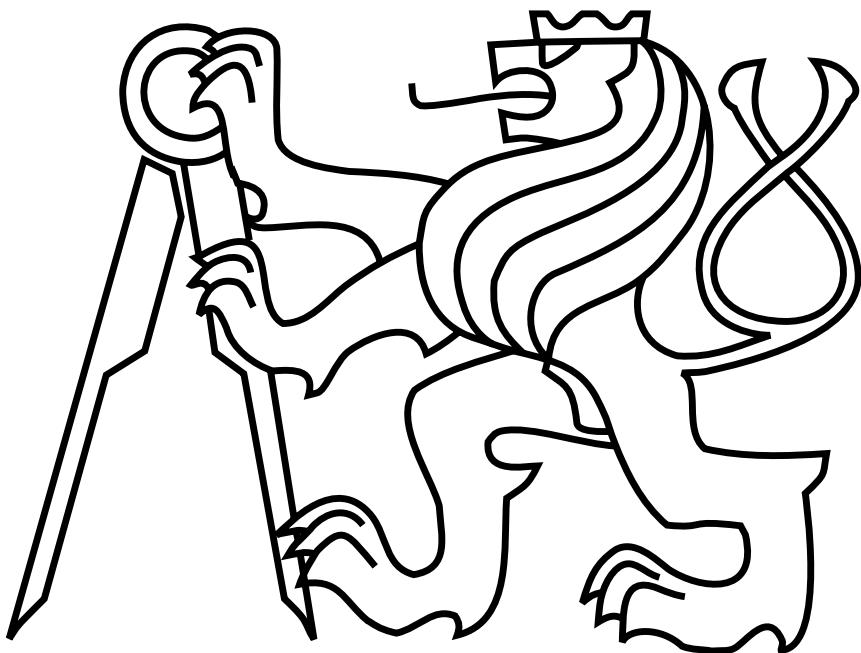


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická  
Katedra řídicí techniky



Bakalářská práce

**BEZDRÁTOVÝ SBĚR DAT  
ZE SÍTĚ BEZDOTYKOVÝCH ČTEČEK**

Praha, 2013

Autor: Ondřej Vinkler

České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra řídicí techniky

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Ondřej Vinkler**

Studijní program: Kybernetika a robotika  
Obor: Systémy a řízení

Název tématu: **Bezdrátový sběr dat ze sítě bezdotykových čteček**

Pokyny pro vypracování:

1. Seznamte se s bezdrátovou platformou IQRF respektive s jejím operačním systémem, možnostmi programování a proprietárním komunikačním protokolem.
2. Prostudujte protokol pro řízení a komunikaci s terminálem pro kontrolu přístupu HotMAX 914.
3. Zprovozněte komunikaci mezi modulem IQRF a přípravkem HotMAX 914 pomocí protokolu UART (RS-232) a realizujte program do modulu IQRF pro vyčítání specifických dat z jednotky HotMAX 914 spojené s RFID čtečkou.
4. Realizujte síť (včetně odpovídajícího programového vybavení) typu master/slave IQRF jednotek s připojenými přípravky HotMax 914 a RFID čtečkami.
5. Realizujte program na PC (včetně SQL databáze), který bude přes síť jednotek IQRF vyčítat RFID data z terminálů HotMAX 914 a ukládat je do databáze.

Seznam odborné literatury:

- [1] podklady k platformě IQRF – popis operačního systému, komunikačního protokolu a všech modulů
- [2] podklady ke komunikačnímu protokolu UART (RS-232) pro jednotku HotMAX 914
- [3] [www.iqrf.eu](http://www.iqrf.eu), [www.ivar.cz](http://www.ivar.cz)

Vedoucí: Ing. Ondřej Nývlt

Platnost zadání: do konce zimního semestru 2012/2013

prof. Ing. Michael Šsbek, DrSc.  
vedoucí katedry



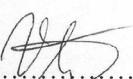
*M. Šebek*  
prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 10. 11. 2011

# PROHLÁŠENÍ

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne ..... 20.5.2013

.....  


Podpis

# **PODĚKOVÁNÍ**

Rád bych poděkoval vedoucímu své bakalářské práce Ondřeji Nývltovi, za jeho pomoc při konzultacích. Dále bych chtěl poděkovat firmě Microrisc a jejich support, který vždy pomohl, když se vyskytl nějaký problém při programování bezdrátové sítě. Velký dík patří firmě Ivar, za dodání hardwarových komponent a podporu při práci s terminálem 914 HotMAX. V neposlední řadě bych rád poděkoval všem, kteří mě podporovali při studiu a tvorbě této práce, zejména mým rodičům.

# **ABSTRAKT**

V této bakalářské práci je navržen a realizován systém, pro sběr informací z bezdrátové sítě bezdotykových čteček. Základem systému je bezdrátová síť s hvězdicovou topologií s hlavním uzlem typu koordinátor a ostatními uzly typu node. K jednotkám node je přes zařízení 914 HotMAX připojená bezdotyková čtečka. Koordinátor je propojený s řídicím počítačem pomocí rozhraní RS-232. Síť a všechna ostatní spojení musela být naprogramována tak, aby při komunikaci nedocházelo ke kolizím packetů. Pro řídící počítač byl navržen program, který umožňuje komunikaci s koordinátorem a ukládá příchozí data z bezdotykových čteček do MySQL databáze. V laboratorním prostředí je systém funkční, ale pro instalaci do reálných podmínek je třeba provést další testování a malé hardwarové úpravy.

# **ABSTRACT**

In this bachelor thesis a system, that can collect data from a wireless network of proximity readers, is designed and programmed. The core of the system is a wireless network which has a star topology with main node called coordinator and other nodes. The proximity readers are connected to nodes using a device called 914 HotMAX. The coordinator is connected to a computer using a RS-232 interface. The network and all other connections are programmed to avoid a packet collision during the communication. A program for main computer realises a communication with the coordinator and sends incoming data from contactless readers to a MySQL database. The system is fully functional in a laboratory environment, but for a installation in real conditions it needs more testing and some minor hardware adjustments.

# OBSAH

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Cíl práce	1
<b>2</b>	<b>Použité technologie</b>	<b>3</b>
2.1	Komunikační protokoly a rozhraní	3
2.1.1	RS-232	3
2.1.2	Universal asynchronous receiver transmitter	5
2.1.3	Serial peripheral interface	5
2.1.4	Bezdrátová komunikace IQRF	6
2.2	Hardwarové komponenty	7
2.2.1	914 HotMAX	7
2.2.1.1	RFID čtečka	8
2.2.1.2	LCD displej	9
2.2.2	IQRF RF moduly	10
2.2.2.1	TR-31B	11
2.2.2.2	TR-53B	12
2.2.2.3	TR-52D	12
<b>3.</b>	<b>Návrh a realizace systému</b>	<b>15</b>
3.1.	Topologie sítě	15
3.2.	Datové toky	15
3.3.	Koordinátor	16
3.4	Node	18
3.5	Propojení Node a 914 HotMAX	18
3.6	Propojení Koordinátora a PC	21
3.7	RS-232 na IQRF modulech	22
3.8	Testování	23
3.9	Software pro PC	24
3.9.1	Připojení k sériovému portu	25
3.9.2	Odesílání zprávy	25
3.9.3	Příjem zprávy	25
3.9.4	MySQL databáze	26
<b>4.</b>	<b>Závěr</b>	<b>27</b>

# SEZNAM OBRÁZKŮ

Obrázek 1.1 Zapojení systému[22]	2
Obrázek 2.1 D-Sub DE-9M konektor a rozložení pinů[2]	4
Obrázek 2.2 Zapojení na sběrnici SPI[5]	5
Obrázek 2.6 914 HotMAX	8
Obrázek 2.3 RFID čtečka karet	9
Obrázek 2.4 LCD displej	10
Obrázek 2.5 Programator CK-USB-02	10
Obrázek 2.6 Terminál IQRF IDE4	11
Obrázek 2.7 RF modul TR-52D	13
Obrázek 3.1 Zapojení systému a datové toky[22]	16
Obrázek 3.2 Kabel pro sériovou komunikaci IQRF	19
Obrázek 3.3 Převodní USB/RS-232	21
Obrázek 3.4 Terminál pro UART komunikaci	24
Obrázek 3.5 Grafické rozhraní programu	25

# **SEZNAM TABULEK**

Tabulka 2.1 Základní zapojení RS-232	3
Tabulka 2.2 Paměť a registry RF komunikace	6
Tabulka 2.3 Základní příkazy pro zařízení 914 HotMAX	7
Tabulka 3.1 Nastavení fukce setRFmode()	17
Tabulka 3.2 Příkazy pro node	18
Tabulka 3.3 Legenda k popisu registrů	19
Tabulka 3.4 Registr TXSTA	20
Tabulka 3.5 Registr RCSTA	20
Tabulka 3.6 Registr T6CON	21
Tabulka 3.7 Funkce pro práci s rozhraním UART	22

# **SEZNAM PŘÍLOH**

A        Obsah přiloženého CD



# 1 ÚVOD

V dnešní době stále více vznikají významné identifikační technologie. Na všemožných místech se můžeme setkat se vstupy do budov, vjezdy, výjezdy, přihlašováním do systémů, zapínáním, vypínáním systémů pomocí čipů. Stačí se podívat do budovy ČVUT na Karlově náměstí, kde je vstup možný jen pomocí karty s čipem. Čipem se zde také otevírají téměř jakékoli dveře. Velmi se také rozvíjejí systémy inteligentních budov. Nejenže místo klíče se dá používat karta, ale také jsou na systém často napojeny další bezpečnostní prvky. Dalším dobrým příkladem jsou bazény, které mají skříňky na čipy. Všechny tyto systémy fungují na nějaké sběrnici, díky které sbírají data ze všech koutů budovy. Bohužel vždy nelze spojit všechny části systému kabelovým spojem. Kvůli tomu se i v této oblasti začaly využívat bezdrátové technologie, které pomůžou překlenout pro kabely neprostupný terén.

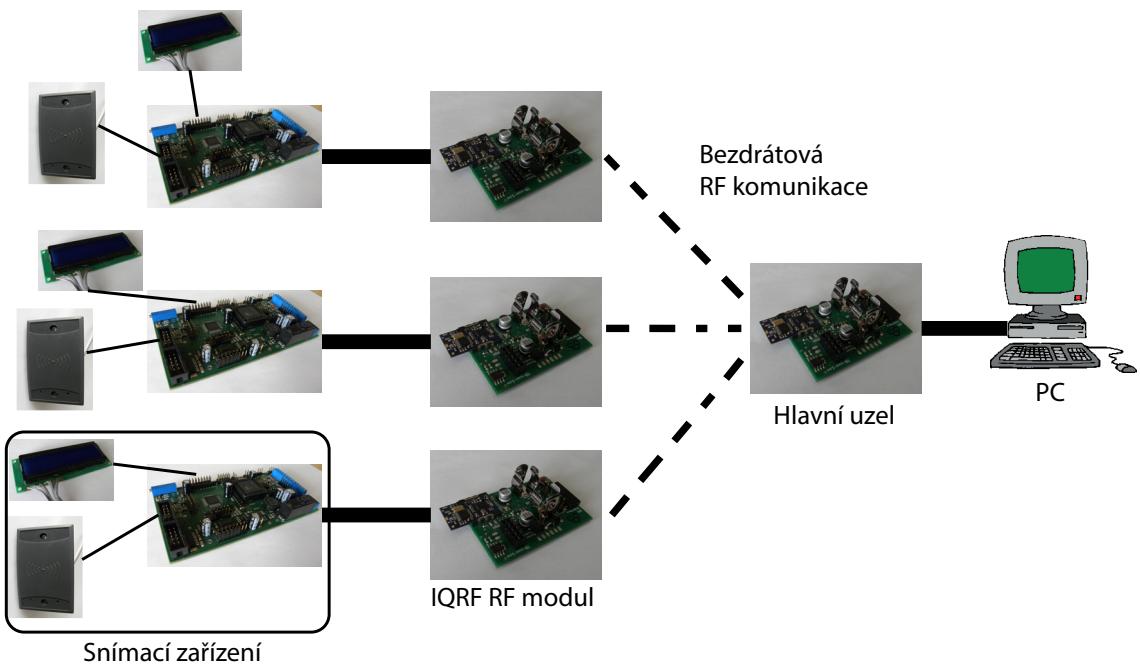
Právě z prostředí plaveckých bazénů vzešel nápad, realizovat systém, který by mohl plavcům sledovat jejich výkony. Představme si, že by systém fungoval takto: Plavec, který má na ruce připevněný čip, by si mohl po každém uplavaném bazénu „pípnout“ a přístroj by mu zaznamenal počet uplavaných kol i čas, za který je uplaval. K tomu je zapotřebí mít na jedné straně bazénu v každé dráze čtečku čipů. Pro bazén v provozu je téměř nemožné nainstalovat tento systém pomocí kabelového připojení. Bazén by se musel na několik dní celý zavřít, aby mohly proběhnout příslušné úpravy v rozvodech kabelů. Naštěstí lze tento problém vyřešit bezdrátovým spojením. Při návrhu takového systému se musí počítat s tím, že snímací přístroj bude napájený akumulátorem. Proto musí bezdrátový vysílač mít malou spotřebu elektrické energie.

V budoucnosti by se systém mohl rozšířit a začít využívat ve všech systémech, jejichž instalace by nedovolovala využít kabelového spoje. Příkladem může být dům, do kterého se instaluje přístupový systém. Před domem je rozsáhlý betonový dvůr se vstupními vraty. Kabelové propojení by bylo velice obtížné, a proto je daleko výhodnější použít bezdrátové spojení. Rozsah použití takového systému, který by šlo napojit na sběrnicový systém, by byl ještě daleko větší než ten bazénový, protože všude, kde by nebylo výhodné realizovat kabelový spoj, by se využilo spojení bezdrátového.

## 1.1 CÍL PRÁCE

Tato práce si klade za cíl vytvořit funkční systém pro plavecké bazény, který by plavcům počítal uplavaná kola a čas, za který je uplavali. Základem bude bezdrátová síť realizovaná pomocí IQRF RF modulů[23]. Tyto moduly se vyznačují velice nízkou spotřebou, a proto jsou vhodné pro tento systém. K uzelům této sítě budou připojeny čtecí zařízení, které budou číst plavcův čip. V síti bude jeden hlavní uzel, který bude sbírat data od všech ostatních uzelů a bude je posílat do řídicího počítače. Veškerá komunikace musí být vytvořena v závislosti na sobě tak, aby nedocházelo ke ztrátám dat. V počítači poběží program, který bude data shromažďovat do databáze.

V kapitole 2 jsou popsány komunikační protokoly a základní komponenty, se kterými se v systému pracuje. V podkapitole 2.1 jsou popsány technické parametry fyzického zapojení se způsobem, kterým komunikace probíhá a jejich rozdíly. V podkapitole 2.2 budou popsány základní vlastnosti použitych komponent. Následuje kapitola 3, jejímž



Obrázek 1.1 Zapojení systému[22]

obsahem je popis návrhu a implementace systému. Je zde vysvětleno programování sítě, jednotlivých uzlů a jejich propojení. Jsou zde uvedeny obsahy důležitých registrů a nejpoužívanějších funkcí. Následuje podkapitola 3.8 o testování systému, včetně popisu problémů, které nastaly při vývoji a jejich řešení. Podkapitola 3.9 obsahuje princip a funkce programu PC, který sbírá data ze systému, včetně databáze, která tato data ukladá.

## 2 POUŽITÉ TECHNOLOGIE

Systém se skládá z mnoha komponent, které jsou propojeny různými komunikačními rozhraními. V této kapitole jsou uvedeny principy těchto rozhraní a technické specifikace použitých komponent. Znalost těchto vlastností pomáhá nejen k lepšímu pochopení funkčnosti celého systému, ale je nezbytná k práci s jednotlivými komponentami při jejich propojování mezi sebou.

### 2.1 KOMUNIKAČNÍ PROTOKOLY A ROZHRANÍ

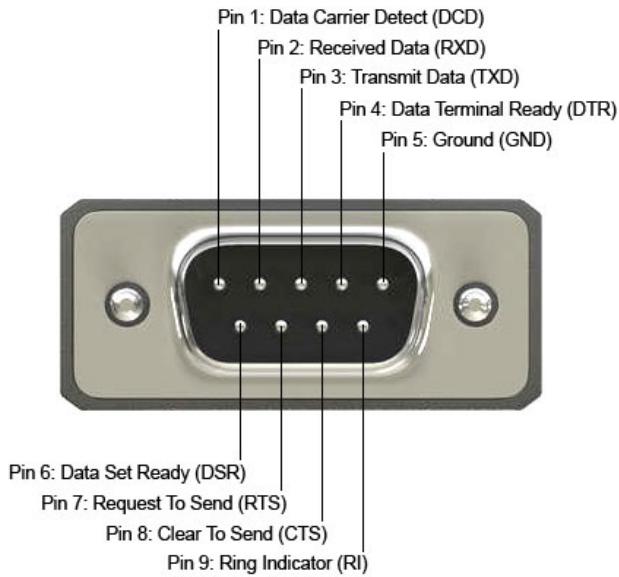
Komunikační protokoly, které byly využity při tvorbě systému, se mezi sebou velmi liší. Proto je v této podkapitole psáno o technických parametrech fyzického zapojení, včetně napěťových úrovních na sériových linkách, rozložení pinů na konektorech a jakým způsobem komunikace probíhá (aktivování komunikace, synchronizační a datové bity, ...). V systému jsou použity následující komunikační protokoly: RS-232, UART, SPI a Bezdrátová komunikace IQRF.

#### 2.1.1 RS-232

Standard RS-232[1] se používá jako komunikační rozhraní počítačů a další elektrotechniky. Umožňuje vzájemné propojení pouze dvou zařízení najednou tak, že se jednotlivé bity vysílají postupně za sebou. Data mohou proudit oběma směry zároveň, po nezávislých párech vodičů a proto jde o zcela bezkolizní komunikaci. Standard RS-232 se zabývá

Tabulka 2.1 Základní zapojení RS-232

Zkratka	Název	Funkce	Pin
TxD	Transmit data	Data posílaná z DTE do DCE	3
RxD	Receive data	Data příjmaná v DTE z DCE	2
RTS	Request to send	Logická jednička signalizuje, že DTE chce vysílat data	7
CTS	Clear to send	Logickou jedničkou protistrana signalizuje, že DTE může vysílat data	8
DSR	Data set ready	Logickou jedničkou protistrana signalizuje že je připravena. Neznamená to že DTE může vysílat data	6
GND	Ground	Signálová zem. Hodnoty napětí na pinech je určena vůči této zemi	5
DCD	Carrier detect	Protistrana signalizuje, že na vedení detekovala nosný signál	1
DTR	Data terminal ready	DTE signalizuje, že je připraven. Protistrana odpovídá nastavením DSR	4
RI	Ring indicator	signalizuje příchozí hovor do DTE	9



Obrázek 2.1 D-Sub DE-9M konektor a rozložení pinů[2]

pouze tím, jak přenést určitou sekvenci bitů, proto se jedná pouze o fyzickou vrstvu. Linka většinou bývá vyvedena pomocí konektoru D-Sub DE-9. Je to devítí pinový konektor rozlišovaný jako samec (M) a samice (F). Minimální zapojení je třívodičové (příjem (RXD), vysílání (TXD) a společná zem (GND)). K nim může být přidáno až 6 dalších vodičů sloužících k řízení přenosu, případně mohou být využity jako napájení přístroje. V tabulce 2.1 je uvedeno kompletní standartní devítivodičové zapojení na konektor D-Sub DE-9. DTE, neboli Data terminal equipment, je většinou terminál nebo počítač emulující terminál. Má konektor typu samec. DCE, neboli Data circuit-terminating equipment, může být například modem, který má konektor typu samice. Rozdělení DTE/DCE bylo poprvé představeno firmou IBM. Na obrázku 2.1 je vidět standartní samčí konektor spolu s popisem jednotlivých pinů.

Kvůli asynchronní komunikaci je potřeba mít v přenosu synchronizační byty. Zde se používá start bit, který přepne napěťovou úroveň z klidové hladiny do hladiny opačné. Poté následují datové byty, kterých je nejčastěji 8, ale výjimečně jich může být i 7 nebo 9. Poté následuje paritní bit, pokud je zvolen a nakonec jeden nebo více stop bitů, které označují konec zprávy. Datové byty se postupně vysílají od nejméně významného bitu (LSB) po nejvíce významný bit (MSB).

Na datových linkách je logický stav bitu určován podle napěťové úrovně, která je vztahována ke společnému zemnícímu vodiči. RS-232 není odolný vůči vnějšímu rušení a vyžaduje, aby měl na obou stranách kabelu zemnící vodič stejný potenciál. Pokud se na jedné straně změní zemní potenciál, začne zemnícím vodičem protékat proud. Pokud proud nestačí rozdíl potenciálů vyrovnat, objeví se tato změna na datových vodičích jako falešná informace. Nejpoužívanější napěťová úroveň je  $\pm 12V$ , ale můžeme se setkat i s úrovněmi  $\pm 5V$ ,  $\pm 10V$ , nebo  $\pm 15V$ . Kladné hodnoty se pak nejčastěji považují za logickou 0 a záporné za logickou 1. U sériových portů v počítačích lze dosáhnout baudové rychlosti až 115200 bd. Další rychlosti jsou odvozeny dělením. Nejčastěji používané jsou baudové rychlosti 57600 bd, 19200 bd, 9600 bd, 4800 bd a 2400 bd. Přenosová rychlosť je vždy menší než baudová, protože k datovým bitům se ještě musí přičíst start bit, parita a stop byty.

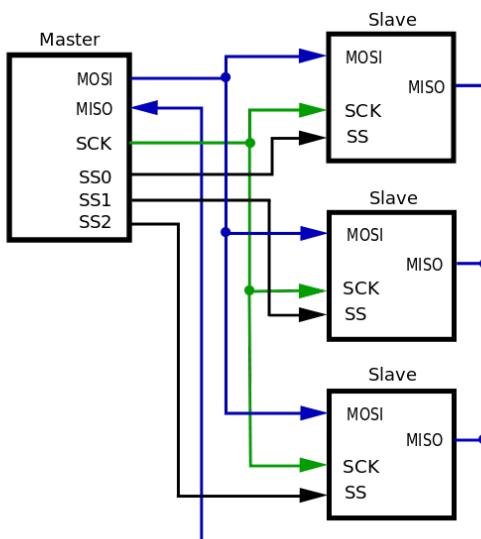
## 2.1.2 UNIVERSAL ASYNCHRONOUS RECIEVER TRANSMITTER

UART[3] nebo univerzální asynchronní vysílač a přijímač je rozhraní pro asynchronní sériovou komunikaci. Obvykle je to samostatný integrovaný okruh, který je nyní často zahrnutý v mikrokontrolerech a používá se pro komunikaci po linkách RS-232, RS-422 nebo RS-485. Po správném nastavení v řídících registrech je veškeré časování, počet bitů a parita stejné. Jediný rozdíl je v úrovních napětí. Zatímco běžná RS-232 nejčastěji používá napětí  $\pm 12V$ , tak UART používá jiné úrovně. Jako logická 0 je považováno napětí 0V a za logickou 1 je považováno napětí 3.3V. K převodu mezi nimi se musí používat převodník napětí. Mezi nejznámější integrované obvody, které poskytují tuto možnost převodu je např.: MAX232[25].

## 2.1.3 SERIAL PERIPHERAL INTERFACE

SPI[4] je sériové periferní rozhraní, které se využívá ke komunikaci mezi řídícími mikroprocesory a dalšími zařízeními jako např.: EEPROM, A/D převodník, atd. Je to sběrnicové rozhraní s jedním master zařízením a možností více slave zařízení. Ke komunikaci slouží 4 vodiče (obrázek 2.2). Vodič SCK je vyhrazený pro hodinový signál, který generuje master. Dva vodiče jsou využívány jako datové, přičemž MOSI je vždy master výstup a MISO je slave výstup. Poslední vodič SS, který slouží k adresaci zařízení. Pomocí tohoto vodiče se aktivuje příjem a vysílání vybraného zařízení a proto se jedná o synchronní komunikaci řízenou masterem. Ke každému slave zařízení vede jeden SS vodič. To znamená, že od masteru jich může vést více. Komunikace probíhá v takovémto pořadí:

- Na SS vodiči zařízení, se kterým chce master komunikovat, nastaví logickou 0
- Master začne vysílat hodinový signál a v té chvíli vyšlou obě strany svá data
- Následně mohou nastat dvě možnosti:
  - master dále dodává hodinový signál, hodnota SS se nemění a komunikace pokračuje
  - master přestane vysílat hodinový signál, hodnota SS se změní na logickou 1 a komunikace se ukončuje



Obrázek 2.2 Zapojení na sběrnici SPI[5]

## 2.1.4 BEZDRÁTOVÁ KOMUNIKACE IQRF

Bezdrátová komunikace v modulech IQRF[7] je z velké části podporována operačním systémem. To znamená, že uživatel bude pracovat primárně pouze s pamětí a buffery. IQRF operační systém zajišťuje všechna potřebná nastavení, včetně plné implementace protokolu:

- na přenosové úrovni - nastavení HW, kódování přenosu, ...
- na packetové úrovni - preamble, kódování, ...
- na síťové úrovni - informace o síti a zařízeních, filtrování, směrování, ...

Jsou podporována dvě nastavení bezdrátové komunikace:

- Peer-to-peer - propojení dvou nebo více zařízení bez řídicího uzlu. Všechny packety jsou dostupné pro všechna zařízení.
- IQMESH - topologie s jedním koordinátorem, který řídí síť až pro 65000 koncových zařízení. Ve směrovací struktuře může být 239 zařízení. Každé zařízení musí být připojeno do sítě. Peer-to-peer packety jsou také povoleny.

Paměť a registry, které se váží ke komunikaci jsou popsány v tabulce 2.2

RF vysílání:

Pro vysílání se nejprve musí připravit data do komunikačního bufferu pro bezdrátovou komunikaci a poté se nastaví délka odesílaných dat v registru DLEN. V případě IQMESH je potřeba nastavit příjemcovu adresu v registru TX. Poté se data už jednoduše odešlou pomocí funkce RFTXpacket(). V případě oboustranné komunikace musí být registry PIN a DLEN aktualizovány před každým odesláním.

RF příjem:

Pro příjem dat slouží funkce RFRXpacket(). Funkce se snaží přjmout packet dat a vrací kontrolu nad aplikací dokud ji úspěšně nepříjme nebo nevyprší timeout. Po přijetí packetu jsou data uložena do bufferu pro bezdrátovou komunikaci. Pro snížení spotřeby je dostupných několik módů:

- STD - standartní mód s nejvyšší spotřebou 13mA. Vysílač má preamble 3ms
- LP(low power) - modul je periodicky přepínaný mezi příjemem a spánkem. To vyústí v cykly o délce 46 ms. Spotřeba je snížena na 330uA. Vysílač zvyšuje preamble na 50ms

Tabulka 2.2 Paměť a registry RF komunikace

bufferRF[64]	buffer, se kterým se pracuje při RF komunikaci. Velikost 64B
PIN	informace o packetu: směrování, síťový mód(peer-to-peer nebo IQMESH), žádost o potvrzení přijetí a šifrování
RX	adresa příjemce, kterou je u IQMESH nutné specifikovat před odesláním
TX	adresa vysílače, která je u IQMESH automaticky aktualizována operačním systémem
DLEN	délka dat v packetu, 0-64B, nutné nastavit před odesláním packetu, automaticky aktualizováno po příjmu
toutRF	timeout pro přijetí packetu: doba, po kterou se funkce RFRXpacket() snaží přjmout packet

- XLP(extra low power) - modul je periodicky přepínáný mezi příjemem a spánkem. To vyúsťuje v cykly o délce 770 ms. Spotřeba je snížena na 25uA. Vysílač zvyšuje preamble na 1s
- RFIM(RF immunity mode) - Příjem je předčasně ukončen pokud síla signálu klesne pod určitou úroveň

Tyto módy musí být nastaveny tak, aby vzájemně odpovídaly mezi sebou u vysílače a příjmače. To znamená, že pokud je na jedné jednotce node nastavený LP mód vysílání, u druhé jednotky musí být nastavený LP mód příjmu. Komunikace v opačném směru může být nastavená do jiného módu, ale opět musí vzájemně odpovídat. Každý modul může být připojený do dvou sítí. V jedné bude figurovat jako obyčejný node a ve druhé musí být jako koordinátor. To umožňuje propojování jednotlivých sítí až do neomezeného počtu zařízení.

## 2.2 HARDWAROVÉ KOMPONENTY

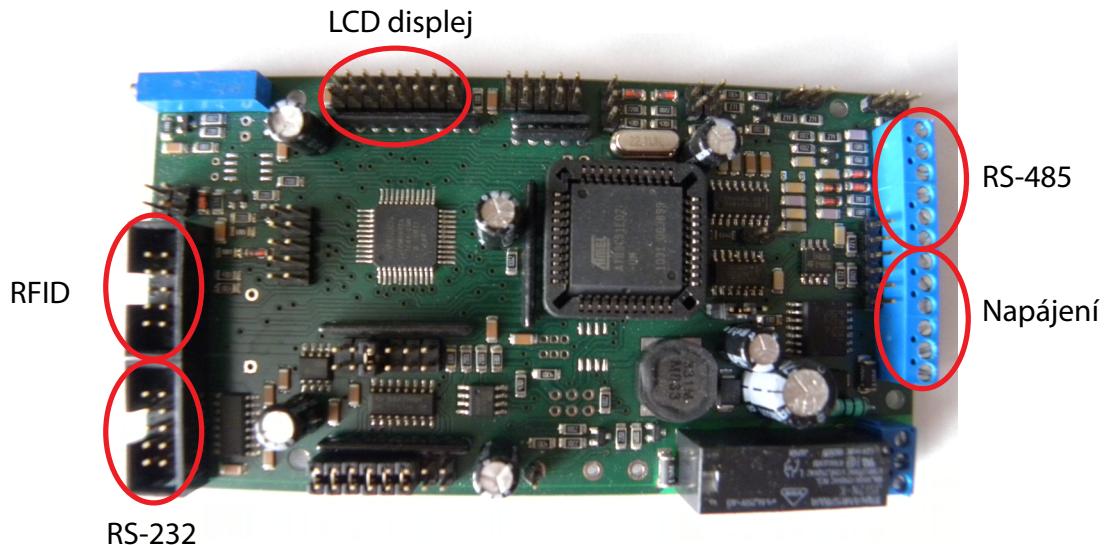
V této podkapitole se popisují jednotlivé komponenty, které jsou v systému použity. Je důležité znát jejich technické parametry pro lepší pochopení fungování systému. Bez těchto znalostí by vývoj systému nebyl možný. U některých komponent je uvedený princip funkčnosti. Popisují se také možnosti připojení, osazení, ovládání prvku a jeho využití.

### 2.2.1 914 HotMAX

914 HotMAX je základní přístupový terminál od firmy Ivar. Je osazený 8-bit flash mikrokontrolerem od firmy Atmel typu AT89C51ED2[15], ve kterém je nahraný vlastní systém. Zařízení je vysoce variabilní z hlediska zapojení a nastavení. Disponuje zapojením point-to-point pomocí RS-232 nebo zapojením do sběrnice RS-485 s protokolem NET92. Výběr zapojení se provádí příkazem, který se odešle do zařízení. RS-232 je možné nastavit hardwarově přímo na zařízení do dvou baudových rychlostí: 9600bd nebo

Tabulka 2.3 Základní příkazy pro zařízení 914 HotMAX

Příkaz	Odpověď	Funkce
%%	!!!	Restartování jednotky
%XI		Identifikace jednotky a její verze
%S	Sabcd	Stav vstupů jednotky: 0 - kontakt rozpojen, 1 - spojen a - 0 b - ochranný kontakt c - vstup IN1 d - vstup IN2
%h	+tt,tt	Teplota ve °C změřená čidlem DS1820
%L1		Pošle přesně 16 znaků na první řádek displeje
%L2		Pošle přesně 16 znaků na druhý řádek displeje



Obrázek 2.6 914 HotMAX

57600bd. Pro komunikaci RS-232 se používá integrovaný obvod SP202EEN od firmy Sipex[26]. K zařízení 914 HotMAX lze připojit RFID čtečku, dvourádkový, šestnáctimístný, alfanumerický LCD display a další komponenty jako například sirénu, anténu, maticovou klávesnici, LED diody a mnoho různých dalších typů čteček. Celé zařízení se ovládá speciálními příkazy, které jsou uvedeny v tabulce 2.3. Všechny příkazy se musí odesílat s koncovým znakem, který odpovídá hodnotě 0x0D.

### 2.2.1.1 RFID ČTEČKA

Technologie Radio Frequency Identification (RFID)[18], byla původně navržena k identifikaci zboží. Jde o bezkontaktní komunikaci na krátkou vzdálenost. Ke správné funkčnosti je zapotřebí čtečka a čipy, které existují ve dvou variantách:

- Pasivní čip:

RFID čtečka pravidelně vysílá elektromagnetické pulsy. Když se v blízkosti objeví pasivní RFID čip, využije energii pulsu k nabítí napájecího kondenzátoru a poté odešle odpověď. Pasivní čipy dokáží vyslat buď číslo určené při jejich výrobě, nebo mají ještě dodatečnou paměť, do které lze zapisovat a číst další data. Pasivní čipy se nejčastěji využívají k:

- Identifikaci zboží
- Řízení přístupu osob
- Bezhmotovostním platbám(elektronická peněženka)

- Aktivní čip:

Využívá se méně než pasivní systém. Je dražší a složitější, protože kromě antény a čipu obsahuje i zdroj energie. Je schopen sám vysílat svou identifikaci, a proto se využívají k aktivní lokalizaci.

K zařízení 914 HotMAX se používá čtečka karet MF7[19] od firmy Gigatek spolu s



Obrázek 2.3 RFID čtečka karet

pasivními RFID čipy, které dokáže číst na vzdálenost až 5cm. Čtečka je vysoce kvalitní a robustní a díky plastovému je obalu odolná proti povětrnostním vlivům. Napájecí napětí se může lišit od 5V do 18V. Typické napájení je 12V se spotřebou 150mA. Disponuje třemi různými komunikačními rozhraními, kterými se připojuje k dalším komponentám: RS-232, MSR ABA, Wiegand 26-bit. Čtečka umožňuje pouze čtení, proto nejsou potřeba žádné řídící příkazy ze zařízení 914 HotMAX. Čtečka pouze čte kódy čipů a posílá je do zařízení 914 HotMAX ve tvaru:

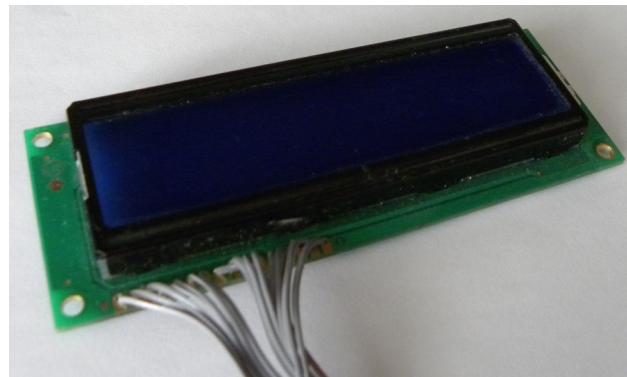
;123456789012345?>#010#013

Délka kódu se může lišit v závislosti na typu použitého čipu. #010 a #013 jsou decimální hodnoty posledních znaků.

### 2.2.1.2 LCD displej

LCD[16] je ploché zobrazovací zařízení, které využívá technologii tekutých krystalů. LCD je energeticky nenáročný, a proto se hodí do systémů, které jsou napájeny z baterií. Navíc nemá negativní vliv na lidský zrak. Zařízení se skládá z vrstvy molekul tekutých krystalů, která je uložená mezi průhlednými elektrodami a polarizačními filtry. Elektroda je sklo na jehož jedné straně je nanesený polarizační film a na druhé je nanesený speciální polymer, který vytváří mikroskopické drážky. Směr drážek musí být stejný jako směr polarizačního filtru. Avšak směry na obou elektrodách musí být vůči sobě pootočeny o 90°. Molekuly tekutých krystalů jsou bez elektrického pole ovlivněny směrem drážek, proto jsou uspořádány ve spirálovité struktuře. Když světlo prochází diplejem, nejprve se polarizuje na prvním polarizačním filtru. Díky tekutým krystalům se rovina polarizace stáčí, a když světlo dospěje ke druhému filtru, je rovina otočena o 90° a projde jím bez dalších ztrát. Celá soustava je průhledná. Po vzniku elektrické pole jsou molekuly taženy souběžně s ním, což snižuje míru otáčení polarizační roviny. Pokud krystaly nejsou vůbec stočené, bude polarizační rovina kolmo k druhému filtru a žádné světlo přes něj neprojde. Když je vrstva molekule tekutých krystalů stočena o 90° jedná se o TN(Twisted nematic) LCD, ale existují i jiné typy displejů s vrstvou, která je stočena o jiný úhel. Z hlediska podsvícení se LCD může dělit na tři druhy:

- Transmisivní - zdroj světla je umístěn na zadní straně



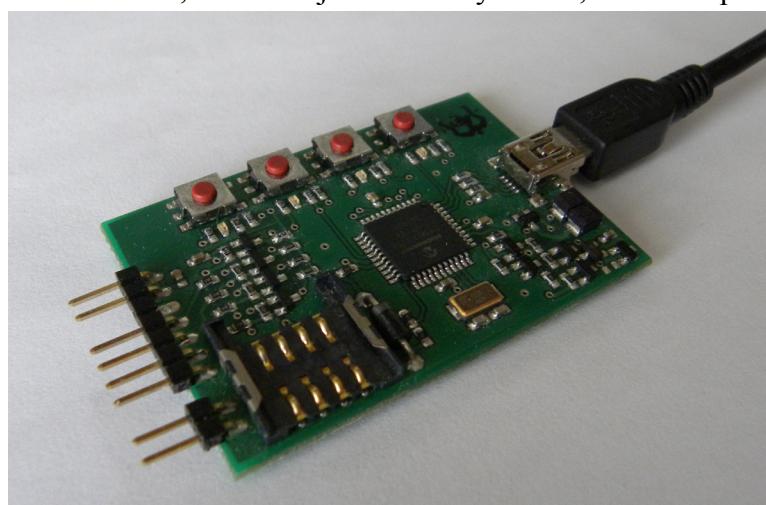
Obrázek 2.4 LCD displej

- Reflexní - na zadní straně je umístěna reflexní vrstva, která odráží příchozí světlo. Podsvícení je umístěno z boku displeje
- Kombinované - na zadní straně je umístěna vrstva, která z jedné strany světlo propouští a z druhé odráží. To umožňuje umístit podsvícení na zadní stranu, ale zároveň dobrou čitelnost displeje bez podsvícení s odrazem okolního světla.

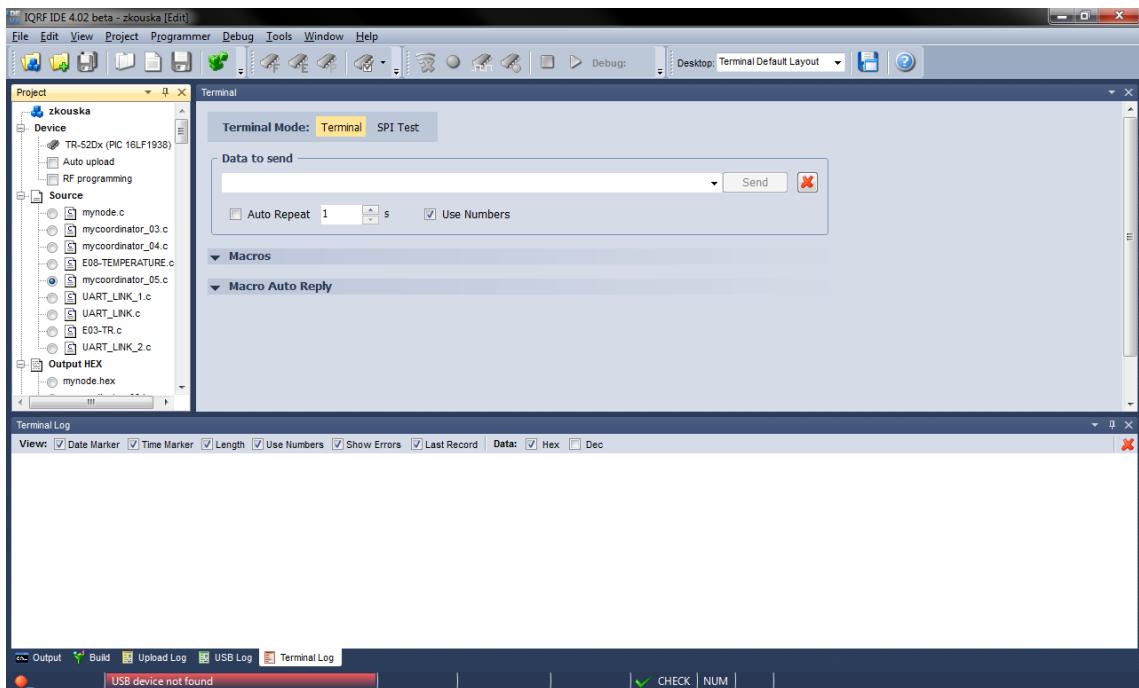
K zařízení 914 HotMAX se využívá LCD typu MC1602E[17]. Jedná se šestnáctimístný dvouřádkový alfanumerický displej s vlastním řadičem. Je to typ STN(Super twisted nematic), který má vrstvu molekul tekutých krystalů stočenou o 180-270°. To umožňuje větší pasivní-maticovou adresaci s větším počtem řádků a sloupců. Displej je transmisivní s modrým pozadím a bílým podsvícením. Pro ovládání displeje jsou v zařízení 914 HotMAX dva příkazy, které stačí k veškeré práci s displejem.

## 2.2.2 IQRF RF MODULY

IQRF moduly jsou zařízení pro bezdrátový přenos dat vyvinuté firmou Microrisc[23]. Mají širokou škálu využití od telemetrie, přes bezdrátové řízení a regulaci, po kontrolu přístupů a vzdálené získávání dat. Disponují vlastním zabudovaným operačním systémem se spoustou funkcí, které výrazně zjednodušují programování. Firma Microrisc vyvinula několik druhů RF modulů, které vzájemně liší výkonem, velikostí paměti a spotřebou



Obrázek 2.5 Programator CK-USB-02



Obrázek 2.6 Terminál IQRF IDE4

energie. Většina z nich je vyrobena ve formátu SIM karty. Při instalaci na jiná zařízení je možné na PCB přidělat pouze konektor na SIM karty. To má výhodu v tom, že modul není připevněný napevno a umožňuje jednoduché vyjmutí a výměnu za nový model (v případě upgradu nebo poškození starého). Firma Microrisc také dodává vývojové kity, na kterých lze testovat aplikace. Tyto kity neposkytují nic jiného než vyvedení pinů na konektory pro jednodušší zapojení a poskytnutí základního napájení (baterie nebo akumulátor). K programování jednotlivých modulů se využívá terminálu IQRF IDE4. Prostředí terminálu je vidět na obrázku 2.6. Před nahráním příslušného programu do RF modulu je nutné nejprve zkompilovat zdrojový kód. Při komplikaci je potřeba dbát na typ modulu a verzi operačního systému. Samotné nahrání do RF modulu může probíhat dvěma způsoby:

- Standartní - modul je do počítače připojen přímo přes USB port. Komunikace probíhá na protokolu SPI.
- RFPGM - bezdrátové nahrávání programu do modulu. Samotný modul není potřeba připojovat do počítače. Stačí, když je v dosahu jiného, do počítače připojeného RF modulu.

K připojení modulu do počítače pomocí USB portu se využívá zařízení CK-USB-02 (obrázek 2.5). V této práci se postupně pracovalo se třemi druhy modulů: TR-31B, TR-53B a TR-52D.

### 2.2.2.1 TR-31B

TR-31B[11] je nejstarší a nejméně výkonný z používaných modulů. Je osazen mikroprocesorem PIC16F886[24]. Pro RF komunikaci využívá integrovaný obvod TR1001. V otevřených prostorech má dosah do 100m, ale naproti tomu maximální spotřeba dosa-

huje 12,3 mA při vysílání na nejvyšší výkon. Obsahuje zabudovaný teplotní senzor MCP9700A, kterým lze snímat okolní teplotu. Vnitřní operační systém je verze 2.09. Flash paměť se využívá pro ukladání programu a má místo na 1024 instrukcí. Na RAM paměti je 98B pro uživatele a 140B v komunikačních bufferech. V EEPROM paměti je 32B určených k uživatelské konfiguraci, ale pokud je jednotka v režimu node je k dispozici dalších 160B. U koordinátora těchto 160B k dispozici není, protože je tam uložená topologie sítě. Komunikace po SPI je vyřešená speciálními funkcemi v OS. To neznamená, že není podporováno žádné další rozhraní. Komunikaci přes UART nebo I2C je možné realizovat uživatelskými funkcemi. V této verzi operačního systému není povolené uživatelské přerušení.

### 2.2.2.2 TR-53B

TR-53B[10] je novější a výrazně výkonnější než předchozí verze. Je také osazen mikroprocesorem PIC16F886[24], ale má rozšířenou FLASH paměť a přídavnou sériovou EEPROM paměť. Pro RF komunikaci využívá integrovaný obvod TRC102. V otevřených prostorech má dosah až 700m, ale spotřeba stoupla na 24mA při vysílání na nejvyšší výkon. Vnitřní operační systém je verze 3.00. Flash paměť se využívá pro ukládání programu a má místo na 1024 instrukcí. V RAM paměti je 40B pro uživatele a 172B v komunikačních bufferech. V EEPROM paměti je 32B určených pro uživatelskou konfiguraci, ale pokud je jednotka v režimu node je k dispozici dalších 160B. U koordinátora těchto 160B k dispozici není, protože je tam uložená topologie sítě. Komunikace po SPI je vyřešená speciálními funkcemi v OS. To neznamená, že není podporováno žádné další rozhraní. Komunikaci přes UART nebo I2C je možné realizovat uživatelskými funkcemi. V této verzi operačního systému není povolené uživatelské přerušení.

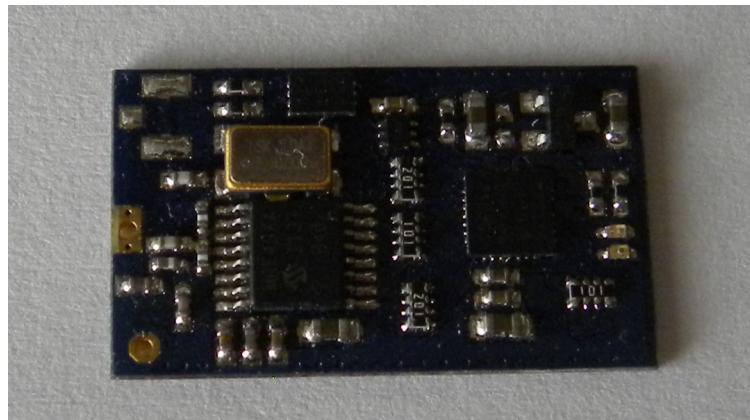
### 2.2.2.3 TR-52D

Nejnovější z používaných modulů[9] je osazený mikroprocesorem PCI16LF1938[14]. Pro RF komunikaci využívá integrovaný obvod MRF49XA[13], který má větší výkon a proto dokáže v otevřených prostorech udržet spojení až na vzdálenost 850m. Spotřeba při vysílání na nejvyšší výkon však zůstala na 24mA. Vnitřní operační systém je verze 3.02D.

Flash paměť se využívá pro uložení uživatelského programu a plug-inů, protože zůstává zachována i po vypnutí napájení. Počet přepsání paměti není neomezený a typicky dosahuje počtu 10000 přepsání. Flash paměť je rozdělena do tří oblastí:

- Oblast 0x3A00 - 0x3FFF - je určena pro uživatelský program, disponuje místem na 1536 instrukcí
- Oblast 0x3000 - 0x37E0 - je určena pro plug-iny, ale může být také využita jako dodatečné místo pro uživatelský program, disponuje místem na 2016 instrukcí
- Zbývající místo je vyhrazeno pro operační systém a jeho plug-iny

Paměť RAM je plně pod kontrolou běžícího programu. Její obsah se po vypnutí napájení ztrácí, proto se využívá pouze za běhu programu a ne jako dlouhodobé uložiště dat. Paměťové místo RAM je rozděleno na 32 RAM bank, každá po 128B. Některé registry



Obrázek 2.7 RF modul TR-52D

jsou přístupné přímo u dalších musí být nejprve vybrána příslušná RAM banka. RAM pamět je rozdělena na následující oblasti:

- bufferRF (0x4A0 - 0x4EF) - 128B - pro RF komunikaci, funkce RFTXpacket() a RFRXpacket pracují jen se 64B toho bufferu
- bufferCOM (0x3A0 - 0x3DF) - 64B - pro sériovou komunikaci (SPI, UART, ...)
- bufferINFO (0x320 - 0x35F) - 64B - pro operační systém, uživatelské blokové operace
- bufferAUX (0x420 - 0x45F) - 64B - pomocný buffer, vhodný pro dočasné skladování obsahu jiných bufferů
- networkInfo (0x2A0 - 0x2BF) - 32B - pro systémové síťové informace, částečně přístupné uživateli při specifikaci bezdrátové sítě
- Systémové proměnné - některé jsou plně přístupné uživateli (toutRF, userStatus, ...), další jsou jen pro čtení (RFchannel, lastRSSI) a některé nejsou přístupné vůbec (nejsou jmenované, protože je uživatel nepotřebuje)
- Oblast přístupná pro data uživatele - 0x5C0 - 0x5EF (48B) v RAM bance 11 a 0x620 - 0x64F (48B) v RAM bance 12. Těchto 96B může být také přístupných v jednom bloku pomocí lineární adresace v rozmezí 0x2390 - 0x23EF. Dají se také využít dva uživatelské registry userReg (0x70 a 0x71) bez nutnosti výběru RAM banky.

Registry INDFx a FSRx slouží k nepřímému adresování RAM paměti v mikrokontrolerech. Do registru FSRx se vloží požadovaná adresa registru, se kterým se chce pracovat. S odpovídajícím registrum INDFx je poté možné pracovat, jako by se jednalo o registr se zvolenou adresou. Z bezpečnostních důvodů je nepřímý přístup přes registr INDFx zakázán. Z toho důvodu nemohou být používány proměnné jako indexy pole ( $A[i] = 0$  je zakázáno kvůli použití registru INDFx C kompilatorem). Místo toho operační systém dovoluje práci s těmito registry pomocí funkcí setINDF0(), setINDF1(), getINDF0() a getINDF1(). V FSRx registru se stále musí nastavit požadovaná adresa pro zápis a potom pomocí funkce setINDFx() se byte zapíše do registru INDFx a následně je zkopirován na požadované místo. Funguje to i naopak když se čte byte z RAM paměti, jen se musí použít funkce getINDFx(). Další možností je použití funkcí readFromRAM() a writeToRAM(). Existuje také několik dalších registrů, které se mohou číst přímo, ale zápis do nich lze provádět jen pomocí funkce writeToRAM().

V mikrokontroleru je také paměť typu EEPROM. Data v této paměti zůstávají i po vypnutí napájení a proto se využívá k uchování konfiguračních parametrů. Počet přepisů

je garantovaný na  $10^6$ , typicky však dosahuje až  $10^7$ . Rozdělení paměti EEPROM:

- Uživatelská data (0x00 - 0x9F) - 160B přístupných pouze pro Node. Koordinátor toto místo využívá pro informace o síti.
- Aplikační data (0xA0 - 0xBF) - 32B pro speciální potřeby uživatele. Jednoduché čtení zajišťuje funkce appINFO().
- Zbývající oblast (0xC0 - 0xFF) - je určena pro operační systém, není uživatelsky přístupná

V případě nedostatku trvalé paměti, lze pořídit modul s přídavnou sériovou EEPROM pamětí, která má velikost 16kB.

V IQRF modulech je SPI komunikace[6] zčásti realizovaná operačním systémem, který podporuje funkčnost ve slave režimu s full nebo halfduplex spojením. Master režim může být také realizován, ale není podporovaný operačním systémem a musí být naprogramován uživatelem. Komunikace je plně synchronní a běží na pozadí operačního systému. Ve zdrojovém kódu je příjem řešen kontrolou bitu příznakového registru \_SPIRX, který změní hodnotu, když přijdou data do modulu. Přijímaná data jsou ukládána do bufferCOM. K odesílání dat slouží funkce startSPI(x), kde x je počet odesílaných bytů. Data jsou odesílána z bufferCOM a délka zprávy je omezená pouze na 64B. Komunikaci přes UART nebo I2C je možné realizovat uživatelskými funkcemi.

V operačním systému verze 3.02D je přístupné uživatelské přerušení. Všechny přerušení se ovládají stejně. Každé může být povoleno nebo zakázáno pomocí bitu příznakového registru, který končí písmenem E (např.: GIE, TXIE, atd.). Samotné přerušení je pak vyvoláno pomocí bitu příznakového registru, který končí písmenem F (např.: RCIF, TXIF, TMR6IF, atd.). Všechny přerušení v programu se mohou povolit nebo zakázat pomocí bitu příznakového registru GIE (global interrupt enable), avšak přerušení musí být vypnuto jen na krátkou dobu, jinak se mohou objevit problémy ve funkčnosti operačního systému (RF komunikace). Při využívání uživatelského přerušení se musí dodržovat tato pravidla:

- uživatelské přerušení musí být povoleno (\_enableUserInterrupt = 1)
- obsluha uživatelského přerušení musí začínat na adrese 0x3F00
- obsluha uživatelského přerušení musí trvat maximálně 50us
- obsluha uživatelského přerušení se nesmí volat častěji než jednou za 800us

Přerušení operačního systému má vyšší prioritu než uživatelské a proto je obsluha uživatelského přerušení volána až po obsluze veškerých přerušení, které byly vyvolány operačním systémem. To znamená, že RF komunikace bude vždy obsloužena dříve než komunikace přes UART.

## 3. NÁVRH A REALIZACE SYSTÉMU

V této kapitole je popsáný postup při vývoji systému. Nejprve je navržena topologie sítě spolu se řídicím algoritmem posílání paketů. Z návrhu se odvíjejí datové toky, které jsou načtrnuty v následující podkapitole 3.2. Následuje vysvětlení implementace a funkčnosti programu koordinátora a jednotky node a jejich propojení s dalšími částmi systému. Poté je zde popsáno testování a postupný vývoj systému včetně problémů, které se vyskytly. Nakonec je popsán princip a ovládání programu na PC.

### 3.1. TOPOLOGIE SÍTĚ

Vzhledem k využití systému, bylo nejlepší zvolit hvězdicovou topologii sítě. Hvězdicová topologie se vyznačuje jedním hlavním řídicím uzlem, který se nazývá koordinátor. K němu jsou připojené ostatní uzly, které se jmenují node. Komunikace vždy probíhá pouze mezi koordinátorem a jednou jednotkou node, které samy mezi sebou komunikovat nemohou. Tento typ zapojení má své výhody:

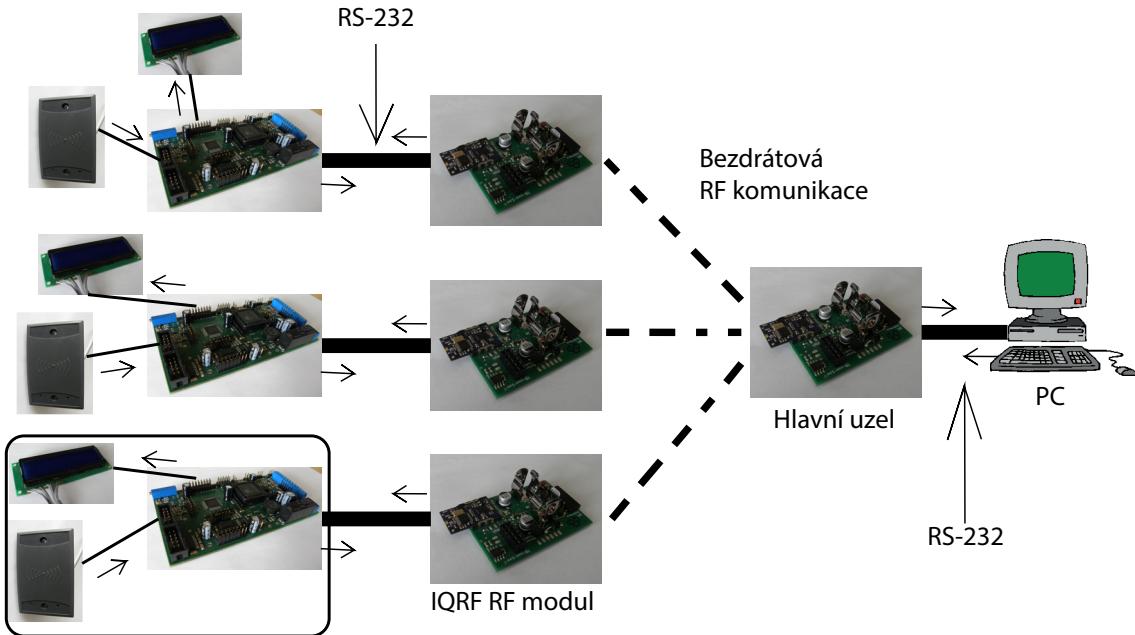
- Pokud selže jeden node, tak ostatní mohou fungovat bez jakéhokoliv omezení
- Jednoduše se rozšiřuje o další jednotky
- Závady se dají snadno nalézt

Bohužel má jednu velkou nevýhodu: V případě selhání centrálního síťového prvku přestane fungovat celá síť.

Při využití hvězdicové topologie v bezdrátové síti je potřeba řídit posílání datových paketů, protože kdyby každý uzel v síti vysílal libovolně podle svého uvážení, tak by mohlo docházet ke kolizím paketů. Například kdyby vysílaly dvě jednotky node zároveň, koordinátor by přijal první paket. V průběhu přijímání tohoto paketu by přišel druhý, ale ten by vůbec nemohl být koordinátorem zaznamenán a data v něm by se nenávratně ztratila. Ke kolizím paketů vůbec nemůže dojít pokud se využije deterministická metoda řízení přístupu k médiu. V systému je použita deterministická metoda Master-slave. Hlavní řídící uzel (Master) se postupně dotazuje jednotlivých uzlů typu slave. Uzly typu slave nesmí samostatně vysílat. Uzel slave vysílá pouze po vyzvání uzlem master. Komunikace probíhá vždy jen mezi uzlem Master a jednotlivými uzly slave. Nevýhodami tohoto přístupu je závislost délky komunikačního cyklu na počtu uzlů slave a naprostá nefunkčnost sítě při výpadku uzlu master. Výhodou je velmi jednoduchá implementace. V systému uzel master odpovídá koordinátoru a uzly slave jsou jednotlivé jednotky node.

### 3.2. DATOVÉ TOKY

Z PC je realizovaná datová linka RS-232 do koordinátora a data mohou proudit oběma směry. Z PC jdou příkazy do koordinátora, který je dále zpracovává. Z koordinátora do PC chodí odpovědi na příkazy nebo data, která byla shromážděna ze sítě. Mezi jednotlivými jednotkami node a zařízeními 914 HotMAX je realizovaná datová linka RS-232. Směrem od jednotky node chodí pouze příkazy k zařízení 914 HotMAX. V opačném



Obrázek 3.1 Zapojení systému a datové toky[22]

směru chodí odpovědi na příkazy a také data, které byly získány od RFID čtečky karet. Všechna tato data se v node ukládají. Mezi koordinátorem a jednotlivými jednotkami node probíhá bezdrátová komunikace. Veškerou bezdrátovou komunikaci aktivuje koordinátor. Má dvě části:

- synchronní část, jejímž základem je komunikační cyklus obvolávání jednotlivých jednotek node a následné čekání na jejich odpověď.
- asynchronní příkazy, po kterých koordinátor může, ale také nemusí, čekat na odpověď.

Podrobný popis synchronní komunikace bude v následující kapitole.

### 3.3. KOORDINÁTOR

Na začátku programu koordinátora je nastavení základních vlastností. Důležitá je funkce `setRFmode()`, která nastavuje přijímací i vysílací napájecí mód a filtraci. Je nutné, aby všechny uzly v síti měly vzájemně odpovídající nastavení, tedy i koordinátor. Naše nastavení odpovídá hodnotě `0b01000000`. Podrobné vysvětlení nastavení funkce `setRFmode()` je v tabulce 3.1.

Hlavní smyčka programu obsahuje tři části, z důvodu oddělení příjmů jednotlivých typů komunikace. Kdyby bylo přijímání vnořené do sebe, mohlo by docházet k nepříjemným situacím, kdy by se jeden typ komunikace vyhodnocoval častěji než ostatní. Takhle se v každém cyklu programu zkонтrolují všechny typy. V první části je synchronní komunikační cyklus, který lze zapínat a vypínat nastavením proměnné `autocall`. Pro obvolávání správného počtu jednotek node se z paměti EEPROM z adresy `0x00` vyčítá počet připojených jednotek. V každém cyklu se posílá žádost další jednotce node dokud se neobslouží poslední. Poté se opět začíná od první jednotky.

V další části kódu je realizovaný příjem RF komunikace. Pomocí funkce `startDelay()`

Tabulka 3.1 Nastavení fukce setRFmode()

setRFmode(mode)	uns 8 mode SWTTFFRR (binárně)
S	zůstat ve vysílacím módu. 1 - zůstává, 0 - nezůstává
W	počkání na konec packetu. 1 - když byl zahájen příjem, počká se na dokončení, i když vypršel čas stanovený toutRF, 0 - příjem je bezpodmínečně ukončen po vypršení času toutRF
TT	nastavení vysílacího módu. Nastavení musí odpovídat přijímacímu módu na druhé straně. 00 - STD RX mód, 01 - LP RX mód, 10 - XLP RX mód, 11 - LP/XLP RX asynchronní ukončení při změně na pinu
FF	nastavení hodnoty filtrování
RR	nastavení přijímacího módu. 00 - STD mód, 01 - LP mód, 10 - XLP mód, 11 - RFIM mód

začne odpočet doby stanovený pro příjem zprávy. Pomocí funkce isDelay() se ve smyčce kontroluje trvání prodlení. Pokud během této doby není přijmutý žádný RF packet pokračuje se dál v programu. Toto čekání je důležité kvůli tomu, aby se při čekání na příjem neodesíaly žádné jiné bezdrátové zprávy, protože by mohlo dojít ke kolizím packetů. Pokud jsou data přijmutá, zkopírují se do bufferINFO a jsou odeslána přes UART do PC.

V poslední části je realizovaný příjem přes UART. Přijmutí celé zprávy je detekováno pomocí časovače timer6, který dokáže vyvolat přerušení, ale hlavně nastavuje bit příznakového registru TMR6IF pokud dosáhne požadované hodnoty. Nastavení časovače je provedeno ve funkci openUART() pomocí registrů T6CON a PR6. Za konec zprávy se považuje mezera mezi přijímanými byty, která je větší než 20ms. Když časovač zaznamená takovou mezeru, tak nastaví bit příznakového registru MTR6IF, který je ve zdrojovém kódu pravidelně kontrolován, a spustí se obsluha komunikace přes UART. Dále se kontroluje přetečení bufferu. Získání počtu přijatých bytů se provádí funkcí getRxStatUUART() a poté se požadovaný počet bytů překopíruje na bezpečné místo do bufferAUX pomocí funkce copyRxDataUART(). Dále se kontroluje první byte zprávy a podle jeho hodnoty se v příkazu switch rozhoduje o dalším postupu v programu. Dostupné příkazy pro ovládání koordinátora:

- ,c` - z paměti koordinátora vymaže všechny připojené nody
- ,I` - jako odpověď se vrátí kompletní konfigurace, včetně připojených nodů
- ,i` - jako odpověď se vrátí nastavení koordinátora
- ,h`<bit> - nastavení proměnné autocall
- ,a`<addr> - pošle dotaz pro node na adresu <addr>, zda nepřišly data od zařízení 914 HotMAX. Jestliže nic nepřišlo jako odpověď, vrátí se hodnota 0x00.
- ,e`<addr><command> - odešle příkaz <command> pro node s adresou <addr>
- ,d`<addr><1line><2line> - odešle data, která se mají zobrazit na LCD displeji, na node s adresou <addr>. <1line> je 16B, které se zobrazí v prvním řádku LCD displeje. <2line> je 16B, které s zobrazí na druhém řádku LCD displeje.
- ,b`<addr> - připojí node s adresou <addr> do sítě
- ,u`<addr> - odpojí node s adresou <addr> od sítě
- ,r`<addr> - na node s adresou <addr> odešle příkaz k jeho resetování

Každý příkaz je kontrolovaný z hlediska délky. Pokud je délka příkazu špatná, provede se funkce restartUART(). U některých příkazů se před odesláním doplňují důležité informace, aby je node po přijetí nemusel dále zpracovávat. Např.: u příkazu „d‘<addr><1line><2line>“ je doplnován koncový znak a řídící příkaz pro 914 HotMAX. U příkazů, u kterých se očekává odpověď, je na konci skoková instrukce na návští příchozí RF komunikace goto RX\_comm.

## 3.4 NODE

Na začátku programu je jednoduché nastavení jednotky a kontrola, jestli je node připojený do sítě. Důležitá je funkce setRFmode(), která nastavuje přijímací a vysílací napájecí mód a filtraci. Je nutné aby všechny uzly v síti měly vzájemně odpovídající nastavení, tedy i koordinátor. Nastavili jsme standartní napájecí mód a nejmenší filtraci. Podrobné vysvětlení nastavení funkce setRFmode() je v tabulce 3.1.

Následuje hlavní programová smyčka, která je opět rozdělená na tři části ze stejného důvodu, jako je popsáno v kapitole 3.3. V první části je možnost odpojit node od sítě, nebo ho přepnout do spánkového režimu. Vše je vykonáváno pomocí tlačítka.

V druhé části je obsluha příjmu přes UART, která je také stejná jako v kapitole 3.3. Jediný rozdíl je v tom, jak se nakládá s příchozími daty. Pokud se první znak zprávy rovná hodnotě 0x3B, znamená to, že přišel kód ze čtečky karet. Kód se poté překopíruje na místo RAM paměti s počáteční lineární adresou 0x23B0 a nastaví se proměnná cardCount na 1. V případě, že se první znak rovná čemukoliv jinému, znamená to, že přišla odpověď na řídící příkaz. Příchozí zpráva se překopíruje do bufferRF a odešle do koordinátora.

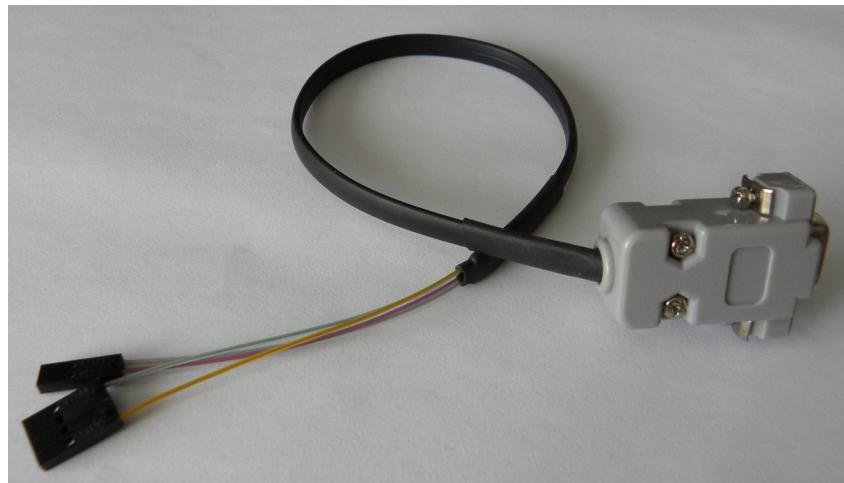
V poslední části je realizovaný RF příjem. Příchozí packety se indikují pomocí funkce checkRF() a samotný příjem pomocí funkce RFRXpacket(). Následně se kontroluje první byte z příchozí zprávy a podle tabulky 3.2 se rozhoduje, jak se dále naloží s příchozími daty.

Tabulka 3.2 Příkazy pro node

Znak	Funkce
'a'	odeslání kódu karty přes RF; v případě že nejsou data k odeslání, odešle se znak '0'
'e'	odeslání dat přes UART do zařízení 914 HotMAX
'd'	odeslání dat přes UART do zařízení 914 HotMAX, speciálně pro displej
'r'	restartování jednotky
'u'	pokyn od koordinátora na odpojení ze sítě

## 3.5 PROPOJENÍ NODE A 914 HOTMAX

Zařízení 914 HotMAX disponuje klasickou linkou RS-232. IQRF RF modul má k dispozici UART, a proto je potřeba k jejich spojení využít napěťového převodníku. Byl využit převodník od firmy Microrisc *IQRF RS-232 cable* (obrázek 3.2)[27]. K RF modulu



Obrázek 3.2 Kabel pro sériovou komunikaci IQRF

se připojuje čtyřmi vodiči. Dva jsou datové (RxD a TxD) a dva napájecí (GND a +3.3V). Na opačné straně je zakončený konektorem D-Sub DE-9F, který odpovídá klasickému třívodičovému zapojení. Zařízení 914 HotMAX komunikuje rámci, které obsahují jeden start bit, 8 datových bitů, žádnou paritu a jeden stop bit. Rychlosť je pomocí jumperu nastavena na 9600 bd. Stejnou konfiguraci musíme softwarově nastavit i v RF modulu. Nejdříve je potřeba nastavit správné I/O registry:

- Registr TRISC.7 - nastavený na 1 jako vstup (RxD)
- Registr TRISC.5 - pin, který je připojený k RxD
- Registr TRISC.6 - nastavený na 0 jako výstup (TxD)

Dále se IQRF modwulech nastavují registry, které zajišťují nastavení přenosu:

- Registr BAUDCON - v tomto registru se nastavují možnosti spojené s baudovou rychlosťí a časováním přenosu. Pro propojení se zařízením 914 HotMAX je nastavený na hodnotu 0.
- Registr SPBRGL - registr, ve kterém se nastavuje baudová rychlosť. Správná hodnota lze vyčíst z tabulky v datasheetu. Pro baudovou rychlosť 9600 bd je registr v tomto systému nastavený na 51.
- Registr TXSTA - nastavuje vlastnosti odesílání. Umožňuje nastavení počtu datových bitů na 8 nebo 9, výběr mezi synchronním a asynchronním módem. V tomto systému je registr nastavený na hodnotu 0b00100100. Podrobný popis registru je v tabulce 3.4.

Tabulka 3.3 Legenda k popisu registrů

R - bit pro čtení	W - bit pro zápis	U - neimplementovaný bit
'0' - bit je standarně nastavený na 0	'1' - bit je standartně nastavený na 1	'x' - hodnota bit je neznámá

Tabulka 3.4 Registr TXSTA

R/W-/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN	SYNC	SENDDB	BRGH	TRMT	TX9D
bit 7							bit 0

- CSRC - bit pro výběr zdroje hodinového signálu. V asynchronní komunikaci není významný
  - TX9 - povolení vysílání 9. bitu. 1 - povolený, 0 - zakázany
  - TXEN - povolení vysílání. 1 - povoleno, 0 - zakázáno
  - SYNC - výběr přenosového módu. 1 - synchronní, 0 - asynchronní
  - SENDDB - odeslání přerušovacího znaku: 1 - pošle přerušovací znak při příštím vysílání, 0 - odeslání přerušovacího znaku je dokončeno
  - BRGH - výběr módu vysoké rychlosti: 1 - vysoká rychlosť, 0 - nízká rychlosť
  - TRMT - status vysílacího posuvného registru: 1 - je prázdný, 0 - je plný
  - TX9D - devátý datový bit k odeslání
- Registr RCSTA - nastavuje vlastnosti příjmu. Umožňuje nastavení počtu datových bitů na 8 nebo 9, povolení sériového portu. Lze z něho vyčíst chybová hlášení. V tomto systému je nastavený na hodnotu 0b10010000. Podrobný popis registru je v tabulce 3.5.

Tabulka 3.5 Registr RCSTA

R/W-/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R - 0/0	R - 0/0	R/W-x/x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

- SPEN - povolení sériového portu. 1 - povolený, 0 zakázany
- RX9 - povolení příjmu 9. bitu. 1 - povolený, 0 - zakázany
- SREN - V asynchronní komunikaci není významný
- CREN - nastavení nepřetržitého příjmu. 1 - povolený, 0 - zakázany
- ADDEN - nastavení detekce adresy. Pro asynchronní 8-bitový přenos není důležitý
- FERR - chyba rámce: 1 - chyba, 0 - žádná chyba
- OERR - chyba přetečení: 1 - chyba, 0 - žádná chyba
- RX9D - devátý datový bit

Dále se musí nastavit registry pro časovač timer6, kterým se u UART komunikace detekuje konec zprávy. Nastavení je provedeno ve funkci openUART() pomocí registrů T6CON a PR6.

- Registr T6CON - zapíná časovač timer6 a nastavuje vstupní hodnoty hodinového signálu a výstupní hodnoty časovače. Podrobný popis registru je v tabulce 3.6.

Tabulka 3.6 Registr T6CON

U - 0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>	
bit 7							bit 0

- bit 7 - neimplementovaný, nastavený na nulu
- T6OUTPS<3:0> - čtyři bity, které upravují hodnotu výstupního signálu časovače v poměru 1:1 až 1:16. Snižuje četnost vyvolání přerušení.
- TMR6ON - zapíná a vypíná časovač timer6: 1 - zapnutý, 0 - vypnuty
- T6CKPS<1:0> - dva bity, které upravují hodnotu hodinového signálu v poměru 1:1, 1:4, 1:16 a 1:64
- Registr PR6 - v tomto registru se nastavuje hodnota pro porovnání. Když časovač timer6 dosáhne této hodnoty, komparátor vygeneruje signál, který generuje přerušení.

## 3.6 PROPOJENÍ KOORDINÁTORA A PC

Na propojení koordinátora a PC se původně používalo rozhraní SPI. Ale vzhledem ke změně zadání a možnosti rozšíření do dalších systémů, bylo vhodnější využít spojení přes RS-232. Do PC, které disponují sériovým portem, je připojení jednoduché pomocí převodníku *IQRF RS-232 cable*, tak jako při propojení jednotky node a zařízení 914 HotMAX. Většina moderních notebooků však tímto portem nedisponuje. Tady lze využít další převodník z USB na RS-232 (obrázek 3.3). Je to trochu nešikovné řešení, ale v případě připojení k notebooku je to nutné. V reálném systému tento převodník nemusí být potřeba, protože v průmyslových systémech je komunikace přes RS-232 více častá. Konfigurace jednotlivých registrů je stejná jako v kapitole 3.5.



Obrázek 3.3 Převodník USB/RS-232

### 3.7 RS-232 NA IQRF MODULECH

V kapitole 3.5 bylo řečeno jak je potřeba nastavit jednotlivé registry pro správné připojení UART. Nyní bude vysvětleno, jak se přijímají a odesílají jednotlivé byty přes RS-232. Jak již bylo řečeno komunikace přes UART není v RF modulech podporována operačním systémem. Proto se celá obsluha musí řídit uživatelsky. Původně byla komunikace přes UART vyřešena bez využití přerušení. Bohužel se toto řešení ukázalo jako nepraktické. Důvodem je RF komunikace, která je řešena pomocí přerušení s nejvyšší prioritou. To někdy způsobovalo, že přes UART nepřišla celá zpráva, protože v průběhu příjmu modul začal zároveň přijímat i RF zprávu. Komunikace přes UART byla přerušena a nepřijaté byty byly už nenávratně ztraceny. Tato chyba se vyskytla při programování na module typu TR-53B s operačním systém verze 3.00. Tato verze operačního systému nepodporovala uživatelské přerušení, a proto se muselo přejít na nové moduly typu TR-52D s operačním systémem 3.02D, které už uživatelské přerušení podporují.

Obsluha uživatelského přerušení probíhá v uživatelské rutině, která musí vždy začínat na adrese 0x3F00. Je rozdělená na dvě části. V první části se kontroluje, jestli nepřišla nějaká data přes UART pomocí bitu příznakového registru RCIF. Pokud přišla, do registru FSR0 se nastaví adresa pro zapsání příchozího bytu, která se skládá z adresy na začátek bufferu a pointeru, který v něm ukazuje na první volné místo, a pomocí funkce setINDF0() se příchozí byte, který je uložený v registru RCREG, zapíše na správné místo. Následně se upravuje hodnota ukazatele, aby ukazoval na další prázdné místo v bufferu, a kontroluje se zda buffer nepřetekl. V této části také nuluje časovač timer6, který měří čas mezi příchozími byty. Ve druhé části se kontroluje bit příznakového registru TXIF, který udává jestli je předchozí byte odeslaný, a také jestli zbývají nějaká další data k odeslání. Pokud ano, tak se nejdříve pomocí registru FSR0 nastaví adresa na místo, kde je byte k odeslání uložený. Následně se pomocí funkce getINDF0() načte příslušný byte do registru TXREG, ze kterého se odesílá. Poté se upravuje ukazatel do odesílacího bufferu, aby ukazoval na správný byte. Kvůli kontrole ukončení vysílání se musí délka dat k odeslání snížit o jedna. Pokud už nejsou žádná data k odeslání, tak se vynuluje bit příznakového registru TXIE.

Samotné odesílání dat se v kódu inicializuje pomocí funkce sendDataUART() a argumentem délky odesílaných dat. V této funkci se nejprve kontruje délka zprávy, jestli nepřesahuje délku bufferu. Následně se nastavuje ukazatel do bufferu a délka dat do vnitř-

Tabulka 3.7 Funkce pro práci s rozhraním UART

openUART()	otevření a nastavení sériové komunikace
restartUART()	restartování sériové komunikace
closeUART()	ukončení sériové komunikace
sendDataUART(length)	odeslání dat o délce length přes UART
getTxStatusUART()	vrací počet bytů, které čekají na odeslání
getRxStatusUART()	vrací počet přijatých bytů
getRxByteUART()	vrací první nepřečtený byte z příchozího bufferu
copyRxDataUART(addr, length)	kopíruje přijatá data o délce length z příchozího bufferu na adresu addr
errorsRecoveryUART()	nulování chybových hlášení

ní proměnné. Pomocí bitu příznakového registru TXIE se povolí přerušení na odesílání pomocí UART.

Pro čtení přijatých dat slouží funkce copyRxDataUART(), která kopíruje přijatá data z bufferu na námi zvolené místo pomocí prvního argumentu. Druhým argumentem se nastavuje délka kopírovaných dat. Délka dat je omezena velikostí bufferu, proto se pokaždé kontroluje. Následně se pomocí funkce getRxByteUART() načítá byte po bytu a ukládá na požadované místo. Funkce getRxByteUART() je další z uživatelských funkcí, která vrací byte z příchozího bufferu pro UART komunikaci. Během tohoto čtení z bufferu se na chvíli vypíná možnost přerušení na globální úrovni, aby nebyl proces čtení z bufferu přerušen RF komunikací. Seznam všech funkcí včetně jejich popisu je vypsán v tabulce 3.7.

## 3.8 TESTOVÁNÍ

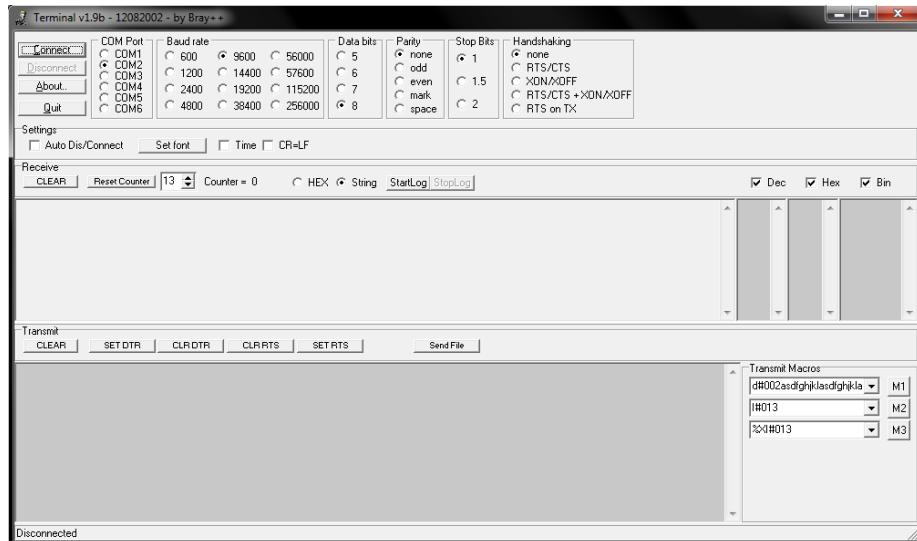
Testování systému probíhalo v několika fázích. V první fázi se testoval dosah RF modulů, jejich spotřeba, atd. V té době se používal RF modul typu TR-31B. U tohoto modulu se zjistil nedostačující dosah. V otevřeném prostoru byl dosah až 100m, ale jakmile se zkoušelo spojení v budově dosah rapidně klesal. RF modul byl také starý se zastaralým operačním systémem. Proto se přešlo na moduly novější typu TR-53B. Tyto moduly jsou z hlediska dosahu dostačující. Následně se testovalo jednoduché spojení mezi RF modulem a zařízením 914 HotMAX. Po vytvoření stabilního spojení byl započat vývoj síťového zapojení. Nejdříve byla vytvořena jednoduchá síť pouze s asynchronními příkazy. Testování bylo prováděno z počítače, ke kterému byl koordinátor připojený pomocí SPI. Po zapojení synchronní komunikace, byla zjištěna chyba v komunikaci z jednotky node přes UART. Do té doby byla komunikace řešena bez přerušení, jednoduchým testováním podmínky, jestli přišla data. Jelikož je RF komunikace řešena pomocí přerušení s nejvyšší prioritou, stávalo se, že v době příjmu UART packetu, přišel zároveň i RF packet. Ten vyvolal přerušení a po návratu k běhu programu byly nepřijaté byty z UART už ztraceny. Protože v používaném RF modulu TR-53B není povolené uživatelské přerušení, nabízely se dvě možnosti, jak řešit tento problém:

- Vytvořit nové řešení síťové komunikace, které by nebylo založeno na synchronním komunikačním cyklu. Jedna taková možnost je síť, ve které může každý node vysílat kdy chce, ale potom se musejí detekovat kolize packetů.
- Přejít na nový typ RF modulů TR-52D. Tento typ má nový operační systém verze 3.02D, který uživatelské přerušení podporuje. Potom není problém naprogramovat příslušné funkce.

Jako lepší bylo vybráno řešení pomocí uživatelského přerušení z důvodu jednoduché implementace síťové komunikace pomocí synchronního obvolávání, oproti složité detekci kolize packetů. Dalším důvodem byla skutečnost, že v budoucnosti by se na nové moduly stejně přešlo. Testování s novými moduly opět pokračovalo v pořadí:

- Otestovat jednoduché spojení mezi RF modulem a zařízením 914 HotMAX
- Vytvořit jednoduchou síť, bez synchronní komunikace
- Přidat do sítě synchronní komunikaci

Následně se kvůli větší variabilitě systému přešlo z připojení do počítače pomocí SPI



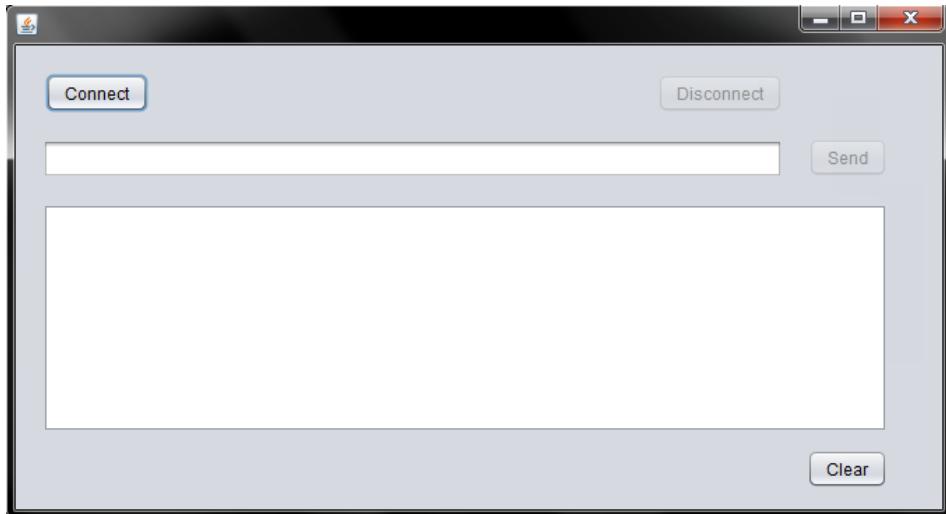
Obrázek 3.4 Terminál pro UART komunikaci

na UART. Při testování tohoto zapojení se narazilo na další chybu. Po zapnutí modulu, probíhala UART komunikace bez problému, ale po chvíli se začaly špatně přenášet konce odchozích paketů. Příchozí komunikace byla pořád bezproblémová. S rostoucím časem bylo v odchozích paketech chybných dat stále více. Po vypnutí a zapnutí IQRF modulu se proces opakoval. Po dlouhém zkoumání různých možností, které by tento problém způsobovaly (špatné nulování bufferů, ...), se došlo k závěru, že je na vině klesající napěťová úroveň v modulu. Do této chvíle byly moduly napájeny pomocí baterie, dodávané firmou Microrisc. Zjistilo se, že při současném zapnutí RF a UART komunikace na nových modulech, klesá napěťová úroveň pod hranici 3.3V, která je nutná ke správné komunikaci rozhraním UART. Problém byl vyřešen změnou napájecího zdroje (akumulátor, síťový adaptér). Testování sítě probíhalo na sestavě koordinátora a dvou jednotek node, ke kterým bylo připojeno zařízení 914 HotMAX. Pro komunikaci přes SPI bylo využíváno prostředí terminálu IQRF IDE4. Pro komunikaci přes UART bylo využíváno prostředí Terminal v1.9b (obrázek 3.4).[28]

### 3.9 SOFTWARE PRO PC

K obsluhování systému z počítače bylo nutné vytvořit program pro komunikaci přes UART. Program je vytvořen v programovacím jazyce Java. Pro komunikaci se sériovým portem byla využita knihovna RXTX[21]. Grafické uživatelské rozhraní, které je vidět na obrázku 3.5, obsahuje čtyři tlačítka a dvě textová pole.

- Tlačítko Connect - slouží k připojení k sériovému portu, nastavení komunikačních parametrů a nastavení posluchače na příchozí kanál. Posluchač je třída, která dokáže detektovat příchozí zprávy.
- Tlačítko Disconnect - slouží k odpojení od sériového portu a jeho uvolnění.
- Tlačítko Send - slouží k odesílání zprávy. Text zprávy je načítán z textového pole.
- Tlačítko Clear - slouží ke vymazání druhého textového pole, do kterého jsou vypisovány příchozí zprávy



Obrázek 3.5 Grafické rozhraní programu

### 3.9.1 PŘIPOJENÍ K SÉRIOVÉMU PORTU

K sériovému portu se program připojí po stisknutí tlačítka Connect. Zavolá se funkce connect(), jejíž argumentem je název portu, ke kterému se chceme připojit. Jméno portu lze zadat jako první argument při spouštění programu. Následně se port identifikuje a zkontroluje se, jestli není obsazený jinou aplikací. Port se otevírá funkcí open() a pomocí funkce setSerialPortParam() se nastavují parametry připojení. Je zvoleno klasické nastavení: rychlosť 9600bd, 8 datových bitů, 1 stop bit a žádná parita. Následně se nastavují vstupní a výstupní kanály a přiřazuje se posluchač příchozích zpráv.

### 3.9.2 ODESÍLÁNÍ ZPRÁVY

Po stisknutí tlačítka Send, se z textového pole načte text do pomocné proměnné typu String. Následně se zjistí délka textu a ve smyčce se zkouší zapisovat na odchozí kanál pomocí funkce write(). Zapisovat se musí po jednotlivých bytech. Proto se na pomocnou proměnnou používá funkce charAt(), která vrací požadovaný znak ze Stringu. Tento znak se ještě musí přetypovat na typ byte. Program je nastavený tak, že dokáže odesílat řídicí příkazy pro koordinátora. Pokud je u některého příkazu potřeba adresa jednotky node, je potřeba ji psát ve tvaru třech číslic. To znamená, že pro připojení jednotky node s adresou 10, je nutné napsat příkaz ve tvaru b010.

### 3.9.3 PŘÍJEM ZPRÁVY

Příjem zprávy je řešený pomocí posluchače, který je přiřazen k sériovému portu. Pokud posluchač zaznamená příchozí data, začne číst ze vstupního kanálu. Čte se tak dlouho dokud jsou na něm data, nebo dokud není zaznamenán znak posunu na nový řádek. Jednotlivé byty jsou ukládány do bufferu a po skončení příjmu jsou vypsáná do textového pole.

### 3.9.4 MySQL DATABÁZE

V programu je implementováno připojení k MySQL databázi, která slouží k ukládání příchozích kódů. Ve funkci connect() je implementováno připojení k databázi, ke kterému se využívá JBCD MySQL driver. Spojení se vytvoří funkcí getConnection(), která má tři argumenty. První argument je adresa databáze, druhý argument je jméno uživatele, třetí argument je heslo a čtvrtý argument je název databáze. Tyto parametry lze zadat jako druhý až pátý argument při spouštění programu. Po vytvoření připojení je proveden příkaz CREATE TABLE IF NOT EXISTS, který vytváří tabulku seznamKodu. Tabulka má 4 sloupce:

- id - typu int, nastavený jako primární klíč, zapnutá auto inkrementace
- hodnotaKodu - typu varchar o délce 45 znaků
- pocetPipnuti - typu int
- addrNode - typu varchar o délce 4 znaky

Do databáze se zapisuje pouze po příchodu kódu ze čtečky karet. Pokud kód v databázi neexistuje, zapíše se na novou pozici, pocetPipnuti se nastaví na 1 a do addrNode se zapíše adresa, od které jednotky node kód přišel (1. znak zprávy). Pokud je kód v databázi již zaznamenaný, inkrementuje se pouze pocetPipnuti v příslušném řádku.

## 4. ZÁVĚR

Byl vytvořen systém, který dokáže sbírat data z přístupového terminálu, který je tvořený zařízením 914 HotMAX a RFID čtečkou karet. Bylo k tomu využito IQRF bezdrátové sítě nastavené do hvězdicové topologie se řízením přístupu pomocí deterministické metody master-slave. Koordinátor byl naprogramován tak, že on jediný aktivuje bezdrátovou komunikaci, a proto je v síti zamezeno kolizím packetů. Koordinátor dokáže sbírat data od všech jednotek node a zároveň dokáže vysílat řídící příkazy pro jednotky HotMAX. Vývoj bezdrátové sítě doprovázela řada hardwarových problémů, jejichž řešení velmi brzdilo postup práce, protože hardware byl považovaný za plně funkční, a proto byly chyby nejprve hledány ve zdrojových kódech a až později v hardwaru. Další velký problém byl způsobený komunikací UART, která zprvu nebyla řešena pomocí uživatelského přerušení a docházelo ke ztrátám dat. Po předělání komunikace s využitím uživatelského přerušení se tento problém vyřešil.

Byla vytvořena počítačová aplikace, která umožnuje řízení koordinátora a navíc dokáže ukládat příchozí data, která byla přečtena čtečkou karet, do jednoduché MySQL databáze.

Během testování poslední verze systému nebyly pozorovány chyby v žádném typu komunikace. Testování probíhalo v laboratorních podmínkách. Samotný systém zatím není po hardwarové stránce k instalaci plně připravený. Je nutné vyřešit napájení přístupového terminálu pomocí akumulátoru a vytvořit ochranný obal, do kterého budou vloženy příslušné části systému.

Nabízí se několik možnosti ke zdokonalení systému v závislosti na způsobu jeho využití. Například pro použití v plaveckých bazénech vytvořit sofistikovanější databázový systém, který by každé jméno spojil s příslušným kódem. Nabízí se také možnost ukládat čas, ve který daný čipový kód přišel. To by umožňovalo měřit čas mezi jednotlivými záznamy.

# PŘÍLOHA A

## OBSAH PŘILOŽENÉHO CD

2013_vinkler_BP.pdf	Bakalářská práce v pdf formátu
nodeBP.c	Zdrojový kód pro jednotku node
coordinatorBP.c	Zdrojový kód pro jednotku koordinátor
TerminalGUI.java	Zdrojový kód PC aplikace
PCapp.zip	Zkompilovaná PC aplikace s knihovnami

# REFERENCE

[1] RS-232 komunikace

<http://cs.wikipedia.org/wiki/RS-232>

[2] RS-232 komunikace

[http://www.usconverters.com/index.php?main\\_page=page&id=61&chapter=0](http://www.usconverters.com/index.php?main_page=page&id=61&chapter=0)

[3] Rozhraní UART

<http://cs.wikipedia.org/wiki/USART>

[4] Rozhraní SPI

[http://cs.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface](http://cs.wikipedia.org/wiki/Serial_Peripheral_Interface)

[5] Rozhraní SPI

[http://commons.wikimedia.org/wiki/File:SPI\\_Stern.svg](http://commons.wikimedia.org/wiki/File:SPI_Stern.svg)

[6] Microrics. SPI Implementation in IQRF TR modules

Dostupné z <http://www.iqrf.org/weben/index.php?sekce=support&id=download>

[7] Microrisc. IQRF Operating System Version 3.02D for TR-52D and TR-54D User's Guide

Dostupné z <http://www.iqrf.org/weben/index.php?sekce=support&id=download>

[8] Microrisc. IQRF Operating System Version 3.02D for TR-52D and TR-54D Reference Guide

Dostupné z <http://www.iqrf.org/weben/index.php?sekce=support&id=download>

[9] Microrisc. TR-52D Transceiver Module Datasheet

<http://www.iqrf.org/weben/index.php?sekce=products&id=tr-52d&ot=transceivers&ot2=tr-52d>

[10] Microrisc. TR-53B Transceiver Module Datasheet

<http://www.iqrf.org/weben/index.php?sekce=products&id=tr-53b&ot=transceivers&ot2=tr-53b>

[11] Microrisc. TR-31B Transceiver Module Datasheet

<http://www.iqrf.org/weben/index.php?sekce=support&id=archive>

[12] Microrisc. DK-PGM-01 IQRF Universal Development kit User's Guide

<http://www.iqrf.org/weben/index.php?sekce=support&id=archive>

[13] Microchip. MRF49XA Datasheet

Dostupné z <http://www.microchip.com/TechDoc.aspx?type=datasheet>

[14] Microchip. PIC16(L)F1938/9 Datasheet

Dostupné z <http://www.microchip.com/TechDoc.aspx?type=datasheet>

- [15] Atmel. AT89C51ED2 Datasheet  
<http://www.datasheetcatalog.org/datasheet/atmel/doc4235.pdf>
- [16] LCD displej  
[http://cs.wikipedia.org/wiki/Displej\\_z\\_tekut%C3%BDch\\_krystal%C5%AF](http://cs.wikipedia.org/wiki/Displej_z_tekut%C3%BDch_krystal%C5%AF)
- [17] Shenze av-display. Specification for LCM module
- [18] RFID čtečka karet  
<http://cs.wikipedia.org/wiki/RFID>
- [19] Promag. MF7 Data Sheet
- [20] J. Holub, J. Novák. Přednáška předmětu Distribuované systémy a počítačové sítě(A3B38DSY). Téma 7: Typy datových přenosů, multiplexování, metody řízení přístupu ke sdílenému médiu
- [21] Knihovna pro sériovou komunikaci TRTX  
[http://rxtx.qbang.org/wiki/index.php/Main\\_Page](http://rxtx.qbang.org/wiki/index.php/Main_Page)
- [22] [http://www.hairymaggot.co.uk/cdn/files/computer\\_clipart1.gif](http://www.hairymaggot.co.uk/cdn/files/computer_clipart1.gif)
- [23] Microrisc  
<http://www.iqrf.org/>
- [24] Microchip. PIC16F886 Datasheet  
Dostupné z <http://www.microchip.com/TechDoc.aspx?type=datasheet>
- [25] Texas Instrument. MAX232 Datasheet  
<http://www.ti.com/lit/ds/symlink/max232.pdf>
- [26] Sipex, SP202EEN Datasheet  
[http://www.datasheetcatalog.org/datasheet/sipex/SP310ECT\\_TR.pdf](http://www.datasheetcatalog.org/datasheet/sipex/SP310ECT_TR.pdf)
- [27] Microrisc. IQRF RS-232 cable  
<http://www.iqrf.org/weben/index.php?sekce=products&id=cab-ser-02&ot=accessories&ot2=cables>
- [28] Terminal v1.9b  
<https://sites.google.com/site/terminalbpp/>