je na technologie drátové a bezdrátové. Na následujících stránkách jsou některé varianty představeny.

### 4.1 Komunikační sběrnice

Komunikační sběrnice jsou rozdělovány na sériové a paralelní. Jiné kritérium je dělení na spojení typu poloviční a plný duplex (half/full duplex) dle toho, zda je umožněn simultánní obousměrný přenos. Zástupců je nepřeberné množství a tato kapitola se věnuje několika vybraným.

#### 4.1.1 SPI

SPI (Serial Peripheral Interface) je zástupce synchronních komunikačních sběrnic. Je určen pro full duplex přenos informací mezi dvěma a více zařízeními v režimu master-slave. Synchronní přenos je zajištěn pomocí hodinového signálu, který je generován master prvkem. Propojení je realizováno pomocí čtyř vodičů: MISO (Master In Slave Out), MOSI (Master Out Slave In), SCK (Serial Clock) a SS (Slave Select). Příklad propojení několika zařízení je znázorněn na obr. 7.



Obr. 7: Příklad propojení více zařízení pomocí sběrnice SPI

Slave prvky mohou komunikovat pouze s nadřazeným členem, který iniciuje a řídí přenos a generuje hodinový signál. Pro přenos vlastních dat jsou určeny vodiče MOSI a MISO. Hlavní prvek vysílá data po vodiči MOSI a podřízená zařízení je na tomto vodiči přijímají. Naopak pro komunikaci, při níž jsou přenášena data ze slave do master zařízení, je používán vodič MISO. Vodič SS slouží k výběru konkrétního podřízeného prvku masterem. Tento

vodič musí být samostatný pro každé další připojené zařízení, a proto použití sběrnice s sebou nese nutnost dostatku volných vývodů u master zařízení. Další možností je propojení všech slave zařízení, které pro komunikaci používají posuvný registr, za sebe tak, že je vodič MISO připojen vždy na vývod MOSI následujícího členu. Přenášená data jsou poté rozdělena do segmentů dle počtu slave zařízení, která pro čtení či zápis využívají pouze jim přidělenou oblast zprávy [16].

#### 4.1.2 I<sup>2</sup>C

Sběrnice I<sup>2</sup>C (IIC, Internal-Integrated-Circuit Bus) zastupuje dvouvodičovou, synchronní, half duplex sběrnici. Prvky jsou zapojovány dle schématu master-slave. Propojení se provádí pomocí dvou vodičů: SCL (Serial Clock) slouží k přenosu hodinového signálu masterem pro časovou synchronizaci a SDA (Serial Data) je určený pro přenos dat. Klidový stav je definován jako logická jednička na obou vodičích, a proto musí být na oba připojeny pull-up rezistory [16]. V základní formě jsou slave zařízení adresována 7bitově, v rozšířené 10bitově, což umožňuje propojení až 128/1024 zařízení na jednu sběrnici. V praxi je však toto číslo menší, jelikož je omezeno parazitní kapacitou linek a zařízení a některé adresy jsou vyhrazeny pro jiné účely (například vysílání všem připojeným zařízením - broadcast).

Sběrnice SPI podporuje režim multimaster. To znamená, že každý člen sítě může na určitý časový úsek převzít úlohu řídicího prvku. Implementována je rovněž detekce kolize [17].

#### 4.1.3 UART

Sériová sběrnice UART (Universal Asynchronous Receiver and Transmitter) využívá asynchronní přenos informací, což spočívá v absenci synchronizačního signálu přenášeného spolu s daty. Pro úspěšnou komunikaci a správnou interpretaci zprávy je nutné, aby vysílací i přijímací zařízení znala rychlost přenosu. K detekci jednoduchých chyb může být použit paritní bit vyjadřující sudou či lichou paritu. Pro propojení dvou zařízení jsou používány pouze dva vodiče. Jeden z prvků je svým vývodem Tx, který je používán pro posílání dat, připojen na vývod Rx, který slouží pro příjem dat, druhého prvku a naopak [18]. Jedná se tedy o full duplex sběrnici, ze které vychází další sběrnice, jako je USART (Universal Synchronous/Asynchronous Receiver Transmitter). Ta umožňuje jak asynchronní full duplex, tak i synchronní half duplex přenos. Dále například sběrnice RS-232 a RS-422 [19].

#### 4.1.4 CAN

Mezi sériovou komunikační technologií se řadí sběrnice CAN (Controller Area Network). Jedná se o sběrnici podporovanou mnohými výrobci, kteří do svých elektronických zařízení implementují CAN protokol. Původním polem působnosti byl rozvod v automobilovém průmyslu, tedy na krátké vzdálenosti. Norma CAN obsahuje dvě varianty: CAN 2.0A a CAN 2.0B, která disponuje možností většího objemu přenášené zprávy.

Přenosová rychlost dosahuje hodnoty až 1 Mb/s (délka vodičů do 40 m, hodnota klesá s narůstající vzdáleností). Prvky propojené do sítě pomocí sběrnice CAN jsou uspořádány dle schématu multimaster. Teoreticky je počet propojených prvků neomezený, ale z důvodu

zatížení sběrnice je uváděna hodnota 64 účastníků na segment. Komunikace je realizována pomocí zpráv, které neobsahují adresu příjemce a jsou přijímány všemi připojenými prvky. Přenášena data začínají identifikátorem typu zprávy a hodnotou značící její prioritu. Kolize je řešena na základě prioritního rozhodování. Protokol CAN zajišťuje v případě kolize přenos zprávy s nejvyšší prioritou. Dále umožňuje definovat, jaké typy zpráv bude dané zařízení přijímat (acceptance filtering) [20].

#### **4.1.5 MODBUS**

Technologie MODBUS je v dnešní době v automatizaci široce používána. Umožňuje vytvářet komunikační spojení typu master-slave. Protokol MODBUS definuje pouze strukturu zpráv a je nezávislý na použitém médiu. Jedná se o jednoduchou a otevřenou technologii, jejíž definice je volně dostupná na internetu [20]. Výhodou je také jeho flexibilita a jednoduchá implementace. Celá řada zařízení jako jsou PLC, mikropočítače, ale i senzory používané v průmyslové automatizaci, jsou vybavena komunikačním rozhraním pro MODBUS [21]. Nevýhodou sběrnice je její zastaralost. Přestože byla vyvinuta roku 1978, do dnešní doby nebyla téměř změněna.

Původně byl zaměřena na přenos pomocí sériové linky typu RS-232, později však došlo k rozšíření i na typy RS-422 a RS-485. Nejnovější specifikace MODBUS TCP umožňuje využití prostředků protokolu TCP/IP tak, že je MODBUS zpráva vložena do TCP paketů. Tato možnost rozšiřuje pole působnosti například na Ethernetové rozhraní [20].

### **4.2 Bezdrátové komunikační technologie**

Bezdrátový přenos informací se rozvíjí zejména s pokrokem mobilních zařízení, ale je používán i pro účely automatizace. O pokroku svědčí standardizace například WLAN (Wireless Local Area Network) nebo Bluetooth. Tato kapitola se věnuje několika málo vybraným představitelům z rozsáhlých řad bezdrátových komunikačních technologií [20].

#### **4.2.1 IQRF**

Platforma IQRF byla představena firmou MICRORISC v roce 2004. Jedná se o bezdrátovou komunikační technologii, která se vyznačuje především nízkou spotřebou energie pro komunikaci nevyžadující vysokou rychlost ani objem přenášovaných dat. Umožňuje propojení zařízení na vzdálenost desítek až stovek metrů (dle prostředí). Tato technologie míří zejména na rozvíjející se obor Internet věcí [22]. IQRF Aliance vznikla za účelem vytvoření komplexního souboru hardwaru a softwaru využívající technologii IQRF. Jejími členy jsou firmy jako například MICRORISC, IQ Home, Protronix a další [23].

Vysílání probíhá v bezlicenčních ISM (Industrial, Scientific and Medical) pásmech 433 MHz, 866 MHz nebo 916 MHz [22]. ISM pásmo je určeno k bezplatnému volnému použití za předpokladu dodržení podmínek pro vyzářený výkon a technické parametry vysílače i přijímače [20]. Významnou vlastností IQRF modulů jsou velmi kompaktní rozměry dosahující velikosti klasické SIM karty. Moduly jsou osazeny mikrokontroléry s vlastním operačním systémem. Výrobce uvádí, že hodnota odběru energie se může pohybovat

za nečinnosti pod 1  $\mu\text{A}$  a za příjmu okolo 15  $\mu\text{A}$ , zatímco přenosová rychlost dosahuje zhruba 20 kb/s. Přenos dat je uskutečněn pomocí až 64Bytových packetů. Doba přenosu mezi dvěma zařízeními je uváděna okolo 40 ms.

Zařízení komunikující pomocí IQRF technologie je možné propojit několika způsoby. Základní topologie je typu Mesh, kde není kladeno žádné omezení. Další možností je využití sítě IQMESH, která byla vyvinuta speciálně pro platformu IQRF a výrazně zvyšuje spolehlivost a možné rozměry sítě. Do IQMESH sítě může být naráz připojeno až 240 zařízení, přičemž každý člen může pracovat jako koordinátor nebo jako běžný uzel. Uzly mohou navíc pracovat jako pomocný koordinátor a předávat zprávy jiným uzlům za účelem zvýšení maximálních rozměrů sítě. Každý člen může pracovat až ve dvou nezávislých sítích. Podpora IQMESH operačním systémem modulů obsahuje mnoho pokročilých funkcí, které zahrnují hned několik routovacích algoritmů, proceduru nastavení sítě či automatickou tvorbu spojení.

IQRF moduly disponují tzv. data flow režimem, který umožňuje přenos informací bez uživatelského kódu pomocí definovaných událostí. K tomu slouží DPA (Direct Peripheral Access) protokol a výrobcem předpřipravené HWP (Hardware Profile). Výrobce uvádí, že s jejich pomocí je implementace bezdrátové sítě značně usnadněna. Navíc je možné rozšířit funkcionalitu pomocí Custom DPA Handleru. Druhou možností je nahrání uživatelského programu, který využívá předdefinované funkce operačního systému IQRF modulu [22].

#### 4.2.2 Bluetooth

Bluetooth standard označuje bezdrátovou komunikační technologii sloužící pro přenos dat na krátké vzdálenosti. Definována je standardem IEEE (Institute of Electrical and Electronics Engineers) 802.15.1 a mezi propojenými zařízeními vytváří síť typu WPAN (Wireless Personal Area Network) [24]. Vysílání probíhá v ISM pásmu na frekvenci 2,4 GHz za použití techniky přeskakující radiové frekvence (frequency hopping), čímž je zajištěn bezpečný a rychlý přenos dat [20]. Jsou definovány dvě různé linky pro přenos informací. Jedná se o linku asynchronní (ACL, Asynchronous Connectionless) a synchronní (SCO, Synchronous Connection Oriented).

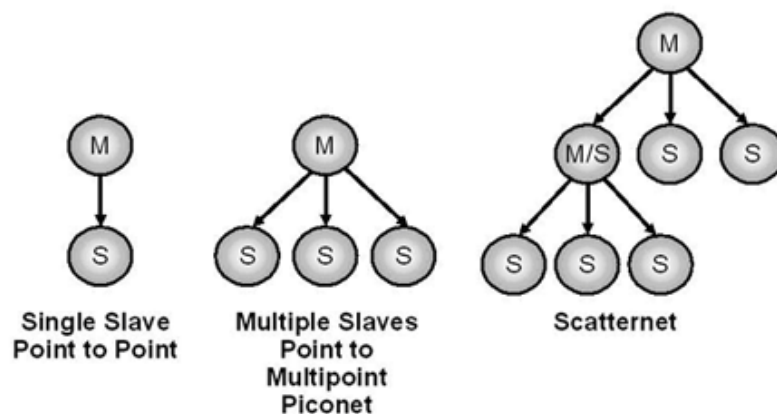
V roce 1998 byla založena organizace Bluetooth SIG (Special Interest Group) firmami Ericsson, Intel, IBM, Nokia a Toshiba a dnes má více než 13000 členů. Toto uskupení se stará o zajištění kompatibility napříč co nejvíce produkty a o vývoj nových verzí.

V současné době je vyvinuto hned několik verzí počínaje verzí 1.0 a konče verzí 5.0. Mezi sebou se liší přenosovou rychlostí, ale i použitými algoritmy. Například verze 2.0 přinesla novinku v podobě technologie EDR (Enhanced Data Rate), která umožňuje několikanásobně zvýšit rychlost přenosu, která se pohybuje u verze 1.2 do 1 Mb/s, u verze 3.0 až do 24 Mb/s. Další novinkou je algoritmus SSP (Secure Simple Pairing) zavedený od verze 2.1, který zrychluje párování mezi zařízeními. Od verze 4.0 je dostupná technologie Bluetooth low energy (BLE, Bluetooth LE), která značně snižuje spotřebu elektrické energie.

Zařízení Bluetooth je možné podle výstupního výkonu dělit do tří tříd. Class 1 je definována pro zařízení o výstupním výkonu do 100 mW, přičemž se maximální dosah pohybuje do 100 m v otevřeném prostoru. Class 2 zahrnuje přístroje o výkonu do 10 mW

a dosahuje až do vzdálenosti 10 m. Class 3 je určena pro prvky s výstupním výkonem do 1 mW a vzdálenost pro přenos menší než 1 m.

Standard umožňuje mezi zařízeními komunikujícími pomocí technologie Bluetooth vytvořit spojení typu point-to-point i point-to-multipoint. Pro řízení komunikace je vždy jeden prvek nadřazený ostatním a určuje průběh komunikace: master-slave model. S jedním masterem může komunikovat až 7 prvků a toto uskupení tvoří tzv. Piconet buňky. Buňky se mohou sdružovat a vytvořit tak větší síť zvanou Scatternet [25]. Na obr. 8 jsou zobrazeny jednotlivé možnosti.



Obr. 8: Konfigurace sítí Bluetooth [26]

### 4.2.3 ZigBee

ZigBee je bezdrátový komunikační standard, jehož primární oblastí použití je sběr dat ze snímačů v síti typu WPAN. Jedná se o technologii umožňující přenos informací na vzdálenost do 70 metrů při nízkých přenosových rychlostech (do 250 kb/s) a slibuje velmi nízkou energetickou náročnost. Pracuje na bezlicenčních frekvencích 868 MHz a 915 MHz dle kontinentu (Evropa, Severní Amerika) a celosvětově pracuje v pásmu 2,4 GHz. ZigBee technologie dosáhla roku 2003 certifikace IEEE jako norma 802.15.4 [20]. O vývoj ZigBee se stará sdružení ZigBee aliance s asi 50 členy se značným zastoupením výrobců automatizační techniky (NXP, Texas Instruments, ARM, atp. [27]), a proto je tato technologie udržována tak, aby co nejvíc vyhovovala požadavkům automatizačních aplikací.

Cílové aplikace technologie ZigBee jsou podle ZigBee aliance a sdružení IEEE předpokládány pro domácí automatizaci (bezdrátové zabezpečovací prvky, kontrola termostatů a osvětlení, univerzální ovladače, a další), průmyslové sítě (průmyslová automatizace a automatizace budov), počítačové periferie (bezdrátové klávesnice, myši, aj.) a automobilové sítě.

Dle konkrétní aplikace je možné vybrat mezi třemi definovanými režimy přenosu informací: periodicky se opakující přenosy (např. data ze snímačů), nepravidelné přenosy (např. uživatelem vyvolané události) a opakující se přenosy s důrazem na malé zpoždění (např. dálkové ovládání) [20].

Struktura ZigBee sítě se skládá z několika typů prvků: koordinátor, router a koncové zařízení. Koordinátor je centrálním prvkem a stará se o založení, nastavení, spuštění



a ochranu sítě. Router umožňuje rozšíření sítě na větší vzdálenosti či připojení k internetu. Koncová zařízení mohou být vlastní senzory a přístroje. Mnozí výrobci vyrábí zařízení, které zvládá víc funkcí najednou, jako například senzor, který zároveň přeposílá zprávy jiným prvkům [28]. Běžně je používána hvězdicová topologie sítě s centrálním řídicím prvkem. Je však možné využít možnosti peer-to-peer přenosu mezi zařízeními, kterých může být aktivních až 255 v rámci jednoho koordinátora. Maximální počet členů v rámci jedné sítě je uváděn až 4000.

Minimální spotřeba celého komunikačního systému je dosažena tím, že pouze centrální prvek musí být neustále na příjmu, zatímco ostatní zařízení mohou být po dlouhý časový úsek v režimu nízké spotřeby. Cílem je umožnit pracovat některým zařízením i několik let bez výměny baterie [20].

#### 4.2.4 Wi-Fi

Technologie Wi-Fi (Wireless Fidelity) je zástupce bezdrátové komunikační technologie vytvářející síť typu WLAN. Spadá do specifikace IEEE 802.11 a nejrozšířenější verze jsou v současné době 802.11a, b, g a n. Zařízení splňující požadavky stanovené standardem jsou označovány Wi-Fi Certified a zaručují kompatibilitu i se zařízeními jiných výrobců. Tato technologie dosahuje poměrně velkého dosahu a objemu přenášených dat oproti výše zmíněným technologiím.

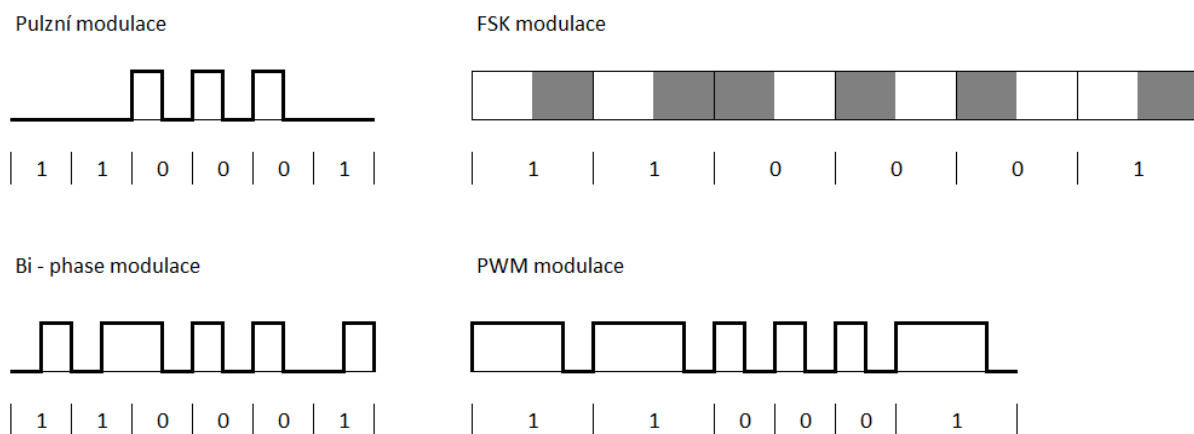
Standard 802.11a je z vyjmenovaných nejstarší a pracuje na frekvencích 5 GHz až 6 GHz. Nabízí maximální přenosovou rychlost (dle použité modulace a kódování) až 54 Mb/s za cenu vyšší spotřeby energie a kratšího dosahu. Standard 802.11b vysílá v pásmu 2,4 GHz až 2,497 GHz. Dosahuje nižších přenosových rychlostí (až 5,5 Mb/s) v porovnání se svým předchůdcem, ale disponuje nižší spotřebou a vyšším dosahem. Nový standard 802.11g využívá pro vysílání frekvenční pásmo 2,4 GHz až 2,497 GHz, spojuje nejlepší vlastnosti svých předchůdců (802.11a a 802.11b) a zároveň s nimi zachovává zpětnou kompatibilitu. Maximální přenosová rychlost je uváděna až 54 Mb/s a zároveň je udržena nízká spotřeba a velký dosah [20]. Standard 802.11n pracuje v pásmech 2,4 GHz a 5 GHz a umožňuje přenášet data rychlostí reálně až 130 Mb/s (teoreticky až 600 Mb/s). Je zpětně kompatibilní s verzemi a, b i g. Oproti předchozím verzím, u nichž byla použita technologie SISO (Single Input – Single Output), byla pro zvýšení přenosové rychlosti u standardu 802.11n zavedena technologie MIMO (Multiple Input – Multiple Output). Princip technologie spočívá v zavedení více antén u vysílače a přijímače. Mezi těmito anténami je poté možné vysílat více datových toků naráz [29].

#### 4.2.5 IrDA

Specifikace IrDA (Infrared Data Association) zastupuje technologii bezdrátového přenosu dat pomocí záření v infračervené oblasti spektra. O standardizaci se stará IrDA konsorcium definující jak hardwarovou podobu vysílačů a přijímačů, tak i používaný komunikační protokol. Vysílání signálu o vlnové délce okolo 875 nm mají na starost infračervené LED. Pro příjem se používají PIN fotodiody s filtrací, která propouští pouze požadované frekvence

[30]. V běžném životě se s touto technologií můžeme setkat například u dálkových televizních ovladačů nebo některých typů mobilních zařízení.

Data jsou přenášena v digitální podobě sériově a existuje více druhů používaných modulací signálu. Znázorněné jsou na obr. 9. První možností je *pulzní modulace*, při níž je každému přenášenému bitu přidělen stejně velký časový úsek. Konstantní hodnota signálu je vyhodnocena jako logická jednička, pokud se v bloku objeví pulz, je tento stav vyhodnocen jako log. nula. Tyto pulzy mají definovanou konstantní délku. *Modulace FSK* (Frequency Shift Keying) spočívá v přenosu log. jedniček a nul pomocí dvou různých frekvencí. Při použití *Bi-phase modulace* jsou, stejně jako v případě použití pulzní modulace, časové bloky pro přenos jednotlivých bitů stejné. Pro určení, zda se jedná o log. jedničku či nulu, se používá změna úrovně signálu uvnitř daného časového úseku. Kladná změna zastupuje log. jedničku a změna záporná log. nulu. Pulzně šířková modulace (PWM, Pulse Width Modulation) spočívá v rozdělení přenášených informací na různě velké časové úseky. Delší bloky jsou vyhodnoceny jako log. jednička a kratší jako log. nula [31].



Obr. 9: Typy modulací IrDA signálu





## 5 REALIZACE

Určení polohy robota je realizováno pomocí sekvence, při níž lokalizační věž vyšle dva druhy signálu s rozdílnou rychlostí šíření v prostoru. Nejprve je vyzařován světelný paprsek a v okamžiku ukončení jsou vyslány ultrazvukové vlny. Díky znalosti rychlosti šíření obou signálů prostředím je vzdálenost od daného majáku spočítána pomocí časového intervalu mezi jejich přijetím. Tento proces je nutné opakovat minimálně třikrát pro jednoznačné určení polohy. Je předpokládáno, že robot zná polohu všech lokalizačních věží a je schopen rozpoznat konkrétní stanici dle vysílací sekvence, která je u všech majáků unikátní. Díky tomu odpadá nutnost další komunikace ohledně identifikace věže. Z použitého principu lokalizace vyplývá nutnost zajistit, aby v daný okamžik vysílala sekvenci maximálně jedna věž. V opačném případě by mohlo dojít k interferenci ultrazvukových vln a chybným výsledkům.

Pro účely aplikace je dále potřeba umožnit sdílení polohy všech robotů dle potřeby. Tyto informace umožňují například rozdělovat úkoly robotům dle jejich aktuální (poslední známé) pozice, plánovat trasu robota či předcházet kolizím. Dalšími doplňujícími avšak důležitými charakteristikami jsou rychlost a spolehlivost komunikační struktury.

Úkolem je tedy navrhnout princip a strukturu komunikace pro daný problém. Tato kapitola se věnuje realizaci zadané aplikace. Nejprve je uveden postupný rozbor situací, který slouží k rozvržení problematiky. Dále jsou představeny použité hardwarové prvky a následuje popis konkrétního zvoleného řešení.

### 5.1 Rozbor

Tato kapitola se zabývá rozbohem ukázkových situací a na závěr je popsáno vybrané schéma komunikace.

#### 5.1.1 Ukázkové situace

Ukázkové situace jsou seřazeny od jednoduchých po složité. Je zde uveden postupný rozbor, z něhož je patrná problematika úlohy.

##### Situace A

Nejjednodušší možností je situace, ve které uvažujeme jen jednoho robota a jednu lokalizační věž. V případě, že je v dosahu pouze jedna lokalizační stanice, je určení polohy rovněž možné za předpokladu, že robot je schopen relativně určit změnu své polohy, tj. disponuje diagnostikou svého pohybu na základě natočení a ujeté vzdálenosti (odometrií).

Pro určení polohy vozidla není nutná komunikace mezi robotem a věží. V tomto případě by věž opakovala sekvenci pro určení polohy neustále. Robotu je poté umožněna lokalizace kdykoliv, bez zpoždění.

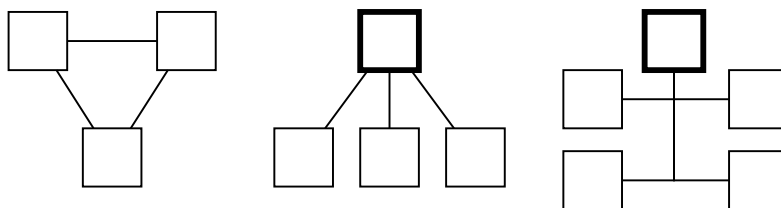
Jiné možnosti mohou být realizovány v závislosti na zavedení komunikace mezi mobilním a stacionárním prvkem. Takové řešení již omezuje vysílání sekvence pouze na případy, kdy je spuštěna lokalizační procedura. V tomto případě je možné sekvenci vysílat

pouze v daném okamžiku na základě žádosti robota, na což je dostatečná jednosměrná komunikace, nebo se věž může periodicky dotazovat robota, zda žádá lokalizační proceduru, což by vyžadovalo přenos obousměrný. Informace o poloze robota může zaznamenávat robot sám nebo lokalizační stanice a podle potřeby tyto údaje sdílet.

### Situace B

Při situaci B uvažujeme jednoho robota a více lokalizačních věží. V tomto případě rovněž není pro určení polohy nutná výměna informací mezi prvkem mobilním a majáky. To by znamenalo, že věže, rovněž jako v Situaci A, periodicky vysílají lokalizační sekvenci bez ohledu na požadavek určení polohy.

Avšak zde je již nutné zajistit, aby nevysílalo více věží naráz. Jednou z možností je kontrolovat vysílání věží pomocí časového multiplexu. Každá stanice by měla přidělené časové okno, ve kterém může odvysílat sekvenci. Toto řešení však vyžaduje časovou synchronizaci pro dodržení časových intervalů. Dále je možné propojit věže do kruhové sítě tak, že každá věž bude znát svého následníka i předchůdce. Stanice si poté budou periodicky předávat právo na vyslání lokalizační sekvence (token). Jiná varianta počítá s určením jedné věže jako věže nadřazené, master věže. Stacionární stanice by byly poté propojeny do hvězdicové či sběrníkové topologie. Hlavní stanice by poté řídila ostatní slave věže. Příklady propojení jsou znázorněny na obr. 10.



Obr. 10: Propojení lokalizačních věží: kruhová síť, hvězdicová a sběrníková topologie

V případě realizace komunikace mezi robotem a věžemi se otevírají další možnosti. Robot může například navazovat spojení se všemi věžemi individuálně (peer-to-peer) a vysílání sekvence by bylo realizováno v případě potřeby. Veškeré řízení věží by měl na starost robot. Senzory robota sloužící k lokalizaci však musí být natočeny k věži, od které je měřena vzdálenost, a pravděpodobně by docházelo k velkým časovým prodlevám při hledání konkrétní věže. V případě uspořádání věží dle schématu master-slave, by robot navazoval spojení pouze s hlavní věží a vysílání lokalizační sekvence by probíhalo pouze v případě potřeby. Jinou možností je navázání komunikace s kteroukoliv věží v dosahu, která by na omezenou dobu vystupovala jako master věž a řídila vysílání lokalizačních signálů.

Informace o poloze robota může opět zaznamenávat samotný robot, případně mohou být předávány stacionárním prvkům. V případě uspořádání věží dle schématu master-slave je vhodné, aby tyto údaje shromažďovala hlavní stanice.

### Situace C

Ukázková situace C se skládá z většího počtu robotů a jedné lokalizační věže. Jak již bylo zmíněno, určení polohy robotů je možné i pomocí pouze jedné stanice.

Vysílání sekvence pro určení polohy může být řízeno více způsoby. Stejně jako v situacích A a B je jedním z nich periodické opakování sekvence, přičemž ji roboti využívají dle potřeby, druhou možností je spouštění lokalizační procedury na základě požadavku robota.

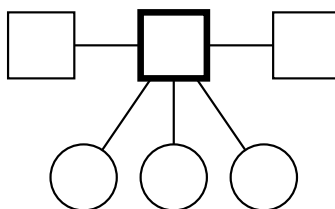
Dále je potřeba zvážit situaci, při které je požadavek na určení polohy více robotů najednou. V případě nepřetržitého vysílání sekvence tento problém odpadá, jelikož více robotů může využít k určení vzdálenosti stejný signál. Za předpokladu, že vysílání sekvence je zahájeno na pokyn od robota, simultánně může určovat svou polohu i více robotů najednou. Sekvence může být jednoduše opakována od prvního požadavku, dokud by všichni aktivní roboti nesignalizovali úspěšné dokončení procesu či chybový stav (například nenalezení žádné věže).

S ohledem na požadavek o sdílení poslední známé pozice robotů musí mít roboti přidělené své unikátní identifikační číslo (ID). Jednou z možností je sdílení těchto informací mezi roboty navzájem a v případě vyžádání předat údaje o poloze i věži. To by ale vedlo k velkému množství přenosu s rostoucím množstvím robotů. Další variantou je předávání polohy robota pouze lokalizační věži, která by shromažďovala informace o všech mobilních prvcích.

### Situace D

Ukázková situace D sestává z více lokalizačních věží a více robotů. Jedná se o kombinaci situací B a C. Řízení věží může být situováno dle situace B, buď pomocí časového multiplexu, periodicky předáváním tokenu v kruhové síti či určením master věže. Při větším počtu prvků je komunikace typu peer-to-peer prakticky nepoužitelná z důvodu enormní časové náročnosti. Vhodnější se zde proto jeví určení master věže.

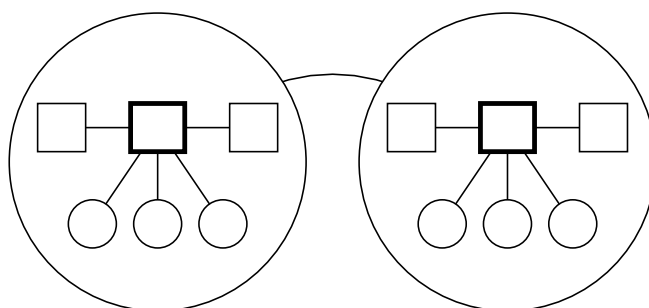
Předání informací o poloze robotů může být realizováno dle Situace C jak mezi roboty tak s věžemi. V případě master-slave struktury je vhodné, aby polohy shromažďovala právě tato nadřazená autorita. Příklad komunikace je znázorněn na obr. 11.



Obr. 11: Příklad komunikace mezi roboty a věžemi

## Situace E

Případným rozšířením problému na více oddělených oblastí, mezi kterými mohou roboti přejíždět, vzniká situace E (obr. 12). V tomto případě mohou oddělené sítě pracovat dle situace D, ale musí být zajištěna výměna informací mezi jednotlivými oblastmi. Příkladem může být situace, kdy je potřeba zjistit polohu určitého robota. Tento požadavek může být vznesen z jakéhokoliv místa, a proto je potřeba vhodná komunikace mezi oblastmi. Propojení všech prvků do jedné sítě může být z fyzikálních důvodů (např. dosah signálu) nereálné. Vyčleněním jedné master věže v každé oblasti, která by měla informace o robotech ve svém okolí, a jejich vzájemným propojením vzniká strukturovaná síť umožňující pokrýt více oblastí. Takto vybudovaná struktura by poté umožňovala snadnou výměnu informací o polohách všech robotů v rámci celé aplikace.



Obr. 12: Schéma situace E

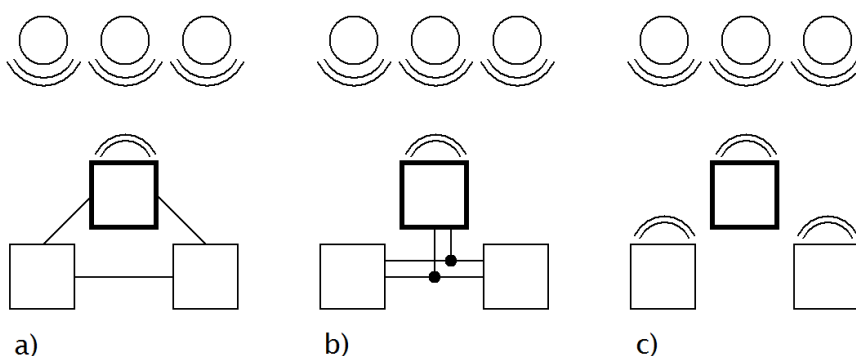
### 5.1.2 Vybrané řešení

Při řešení úlohy této práce byla uvažována situace D, při které je předpokládán větší počet mobilních i stacionárních prvků v rámci jednoho celku. Z věží je vybrána jedna stanice a je označena jako hlavní věž (master) a zbylé stanice vystupují jako slave věže (podřízené master věži). Toto řešení umožňuje pokrýt prostor v dosahu hlavní věže. Při požadavku na rozšíření (situace E) je možné vytvořit další celek s vlastní master věží a výměna informací by poté probíhala mezi těmito hlavními věžemi. Roboti i podřízené věže komunikují pouze s hlavní věží, která zpracovává požadavky robotů a navíc obstarává režii spojenou s řízením slave stanic. To znamená, že zajišťuje, aby v daný okamžik vysílala lokalizační sekvenci maximálně jedna stanice.

Pro komunikaci mezi robotem a hlavní stanicí je vhodné, aby byla realizována pomocí bezdrátové technologie. Mezi věžemi navzájem může být výměna informací realizována pomocí vodičů i bezdrátově. Pro zajištění funkcionality a výměny informací bylo potřeba zvolit vhodný výpočetní výkon a bezdrátovou technologii. Při výběru bylo nutné zohlednit, zda je vhodnější provedení, ve kterém jsou výpočetní jednotka a bezdrátový modul implementovány na jedné desce plošných spojů (DPS), nebo externí modul připojený k hlavní desce s výpočetní jednotkou. První varianta vyniká z důvodu, že bezdrátový modul umístěn přímo na DPS mívá již implementovanou podporu v podobě definovaných funkcí pro práci s ním. Výhoda druhé možnosti spočívá v tom, že externí komunikační modul umožňuje velmi snadnou záměnu za jiný model či dokonce jinou technologii. Proto byla zvolena druhá,

modulární varianta v podobě externího bezdrátového modulu IQRF DCTR-72DAT. Jako výpočetní jednotka byl zvolen mbed ST-Nucleo F303-K8.

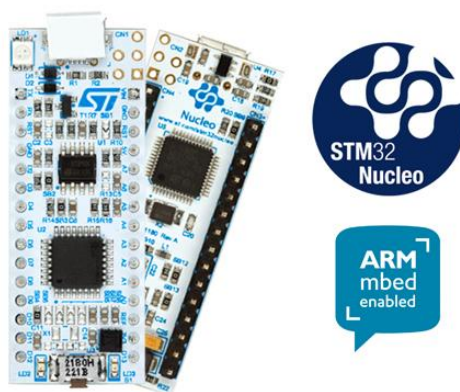
Roboti komunikují pomocí technologie IQRF s master věží. Pro komunikaci mezi hlavní stanicí a podřízenými věžemi jsou realizovány tři možnosti: propojení do logického kruhu, pomocí sběrnice a bezdrátově. První varianta spočívá v propojení všech věží do logického kruhu, ve kterém si vzájemně posílají pouze signál značící ukončení vysílání předcházející stanice. Při výběru sběrnice bylo potřeba si uvědomit, že k jedné master věži může být připojeno mnoho slave věží. Z tohoto důvodu byla zvolena dvou vodičová sběrnice I<sup>2</sup>C. Bezdrátová implementace využívá již zmíněnou technologii IQRF. Realizovány a otestovány jsou tři samostatné projekty, které se liší vzájemným propojením lokalizačních stanic. Na obr. 13 je pro tyto varianty znázorněno propojení jednotlivých prvků: roboti s hlavní věží komunikují pomocí bezdrátové technologie IQRF, věže jsou propojeny do logického kruhu (a), pomocí sběrnice I<sup>2</sup>C (b) a rovněž bezdrátově (c).



Obr. 13: Zjednodušené schéma propojení prvků pro jednotlivé varianty

## 5.2 mbed ST-Nucleo F303-K8

Platforma ST-Nucleo F303-K8 (obr. 14) patří mezi mbed enabled vývojové desky. Svými rozměry a rozložením pinů je označována jako kompatibilní s platformou Arduino. Rozměry činí 50,3 x 18,5mm. Může být napájena přes integrovaný micro-USB konektor nebo z externího zdroje napětím 3,3V, 5V nebo 7 - 12 V [32].



Obr. 14: Vývojová deska ST-NUCLEO F303K8 [32]

Osazený 32bitový mikroprocesor Cortex-M4 je postaven na ARM architektuře a umožňuje maximální taktovací frekvenci 72 MHz. Dále disponuje 64 KB Flash pamětí pro program, čtyřmi oscilátory, jedenácti časovači, 51 I/O piny. Je možné využít až dva A/D převodníky s až 21 kanály a nastavitelnou přesností 12, 10, 8, nebo 6 bitů, nebo až tři 12bitové D/A převodníky. CPU dále disponuje několika režimy provozu. Jedná se o Run, Sleep, Stop a Standby módy. V režimu Run jsou všechny součásti v provozu, Sleep mód zastaví pouze CPU, zatímco periférie zůstávají v provozu a mohou probudit CPU v případě přerušení nebo určité události. Zbylé dva módy dosahují ještě větší úspory, ale vyžadují speciální podmínky pro probuzení [33]. Deska dále podporuje na definovaných pinech sběrnice CAN, SPI, I<sup>2</sup>C (případně SMBus či PMBus) a UART. Po připojení k počítači pomocí USB se hlásí jako Flash paměť a vytváří vlastní COM port. Pro uživatele je dále k dispozici jedna LED [32].

### 5.3 IQRF modul DCTR-72DAT

IQRF modul DCTR-72DAT (obr. 15) spojuje vysílač i přijímač na jedné desce. Modul zahrnuje 8bitový mikrokontrolér PIC16LF1938-I/MV od firmy Microchip, RF (Radio Frequency) obvod SPIRIT1 od firmy STMicroelectronics, EEPROM paměť o velikosti 32 KB, anténu pro bezdrátový přenos signálu, snímač teploty, atd.

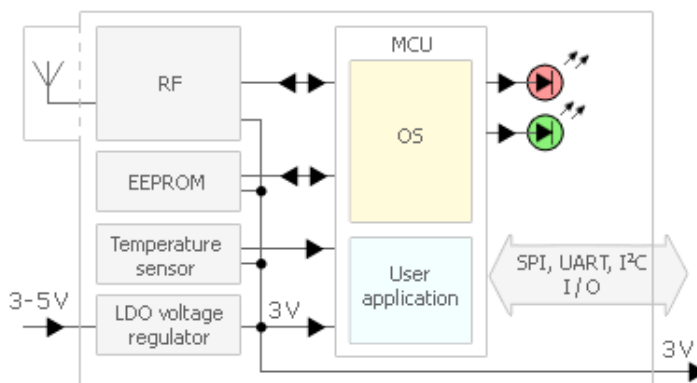


Obr. 15: Bezdrátový IQRF modul DCTR-72DAT [34]

Mikrokontrolér má implementovaný vlastní operační systém, který je možné rozšířit pomocí DPA (Direct Peripheral Access) frameworku, který usnadňuje práci s modulem v případě použití IQMESH sítě. Dále je možné využít možnosti nahrání vlastního uživatelského programu, který může obsahovat mnoho funkcí OS mikrokontroléru [34]. Pro přenos signálu je použita GFSK (Gaussian Frequency Shift Keying, FSK s použitím Gaussova filtru) modulace. Jedná se o digitální modulaci, tzv. kmitočtové klíčování, vhodné pro přenos binárních datových signálů [35]. Přenos dat probíhá na frekvencích 868 MHz nebo 916 MHz dle nastavení uživatele. Jedná se o ISM pásmo, které dovoluje schváleným zařízením provoz bez licenčních poplatků. Uživatel si dále může zvolit kanál v daném pásmu a sílu vysílání signálu. Rychlost (bit rate) je napevno nastavena na 19,8 kb/s, ale výrobce v budoucnu plánuje umožnit tento parametr programově měnit. Deska disponuje šesti programovatelnými



I/O piny. Mezi další vlastnosti patří například možnost využití časovače, PWM výstupu, 2 LED a A/D převodníků či analogového komparátoru. Zmíněných 6 programovatelných I/O pinů může být použito ke komunikaci přes sériovou linku, SPI, či jako uživatelské I/O piny. Jeden z nich může být aplikován pro napájení periférií napětím 3V. Schéma modulu je znázorněno na obr. 16.



Obr. 16: Schéma bezdrátového IQRF modulu série TR-72D [34]

Modul dosahuje rozměrů pouhých 31.8 x 14.9 x 3.3 mm a k připojení slouží klasický SIM konektor. Modul lze napájet napětím v rozsahu 3,1 až 5,3V a o přizpůsobení na napájecí napětí 3V se stará LDO (Low-DropOut) regulátor napětí. Spotřeba modulu se odvíjí od zvoleného módu. Při vysílání se spotřeba pohybuje mezi 8,3 až 19 mA dle zvolené síly výstupního signálu a módu (STD - Standard, LP - Low Power nebo XLP - Extra Low Power). Při přijímání se odběr pohybuje od 16  $\mu$ A až 12,3 mA dle zvoleného módu. Módy LP a XLP snižují spotřebu na straně přijímače tím, že při kontrole, zda je možné přijmout data, je modul střídavě přepínán mezi přijímáním a sleep módem. Doba kontroly je rovněž nastavitelná. Tímto způsobem je však prodloužena doba přenosu packetu, jelikož počátek přenosu může být takto podstatně zdržen, a s tím spojena zvýšená spotřeba na straně vysílače. Vysílač i přijímač musí být nastaveny na stejný mód. Pokud je modul uveden do stavu spánku a všechny periférie vypnuty, je možné dosáhnout spotřeby 2,9  $\mu$ A. Při běhu programu může být RF obvod uspán (1,4 mA) nebo připraven (2,8 mA). Výrobce uvádí maximální dosah v ideálních podmínkách až 1100 m, zatímco módy pro úsporu energie LP a XLP snižují dosah až o 25%. Maximální přenosová vzdálenost dále závisí na mnoha okolnostech. Modul má nejlepší vlastnosti z hlediska dosahu v případě, že je provozován (resp. jeho anténa) ve vertikální poloze. Při horizontální montáži je dosah v určitých směrech značně omezen. Dále závisí na okolním prostředí, zda jsou v těsné blízkosti umístěny vodivé objekty, a výšce montáže nad zemí, která aby měla být, dle informací výrobce, ideálně alespoň 1,6 m pro maximální dosah. Důležitá je i vzájemná orientace komunikujících IQRF modulů. Pro zvýšení odolnosti proti rušení může být přichodzí signál filtrován, čímž je však snížen dosah [34, 36].

## 5.4 Průběh komunikace

Tato kapitola se věnuje samotné komunikaci mezi prvky. Nejdříve je specifikován obecný průběh komunikace pomocí technologie IQRF a dále je popsána komunikace mezi jednotlivými členy.

### 5.4.1 IQRF

Pro účely této práce by bylo velmi vhodné prvky komunikující pomocí bezdrátové technologie IQRF propojit do sítě typu IQMESH, která podporuje mnoho pokročilých funkcí a značně usnadňuje implementaci bezdrátové komunikace. Bohužel bylo po konzultaci s externími pracovníky zjištěno, že se tento typ sítě nehodí pro dynamicky se měnící síť, ale pouze pro statické prvky, které za provozu nemění svou polohu. Z tohoto důvodu musela být zvolena druhá varianta, kterou je komunikace typu broadcast. Vyslanou zprávu může zachytit a přijmout kterýkoli prvek (i více). Adresování musí být tedy řešeno v obsahové části přenášených dat, jelikož nelze použít pokročilé integrované funkce sítě IQMESH.

Komunikace mezi mbed deskou a IQRF modulem je realizována pomocí sběrnice SPI. Mbed figuruje na sběrnici SPI jako master a IQRF modul vystupuje jako slave. Jeden ze zbylých pinů bezdrátového modulu je navíc připojen k mbed za účelem signalizace přijetí zprávy modulem. Data musí být vyčtena z bufferu IQRF modulu předtím, než je přijata nová zpráva, jinak dojde k nenávratné ztrátě informací.

Mbed kontroluje stav IQRF modulu (SPI\_CHECK) pomocí jednoho Byte 0x00 (tzv. Dummy Byte), na který dostává odpověď (SPI\_STATUS) dle stavu modulu. Výčet možných odpovědí na SPI\_CHECK, tedy SPI\_STATUS, je uveden v tab. 4.

| HEX                         | SPI status  |
|-----------------------------|---|
| 00                          | SPI neaktivní   |
| 07                          | SPI pozastaveno   |
| 3F                          | SPI není připraven, buffer plný, poslední kontrolní součet správný                                  |
| 3E                          | SPI není připraven, buffer plný, poslední kontrolní součet nesprávný                                |
| 40 – (40+N <sub>max</sub> ) | SPI data připravena<br>41 až 40+N <sub>max</sub> : délka dat = hodnota – 0x40<br>40: délka dat = 64 |
| 80                          | SPI připraven – komunikační mód   |
| 81                          | SPI připraven – programovací mód  |
| 82                          | SPI připraven – debug mód   |
| FF                          | SPI neaktivní (HW error)  |

Tab. 4: SPI status



Obecná struktura vysílaného packetu i odpovědi IQRF modulu je znázorněna na obr. 17, který zahrnuje jak komunikaci při kontrole stavu modulu, tak i obecný formát packetu při odesílání či přijímání zprávy. Packet se skládá z několika bloků o délce 1 Byte.

|           |         |         |                 |                 |     |                        |      |           |
|-----------|---------|---------|-----------------|-----------------|-----|------------------------|------|-----------|
| SPI_CHECK | SPI_CMD | PTYPE   | DM <sub>1</sub> | DM <sub>2</sub> | --- | DM <sub>SPIIDLEN</sub> | CRCM | SPI_CHECK |
| SPI_STAT  | SPISTAT | SPISTAT | DS <sub>1</sub> | DS <sub>2</sub> | --- | DS <sub>SPIIDLEN</sub> | CRCS | SPI_STAT  |

Obr. 17: Kontrola stavu IQRF modulu (vlevo), struktura packetu (vpravo)

Blok SPI\_CMD může obsahovat tři hodnoty, ale pro účel této práce je používána pouze hodnota 0xF0 značící čtení nebo zápis dat. Byte PTYPE (obr. 18) je rozdělen na dvě části. MSB (Most Significant Bit) je nazýván CTYPE a určuje, zda bude buffer modulu sloužící ke komunikaci přes sběrnici SPI měněn nebo ne. Zbylých 7 bitů označuje délku přenášené zprávy (SPIDLEN).

|       |         |    |    |    |    |    |    |
|-------|---------|----|----|----|----|----|----|
| b7    | b6      | b5 | b4 | b3 | b2 | b1 | b0 |
| CTYPE | SPIDLEN |    |    |    |    |    |    |

Obr. 18: Struktura PTYPE Bytu

Následuje vlastní zpráva (Byty DM<sub>1</sub> až DM<sub>SPIIDLEN</sub>) a na závěr je umístěn kontrolní součet. Kontrolní součty od master/slave (mbed/IQRF modul) prvku (CRCM/CRCS) jsou vypočítány dle následujících vztahů:

$$\begin{aligned} \text{CRCM} &= \text{SPI\_CMD} \text{ xor } \text{PTYPE} \text{ xor } \text{DM}_1 \text{ xor } \text{DM}_2 \text{ xor } \dots \text{ xor } \text{DM}_{\text{SPIDLEN}} \text{ xor } 0xF5 \\ \text{CRCS} &= \text{PTYPE} \text{ xor } \text{DS}_1 \text{ xor } \text{DS}_2 \text{ xor } \dots \text{ xor } \text{DS}_{\text{SPIDLEN}} \text{ xor } 0xF5 \end{aligned}$$

CRC je počítáno a kontrolováno jak v mbed, tak v IQRF modulu na vysílací i přijímací straně. Tímto způsobem je zajištěna indikace porušení dat [37].

Jak již bylo zmíněno, program IQRF modulů je možné přizpůsobit dle potřeby. Této vlastnosti je využito k iniciaci modulu. Vysílané zprávy jsou přijímány všemi IQRF moduly v dosahu a tudíž by byly mbed velice často vyzývány k přijetí této zprávy i v případě, že nejsou jejími adresáty. Jelikož jsou zprávy adresovány vždy právě jednomu prvku, docházelo by k této situaci s rostoucím počtem prvků stále častěji. Z tohoto důvodu je na začátek programu zařazena iniciační procedura, která zaručí, že IQRF modul bude předávat své nadřazené vývojové desce pouze zprávy, které jsou této desce adresované.

Uživatel musí každému prvku komunikujícímu pomocí technologie IQRF nastavit jeho unikátní ID. Typ je již určen vlastním programem, který je různý pro všechny druhy prvků. Tyto údaje jsou poté přeneseny do IQRF modulu pomocí zprávy (tečky mezi znaky v této i dalších ukázkách jsou použity pouze pro zpřehlednění zápisu):

I.N.I.T.*typ.id*

Byte *typ* je nahrazen znakem 'R', 'M', nebo 'S' dle typu prvku. Byte *id* určuje identifikační číslo nastavené uživatelem.

Dále je implementováno potvrzování zpráv. IQRF modul, který přijme zprávu adresovanou příslušnému prvku, odpoví potvrzovací zprávou. Tato potvrzovací zpráva je po přijetí předána vysílacímu mbed. V případě nepřijetí potvrzení IQRF modulem je po uplynutí 50 ms vyslání zprávy modulem opakováno. Pokud ani po druhém opakování nepřijde potvrzení, je nadřazené vývojové desce poslána informace o neúspěšném přenosu zprávy. Chování IQRF modulu je znázorněno v následující ukázce:

```
//IQRF modul M0 přijme od nadrazene mbed desky zprávu pro R0 a posle ji
//zápis M.0.R.0 označuje odesílatele (master 0) a adresáta (robot 0)
SPI_IN: M.0.R.0.W
IQRF_OUT: M.0.R.0.W

//IQRF modul R0 tuto zprávu přijme, preda nadrazene
//mbed desce a zaroven vysle potvrzovací zprávu
SPI_OUT: M.0.R.0.W
IQRF_OUT: R.0.M.0.K

//IQRF modul M0 posila po prijeti potvrzeni nadrazene mbed desce
//potvrzeni uspesneho prenosu
IQRF_IN: R.0.M.0.K
SPI_OUT: R.0.M.0.O

//v pripade neprijeti potvrzeni modul M0 opakuje vysilani a v pripade
//opetovneho neprijeti potvrzeni odesila zprávu znacici neuspesny
//prenos
SPI_OUT: R.0.M.0.N
```

#### 5.4.1 Master věž – robot

Komunikace mezi master věží a robotem je oboustranná, řízena věží. Master věž se periodicky dotazuje všech robotů, které má v paměti. Pouze po určitou dobu po obdržení tohoto dotazu může robot odpovědět. Robot komunikaci nikdy neiniculuje. Po odeslání každého dotazu je nejprve přijata zpráva od IQRF modulu značící obdržení či neobdržení zprávy příjemcem a případně posléze master věž čeká určitý časový interval na odpověď.

#### 5.4.2 Master věž – slave věž

Pro zajištění lokalizační procedury je potřeba, aby v jeden okamžik vysílala nanejvýš jedna věž svou sekvenci. Za tímto účelem je implementována komunikace mezi mbed, které představují master a slave věže, ve třech variantách. Jedna z možností spočívá v propojení všech věží do kruhové sítě. Dále jsou stanice propojeny pomocí sběrnice I<sup>2</sup>C a ve třetí možnosti je spojení provedeno pomocí technologie IQRF.

#### Token ring

Propojení věží do topologie typu kruh je nejjednodušší z uvedených variant. Každá věž (včetně master věže) zná pouze svého předchůdce a následníka. Konkrétně je každá věž propojená jedním vodičem se svým následníkem a předchůdcem. Při aktivním požadavku na lokalizaci si věže posílají impuls (token), který představuje výzvu k odvysílání sekvence.

Tuto proceduru zahajuje i ukončuje hlavní věž, tudíž i v případě, že požadavek na lokalizaci již není aktivní, pokračuje předávání tokenu až po master stanici, kde je zadržen. Výhodou této možnosti je velice snadná implementace i změna počtu či pořadí věží, jelikož není zavedena adresace. Nevýhodou je ovšem závislost celku na každém jednotlivém prvku.

Master věž má mezi iniciačními proměnnými i informaci o počtu podřízených věží a tudíž je schopna poruchu logického kruhu odhalit měřením doby od vyslání pulzu do jeho přijetí. Kritická doba pro přijetí  $t_{TIMEOUT}$  je počítána dle vztahu (1):

$$t_{TIMEOUT} = t_{LS} * (n_{SV} + 1) \quad (1)$$

|          |                                    |
|----------|------------------------------------|
| $t_{LS}$ | doba vysílání lokalizační sekvence |
| $n_{SV}$ | počet slave věží                   |

K počtu slave věží je přičtena jedna navíc, jelikož je nutné počítat s konečnou rychlostí reakce a provedení příkazu a probuzení podřízených stanic z režimu snížené spotřeby energie.

## I<sup>2</sup>C

Další z implementovaných variant je propojení stanic sběrnici I<sup>2</sup>C. Všechny věže jsou propojeny pouze pomocí dvou vodičů (SDA a SCL). Master věž má pro toto řešení mezi iniciačními proměnnými i seznam adres slave věží. Při aktivní žádosti o lokalizaci všechny věže postupně vysílají lokalizační sekvenci. Řízení věží obstarává nadřazená stanice. Oslovení jedné podřízené věže se skládá z několika částí. Nejprve je daná slave věž vyzvána k zahájení lokalizační sekvence znakem 'y'. Současně je kontrolována úspěšnost přenosu. Po uplynutí doby určené k vysílání lokalizační sekvence s+e master věž periodicky dotazuje dané věže, zda je sekvence ukončena. Slave věž odpovídá znakem 'n' v případě, že stále ještě vysílá sekvenci, či znakem 'y' značícím dokončení úkolu. Po obdržení zprávy o ukončení, nebo uplynutí doby  $2 * t_{LS}$ , což je vyhodnoceno jako porucha dané věže, je oslovena další podřízená stanice. V případě, že není žádná aktivní žádost o lokalizaci, master věž již neoslovuje žádné další věže.

Výhodou této varianty je propojení všech věží pouze pomocí dvou vodičů a případný výpadek jedné nebo několika věží neznamená znemožnění provedení lokalizační procedury. Způsobí však její zdržení.

## IQRF

Třetí a poslední variantou je propojení lokalizačních stanic pomocí stejné technologie, jakou jsou propojeny roboti s master věží, a to pomocí IQRF. V zásadě se jedná o stejný postup jako v případě propojení pomocí sběrnice I<sup>2</sup>C, pouze s tím omezením, že hrozí kolize mezi odpověďmi robota a slave věže. Proto je implementován mechanismus, který tomu zabráňuje. Hlavní věž oslovuje robota či podřízenou věž pouze v případě, že již není očekávána odpověď od jiného prvku. Tímto způsobem je zajištěno, že vysílat přes IQRF může pouze jeden prvek naráz, a tudíž je zabráněno kolizi. Příznak značící čekání na odpověď, ať už od slave věže či robota, je udržován aktivní do přijetí odpovědi nebo informace o nepřijetí zprávy.

Master věž předává pokyn k vysílání slave věží zprávou ve formátu  $M.id_M.S.id_S.Y$ , kde  $id_M$  zastupuje ID hlavní věže a analogicky  $id_S$  ID podřízené věže. Po uplynutí doby

pro vysílání sekvence se master věž dotazuje na stav zprávou  $M.id_M.S.id_S.H$ , na kterou obdrží odpověď  $S.id_S.M.id_M.K$  značící stále probíhající vysílání nebo  $S.id_S.M.id_M.Y$  pro dokončení.

Tato možnost je časově nejnáročnější, jelikož časové náklady spojené s režii bezdrátového přenosu jsou nepoměrně větší, než v předchozích dvou variantách. Dále může být způsobeno občasné prodloužení intervalu mezi oslovováním robota během lokalizační procedury, jelikož je upřednostněno oslovení slave věže před robotem, z důvodu zajištění co nejrychlejší lokalizace. To se jeví více prioritní, než vysoká frekvence oslovování robotů. Na druhou stranu s sebou nese tato možnost nesporné výhody, zejména absence nutnosti propojování stanic vodiči. Věže mohou být tedy umístěny kdekoli pouze s omezením dosahu IQRF modulů.

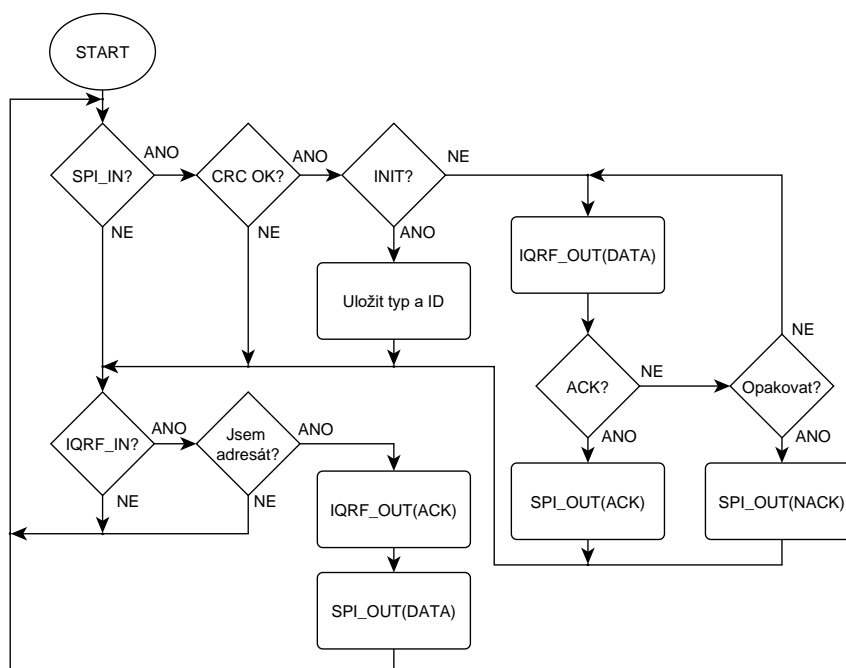
## 5.5 Algoritmus

Tato kapitola se zabývá logikou jednotlivých prvků, snaží se objasnit, jak se který prvek chová a reaguje na nastalé situace. Proti zastavení programu v důsledku neočekávané chyby mají prvky nastaven ochranný timer, který zajišťuje uvedení programu do chodu i v těchto případech.

### 5.5.1 IQRF modul

Program IQRF modulu je navrhnut a napsán tak, aby byl univerzální a bylo možné ho použít u jakéhokoliv prvku nezávisle na jeho typu. Chování IQRF modulů je řízeno nadřazenou mbed deskou.

Program je složen ze dvou částí: příjem dat od nadřazené mbed desky a příjem dat od jiného IQRF modulu. Algoritmus modulu je znázorněn na obr. 19.



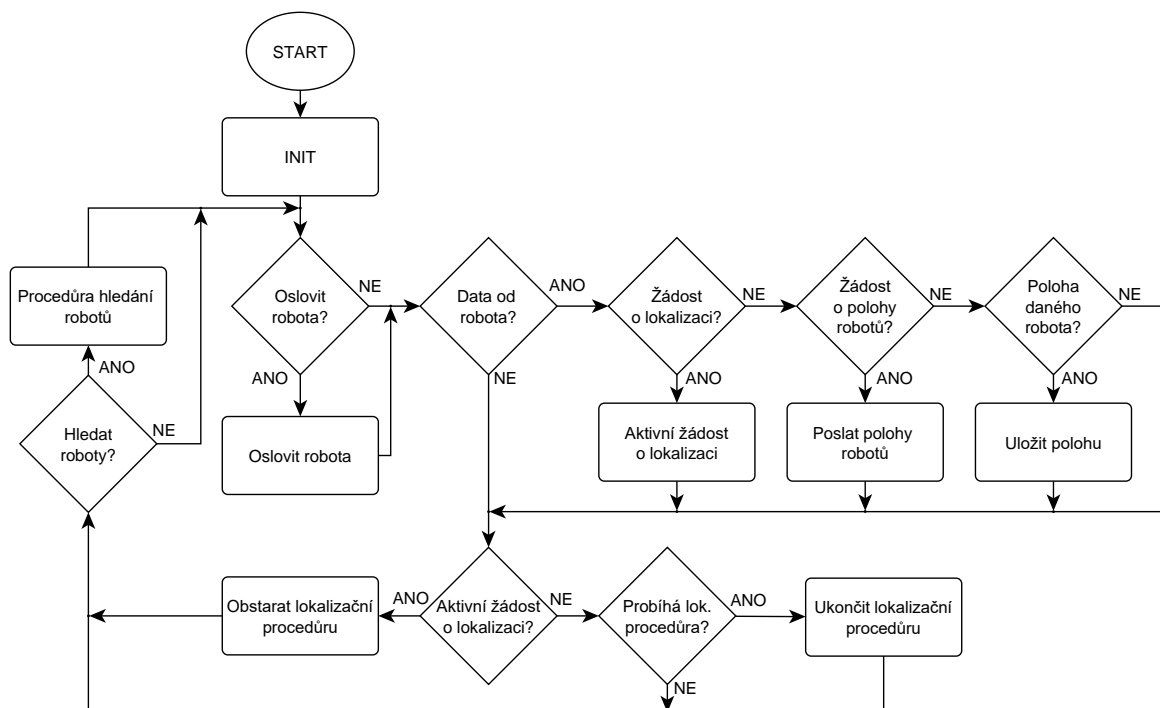
Obr. 19: Algoritmus IQRF modulu

V případě, že jsou přijata data od nadřazené mbed desky (SPI\_IN), je zkoumána kontrolní část zprávy (CRC). Pokud je kontrola úspěšná, následuje zkoumání typu zprávy. Může se jednat o iniciační zprávu (INIT), která udává typ a ID daného prvku, v tom případě jsou tyto informace uloženy, nebo se jedná o zprávu určenou k odeslání a je odvíšována pomocí IQRF (IQRF\_OUT) a spuštěn časovač po dobu 50 ms, což je doba, po kterou je nutné nejpozději přijmout potvrzovací zprávu. Vysílání je jednou opakováno v případě nepřijetí potvrzení přenosu od adresovaného IQRF modulu. Poté je informace o přijetí (ACK) či nepřijetí (NACK) odeslána nadřazené mbed desce. Druhá část kódu se stará o bezdrátový příjem dat (IQRF\_IN). V takovém případě jsou zkontrolovány údaje, pro koho je zpráva určena, a pokud je adresována právě danému prvku, je odeslán ACK signál odesílateli a následně zpráva předána nadřazené mbed desce. Předání zprávy nadřazené mbed desce spočívá ve vyslání krátkého pulzu, který způsobí přerušení v nadřazené vývojové desce, a značí připravená data k přenosu přes sběrnici SPI.

### 5.5.2 Master věž

Algoritmus master věže (obr. 20) začíná definicí potřebných proměnných. Jednou z nich je i seznam robotů, ve kterém jsou obsaženy informace o jejich ID, poslední známé poloze, zda má robot aktivní žádost o lokalizaci atd. Tento seznam je neustále udržován a v případě přijetí relevantních informací aktualizován. V iniciační části programu je mimo jiné nastaven IQRF modul.

Dále je v programu zařazena procedura pro hledání dostupných robotů, která je periodicky spouštěna. Nově nalezení roboti jsou přidáni do seznamu robotů a dříve objevení roboti zůstávají v seznamu beze změny, tedy včetně dosud získaných informací o poloze.



Obr. 20: Algoritmus master věže

Hlavní program master věže se dělí na tři části. Oslovení robota, příjem dat od robota a obsluha slave věží. První část, oslovení robota, je řízena časovačem nastaveným na 1 s. Tento údaj je možné měnit, či dokonce zcela odstranit a dotazovat se následujícího robota ihned po obdržení odpovědi (či NACK) od předchozího mobilního prvku. Toto rozhodnutí je závislé na konkrétní aplikaci a zvýšení frekvence dotazování robotů s sebou nese i navýšení spotřeby energie. Je tedy vhodné při implementaci na určitou situaci zvolit vhodný kompromis. Postupně jsou oslovováni roboti dle pořadí v seznamu aktivních robotů. V případě neúspěšného odesílání výzvy je tato skutečnost zaznamenána, a pokud se tato situace několikrát opakuje, je příslušný robot přesunut do seznamu neaktivních robotů. Neaktivní roboti nejsou nadále dotazováni, ale informace o nich jsou stále drženy v paměti master věže. Neaktivní robot se může dostat opět do seznamu aktivních pouze během procedury hledání dostupných robotů.

Master věž může obdržet 4 různé odpovědi od robota: žádost o lokalizaci, žádost o polohu všech známých robotů, polohu daného robota či zprávu značící, že robot nemá žádný požadavek. Žádost o lokalizaci způsobí, že je aktivována lokalizační procedura. Najednou může mít aktivní žádost o lokalizaci více robotů najednou. Lokalizační procedura je spuštěna při přijetí první žádosti a ukončena až ve chvíli, kdy není žádná žádost aktivní. Žádost o polohy všech známých robotů následuje procedura, při níž jsou robotu poslány poslední známé polohy všech robotů v paměti master věže. Odpověď robota obsahující polohu daného prvku značí úspěšnou lokalizaci, žádost robota o lokalizaci je deaktivována a údaje zaznamenány.

Poslední část programu, obsluha slave věží, se liší dle zvolené varianty (token ring, I<sup>2</sup>C či IQRF). V zásadě jde o různá řešení stejného postupu. V případě, že má alespoň jeden robot aktivní žádost o lokalizaci, se master věž stará o to, aby postupně vysílaly všechny věže (včetně jí samé) svou lokalizační sekvencí. Důležité je dodržet, aby v daný okamžik vysílala maximálně jedna věž. Procedura pro zjištění dostupných věží není implementována, jelikož není předpokládána častá změna struktury věží.

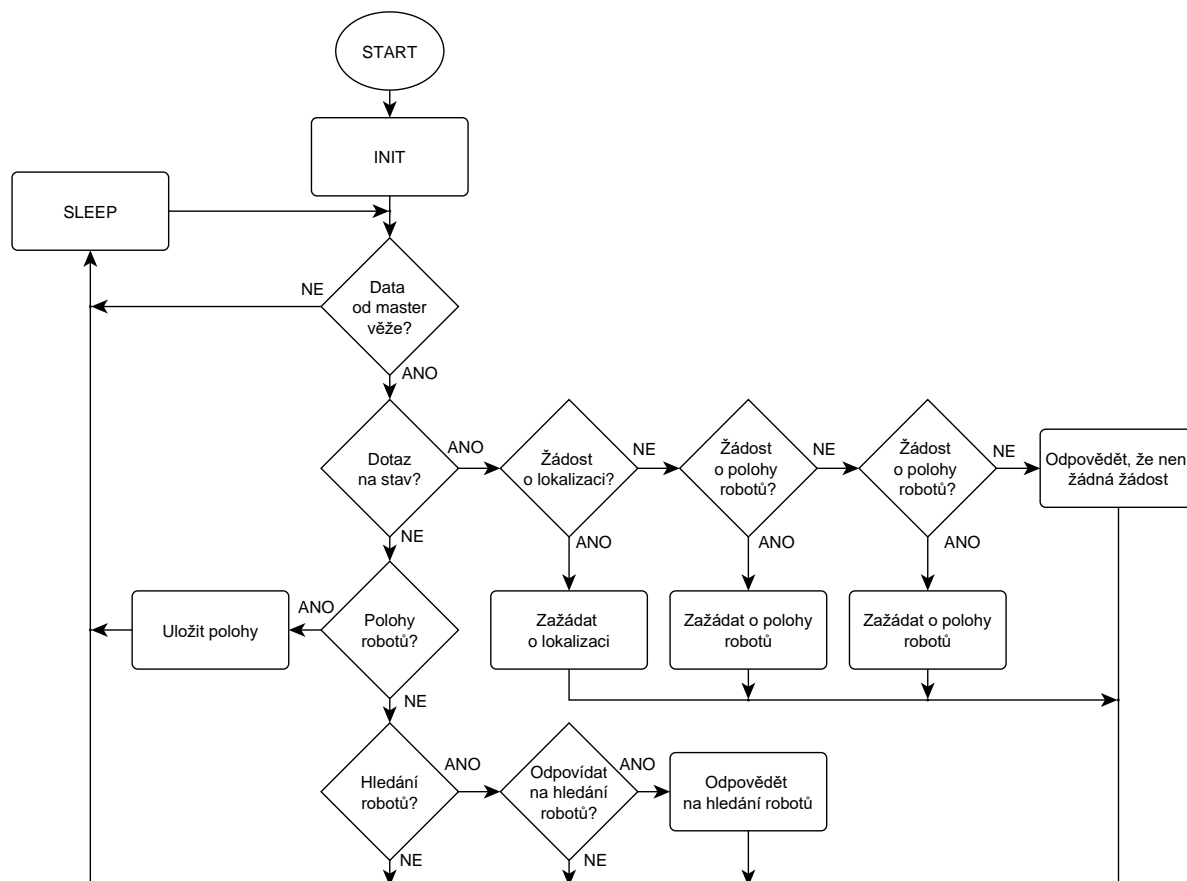
### 5.5.3 Slave věž

Prvek typu slave věž má jediný úkol. Po přijetí pokynu k vysílání lokalizační sekvence dojde k jejímu spuštění. Po dokončení předává slave stanice informaci o dokončení dále (master nebo další slave stanici – dle zvolené varianty). Po dobu nečinnosti je mbed uveden do stavu snížené spotřeby energie, ze které může být vzbuzen přijetím zahajovacího pokynu.

### 5.5.4 Robot

Algoritmus prvku typu robot je znázorněn na obr. 21. Na začátek je zařazena počáteční iniciace potřebných proměnných, nastavení SPI sběrnice a IQRF modulu. Pro úsporu spotřeby energie, která je u mobilního robota žádoucí, je v případě nečinnosti prvek uspán pro snížení spotřeby energie. IQRF modul však neustále přijímá data a v případě přijetí zprávy adresované danému robotu je mbed probuzen.

V případě přijetí dotazu na stav robota je tento stav zkontrolován a následuje vhodná odpověď. Ta může značit žádost o lokalizaci, žádost o polohu všech robotů, v případě předcházející úspěšné lokalizace vlastní polohu či pouze fakt, že je robot na příjmu a žádný požadavek momentálně nemá.



Obr. 21: Algoritmus prvku typu robot

## 5.6 Struktura dat

Mezi prvky, které spolu komunikují pomocí IQRF technologie, je výměna informací realizována prostřednictvím krátkých zpráv, jejichž struktura je znázorněna na obr. 22. Jedná se pouze o datovou část, která je částí celkově přenášených dat do IQRF modulu (Byty DM<sub>1</sub> až DM<sub>SPIDLEN</sub>, vizte kap. 5.4.1). Skládá se z pěti Bytů. Výjimky tvoří procedura hledání robotů masterem a přenos informací o poloze robota, o kterých je pojednáváno dále.

Pro přehlednost a čitelnost zpráv je každý Byte vyjádřen pomocí ASCII znaku. Byte S\_TYPE může nabývat pouze tří hodnot: R – robot, M – master věž, S – slave věž. Tím je určen typ odesílatele. Následující blok S\_ID jednoznačně identifikuje prvek z dané skupiny. Stejným způsobem je identifikován adresát (Byty R\_TYPE a R\_ID). První čtyři Byty tedy slouží k definici odkud je zpráva vysílána a kam je určena.

Pátý člen TM slouží k sémantické identifikaci zprávy. Kromě přenosu polohy slouží rovněž i jako jediný nositel informace zprávy. Každý prvek má definovaný výčet typů zpráv,



které přijímá a vysílá. Pokud je přijat nedefinovaný typ zprávy, je tato skutečnost vyhodnocena jako neznámý požadavek a nadále je ignorován.

|        |      |        |      |    |
|--------|------|--------|------|----|
| S_TYPE | S_ID | R_TYPE | R_ID | TM |
|--------|------|--------|------|----|

Obr. 22: Struktura zprávy

|        |                               |
|--------|-------------------------------|
| S_TYPE | typ odesílatele (Sender_TYPE) |
| S_ID   | ID odesílatele (Sender_ID)    |
| R_TYPE | typ adresáta (Recipient_TYPE) |
| R_ID   | ID adresáta (Recipient_ID)    |
| TM     | typ zprávy (Type of Message)  |

Robot komunikuje pouze s hlavní věží a tudíž je vždy adresát nebo odesílatel právě prvek typu master. Seznam typů zpráv vztahující se k prvku typu robot je znázorněn v tab. 5.

| Příjem |               | Vysílání |                           |
|--------|---------------|----------|---------------------------|
| W      | Dotaz na stav | L        | Žádost o lokalizaci       |
| G      | Příjem poloh  | G        | Žádost o polohy           |
|        |               | P        | Vysílání vlastní polohy   |
|        |               | A        | Žádný požadavek           |
|        |               | H        | Odpověď na hledání robotů |

Tab. 5: Typy zpráv definované pro prvek typu robot

V případě příjmu zprávy robot reaguje (odpovídá) pouze na dva typy zprávy. Příjem znaku 'W' značí dotaz na stav daného robota a znak 'G' signalizuje přenos poloh všech robotů, které má master věž v paměti. Na zprávu značící dotaz na stav robota mobilní prvek odpovídá příslušnou zprávou dle situace.

Slave věž si rovněž jako robot vyměňuje informace pouze s master věží. Výčet typů zpráv slave věže při komunikaci s master věží pomocí IQRF technologie je v tab. 6.

| Příjem |   | Vysílání |                                 |
|--------|---|----------|---------------------------------|
| Y      | Pokyn k zahájení lokalizační sekvence   | Y        | Lokalizační sekvence ukončena   |
| H      | Dotaz na dokončení lokalizační sekvence | K        | Lokalizační sekvence neukončena |

Tab. 6: Typy zpráv definované pro prvek typu slave věž

Příjem 'Y' signalizuje pokyn k zahájení lokalizační sekvence a typ 'H' slouží k zjištění stavu vysílání lokalizační sekvence. Na druhý zmíněný typ zprávy podřízená věž odpovídá dle toho, zda bylo vysílání lokalizační sekvence ukončeno.



Master věž může komunikovat jak s robotem tak se slave věží. Seznam typů zpráv pro hlavní stanici je patrný z tab. 5 a 6. Pro úplnost je tento seznam uveden v tab. 7.

| Master věž – robot     |          |
|------------------------|----------|
| Příjem                 | Vysílání |
| L                      | W        |
| G                      | G        |
| P                      |          |
| A                      |          |
| H                      |          |
| Master věž – slave věž |          |
| Příjem                 | Vysílání |
| Y                      | Y        |
| K                      | H        |

Tab. 7: Typy zpráv definované pro prvek typu master věž

## 5.7 Hledání dostupných robotů

Master věž hledá dostupné roboty pomocí zprávy  $M.id_M.R.X$ , kde  $id_M$  zastupuje ID dané master stanice. Roboti mohou reagovat z několika důvodů. Může se jednat o situaci, kdy je robot zcela nově přidán nebo nedošlo již delší dobu k oslovení robota master věží, což značí, že byl robot označen jako neaktivní. Pokud nenastane ani jedna ze zmíněných situací, prvek nereaguje. V opačném případě robot čeká náhodný časový úsek (0 až 500 ms) a poté odpovídá zprávou  $R.id_R.M.id_M.H$ . Master stanice po přijetí této zprávy přidá daného robota do seznamu robotů, případně změni jeho status na aktivní. Při proceduře hledání dostupných robotů jak na straně master stanice, tak na straně robotů, nedochází k potvrzování zpráv. Ukázková situace může vypadat následovně:

```
//master vez 0 hleda dostupne roboty
M.0.R.X
//robot 0, ktery chce byt dotazovan danym masterem, odpovida
R.0.M.0.H
//robot 5, ktery chce byt dotazovan danym masterem, odpovida
R.5.M.0.H
```

## 5.8 Přenos poloh

Je předpokládán 2-D (dvou-dimenzionální) souřadnicový systém shodný pro všechny roboty. Blok obsahující informace o poloze robota se skládá z pěti Bytů. První Byte (P\_ID, Possition\_ID) obsahuje ID daného robota a následujících 32 Bytů udává polohu daného robota v osách x a y ( $x_1, x_2, y_1, y_2$ ). Jeden nebo více těchto bloků je připojeno za Byte TM

(vizte kap. 5.6). Struktura části zprávy při přenosu informace o poloze robota je znázorněna na obr. 23.

|      |       |       |       |       |
|------|-------|-------|-------|-------|
| P_ID | $x_1$ | $x_2$ | $y_1$ | $y_2$ |
|------|-------|-------|-------|-------|

Obr. 23: Část zprávy sloužící k přenosu polohy robota

Z pohledu master věže mohou nastat dvě situace: robot posílá svou polohu po úspěšné lokalizaci, nebo žádá o poslední známé polohy všech mobilních prvků, které má master věž k dispozici.

U první varianty je Byte P\_ID nadbytečný, jelikož identifikace robota je již obsažena v adresové části zprávy (S\_ID). Pro zachování konzistentní struktury zpráv v rámci aplikace je však tato redundantní část použita. Po úspěšné lokalizaci robot odpoví na dotaz na stav zprávou obsahující jeho určenou polohu. Průběh komunikace může být následující:

```
//master vez 0 se dotazuje na stav robota 0
M.0.R.5.W
//v pripade, ze se robot uspesne lokalizoval, odpovida zpravou
//obsahující jeho polohu
R.5.M.0.P.5.2511.320
```

Druhá možnost se skládá ze dvou částí. V případě potřeby robot odpovídá žádostí o polohu všech robotů. Následně master věž zahajuje proceduru odesílání poloh. Polohy jsou posílány pohromadě v jedné nebo více zprávách. Přenos začíná zprávou, kde do bloku TM je vložen znak 'G' a následuje údaj určující počet přenášených poloh. Poté následují vlastní informace o polohách robotů. Jedna zpráva může obsahovat údaje maximálně o deseti robotech. V případě vyššího počtu jsou údaje rozděleny do více zpráv. Konec přenosu je signalizován znakem 'E'. Robot si po přijetí tohoto znaku zkontroluje počet poloh, které měl přijmout (údaj na začátku první zprávy), a skutečně přijatých. V případě nesouhlasných údajů svůj požadavek na dotaz opakuje. Ukázkový přenos poloh je následující:

```
//master vez 0 se dotazuje na stav robota 0
M.0.R.0.W
//v pripade pozadavku robot odpovida zadosti o polohy všech robotu
R.0.M.0.G
//master vez zahajuje prenos, v paměti má 11 robotů
//posila polohy 10 robotu: 1, 2, 8, ...
M.0.R.0.G.11.1.152.121.2.322.587.8.695.478...
//nasledne posila polohu 11. robota (27) a signalizuje konec prenosu
M.0.R.0.G.27.147.863.E
```

## 5.9 Software

Hlavní program pro mbed je uložen v souboru typu cpp. Dále jsou přidruženy soubory obsahující pomocné třídy a funkce. Tyto soubory obsahují často používané funkce

pro zřehlednění hlavního programu. IQRF modul má svůj program v souboru myTR.c. Všechny zdrojové kódy jsou umístěny v příloze. Při jejich psaní byly použity informace z [38].

### 5.9.1 mbed

Pro vývoj programu pro vývojovou desku mbed byl použit mbed online kompilátor a programovací jazyk C++. V této kapitole jsou představeny části funkcí použitých pro řešení úlohy. Při psaní kódu pro mbed desku byly použity informace z [37].

#### Kontrola stavu IQRF modulu

Funkce *checkSPIStatus(DigitalOut slave, SPI spi)* slouží k zjištění statusu IQRF modulu. Přijímá dva parametry, kterými jsou *slave* označující DigitalOut pin, na který je připojen SS IQRF modulu, a *spi*, což je objekt třídy SPI používaný pro komunikaci po této sběrnici. V následujícím kódu je uveden příkaz sloužící k odeslání a přijetí dat přes sběrnici SPI definovaný ve standardní knihovně mbed.h. Pro zjišťování statusu IQRF modulu je výrobcem doporučeno použít tzv. dummy Byte (0x00).

```
char response = spi.write(0x00);
```

#### Výpočet CRC

Pro zajištění detekce nekonzistence dat je používán kontrolní součet, CRC (Cyclic Redundancy Check, cyklický redundantní součet). Výpočet CRC je prováděn ve funkci *getCRC(vector<char> data)*, jež přebírá zprávu a vrací jeden Byte v podobě hledaného kontrolního součtu.

```
char crc = data.at(0)^0x5F;
for(int i = 1; i < data.size(); i++)
{
    crc = crc ^ data.at(i);
}
```

#### Odesílání zprávy

Pro odeslání dat slouží funkce *sendData(vector<char> data, DigitalOut slave, SPI spi)*. Kromě objektu třídy SPI a pinu SS přebírá navíc vlastní data. Nejprve je provedena kontrola, zda zpráva svou délkou nepřesahuje maximální povolenou hodnotu. Poté jsou na začátek dat přidány Byty SPI\_CMD a PTYPE a na konec zprávy Byty CRCM a SPI\_CHECK (vizte kap. 5.4.1). Tato zpráva je poté odeslána do IQRF modulu a zároveň přijímána odpověď. U odpovědi je zkontrolován CRCS a SPI\_STATUS značící zda proběhla komunikace s IQRF modulem v pořádku.

```
vector<char> dataToSend;
dataToSend.push_back(0xF0);           //SPI_CMD: Data read/write
dataToSend.push_back(128+sizeofMessage); //PTYPE: CTYPE 1 + SPIDLEN
for(int i = 0; i < sizeofMessage; i++)
{
    dataToSend.push_back(message.at(i)); //vlastni zprava
}
char crc = getCRC(dataToSend);
dataToSend.push_back(crc);             //CRCM
dataToSend.push_back(0x00);            //SPI_CHECK
```

### Příjem zprávy

Ke čtení dat z IQRF modulu je určená funkce *readData(char length, DigitalOut slave, SPI spi)*, která přebírá, mimo proměnných vztahujících se ke sběrnici SPI, i délku zprávy, která má být přijata. Tato délka je zjištěna z předchozího SPI\_CHECK. Po kontrole délky zprávy je sestavena sekvence, která je poslána do IQRF modulu a slouží jako pokyn k předání dat z IQRF modulu do mbed. Po odeslání/přijetí zprávy následuje kontrola bloků CRCS a SPI\_STATUS. Následně jsou z přijatých dat odstraněny první dva Byty, jelikož se jedná o Byty vyjadřující SPI\_STATUS před vyčtením dat, a poslední dva Byty (CRCS a SPI\_STATUS). Po této operaci zůstává samotná zpráva, která je zároveň návratovou hodnotou funkce.

```
vector<char> dataToSend;
dataToSend.push_back(0xF0);           //SPI_CMD: Data read/write
dataToSend.push_back(size);           //PTYPE: CTYPE 0 + SPIDLEN
for(int i = 0; i < size; i++) {
    dataToSend.push_back(0x00);        //generovani master clock
}
char crc = getCRC(dataToSend);
dataToSend.push_back(crc);             //CRCM
dataToSend.push_back(0x00);            //SPI_CHECK
```

### Pseudonáhodné generování čísel

Pro generování pseudonáhodných čísel, které slouží k určení doby čekání robota při proceduře hledání dostupných robotů, je využito funkce *rand()*. Jelikož tato funkce generuje pouze pseudonáhodnou posloupnost čísel, je využita funkce *srand(int seed)*, která slouží k iniciaci posloupnosti a tudíž je tím umožněno generování rozdílných řad. Důležité je však získat různé seedy. K tomu je použit A/D převodník mbed vývojové desky, jehož pin není připojen k žádné periférii. Z pinu A/D převodníku je přečtena hodnota napětí, která představuje náhodný šum.

```
int seed = (int)(ain.read()*1000);
srand(seed);
wait_ms(rand()%500); //cekam nahodnou dobu 0-500 ms
```

### 5.9.2 IQRF modul

K nahrání uživatelského programu byl použit programátor CK-USB-04A a volně stažitelný software IQRF IDE volně stažitelný na stránkách výrobce ([39]). Po připojení programátoru s IQRF modulem do USB portu počítače je modul aplikací rozpoznán a umožní uživateli upravit jeho nastavení a vybrat, zkompileovat a nahrát do modulu vlastní program. Dále umožňuje komunikovat s modulem pomocí terminálu a testovat reakce.

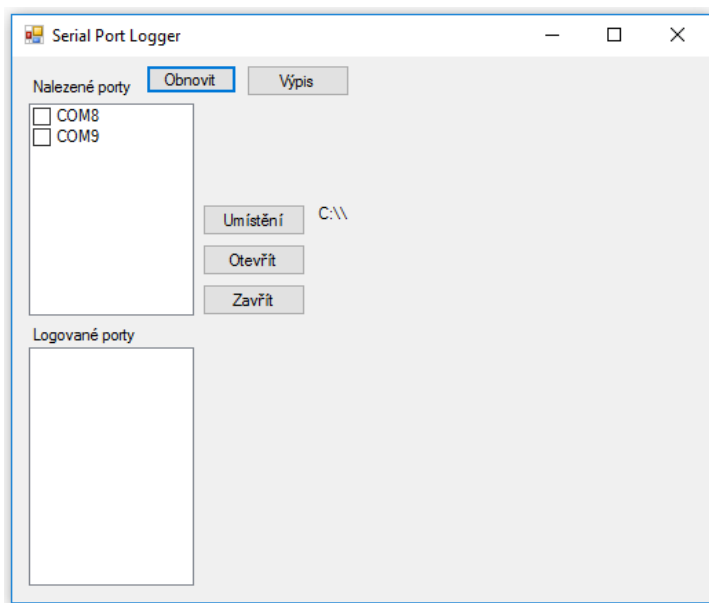
Při psaní programu pro IQRF modul byla čerpána inspirace z dokumentace a ukázkových kódů (IQRF StartUp Package) stažitelných na stránkách výrobce ([39]). Tento balíček je umístěn v příloze. Program pro bezdrátový modul je napsán v jazyce C. V iniciační části jsou nastavovány parametry modulu jako jsou síla vysílaného signálu, mód atd. Všechny použité funkce jsou již implementované v OS modulu od výrobce. Pro kontrolu, zda je možné přijmout data, slouží funkce *checkRF(int RX\_FILTER)*, kde *RX\_FILTER* udává minimální sílu signálu. K samotnému přijetí zprávy pak slouží příkaz *RFRXpacket()*. IQRF modul obsahuje dva buffery. Buffer COM je používán při komunikaci přes sběrnici SPI a Buffer RF slouží pro práci s bezdrátovým přenosem. Zpráva přijata přes IQRF je zkopírována z bufferu RF do COM příkazem *copyBufferRF2COM()* a následný příkaz *startSPI(int DLEN)*, kde *DLEN* značí délku připravených dat, způsobí, že na zprávu SPI\_CHECK modul odpoví údajem o délce připravené zprávy (SPI\_STAT). Funkce *getStatusSPI()* slouží ke zjištění stavu sběrnice SPI a aktualizaci parametru *\_SPIRX* značícím, zda byla přijata data přes SPI, a *\_SPICRCok* udávající korektnost CRCM posledních přijatých dat. Pro odeslání zprávy přes IQRF je potřeba data zkopírovat z COM bufferu do RF bufferu (*copyBufferCOM2RF()*) a následně pomocí funkce *RFTXpacket()* je zpráva odeslána. Poté je nutné zavolat příkaz *startSPI(0)* k restartování komunikace po sběrnici SPI, čímž je umožněn příjem další zprávy od mbed [39].

### 5.9.3 Diagnostický software

Pro účely práce byl vytvořen software v programovacím jazyce C# v prostředí Visual Studio Community 2015. Program slouží k ukládání zpráv od prvku do souboru typu csv. UI (User Interface), který je zobrazen na obr. 24, se skládá z několika částí. Nalevo jsou umístěny dva seznamy COM portů. Horní slouží k zobrazení dostupných portů a spodní indikuje, jaké porty jsou právě zaznamenávány.

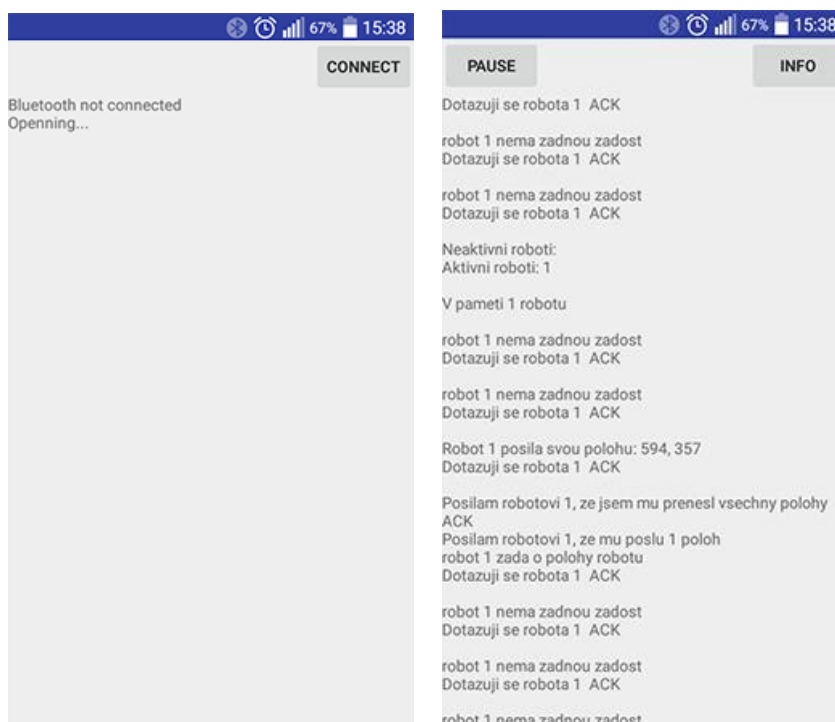
Tlačítko *obnovit* slouží k aktualizaci seznamu dostupných portů. V případě připojení a odpojení zařízení je nutné použít toto tlačítko, aby se změny projevíly. Dále jsou zde umístěna tři další tlačítka. Stisknutím tlačítka *umístění* je otevřen dialog pro výběr ukládání dat. Tlačítko *otevřít* zahrne všechny vybrané COM porty do seznamu zaznamenávaných portů. Tlačítko *zavřít* odstraní všechny vybrané porty ze seznamu zaznamenávaných portů. Soubor je umístěn v uživatelem vybrané složce (defaultně ve složce s programem). V případě, že se soubor v daném umístění nenalézá, je programem založen. V opačném případě jsou další záznamy zapisovány na konec souboru. Pro snazší diagnostiku je do souboru přidán časový údaj, kdy byl záznam přijat. Tlačítko *výpis* pošle stanici pokyn k vypsání informací o počtu robotů v paměti a seznam aktivních a neaktivních robotů, který je proveden, pokud se jedná o hlavní věž.

Spustitelný soubor a projekt pro vývojové prostředí Visual Studio Community 2015 obsahující všechny zdrojové kódy aplikace je umístěn v příloze.



Obr. 24: Software pro záznam dat z COM portů

Pro účely testování bylo rovněž potřeba prvky přemísťovat. Aby nebylo nutné s sebou nosit počítač, byla vytvořena aplikace pro OS Android. Použit byl programovací jazyk Java, programové prostředí Android Studio a vývojové nástroje Android SDK (Software Development Kit).



Obr. 25: Ukázka aplikace pro OS Android

Pro komunikaci byl použit bezdrátový Bluetooth modul HC-06, který byl připojen k mbed přes sběrnici UART. Program mbed nemusí být pro tyto účely téměř upravován, pouze je nutné převést sériovou komunikaci na příslušné piny. Před použitím je nutné spárovat mobilní zařízení s bezdrátovým modulem. Samotná aplikace využívá několika API funkcí OS Android. Po zapnutí programu nalezne v seznamu spárovaných Bluetooth zařízení právě tento modul, a pokusí se k němu připojit. V případě úspěšného spojení je uživateli umožněno spustit výpis přijatých dat, který je možné dle potřeby pozastavit. Dále je zahrnuta možnost poslat hlavní stanici pokyn k vypsání důležitých informací (tlačítko *INFO*). Ukázky z prostředí aplikace jsou na obr. 25.

Aplikace podporuje OS Android od verze 2.1.x. Testování proběhlo na zařízení LG L90 s OS Android verze 5.0.2. Projekt pro prostředí Android Studio včetně zdrojových kódů a vygenerovaný apk soubor je umístěn v příloze.



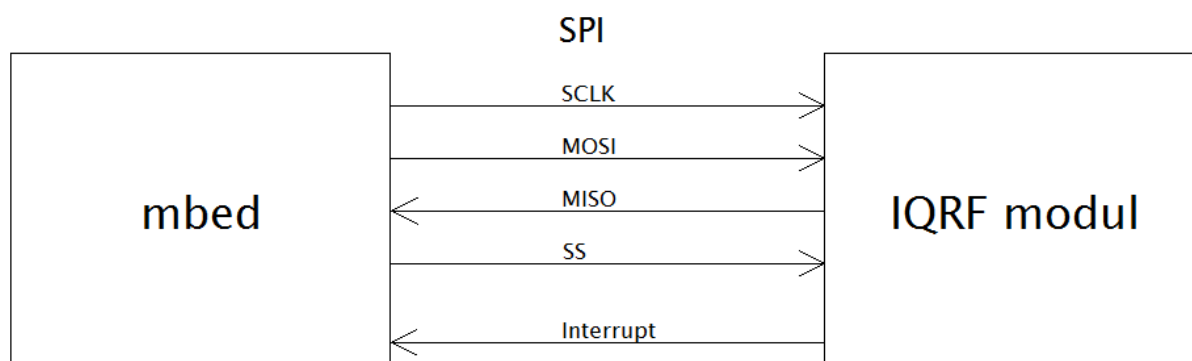


## 6 EXPERIMENTY

Tato kapitola obsahuje popis praktických experimentů, které slouží k ověření funkčnosti a vlastností řešení. Nejprve je představen použitý hardware a software a následně jsou uvedeny provedené experimenty a jejich výsledky.

### 6.1 Použitý hardware a software

Při testech byly využity již zmíněné mbed vývojové desky ST-NUCLEO F303-K8 a IQRF moduly DCTR 72-DAT, které však nebyly umístěny na reálných prvcích, jelikož v době vypracování této práce nebyly zkompletovány. Pro účely experimentu byly tedy prvky umístěny na nepájivém poli. Zjednodušené zapojení je znázorněno na obr. 26. K testování funkčnosti byly dostupné čtyři vývojové kity a stejné množství IQRF modulu. IQRF moduly byly připojeny k mbed pomocí zakoupeného adaptéru BB-02. Jelikož montáž modulu na robota závisí na jeho konstrukci, byl v prostředí Eagle vytvořen návrh podobného adaptéru, který lze dle potřeby upravit pro snadnou montáž, a je umístěn v příloze. Vývojová deska byla navíc připojená k počítači nebo k Bluetooth modulu HC-06, který byl k dispozici, přes sběrnici UART pro sběr a diagnostiku dat. K mbed byly připojeny i tři LED sloužící k signalizaci stavu prvku.



Obr. 26: Zjednodušené schéma propojení mbed a IQRF modulu

K diagnostice funkčnosti byl použit zmíněný software pro zápis sériové komunikace do log souboru a software *putty* sloužící k simulování terminálu v prostředí Windows, nebo aplikace pro Android zařízení.

### 6.2 Čas přenosu

Pro určení časové náročnosti bezdrátové komunikace byl proveden experiment, při kterém spolu komunikovali dva bezdrátové moduly. Jeden z nich představoval koordinátora, který periodicky oslovoval druhý prvek a následně přijímal jeho odpověď. Koordinátor prováděl zároveň vyhodnocování časového průběhu bezdrátového přenosu dat. V okamžiku odeslání výzvy hlavním prvkem byl spuštěn časovač a posléze byl měřen časový úsek do přijetí potvrzovacího signálu (ACK), nebo zprávy signalizující neúspěšný bezdrátový přenos

(NACK). V případě přijetí ACK signálu pokračovalo měření až do obdržení odpovědi, či dalšího oslovení. Naměřené údaje tedy zastupují dobu od odeslání výzvy po konkrétní událost. Test byl proveden pro 1000 zpráv a vyhodnocen byl minimální, maximální a průměrný časový úsek pro každou událost. Pro výpis údajů byl použit vlastní software pro systém Windows vyvinutý pro účely této práce (vizte kap. 5.9.3). Výsledky jsou uvedeny v tab. 8.

|         | počet | min. [ms] | max. [ms] | průměr [ms] |
|---------|-------|-----------|-----------|-------------|
| ACK     | 998   | 25        | 110       | 27          |
| NACK    | 2     | 155       | 163       | 159         |
| Odpověď | 998   | 102       | 270       | 160         |

Tab. 8: Výsledky experimentu pro určení doby bezdrátové komunikace

## 6.2 Dosah IQRF modulu

Experiment slouží k měření reálného dosahu dvou IQRF modulů v různém prostředí. K signalizaci odeslané a přijaté zprávy byly použity dvě LED umístěné přímo na IQRF modulu a vlastní aplikace pro mobilní zařízení Android. Testování proběhlo v prostorách Fakulty strojního inženýrství, Vysokého učení technického v Brně. Měření bylo provedeno ve volném prostoru mimo budovu a v dlouhé chodbě bez překážek simulující rozlehlou halu. Pro otestování komunikace byly použity dva prvky, které si periodicky vyměňovaly krátké zprávy. Přenos byl vyhodnocen jako úspěšný v případě, že na výzvu byla obdržena odpověď. Testovaná zařízení byla umístěna 1,5 m nad zemí.

Ve venkovních prostorách byla naměřena maximální vzdálenost bezdrátové komunikace 170 m. Průměrná úspěšnost přenosu se však pohybovala okolo 10%. Spolehlivá výměna informací byla dosažena na vzdálenost až 130 m. Pro měření ve vnitřních prostorách byla zvolena 70 m dlouhá chodba. Komunikace mezi prvky byla spolehlivá i na tuto maximální vzdálenost.

## 6.3 Reálná situace

Při testování reálné situace byla zvolena situace D (vizte kap. 5.1.1), která zahrnuje více robotů a věží v rámci jednoho celku. Test proběhl pro dvě varianty: jeden robot se třemi věžemi a dva roboti se dvěma věžemi. Robot simuluje svou činnost tím, že v nastavených časových okamžicích požaduje lokalizační proceduru, polohy všech robotů, nebo signalizuje úspěšnou lokalizaci.

Následující záznamy představují výpis z jedné z testovaných možností, kde je umístěn jeden robot, jedna master věž a dvě slave věže. Věže komunikují po sběrnici I<sup>2</sup>C. Jako první je uveden výpis prvku typu master věž. Nejprve je spuštěna procedura pro zjištění dostupných robotů. V tomto případě je nalezen pouze jeden robot. Dále je robot periodicky dotazován, zda nemá nějaký požadavek. V případě, že žádá lokalizaci, je spuštěna lokalizační procedura.

Na žádost o polohy jsou robotu poslány poslední známé polohy všech robotů v paměti master věže, v tomto případě pouze poslední známá poloha daného robota. Po úspěšné lokalizaci je na dotaz na stav přijata poloha robota. Nakonec je poslán pokyn hlavní věži k vypsání informací o robotech v paměti.

```
//hledani dostupnych robotu
Hledam roboty   cas: 21.04.2017 16:21:05:243;
Robot 1 nalezen a pridan   cas: 21.04.2017 16:21:05:340;
Nove nalezeni: 1
//periodicke dotazovani robota na stav
Dotazuji se robota 1   cas: 21.04.2017 16:21:07:584;
    ACK   cas: 21.04.2017 16:21:07:609;
robot 1 nema zadnou zadost   cas: 21.04.2017 16:21:07:767;

Dotazuji se robota 1   cas: 21.04.2017 16:21:08:576;
    ACK   cas: 21.04.2017 16:21:08:670;
robot 1 nema zadnou zadost   cas: 21.04.2017 16:21:08:822;
//robot ma zadost na lokalizaci
Dotazuji se robota 1   cas: 21.04.2017 16:21:13:598;
    ACK   cas: 21.04.2017 16:21:13:698;
Robot 1 zada o lokalizaci   cas: 21.04.2017 16:21:13:862;
//robot zada polohy vsech robotu
robot 1 zada o polohy robotu   cas: 21.04.2017 16:21:23:821;
Posilam robotovi 1, ze mu poslu 1 poloh   cas: 21.04.2017 16:21:23:870;
    ACK   cas: 21.04.2017 16:21:23:977;
Posilam robotovi 1, ze jsem mu prenesl vsechny polohy   cas: 21.04.2017
16:21:23:902;
    ACK   cas: 21.04.2017 16:21:24:074;
//robot se uspesne lokalizoval a posila svou polohu
Dotazuji se robota 1   cas: 21.04.2017 16:21:25:638;
    ACK   cas: 21.04.2017 16:21:25:704;
Robot 1 posila svou polohu: 594, 357   cas: 21.04.2017 16:21:25:884;
//vypis
V pameti 1 robotu   cas: 21.04.2017 16:23:42:473;
Aktivni roboti: 1
Neaktivni roboti:
```

Následuje výpis z prvku typu robot. Oba dva uvedené výpisy byly zaznamenávány simultánně, tudíž si časové údaje odpovídají a je možné sledovat vzájemný „dialog“ obou prvků.

```
//hledani dostupnych robotu
Master 0 hleda roboty, odpovidam   cas: 21.04.2017 16:21:05:328;
    ACK   cas: 21.04.2017 16:21:05:340;
//periodicke dotazovani od master veze na stav robota
Master 0 se dotazuje na stav   cas: 21.04.2017 16:21:07:634;
odpovidam M0: zadna zadost   cas: 21.04.2017 16:21:07:634;
    ACK   cas: 21.04.2017 16:21:07:754;
```

```
Master 0 se dotazuje na stav    cas: 21.04.2017 16:21:08:670;
odpovidam M0: zadna zadost    cas: 21.04.2017 16:21:08:695;
    ACK    cas: 21.04.2017 16:21:08:822;
//robot ma zadost o lokalizaci
Master 0 se dotazuje na stav    cas: 21.04.2017 16:21:13:698;
odpovidam M0: zadost o lokalizaci    cas: 21.04.2017 16:21:13:723;
    ACK    cas: 21.04.2017 16:21:13:862;
//robot zada polohy vsech robotu
Master 0 se dotazuje na stav    cas: 21.04.2017 16:21:23:655;
odpovidam M0: zadost o polohy robotu    cas: 21.04.2017 16:21:23:688;
    ACK    cas: 21.04.2017 16:21:23:821;
Master 0 posila 1 poloh    cas: 21.04.2017 16:21:23:977;
Poloha robota 1: 35, 474
Master 0 poslal 1 poloh    cas: 21.04.2017 16:21:24:040;
//robot se uspesne lokalizoval a posila svou polohu
Master 0 se dotazuje na stav    cas: 21.04.2017 16:21:25:728;
odpovidam M0: uspesna lokalizace, ma poloha 594, 357    cas: 21.04.2017
16:21:25:777;
    ACK    cas: 21.04.2017 16:21:25:884;
```

## 7 ZÁVĚR

Práce se zabývá návrhem a realizací komunikace mezi mobilními autonomními roboty a lokalizačními body, které slouží k určení polohy robotů v prostoru. Samotný proces lokalizace mobilního prvku není předmětem této práce. Cílem je umožnit přenos informací spojený s tímto lokalizačním procesem.

Určení polohy robota spočívá ve vysílání dvou druhů signálu, který se skládá ze světelných paprsků a ultrazvukových vln, lokalizačními stanicemi. Ze znalostí o polohách stacionárních bodů a rychlosti šíření obou signálů v prostoru je určena vzdálenost od konkrétní věže. Po opakování tohoto procesu je umožněno určit pozici robota v prostoru. Z důvodu možné interference ultrazvukových vln a následně chybných výsledků je nutné, aby lokalizační sekvenci nevysílalo více věží naráz. Pro tento účel je ustanovena komunikační síť mezi lokalizačními stanicemi. Věže jsou uspořádány dle schématu master-slave a zvolená nadřazená stanice kontroluje proces vysílání lokalizačních sekvencí tak, aby podřízené věže vysílaly v předem daném pořadí a pouze v případě, že předchozí věž již ukončila svou činnost. Přenos informací mezi stacionárními body je realizován ve třech variantách, a tudíž umožňuje zvolení vhodného řešení pro konkrétní aplikaci. Výsledná činnost lokalizačních věží je totožná. Komunikace s mobilními roboty je realizována pomocí bezdrátové komunikační technologie IQRF a probíhá pouze s hlavní stanicí. Nadřazená stanice se periodicky dotazuje jednotlivých mobilních prvků na jejich stav a dostává odpověď vyjadřující aktuální status či požadavek daného robota.

Vzhledem k cílům práce byl proveden návrh a realizace bezdrátové komunikace. Dále bylo realizováno potvrzování zpráv, o které se stará program bezdrátových IQRF modulů. Pro každou variantu komunikace mezi lokalizačními stanicemi (vizte kap. 5.1.2) byl vytvořen samostatný projekt v prostředí mbed. Jedná se o propojení věží do logického kruhu, pomocí sběrnice I<sup>2</sup>C a pomocí bezdrátové technologie IQRF. Pro účely práce byl vyvinut vlastní diagnostický software pro prostředí Windows (C#) a Android (Java). Program pro OS Windows ukládá data na disk zařízení pro umožnění následné kontroly záznamu. Aplikace pro mobilní zařízení Android je určena pro kontrolu aktuálního stavu prvku pomocí chytrého telefonu či tabletu. Pro ověření funkcionality bylo provedeno testování časového průběhu a dosahu bezdrátové komunikace. Dále byl realizován experiment v podobě simulace reálné situace s několika prvky.

Výsledné realizované řešení umožňuje libovolné připojování robotů do sítě. V případě nového přidání robota či ztráty spojení, například vzdálením z dosahu master věže, je mobilním prvkům umožněno opětovné navázání komunikace. Stacionární body dle potřeby vysílají lokalizační sekvenci pro určení pozice mobilních prvků. Dále je umožněno sdílení poloh robotů v podobě shromažďování posledních známých pozic všech robotů hlavní stanicí a jejich distribuci mezi mobilní prvky. Nedílnou součástí řešení je i řízení lokalizačních věží, aby nedocházelo k jejich vzájemnému rušení. Pro snížení spotřeby energie jsou prvky v době své nečinnosti uvedeny do stavu snížené spotřeby energie.

Do budoucna je možné řešení rozšířit například o sdílení poloh lokalizačních věží, které jsou v současnosti definovány přímo v paměti robotů. Pro výraznější navýšení

funkcionality je vhodné zvolit vývojovou desku s rozměrnější Flash pamětí, jelikož byl při vývoji dosažen limit velikosti uživatelského programu.

## 8 SEZNAM POUŽITÉ LITERATURY

- [1] ŠEBESTA, Jiří. *Globální navigační systémy*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2012. ISBN 978-80-214-4500-0.
- [2] mbed [online]. 2016 [cit. 2017-4-21]. Dostupné z: <https://www.mbed.com/en/>
- [3] mbed LPC1768. *mbed* [online]. [cit. 2017-4-21]. Obrázek. Dostupné z: <https://developer.mbed.org/media/img/boardlogos/lpc1768/pinout.png>
- [4] Boards. *mbed* [online]. [cit. 2017-4-21]. Dostupné z: <https://developer.mbed.org/platforms/>
- [5] mbed Compiler. *mbed* [online]. [cit. 2017-4-21]. Dostupné z: <https://developer.mbed.org/compiler/>
- [6] MEINDL, Jan. *Návrh a realizace vestavěného systému řízení mobilního robotu BENDER II* [online]. Brno, 2015 [cit. 2017-4-21]. Dostupné z: [https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/40811/2015\\_BP\\_Meindl\\_Jan\\_145538.pdf?sequence=1&isAllowed=y](https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/40811/2015_BP_Meindl_Jan_145538.pdf?sequence=1&isAllowed=y). Bakalářská práce.
- [7] VÝBORNÝ, Filip. *Bezdrátová senzorová síť z komponent Arduino* [online]. Brno, 2012 [cit. 2017-4-21]. Dostupné z: <https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/55303/12699.pdf?sequence=1&isAllowed=y>. Bakalářská práce.
- [8] Arduino UNO & Genuino UNO. *Arduino* [online]. [cit. 2017-4-21]. Obrázek. Dostupné z: [https://www.arduino.cc/en/uploads/Main/A000066\\_iso\\_both.jpg](https://www.arduino.cc/en/uploads/Main/A000066_iso_both.jpg)
- [9] Arduino Products. *Arduino* [online]. [cit. 2017-4-21]. Dostupné z: <https://www.arduino.cc/en/Main/Products>
- [10] Software. *Arduino* [online]. [cit. 2017-4-21]. Dostupné z: <https://www.arduino.cc/en/Main/Software>
- [11] ČERMÁK, David. *Návrh jednodeskového řídicího systému pro model vozidla pohybujícího se v autonomním konvoji* [online]. Brno, 2013 [cit. 2017-4-21]. Dostupné z: [https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/24721/cermak\\_david\\_bp\\_hlavni\\_dokument.pdf?sequence=1&isAllowed=y](https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/24721/cermak_david_bp_hlavni_dokument.pdf?sequence=1&isAllowed=y). Bakalářská práce.
- [12] JANÁS, Ondřej. *Zařízení pro sběr dat s Raspberry Pi* [online]. Brno, 2015 [cit. 2017-4-21]. Dostupné z: <https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/38964/BP.pdf?sequence=1&isAllowed=y>. Bakalářská práce.
- [13] Raspberry Pi Zero: The \$5 computer. *Raspberry Pi* [online]. [cit. 2017-4-21]. Dostupné z: <https://www.raspberrypi.org/blog/raspberry-pi-zero/>

- [14] Raspberry Pi Zero. *Raspberry Pi* [online]. [cit. 2017-4-21]. Obrázek. Dostupné z: <https://www.raspberrypi.org/app/uploads/2016/02/Raspberry-Pi-Zero-web.jpg>
- [15] Boards. *BeagleBoard* [online]. [cit. 2017-4-21]. Dostupné z: <https://beagleboard.org/boards>
- [16] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR ATmega16*. Praha: BEN - technická literatura, 2006. ISBN 80-7300-174-8.
- [17] PRAX, Jakub. *Zařízení Využití sběrnice I2C pro komunikaci s externím zařízením* [online]. Brno, 2007 [cit. 2017-4-21]. Dostupné z: <https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/25416/MMSE.pdf?sequence=2&isAllowed=y>. Diplomová práce.
- [18] ŠTRAUS, Pavel. *Vývoj sériových komunikačních periférií Pomocí FGPA* [online]. Brno, 2009 [cit. 2017-4-21]. Dostupné z: [https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/2907/Straus\\_Vyvoj\\_seriovych\\_komunikacnich\\_periferii\\_pomoci\\_FPGA.pdf?sequence=1&isAllowed=y](https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/2907/Straus_Vyvoj_seriovych_komunikacnich_periferii_pomoci_FPGA.pdf?sequence=1&isAllowed=y). Bakalářská práce.
- [19] GIERTL, Juraj. *Dekodér sériových sběrnic na platformě Arduino* [online]. Brno, 2016 [cit. 2017-4-21]. Dostupné z: <https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/61702/Dekoder%20seriovych%20zbernic.pdf?sequence=3&isAllowed=y>. Bakalářská práce.
- [20] ZEŽULKA, František. *Prostředky průmyslové automatizace*. Brno: VUTIMUM, 2004. ISBN 80-214-2610-1.
- [21] ONDRÁŠEK, Michael. *Simulátor a klient pro MODBUS zařízení* [online]. Brno, 2013 [cit. 2017-4-21]. Dostupné z: [https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/27634/BP\\_Ondr%C3%A1%C5%A1ek\\_Michael\\_2013\\_el\\_verze.pdf?sequence=2&isAllowed=y](https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/27634/BP_Ondr%C3%A1%C5%A1ek_Michael_2013_el_verze.pdf?sequence=2&isAllowed=y). Bakalářská práce.
- [22] IQRf - Technology for wireless [online]. [cit. 2017-4-21]. Dostupné z: <http://www.iqrf.org/>
- [23] IQRf Alliance [online]. [cit. 2017-4-21]. Dostupné z: <http://www.iqrfalliance.org/>
- [24] VYMĚTAL, Jan. *Bluetooth GPS Logger* [online]. Brno, 2009 [cit. 2017-4-21]. Dostupné z: <https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/52836/8093.pdf?sequence=1&isAllowed=y>. Bakalářská práce.
- [25] KNYTL, Marek. *Informační moduly komunikující přes Bluetooth* [online]. Brno, 2011 [cit. 2017-4-21]. Dostupné z: <https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/52968/12468.pdf?sequence=1&isAllowed=y>. Bakalářská práce.



- [26] QUADRI, Ishaq. Unwire yourself with Bluetooth Technology. *Binary Spectrum* [online]. [cit. 2017-05-11]. Obrázek. Dostupné z: <http://www.binaryspectrum.com/clients/case-studies/bluetooth.htm>
- [27] ZigBee alliance. *ZigBee* [online]. [cit. 2017-4-21]. Dostupné z: <http://www.zigbee.org/zigbeealliance/our-members/>
- [28] BHUMIKA, Mukesh KUMAR, Sudhir Narayan SINGH a Bhanuj TANEJA. Relative Study of Zig-Bee/IEEE 802.15.4 WPAN Performance under Different Topologies. In: *2015 Fifth International Conference on Advanced Computing*. IEEE, 2015, s. 415-420. DOI: 10.1109/ACCT.2015.129. ISBN 978-1-4799-8488-6. Dostupné také z: <http://ieeexplore.ieee.org/document/7079119/>
- [29] ŠAFAŘÍK, Jan. *Určování polohy Wi-Fi zařízení* [online]. Pardubice, 2010 [cit. 2017-4-21]. Dostupné z: [https://dk.upce.cz/bitstream/handle/10195/37358/SafarikJ\\_UrcovaniPolohy\\_ZN\\_2010.pdf?sequence=1&isAllowed=y](https://dk.upce.cz/bitstream/handle/10195/37358/SafarikJ_UrcovaniPolohy_ZN_2010.pdf?sequence=1&isAllowed=y). Diplomová práce.
- [30] MYSLÍK, Vladimír, Jan ŘEHÁK. IrDa - Kompletní popis. *Vývoj.HW.cz* [online]. [cit. 2017-4-21]. Dostupné z: <http://vyvoj.hw.cz/teorie-a-praxe/dokumentace/irda-kompletni-popis.html>
- [31] RŮŽIČKA, Michal. *Návrh a konstrukce majáků pro indoor navigaci mobilních robotů* [online]. Brno, 2010 [cit. 2017-4-21]. Dostupné z: [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=28611](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=28611). Bakalářská práce.
- [32] *STM32 Nucleo-32 board User manual* [online]. 2016 [cit. 2017-4-21]. Dostupné z: [http://www.st.com/content/ccc/resource/technical/document/user\\_manual/e3/0e/88/05/e8/74/43/a0/DM00231744.pdf/files/DM00231744.pdf/jcr:content/translations/en.DM00231744.pdf](http://www.st.com/content/ccc/resource/technical/document/user_manual/e3/0e/88/05/e8/74/43/a0/DM00231744.pdf/files/DM00231744.pdf/jcr:content/translations/en.DM00231744.pdf)
- [33] *STM32F303x6/x8* [online]. 2016 [cit. 2017-4-21]. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/group3/38/c6/ca/d1/f7/d5/4e/30/DM00092070/files/DM00092070.pdf/jcr:content/translations/en.DM00092070.pdf>
- [34] *(DC)TR-72D RF Transceiver Module Data Sheet* [online]. 2017 [cit. 2017-4-21]. Dostupné z: [iqrf.org/weben/downloads.php?id=337](http://iqrf.org/weben/downloads.php?id=337)
- [35] FREČER, Petr. *Radiokomunikační monitor pro ISM pásmo 868 MHz* [online]. Brno 2012 [cit. 2017-4-21]. Dostupné z: [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=51596](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=51596). Diplomová práce.
- [36] *IQRF OS Operatin Sysem* [online]. 2017 [cit. 2017-4-21]. Dostupné z: [iqrf.org/weben/downloads.php?id=155](http://iqrf.org/weben/downloads.php?id=155)

- [37] *SPI Implementation in IQRF For (DC)TR-7xD Technical guide* [online]. 2016 [cit. 2017-4-21]. Dostupné z: [www.iqrf.org/weben/downloads.php?id=85](http://www.iqrf.org/weben/downloads.php?id=85)
- [38] MANN, Burkhard. *C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMELE AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky ..* Praha: BEN - technická literatura, 2003.  $\mu$ C & praxe. ISBN 80-730-0077-6.
- [39] Examples for TR, StartUp Package. *IQRF* [online]. 2015 [cit. 2017-4-21]. Dostupné z: <http://iqrf.com/support/code-examples/for-tr>

## 9 SEZNAM PŘÍLOH

|           |    |  |
|-----------|----|--|
| Příloha 1 | CD | <ol style="list-style-type: none"><li>1. DP – elektronická verze práce</li><li>2. mbed – zdrojové kódy pro vývojové kity Nucleo v prostředí mbed</li><li>3. IQRf – zdrojový kód pro bezdrátové IQRf moduly</li><li>4. IQRfStartUpPackage – dokumentace a ukázkové kódy od výrobce bezdrátových modulů</li><li>5. Eagle – schéma a výkres konektoru pro připojení bezdrátového modulu v programu EAGLE</li><li>6. Software – zdrojové kódy diagnostického softwaru pro systém Windows a Android v prostředí Visual Studio Community 2015 a Android Studio</li></ol> |
|-----------|----|--|