

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Bakalářská práce

**Použití Zigbee pro komunikaci
mezi procesory Atmel-AVR**

Autor: Radim Sejk

školitel: Ing. Michal Šerý

České Budějovice 2012

Bibliografické údaje

Sejk Radim, 2012: Použití Zigbee pro komunikaci mezi procesory Atmel-AVR

[Application of Zigbee for Atmel-AVR communication. Bc.. Thesis, in Czech.] – 39 p.
Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace:

Zigbee protokol je velice užitečný způsob přenosu dat mezi mikroprocesory na krátkou či středně velkou vzdálenost. Výstupem této práce v teoretické části jsou výsledky měření dosahu a rychlosti odezvy XBee modulů.

V praktické části je sestaven demonstrační přípravek realizující přenos digitálních a analogových veličin. Nastavení a zobrazení těchto přenášených dat zajišťuje vytvořený obslužný program.

Annotation:

Zigbee protocol is a very useful way for data transfer between microprocessors at short or medium distance. Theoretical part of this dissertation comprises results of XBee modules range and speed of response measurements.

Practical part contains demo device which implements transmission of digital and analog variables. Transmitted data setting and visualisation is provided by developed service utility.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě – v úpravě vzniklé vypuštěním vyznačených částí archivovaných Přírodovědeckou fakultou - elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 8.12.2012

.....

Radim Sejk

Poděkování

Děkuji Ing. Michalovi Šerému za konzultace a uvedení do problematiky tohoto tématu a propůjčení logického analyzátoru. Dále firmě Sparkfun za poskytnutí slevy na nákup některých součástí a modulů za účelem této práce a členům fóra Arduino.cc za cenné rady pro oživení komunikace XBee se softwarem X-CTU.

Obsah

1 Úvod.....	1
2 Cíle práce	2
3 Metodika	3
3.1 Moduly XBee	3
3.2 Arduino.....	4
3.3 Wiring.....	5
3.4 Jazyk Processing	6
4 Teorie.....	7
4.1 Komunikace	7
4.1.1 Sériová komunikace	7
4.2 AT příkazy	8
4.2.1 ATAT	10
4.2.2 ATBD	10
4.2.3 ATDT.....	10
4.2.4 ATHP	11
4.2.5 ATID.....	11
4.2.6 ATMK.....	11
4.2.7 ATRR.....	11
4.3 Processing	12
4.4 Arduino.....	13
4.5 X-CTU.....	14
4.5.1 Připojení	14
4.5.2 Nastavení.....	14
4.5.3 Test Dosahu	15
4.5.4 Terminál	16
4.5.5 Nastavení Modulu	17
4.5.6 Shrnutí a možné komplikace	18
5 Výsledky.....	19
5.1 Zapojení.....	19
5.2 Měření dosahu	21
5.2.1 Test č.1	22
5.2.2 Test č.2	23
5.2.4 Test č.3	24
5.2.5 Test č.4	25
5.3 Měření rychlosti přenosu.....	25

6	Možné využití Arduino + XBee	27
6.1	Obslužný program do PC:	28
6.1.1	Nastavení sériového portu:	28
6.1.2	Grafická část:	28
6.1.3	Detekce pozice kurzoru:	29
6.1.4	Odeslání dat:	29
6.1.5	Příjem dat:	30
6.2	Nyní program do Arduino:	30
6.2.1	Setup:	30
6.2.2	Loop:	30
6.2.3	PCtoXBEE:	31
6.2.4	XBEEtoPC:	31
6.2.5	LED:	31
7	Závěr.....	32
	Seznam použité literatury:.....	33

1 Úvod

V současné době je velký trend využívat k přenosu dat bezdrátovou komunikaci. Na trhu je k dispozici velké množství zařízení, která nám přenos dat technicky zajistí, avšak ne všechny přenosové systémy či jejich komunikační protokoly jsou dostatečně rozšířené v povědomí. Proto bych chtěl v této práci vysvětlit některé pojmy z této problematiky a uvést jednu z možností, jak přenášet data digitální i analogová pomocí bezdrátové technologie mezi mikroprocesory a tato data dále zobrazit v PC.

2 Cíle práce

Cílem této práce je představit bezdrátové moduly XBee v kombinaci s mikroprocesorem Atmel a poté realizace demonstračního přípravku, který předvede komunikaci mezi mikroprocesory. Systém musí být schopen přenášet jak digitální tak analogové veličiny a dále tyto veličiny nastavit, případně zobrazit jejich stav pomocí programu v PC.

Bakalářská práce má tři části. V první textové části se seznámíme s použitým hardware a software. V části druhé bude sestaven funkční demonstrační přípravek.

Funkční přípravek musí zajistit přenos digitální i analogové veličiny pomocí bezdrátových modulů XBee v kombinaci s mikroprocesory Atmel. Dále zde bude zdůvodněn výběr jednotlivých použitých součástí a programovacího jazyku.

Třetí část pojednává o provedených měřeních dosahu a doby potřebné k přenosu dat Xbee modulů.

3 Metodika

3.1 Moduly XBee

Bezdrátové moduly XBee od firmy Digi poskytují širokou škálu jednotlivých platforem. Podporují více topologií sítě jako „point to point“, „point to multipoint“ a „peer to peer“ na pod protokolem 802.15.4 zvaným Zigbee. Dále si můžeme vybrat mezi moduly pracujícími na různých frekvencích.

Tyto moduly jsou snadno a rychle použitelné a po rozbalení ihned funkční pro jednoduché aplikace, případně se pomocí AT příkazů dají nastavit další důležité parametry pro složitější použití ve složitějších aplikacích.

Moduly XBee se hodí pro aplikace, které vyžadují rychlou odezvu a předpověditelné časování komunikace. Dosahují rychlé a kvalitní komunikace ať už jako jednoduchá náhrada kabelové linky nebo jako více komplexní hub pro síť s více senzory či cílovými zařízeními.

Na trhu je relativně velká škála jednotlivých modulů s různými vlastnostmi. Výběr musíme přizpůsobit hlavním parametrům, které tyto moduly poskytují:

- Síť 802.15.4/Multipoint
- 2,4 GHz nebo 900 MHz pro větší dosah a prostupnost překážkami
- Nízkoodběrový modul nebo modul s vysokým výkonem pro větší dosah

Pro tuto práci jsem zvolil modul XBee Pro 900 XSC, jelikož pak tento systém dále hodlám použít pro dálkově ovládaného a z větší části i autonomního šestikolového robota. Na frekvenci 900MHz a výkonu 100mW je výrobcem deklarován maximální dosah 15mil (v praxi však dosah znatelně menší viz 6.2 měření dosahu).

3.2 Arduino

Arduino je opensource projekt elektronické vývojové platformy, založené na jednoduchém a flexibilním hardwaru i softwaru. Je určeno umělcům, designérům, domácím kutilům a všem lidem, které zajímá tvorba interaktivních předmětů a prostředí.

Arduino dokáže zjistit změny v prostředí díky přijmutí signálů od obrovské škály senzorů a dokáže pozměnit jejich okolí řízením světel, motorů, reproduktorů či jiných ovládacích prvků.

Mikroprocesor na desce je primárně určen k programování v „Arduino programming language“, které je založeno na jazyce Wiring a prostředí Processing, není však problém psát program i v jiných jazycích (např. Bascom, C, Asembler).

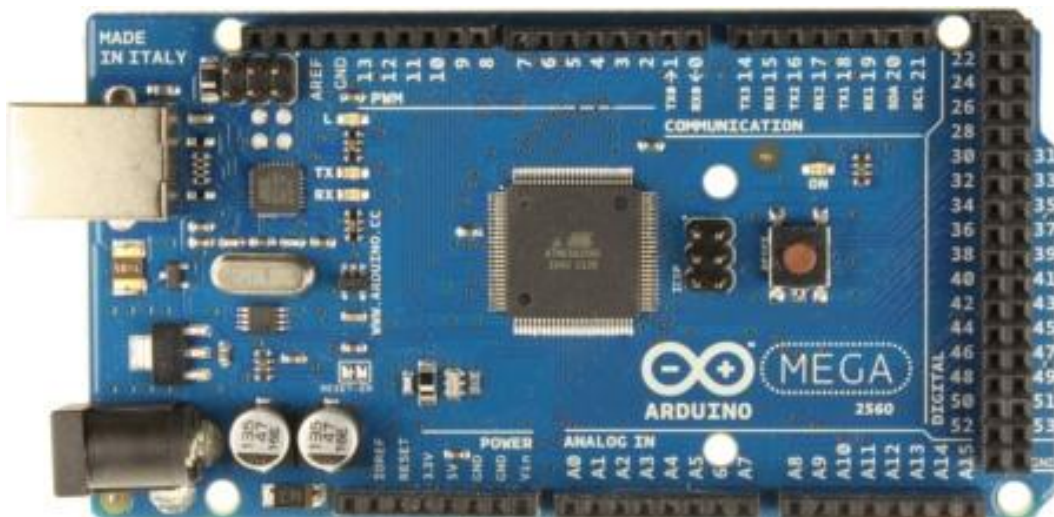
Projekty mohou být buď samostatn , nebo mohou komunikovat s PC a přenechat tak důležité výpočty právě na něm.

Jelikož je tento kit open-source, můžete si jeho schéma a návod na stavbu stáhnout na internetu a postavit doma, nebo zakoupit jako hotový výrobek.

Pro potřeby této práce jsem zvolil Arduino MEGA 2560. Jak již název napovídá, je založen na mikroprocesoru Atmega 2560. Poskytuje 54 digitálních vstupů/výstupů přičemž 14 z nich může být použito pro hardwarové generování pwm signálu. Dále obsahuje 16 vstupů s A/D převodníkem pro detekci analogových veličin , 4 hardwarove sériové linky. K dispozici je zde také 16MHz krystal a Atmega 16U2 pracující jako převodník USB-seriová linka.

Tento typ Arduino MEGA 2560 se mi jeví pro tento projekt jako nejvíce vhodný zejména z toho důvodu, že obsahuje více hardwarových linek, které jsou potřebné ke spojení jak s počítačem tak s XBee modulem. Existují sice způsoby,

jak tyto sériové linky emulovat na jakémkoliv digitálním pinu procesoru, avšak hardwarová linka je mnohem stabilnější a nebude zabírat v procesoru místo ani čas. Další výhodou je právě převodník USB pro snadné nahrávání programu či diagnostiku a pro komunikaci s obslužným programem v PC.



obr. 1: Arduino MEGA 2560[1]

3.3 Wiring

Wiring je open-source programovací jazyk pro mikroprocesory. Umožňuje psát software napříč různými platformami k ovládání zařízení připojených k široké škále mikroprocesorových desek tak, aby vytvořil interaktivní objekty a fyzické zážitky.

Tento jazyk je vytvořen designéry a umělci s myšlenkou vytvořit komunitu, kde začátečníci až po experty z celého světa sdílejí své nápady, vědomosti a jejich společné zkušenosti. Na světě jsou tisíce studentů, vývojářů a domácích kutilů, kteří používají Wiring k výuce a tvorbě prototypů.

K programování Arduina jsem si zvolil tento jazyk, protože s ním mám již předchozí zkušenosti a práce s ním na Arduinu je velice intuitivní a kreativní.

Hlavní výhody Wiringu:

- Velká podpora všech procesorů ATmega a ATiny, TI MSP430, PIC24/32...
- Zdarma ke stažení jako opensource
- Stovky knihoven urychlujících programování
- Funkční pod většinou používaných OS
- Intuitivní způsob programování

Jako programovací prostředí pro tento jazyk byl zvolen software Arduino 0.22. Je to sice dnes již zastaralá verze tohoto programu (k datu této práce je poslední verze Arduino 1.0.3), ale pro tento projekt výhodnější. U novějších verzí totiž nelze použít příkaz `Serial.write`, který je právě pro komunikaci s Processingem velice užitečný. Jedinou nevýhodou bude delší doba kompilace zdrojového kódu pro procesor.

3.4 Jazyk Processing

Processing je opensource programovací jazyk a prostředí pro zájemce, kteří chtějí vytvářet obrázky, animace, a hlavně interakce mezi dalšími zařízeními. Existují do něj knihovny, které přímo spolupracují s Arduinem a dokonce má velice podobnou strukturu a příkazy jako Wiring, který z něho i vychází.

Z těchto výše popsaných důvodů je Processing vhodná volba pro tuto práci.

4 Teorie

4.1 Komunikace

Pod tímto pojmem si představujeme sdělení určitého obsahu pomocí předem dohodnutých symbolů. V počítačové praxi se jedná o komunikaci paralelní nebo sériovou (případně sérioparalelní). V této práci se budeme věnovat především komunikaci sériové jak mezi jednotlivými moduly tak i mezi Arduinem a počítačem.

4.1.1 Sériová komunikace

Sériová komunikace funguje na principu přenosu dat, vždy maximálně jednoho bitu (logické úrovně 1 nebo 0) za jednotku času narozdíl od komunikace paralelní, kde se naráz posílá více bitů najednou.

Pro přenos signálu nám tedy stačí sběrnice, která se skládá pouze ze dvou vodičů: vodič datový a zem.

Existují dva typy sériové komunikace z hlediska fyzikálního přenosu a to sice přenos pomocí změny proudu anebo změny napětí. V případě Arduina se jedná o přenos pomocí změny úrovně napětí, protože je mnohem snáze realizovatelný. Jsou to tzv. úrovně TTL, kdy logické nule připadá napětí mezi 0 V a 0,3 V a logické 1 připadá na 2,7 V až 5 V. Tím jsou jasně daná pravidla a cokoliv mimo tyto meze není bráno v potaz.

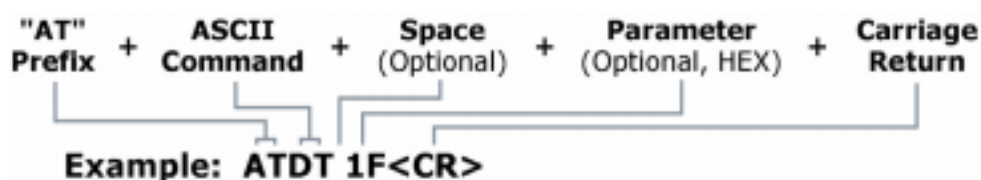
V tomto způsobu komunikace je však nutné jasně vymezit, kde je začátek přenosu a kde je jeho konec, aby nedocházelo k zamíchání dat mezi sebou, každá zpráva je proto označena start a stop bitem.

Pro jednoduchou kontrolu správnosti příjmu je ještě někdy používán bit paritní, který obsahuje informaci o počtu jedničkových bitů a je tedy pak možno v rychlosti zkontrolovat, jestli jsme přijali zprávu celou v pořádku a kompletní .

4.2 AT příkazy

Jsou to jednoduché příkazy používané ke konfiguraci zařízení často různých typů modemů, což je právě náš případ.

Každý AT příkaz se skládá ze tří částí. První část je vždy stejná a to dva znaky „AT“. Dále následuje příkaz složený z několika písmen a volitelně i jeho parametr, většinou hodnota udaná v číslech. Zadání příkazu se provádí stiskem klávesy Enter nebo posláním ASCII hodnoty 13, která odpovídá právě Enteru.



obr. 2: Schéma AT příkazu[2]

Pro nastavení XBee je nutné ještě před samotným zadáním příkazu zaslat sekvenci znaků “+++” a počkat jednu vteřinu (není-li později nastaveno jinak příkazem ATAT), než je modul schopen AT příkazy přijímat. Je to ochrana před náhodným přenastavením modulů při posílání dat shodných právě s AT příkazy.

Zaslaný příkaz je okamžitě zpracován a po jeho úspěšném provedení je odeslána odpověď „OK“, v opačném případě odpověď „ERROR“. Není-li po dobu „Command Mode Time-out“ přijat modulem žádný příkaz, vrátí se automaticky do pohotovostního režimu, kdy je připraven přijímat/odesílat data. Stejného výsledku dosáhneme odesláním příkazu „ATCN“.

U systému XBee máme k dispozici AT příkazy viz tab.1

Příkaz	Název příkazu	Příkaz	Název příkazu
AT	Guart Time After	MK	Adress Mask
BD	Interface Data Rate	NB	Parity
BT	Guard Time Before	PC	Power-up Mode
CC	Command Sequence Character	PW	Pin Wake-up
CD	DO3 Configuration	RE	Restore Defaults
CN	Exit AT command Mode	RN	Delay Slots
CS	DO2 Configuration	RO	Packetization Time-out
CT	Command Mode Time-out	RP	RSSI PWM Timer
DB	RSSI Level in dBm of the most recent packet	RR	Retries
DT	Destination Adress	RS	RSSI
E0	Echo Off	RT	DI2 Configuration
E1	Echo On	SB	Stop Bits
ER	Recieve Error Count	SH	Serial Number High
FH	Force Wake-up initializer	SL	Serial Number Low
FL	Software Flow Control	SM	Sleep Mode
FR	Forces the module to reset	ST	Time before Sleep
FT	Flow Control Treshold	SY	Time before Initialization
GD	Recieve Good Count	TR	Transmit Error Count
HP	Hopping Channel	TT	Streaming Limit
HT	Time before Wake-up Initializer	VR	Firmware Version
ID	Module VID	WR	Write
LH	Wake-up Initializer Timer		

Tab.1: AT příkazy u XBee modulů

Nyní si popíšeme funkce vybraných příkazů AT a jejich parametrů.

4.2.1 ATAT

- Parametr: 0x02-0xFFFF (x100ms)
- Nastavení doby, po kterou je nutno neposílat žádný znak za příkazem CC, aby modul přešel do AT režimu.

4.2.2 ATBD

- Parametr: 0-6
- Nastavení přenosové rychlosti pro posílání dat do/z modulu. Rychlosti přenosu jsou udávány v baudech a v tomto pořadí:
1200-2400-4800-9600-19200-38400-57600
- U modulu 900 XSC maximálně však 9600 tedy parametr 4.

4.2.3 ATDT

- Parametr: 0-0xFFFF
- Nastavení cílové adresy. Pouze moduly se shodnou cílovou adresou jsou schopné mezi sebou komunikovat. Data jsou sice přijata, ale nejsou už posílána dál na TX výstup modulu.

4.2.4 ATHP

- Parametr: 0-6
- Nastavení Hopping channel. Pouze moduly se stejným parametrem jsou schopné přijmout data. Používá se pro oddělení modulů v rámci jedné sítě jako prevence proti zahlcení.

4.2.5 ATID

- Parametr: 0x10-0x7FFFF
- Nastavení Vendor Identification Number. Pouze moduly se stejným VID jsou schopny přijmout data.

4.2.6 ATMK

- Parametr: 0x-0xFFFF
- Nastavení síťové masky. Pro přijetí dat v síti je nutné mít nastavenou shodnou masku.

4.2.7 ATRR

- Parametr: 0-0xFF
- Nastavení počtu opakovaného odeslání. V případě, že odeslaná data nejsou následována potvrzením o přijetí, modul se pokusí data odeslat znovu dokud nejsou přijata, nebo není naplněn počet pokusů.

4.3 Processing

Programovací prostředí Processing má základní strukturu programu podobnou všem ostatním vyšším jazykům vycházejících z C.

Program musí vždy obsahovat dvě základní funkce:

- `void setup()` -Používá se k inicializaci proměnných či některých funkcí. Proběhne pouze jednou při startu programu.
- `void draw()` -Hlavní funkce programu (něco jako main v C++).

Zde je zdrojový kód celého programu případně je samozřejmě možné volat jiné funkce programu.

The image shows a screenshot of the Processing IDE window titled "sketch_121028b | Processing 2.0b5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for play, stop, save, and other functions, along with a "JAVA" button. The main text area contains the following code:

```
void setup() {  
  size(480,800);  
  noStroke();  
  fill(255);  
  rectMode(CENTER); // This sets all rectangles to draw from t  
}  
  
void draw() {  
  background(#FF9900);  
  rect(width/2, height/2, 150, 150);  
}
```

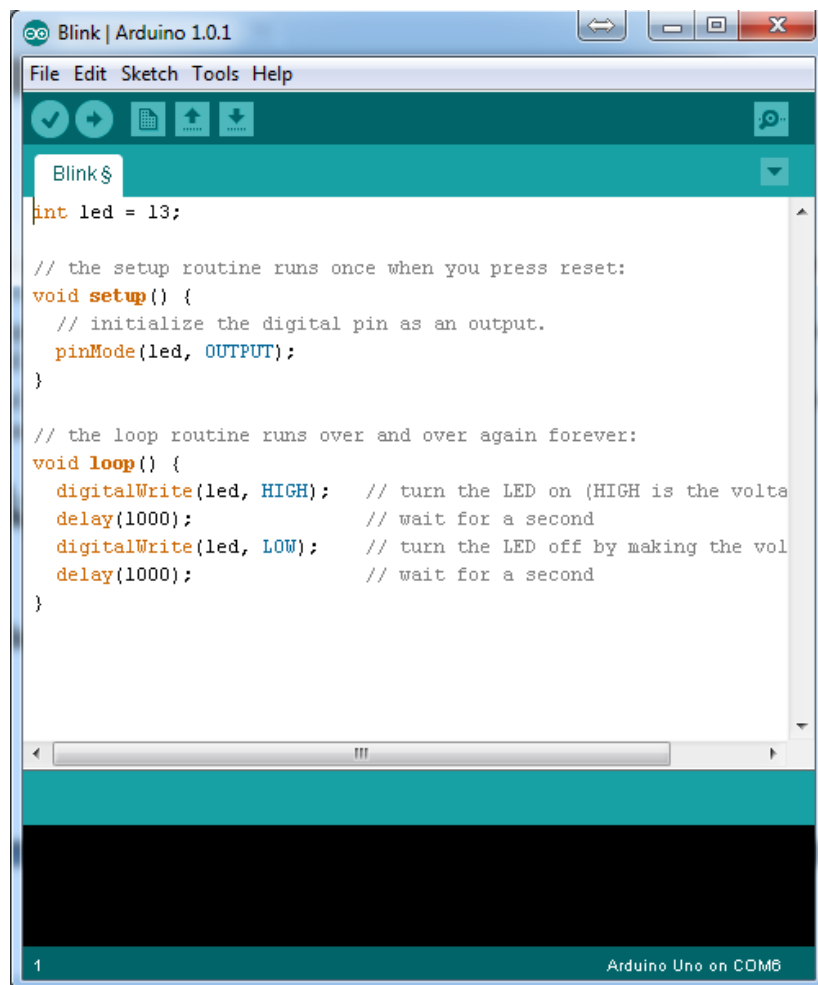
The code is color-coded: keywords in blue, comments in grey, and strings in red. The IDE interface includes a scroll bar on the right and a status bar at the bottom showing the number "3".

obr. 3: prostředí Processing s ukázkou -vykreslí bílý čtverec bez okrajů na oranžovém podkladu

4.4 Arduino

Programovací prostředí Arduino vychází z Processingu a je mu tedy velice podobné. Program musí vždy obsahovat dvě základní funkce:

- `void setup()` - Používá se k inicializaci proměnných či některých funkcí. Proběhne pouze jednou při startu programu.
- `void loop()` - Hlavní funkce programu. Zde je zdrojový kód celého programu případně je samozřejmě možné volat jiné funkce programu.

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, running, and uploading. The main text area contains the following code:

```
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the volta
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the vol
  delay(1000);             // wait for a second
}
```

The bottom status bar shows "1" on the left and "Arduino Uno on COM6" on the right.

obr. 4: Prostředí Arduino s ukázkou programu-blikání LED

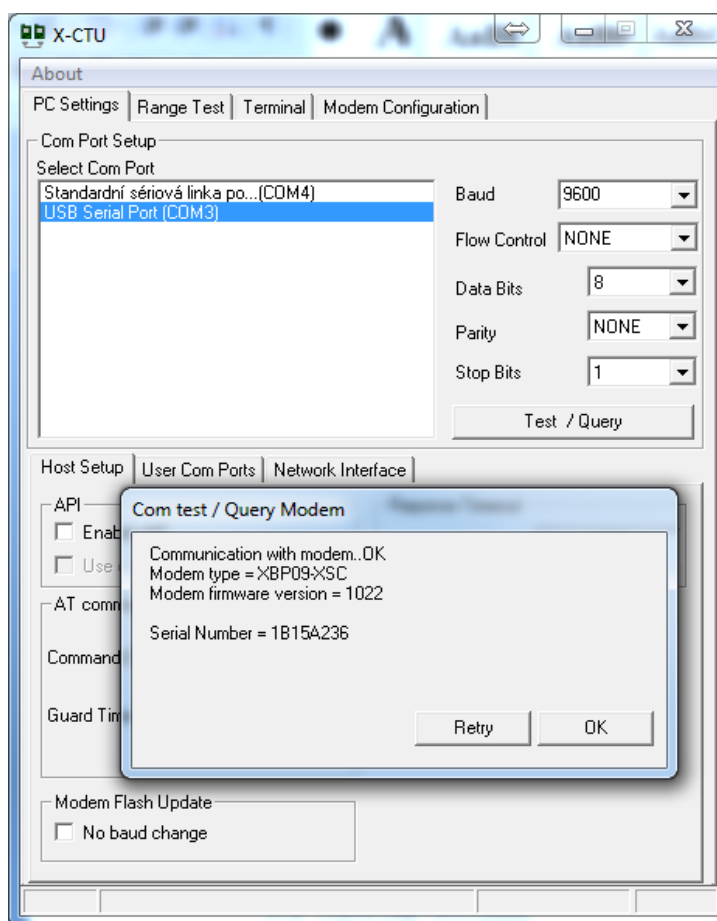
4.5 X-CTU

4.5.1 Připojení

X-CTU je diagnostický program od firmy Digy pro moduly XBee. Po připojení modulu k počítači poslouží jakýkoliv převodník USB \leftrightarrow TTL s vyvedenými CTS,RTS,DTR a samozřejmě RX/TX piny a konvertor úrovní z 5 V na 3,3 V.

4.5.2 Nastavení

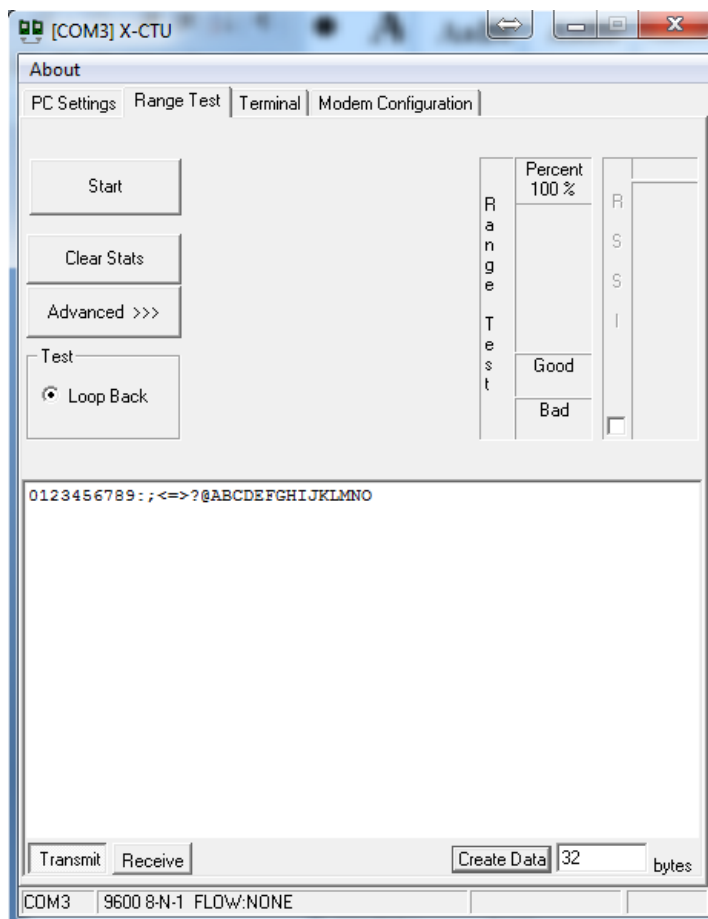
Po připojení modulu k počítači a zapnutí programu zvolíme v záložce PC Settings COM port a stiskneme tlačítko Test/Query. Tím je modul připojen a můžeme pokračovat.



Obr.5: X-CTU nastavení

4.5.3 Test Dosahu

Hned další záložkou je Range Test. Jedná se o velice šikovný nástroj pro zjištění síly signálu a procenta správně přenesených zpráv. Dá se tak jednoduše zjistit funkčnost modulů a dokonce i provést měření dosahu modulů.

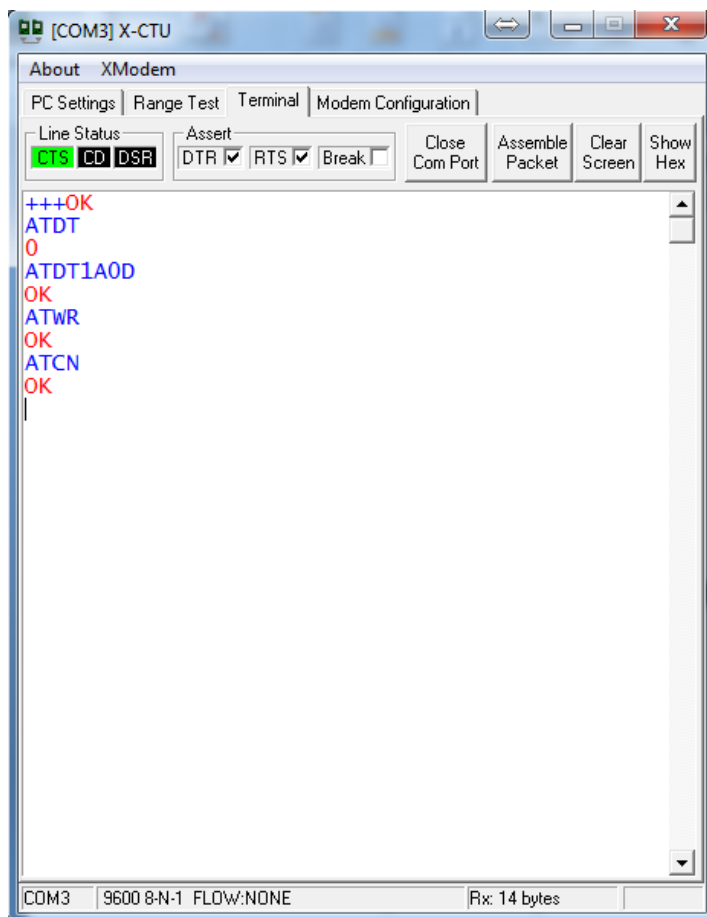


obr. 6: X-CTU Range Test

Program vyšle řetězec znaků o určité velikosti do vzdáleného modulu, kterému jsme předem propojili RX a TX piny. Típnutím se řetězec opět vrátí a program může zjistit sílu signálu a zkontrolovat správnost příjmu.

4.5.4 Terminál

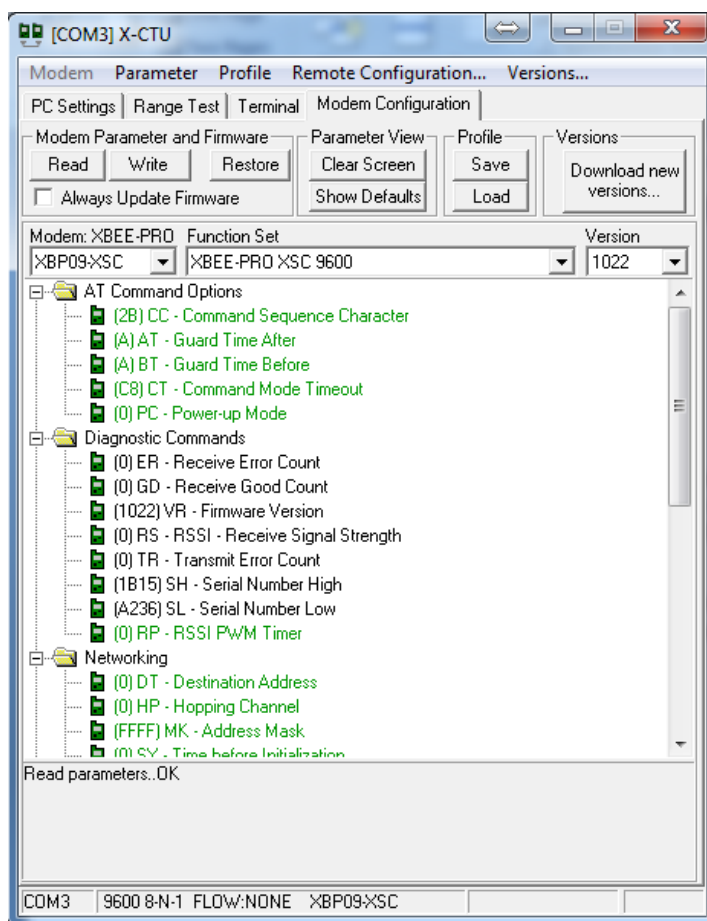
Dále máme na výběr záložku Terminal. Je to jednoduché terminálové okno sloužící k přístupu sériové XBee linky. Toho můžeme jak je patrné na dalším obrázku využít například k AT příkazům, nebo třeba ke kontrole vstupních/výstupních dat z modulu.



obr. 7: X-CTU Terminál

4.5.5 Nastavení Modulu

Posledním v řadě je záložka Modem Configuration. Zde můžeme jednoduše číst, nebo konfigurovat jednotlivé parametry nastavení modulu. Obsahuje vše, co lze nastavit i AT příkazy v přehledně uspořádaném stromu s krátkými vysvětlivkami co které nastavení ovládá. Pro seznámení s moduly však doporučuji naučit se pracovat i s AT příkazy.



obr. 8: X-CTU Nastavení modulu

4.5.6 Shrnutí a možné komplikace

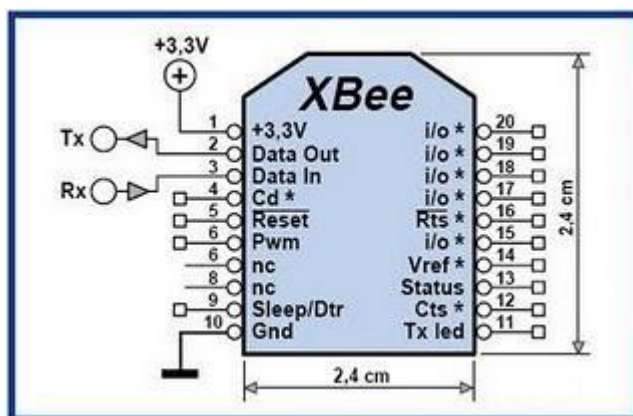
Prací s tímto programem si ušetříte spoustu času a úsilí. Obsahuje totiž vše, co je nutné pro nastavení a komunikaci s XBee a dokonce i pro nahrání nového nebo jiného FW. Přehrávání FW v tomto případě nediskutuji, protože zde použité moduly mají k dispozici pouze jeden typ.

Budete-li však používat například moduly s označením XBee 2, které podporují více verzí FW, musíte dbát na správný výběr. Existují FW, které mají například vypnutou sériovou linku a tím pádem nelze přes ní pak nastavovat modul či vysílat/přijímat data. V takovém případě je pak nutné v záložce Modem Configuration zaškrtnout pole Always Update Firmware a při uploadování několikrát modul resetovat (uzemnit reset pin).

5 Výsledky

5.1 Zapojení

Zapojení Arduina a XBee modulu je teoreticky velice jednoduché. Stačí přivést napájení a zrcadlově propojit RX a TX piny. V praxi však nastává malý problém a to s úrovněmi napětí.



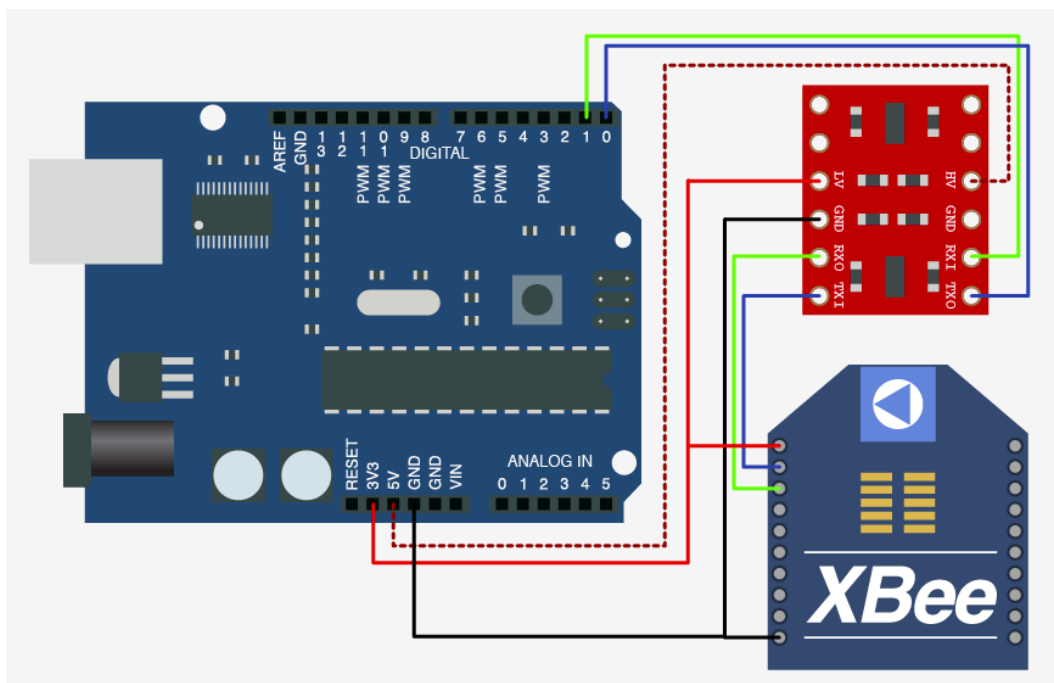
obr. 9: Zapojení XBee [3]

XBee je totiž konstruováno tak, aby pracovalo s 3,3 V napětovými úrovněmi, zatímco Arduino MEGA 2560 pracuje na 5 V. Proto je nutné použít konvertor úrovní. Existuje mnoho tzv. XBee Shieldů-desek plošných spojů, které se nasadí přímo na Arduino a pracují jako konvertor úrovní a obsahují již přímo konektory pro vložení XBee s ne úplně standartní roztečí pinů.

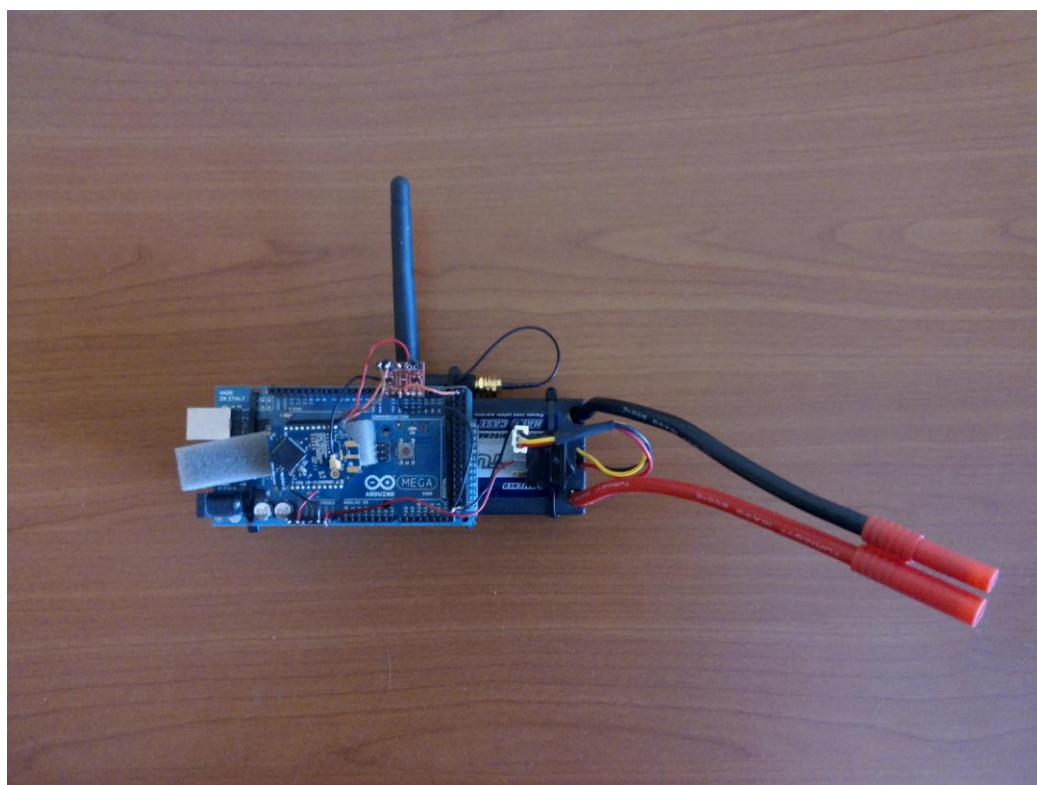
Já jsem však použil pouze úrovnový konvertor, protože nepotřebuji všechny funkce XBee Shieldu a také kvůli jeho ceně. Použité XBee 900 XSC je sice schopné pracovat i s 5 V úrovněmi signálu (je však stále nutné zachovat 3,3 V napájení), nicméně pro univerzálnost tohoto přípravku byly konvertory úrovní použity.

Uvedený konvertor firmy Sparkfun se ukázal jako naprosto dostatečný a funkční. Pro snížení napětí je zde použit napětový dělič a pro zvýšení spínací

tranzistor. Jediným problémem při jeho použití je ne úplně intuitivní označení vstupů a výstupů a proto jsem byl nucen vyhledat blokové schéma zapojení níže.



obr. 10: Blokové schéma propojení všech komponent[4]



obr. 11: Hotový přípravek

5.2 Měření dosahu

Abychom měli reálnou představu, kde všude se dají tyto moduly použít, provedl jsem test dosahu.

Kritéria testu:

- Porovnat sílu signálu vůči vzdálenosti přijímače od vysílače
- Zjistit počet přenesených paket
- Různé terénní podmínky
- Různé druhy antén

Protože jedno z kritérií testu je vyzkoušet různé druhy antén, není proto úplně vyhovující použít Range Test z programu X-CTU, jelikož ten data odešle na vzdálený „router“ a zase přijímá zpátky. Jsou to tedy dvě cesty mezi anténami.

Pro účely tohoto testu jsem sestavil jednoduchý vysílač, který opakovaně odesílá: „TeSt“+“x“, kde x představuje počet takto odeslaných zpráv, abychom měli kontrolu, že se nejedná o předchozí zprávu.

```
int x = 0;
void setup() {
  Serial.begin(9600);
}

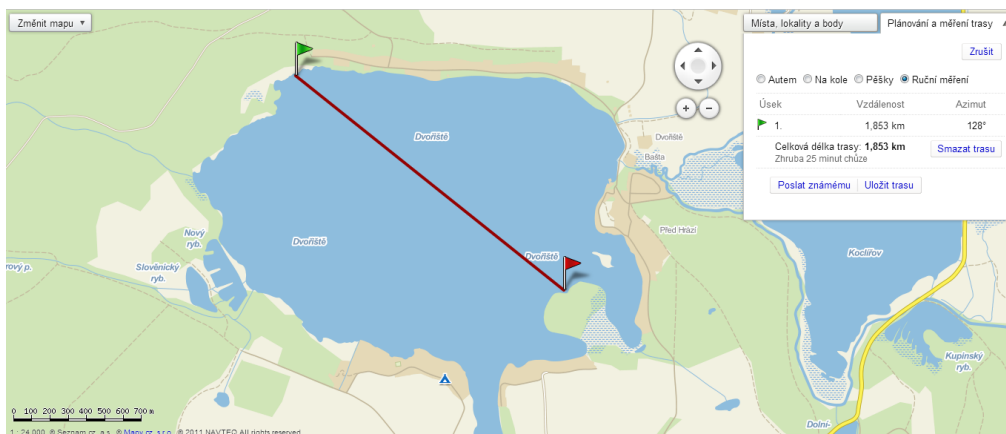
void loop() {
  Serial.print(„TeSt“);
  Serial.print(x);
  delay(500);
}
```

Jako přijímač jsem použil XBee připojené k počítači přes XBee Explorer Dongle. Je to FTDI převodník s konvertorem úrovní na 3,3 V (pro XBee).

K měření vzdálenosti byla použita GPS v mobilním telefonu.

5.2.1 Test č.1

- Test na přímou viditelnost.
- Lokalita: rybník Dvořiště poblíž obce Smržov okres J.Hradec
- Použité antény: 900MHz Duck Antenna RP-SMA na vysílači i přijímači
- Vzdálenost 1853m
- RSSI: 2F
- Přenesené pakety: 100 %

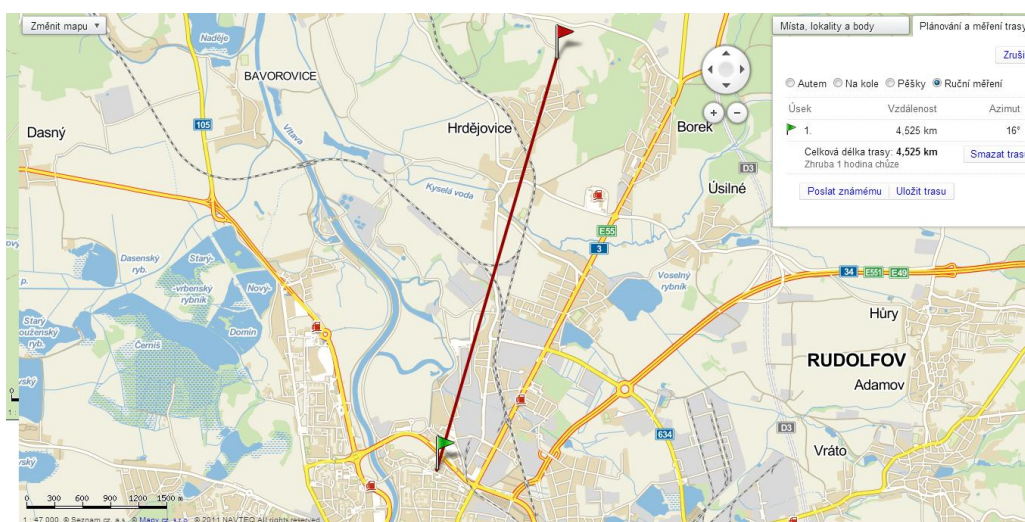


obr.12: Mapa testu č.1

Toto byl první test, kde jsem si chtěl ověřit funkčnost přípravků a jejich správného zapojení. Vysílač byl umístěn na břehu ve vypuštěném rybníku ve výšce 160 cm nad zemí na stativu a přijímač přibližně ve stejné výšce v ruce. Po celou délku trasy nebyla ztracena jediná zpráva a na konečném stanovišti byla síla signálu 2F.

5.2.2 Test č.2

- Test na přímou viditelnost.
- Lokalita: České Budějovice-Hosín
- Použité antény: 900MHz Duck Antenna RP-SMA na vysílači i přijímači a poté Yaggi anténa na GSM pásmo (900-1800 MHz)
- Vzdálenost 4525 m
- RSSI: 1C
- Přenesené pakety: 72 % a 15 %



obr.13: Mapa testu č.2

Tento test byl velice zajímavý. Vysílač byl umístěn na střechu šestipatrového panelového domu v místě mého bydliště a přijímač byl na přímou viditelnost umístěn ve vzdálenosti 4525 m opět v ruce. Nejprve byl proveden test s „pendrekovou“ anténou. Výsledkem je ještě celkem uspokojivých 72 % přenesených paket a síla signálu 1C.

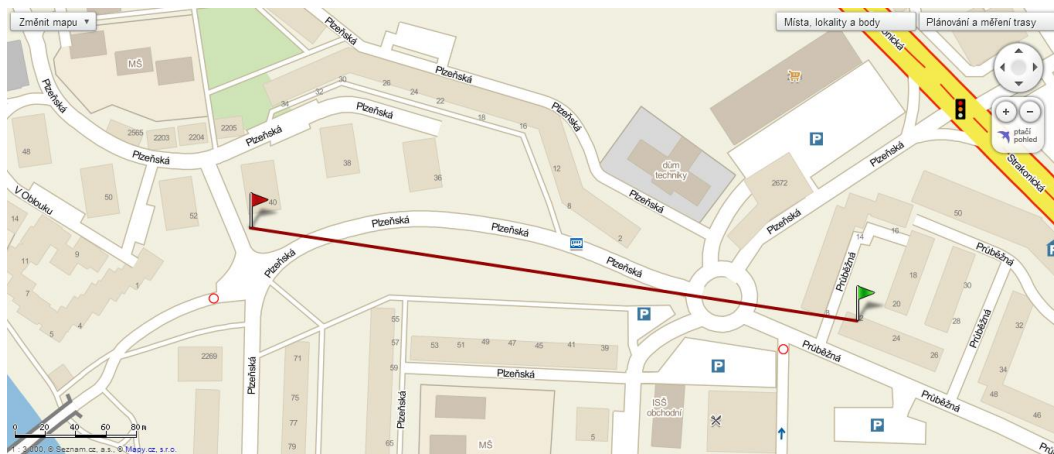
Po výměně antény na přijímači za Yaggi byl však výsledek opravdu špatný. Anténa byla umístěna na stativ, aby se dala snadno a přesně namířit na

vysílač, protože je hodně směrová. Při namíření přesně na vysílač jsem však nechytil žádný signál. Po chvíli zkoušení jsem našel pozici (cca 8° vpravo od vysílače a 10° nahoru), kde byla síla signálu nejsilnější a i přesto bylo přijato pouze 15% vyslaných zpráv.

Tento výsledek přičítám nízké kvalitě Yaggi antény, která byla způsobena zřejmě již ve výrobě.

5.2.4 Test č.3

- Test v zástavbě.
- Lokalita: České Budějovice
- Použité antény: 900 MHz Duck Antenna RP-SMA na vysílači i přijímači
- Vzdálenost 397 m



obr.14: Mapa testu č.3

Vysílač opět na střeše panelového domu a přijímač v ruce. V tomto testu je jasně vidět, že signál se v městské zástavbě rychle ztrácí. V mapě je označen bod naprosté ztráty signálu.

5.2.5 Test č.4

- Test v zástavbě (panelový dům).
- Lokalita: České Budějovice
- Použité antény: 900 MHz Duck Antenna RP-SMA na vysílači i přijímači
- Přenesené pakety: 48 %

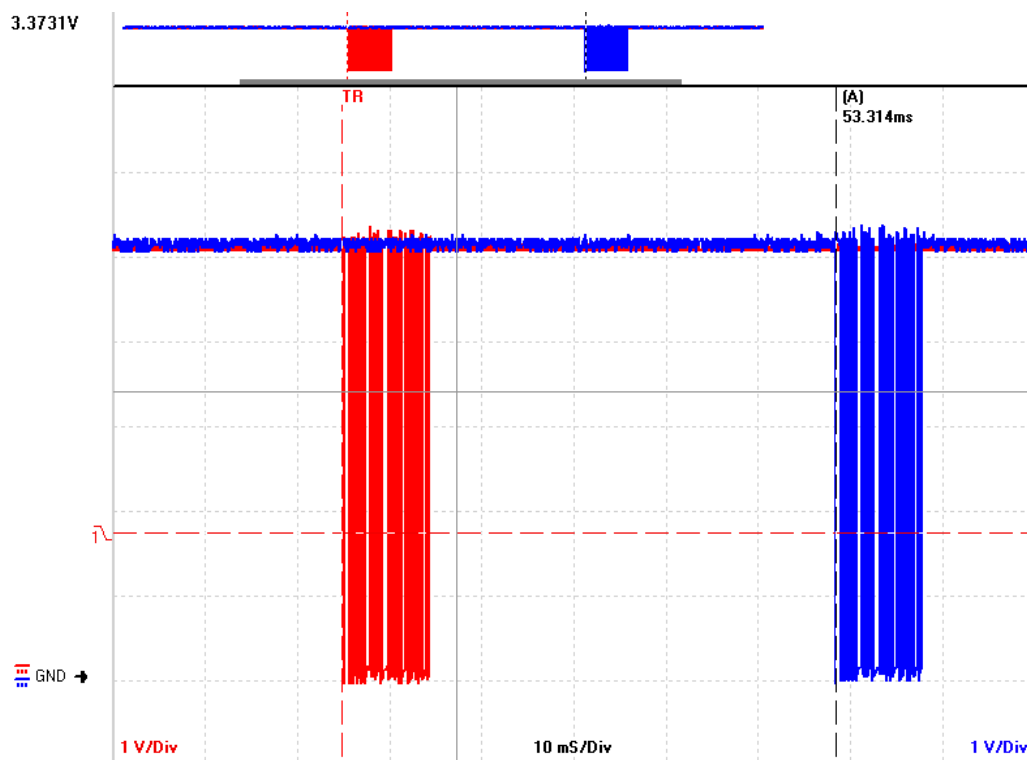
V tomto testu jsem zkoušel prostupnost signálu železobetonovými zdmi panelového domu. Vysílač umístěn v bytě v 5. patře a přijímač ve výtahu s dvojitými kovovými dveřmi a kovovou kabinou ve 2. patře. Signál byl i přes tolik vrstev překážek až překvapivě silný. Při měření ve výtahu v prvním patře jsem už ale nezachytil žádný signál.

5.3 Měření rychlosti přenosu

Při použití modulů pro některé komunikace s kriticky důležitým časováním je důležité znát dobu, za kterou jsou moduly schopny přenést data.

Pro změření rychlosti přenosu byly použity antény 900MHz Duck Antenna RP-SMA na vysílači i přijímači. Vysílač odesílal opakovaně zprávu „TeSt“+“x“ s rychlostí 9600 baudů.

Přijímač byl připojen k logickému analyzátoru. Výsledkem je tato zachycená obrazovka (viz obr.15).



obr.15: Výstupní graf logického analyzátoru

Z obrázku vyplývá, že doba mezi prvním odesláním a prvním přijutím vychází na 53,314 ms. Tato doba je součtem doby samotného přenosu signálu přes vzdálenost z bodu A do bodu B, která při tomto testu byla přibližně půl metru a jelikož radiové vlny se šíří rychlostí světla ve vakuu, tak tuto hodnotu můžeme klidně zanedbat. Druhá složka této doby je doba nutná ke zpracování samotného signálu XBee moduly, které musí kontrolovat při každém příjmu i vyslání zprávy síťové adresy a kontrolního součtu dat.

Dále je také nutno uvést, že při prvním spuštění je potřeba vyčkat než se jednotlivé moduly spárují. Tato doba závisí na topologii sítě a počtu modulů. V případě modulů XBee 900 XSC je tato doba přibližně 20 vteřin. Je-li však síť již vytvořena, vstup dalšího modulu do sítě je téměř okamžitý.

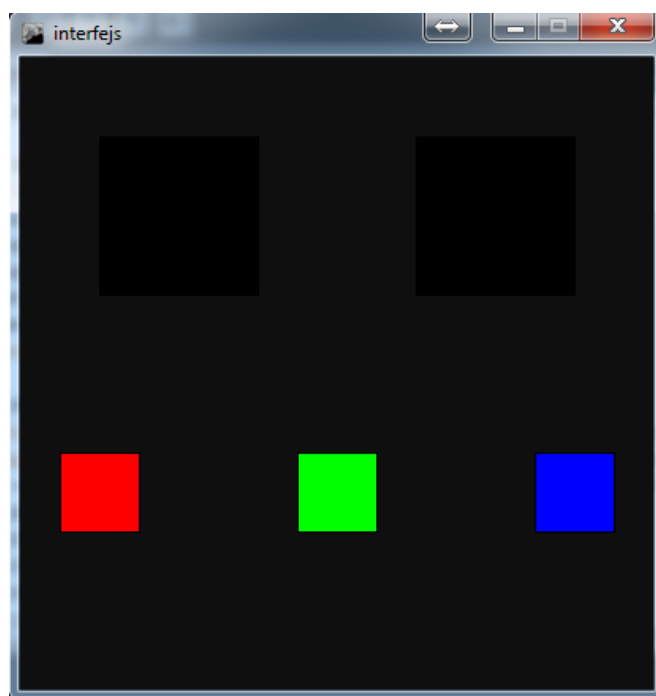
6 Možné využití Arduino + XBee

Využití této kombinace modulů je nespočet. Jistě si dokážete představit použití jako např. senzorové sítě ve velkých budovách pro monitoring nebo řízení teplot, detekce přítomnosti osob nebo třeba ovládání prezentačních místností- ovládání žaluzií, projektoru, ozvučení, osvětlení aj.

Záleží pouze na požadavcích a dostupných senzorech případně ovládacích prvcích.

Pro představu funkčnosti jsem vytvořil tento demonstrační přípravek. Skládá se ze dvou Arduino+XBee modulů připojených k počítači, na kterém běží obslužný program, který simuluje přenos analogových i digitálních veličin pomocí Zigbee protokolu.

Program je velice jednoduchý k použití. Po zapnutí se zobrazí toto okno:



obr.16: Obslužný program

Po najetí myši na horní levý čtverec se na vzdáleném modulu rozsvítí žlutá dioda LED. Naopak po najetí na pravý čtverec dioda zhasne. Stav je indikován barvou horních čtverců. Tímto simulujeme přenos digitální veličiny (1/0 nebo On/Off).

Dolní čtverce slouží k nastavení barvy pozadí vzdáleného programu. Po najetí na čtverec příslušné barvy se změní i barva pozadí. Toto je simulace přenosu tří analogových veličin v rozmezí od 0 až po 255 v RGB formátu.

Nyní si ukážeme funkce programu na jednotlivých částech kódu a popíšeme jejich funkce jak v části pro PC tak i v části pro Arduino.

6.1 Obslužný program do PC

6.1.1 Nastavení sériového portu

První velice důležitá věc v programu je inicializace komunikace mezi PC a Arduinem. Na PC je však možné mít k dispozici více portů COM a proto si je musíme nejprve všechny nechat vypsat. K tomu slouží funkce `Serial.list()`.

Dále nastavíme příslušné číslo portu (v tomto případě port 2) a spojení je zajištěno.

```
println(Serial.list());  
String portName = Serial.list()[2]
```

6.1.2 Grafická část

V dalším kroku provedeme vykreslení všech grafických objektů v okně a nastavení jejich barev. Nejprve začneme příkazem `background()` s parametry udávajícími barvu pozadí. Poté použijeme příkaz na nastavení barvy a vykreslení čtverce na zadaných XY souřadnicích a zadané velikosti.

```
fill(0); //černá  
rect(50, 50, 100, 100); // XY 50,50 a velikost 100x100
```

6.1.3 Detekce pozice kurzoru

Pro zjištění, zda se kurzor nachází nad požadovaným grafickým prvkem použijeme funkci, která kontroluje jestli je kurzor v požadovaném rozmezí XY souřadnic.

```
boolean mouseOverRect1()  
    return ((mouseX >= 50) && (mouseX <= 150) &&  
(mouseY >= 50) && (mouseY <= 150));  
}
```

Tato funkce vrátí `true` v případě, že se kurzor nachází v požadovaném rozmezí. V opačném případě samozřejmě vrátí `false`.

6.1.4 Odeslání dat

K odeslání dat využijeme funkce `port.write()`. Parametrem této funkce jsou přímo data určená k odeslání. Program používá pevně daného formátu odesílaných dat.

```
port.write(ID);  
port.write(hodnotaA);  
port.write(hodnotaB);  
port.write(hodnotaC);  
port.write(kontrola1);  
port.write(kontrola2);
```

Prvním odeslaným bytem je takzvané ID. To nám určuje obsah odesílaných dat. Tento program pro jednoduchost používá pouze dvě:

- ID = 1: pro nastavení stavu LED diody
- ID = 2: pro nastavení barvy pozadí

V praxi však není problém takto nastavený protokol rozšířit až na 255 možných ID. V případě, že je potřeba ID víc, lze odesílat další ID byty a tím zvětšit počet možností.

Druhou částí odeslaného paketu jsou tři byty, které obsahují data. V případě dat o stavu LED je použit pouze první byte a ostatní se rovnají nule. Odesíláme-li data o barvě pozadí, obsahují všechny tři údaje o RGB barevných složkách.

Jako poslední část jsou dva kontrolní byty. Ty obsahují numerický součet všech předchozích údajů. Stane-li se, že přijmeme neúplný nebo chybný paket, kontrolní součet pak nebude souhlasit. V tom případě přijatá data dále nezpracováváme, aby nedošlo k nečekanému stavu.

6.1.5 Příjem dat

Ve stejném formátu je koncipován i příjem dat z Arduina. Nejprve pomocí `port.read()` přečteme všechna příchozí data. Po ověření kontrolního součtu vyhodnotíme ID a uložíme do proměnných příslušná přijatá data.

6.2 Nyní program do Arduino

6.2.1 Setup

Ve funkci `Setup()`, která proběhne pouze při startu programu, provedeme inicializaci sériové linky a nastavení pinu 13 jako digitálního výstupu pro diodu LED.

```
void setup() {  
  Serial.begin(9600);           //inicializace linky k PC  
  Serial3.begin(9600);         //inicializace linky k XBee  
  pinMode(13, OUTPUT);        //pin13 výstup  
}
```

Parametrem 9600 pro příkaz `Serial.begin()` nastavíme přenosovou rychlost.

6.2.2 Loop

Funkce `loop()` běží v programu stále dokola. Zde voláme tři další funkce.

```
void loop() {  
  PctoXBEE();  
  XBEEtoPC();  
  LED();  
}
```

6.2.3 PCtoXBEE

je funkce kontrolující příchozí data z počítače a dále je po zpracování odešle přes XBee. Nejprve vyčká na příchozí data. Provede kontrolu jestli počet přijatých bytů odpovídá nastavení a pak podobně jako v obslužném programu v Processingu ověří kontrolní součet, zjistí ID a uloží data do příslušných proměnných. Čtení příchozích dat je zde zajištěno příkazem `Serial.read()`.

Po přijetí dat z počítače provede kontrolu změny. V případě, že jsou přijatá data shodná s předchozími, vrátí se k čekání na příjem. Jsou-li ale přijatá data rozdílná od předchozího příjmu, odešle je pomocí příkazu `Serial3.print()` do XBee. Po odeslání uloží data pro opětovnou pozdější kontrolu příjmu nových dat.

6.2.4 XBEEtoPC

pracuje velice podobně s tím rozdílem, že kontroluje příchozí data z XBee modulu a po kontrole a zpracování je odesílá přes USB do obslužného programu v PC.

6.2.5 LED

Posledním prvkem tohoto programu je funkce `LED()`, která zajišťuje rozsvícení diody LED v případě příjmu správného příkazu. O to se stará příkaz `digitalWrite()` se dvěma parametry. Prvním je číslo pinu, který chceme ovládat a druhým je stav do kterého chceme pin přepnout. V tomto případě příkaz vypadá takto:

```
digitalWrite(13, stavLED);
```

7 Závěr

V úvodu mé bakalářské práce je uveden její cíl: „Navrhnout a sestrojít demonstrační přípravek pro moduly XBee a procesory Atmel-AVR“.

Jsem přesvědčen, že zadaný cíl jsem naplnil. Na základě předem připraveného projektu jsem sestavil demonstrační přípravek splňující veškeré zadané požadavky na jeho funkce.

Po napsání programů do platformy Arduino a také pro PC jsem přistoupil k „oživení“ přípravku. Toto oživení probíhalo v podstatě očekávaným způsobem. Software jsem postupně v několika krocích upravoval a ladil tak, abych dosáhl plně funkčního systému.

Po zkouškách dosahu v různém terénu a měřeních s přípravkem nebylo nutné přikročit k žádným zásadním úpravám či změnám.

Výsledkem provedených kontrolních měření jsou data, která dokazují spolehlivost tohoto přípravku a předpoklad možnosti širokého využití od řídicích systémů pro tzv. inteligentní budovy, které ovládají osvětlení, vytápění, sluneční clony, bezpečnostní prvky až po detekci přítomnosti osob v daném prostoru.

Jednou z dalších možností využití, kterou dále i hodlám realizovat je návrh a sestrojení systému pro dálkové ovládání pyrotechniky. Hlavní výhodou bude stabilita signálu, zabezpečení linky proti náhodnému (či cílenému) nežádoucímu odpálení roznětky a dále možnost odesílat zpět do ovládacího prvku stav rozbušek případně zbývající kapacitu akumulátoru. Výhodou bude také možnost topologie sítě „point to multipoint“, kdy jedním ovladačem bude možné řídit odpálení roznětek u více koncových zařízení.

Seznam použité literatury:

[1] http://arduino.cc/en/uploads/Main/ArduinoMega2560_R3_Front_450px.jpg

[2] <http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-XCS-Manual.pdf>
str.17

[3] http://www.pyroelectro.com/tutorials/xbee_pan_tilt_servo/img/xbee_pinout.jpg

[4] http://bildr.org/blog/wp-content/uploads/2011/04/5v_arduino_xbee1.png