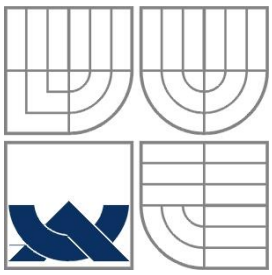
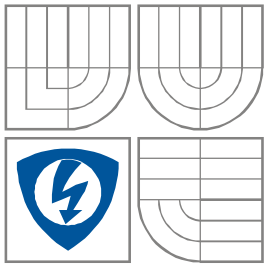


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

SOFTWARE PRO ANALÝZU DÁLKOVÉHO ODEČTU MĚŘICÍCH ZAŘÍZENÍ

SOFTWARE FOR THE ANALYSIS OF THE REMOTE-READING OF THE METERS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. Radek Přívětivý
AUTHOR

VEDOUCÍ PRÁCE Ing. Roman Mego
SUPERVISOR

BRNO, 2015

ABSTRAKT

V první části diplomové práce jsou popsány principy dálkového odečtu měřičů v domácnostech a jednotlivé komponenty inteligentních sítí pro odečet těchto měřidel. Další část práce obsahuje popis standardu Wireless M-Bus a ZigBee. V následující kapitole je popis obvodu Realtek RTL2832U, který je součástí zvoleného přijímače. Další část práce je věnována základnímu popisu modulací se zaměřením na úhlové modulace – FM, FSK. V závěrečné části je detailní popis přijímaného signálu, šifrování AES a vytvořeného softwaru.

KLÍČOVÁ SLOVA

SDR, dálkový odečet měřidel, Wireless M-Bus, RTL2832U, AES šifrování

ABSTRACT

There are describes the principles of remote meter reading in the home and components of smart grids for reading these meters in the first part of master's thesis. The next part of the thesis contains description of standard Wireless M-Bus and ZigBee. The next chapter is description of the circuit Realtek RTL2832U, which is part of the selected tuner. Another part is devoted to basic description of modulation focusing on the angle modulation – FM, FSK. In the final part is a detailed description of the received signal. AES encryption and created software.

KEYWORDS

SDR, remote-reading of the meters, Wireless M-Bus, RTL2832U, AES encryption

PŘÍVĚTIVÝ, R. *Software pro analýzu dálkového odečtu měřících zařízení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2015. 38 s., 2 s. příloh. Diplomová práce. Vedoucí práce: Ing. Roman Mego.

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Software pro analýzu dálkového odečtu měřících zařízení jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Romanu Megovi za účinnou metodickou, pedagogickou a odbornou pomoc při zpracování mé diplomové práce.

OBSAH

Seznam obrázků **vii**

Seznam tabulek **viii**

Úvod **1**

1 PRINCIP DÁLKOVÉHO ODEČTU BYTOVÝCH MĚŘIDEL **2**

1.1 Jednotlivé komponenty inteligentních sítí 2

1.1.1 Vodoměry, plynoměry, kalorimetry 3

1.1.2 Elektroměry 3

1.1.3 Data koncentrátory 3

1.1.4 Servery 4

1.1.5 Monitory, televizory, mobilní telefony atd. 4

2 POPIS NEJPOUŽÍVANĚJŠÍCH STANDARDŮ **5**

2.1 Wireless M-Bus 5

2.1.1 Režimy rádiového přenosu 5

2.1.2 Princip komunikace 6

2.1.3 Formát paketů 7

2.2 ZigBee 7

2.2.1 Struktura komunikačního standardu 8

2.2.2 Topologie sítě 9

3 DVB-T TUNER S DAB+, FM A SDR S TUNEREM R820T **10**

3.1 RTL2832U 11

3.1.1 Software pro RTL2832U 11

3.2 R820T 11

3.3 Softwarově definované rádio 12

4 MODULACE **14**

4.1 Úhlové modulace 15

4.1.1 Kmitočtová modulace – FM 16

4.2 Binární FSK – 2FSK (BFSK) 17

4.3 I/Q data 17

5 PŘIJÍMANÝ SIGNÁL **19**

5.1 Formát paketu 19
5.1.1 Linková vrstva 20
5.1.2 Aplikační vrstva 21
6 ŠIFROVÁNÍ AES **24**
6.1 Šifrování 24
6.2 Dešifrování 25
7 SOFTWARE PRO PŘÍJEM A ZPRACOVÁNÍ DAT **27**
7.1 Popis Softwaru 27
7.1.1 Inicializace 27
7.1.2 Spuštění vlákna pro příjem dat 28
7.1.3 Hlavní smyčka 29
7.2 Změna komunikačního protokolu 32
8 GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ (GUI) **33**
8.1 Analýza funkcí softwaru 33
8.2 Popis GUI 33
ZÁVĚR **36**
A PŘIJATÁ DEKÓDOVANÁ A DEŠIFROVANÁ DATA **39**
B ZOBRAZENÍ PŘIJATÝCH DAT **40**

SEZNAM OBRÁZKŮ

Obrázek 1: Komponenty inteligentních sítí [2]. 2
Obrázek 2: Data koncentrátor od firmy ModemTec [4].....3
Obrázek 3: Možnosti využití sběrnice Wireless M-Bus a M-Bus [6]. 5
Obrázek 4: Struktura komunikace. 6
Obrázek 5: Struktura zprávy nadřazené jednotky [6].....7
Obrázek 6: Kompletní podoba zprávy [6]. 7
Obrázek 7: Zjednodušená forma paketu [6]. 7
Obrázek 8: OSI model ZigBee [7]. 8
Obrázek 9: Topologie sítě ZigBee typu a) hvězda, b) strom a c) síť [8].....9
Obrázek 10: Ukázka použitého tuneru [9].....10
Obrázek 11: Blokové schéma přijímače [10]. 10
Obrázek 12: Obecné schéma ideálního SR [13]. 12
Obrázek 13: Schéma SDR [13].....13
Obrázek 14: Průběhy základních typů digitálních modulací [14].....14
Obrázek 15: Základní typy modulací [14].....15
Obrázek 16: Vztah modulací FM a PM při modulaci harmonickým signálem [14].....16
Obrázek 17: Modulační signál FSK a průběh hodnoty fáze [14]. 17
Obrázek 18: Polární reprezentace sinusové vlny, kde A značí amplitudu a ϕ fázi [15]..18
Obrázek 19: I/Q zobrazení v polární formě, kde A značí amplitudu a ϕ fázi [15]. 18
Obrázek 20: Blokové schéma softwaru. 27
Obrázek 21: Blokový diagram hlavní smyčky.....29
Obrázek 22: Grafický průběh prvních 36 zakódovaných bitů. 30
Obrázek 23: Grafické uživatelské rozhraní. 34
Obrázek 24: Blokové diagramy pro výpis dat. 35

SEZNAM TABULEK

Tabulka 1: Tabulka režimů rádiového přenosu [6]. 6
Tabulka 2: Základní parametry přijímaného signálu. 19
Tabulka 3: První blok vysílaného paketu. 19
Tabulka 4: Druhý blok vysílaného paketu. 20
Tabulka 5: Třetí blok vysílaného paketu. 20
Tabulka 6: Aplikační vrstva. 21
Tabulka 7: Data header pro CI = $7A_h$ [17]. 21
Tabulka 8: Obsah pole DIF [17]. 22
Tabulka 9: Obsah pole VIF [17]. 22
Tabulka 10: Tabulka vybraných jednotek a násobitelů přenášených hodnot [17]. 23
Tabulka 11: Tabulka kódování 3 z 6 [16]. 31

ÚVOD

Dálkový odečet bytových měřidel v současné době představuje velký pokrok pro domácnosti a společnosti. Princip je založen na sbírání informací z měřidel do data koncentrátoru, odkud jsou data přenášena na server.

Domácnostem to přináší výhodu zejména v tom, že jim odpadá povinnost zpřístupnit byt (soukromý pozemek) pro pracovníky k odečtu stavu měřidel. Dále mohou uživatelé díky statistikám vyhodnocovat svou spotřebu, a tím se například vyhnout zbytečné spotřebě. Statistiky mohou také spotřebitele upozornit na různé havárie nebo poruchy, jako je například prasklé potrubí.

Firmám tato technologie ušetří náklady spojené s cestováním pracovníků pro odečet stavu měřidel přímo v konkrétních objektech. Společnosti mohou také lépe vyhodnocovat aktuální zatížení sítě.

Cílem této diplomové práce je realizovat software pro dálkový odečet stavu bytových měřidel. Přijímačem pro tyto informace bude DVB-T přijímač do osobního počítače připojitelný přes USB port s obvodem Realtek RTL2832U. Pro testování je jako vysílač použit Double dongle RF M-bus & ZigBee od firmy ModemTec. Vysílač vysílá paket každých 500 ms. V paketu se inkrementuje počítadlo spotřeby vody s každou vyslanou zprávou o 1 v rozsahu od 0 (zapnutí) do 0x7FFFFFFF a poté znova od 0.

První část práce obsahuje obecný popis dálkového odečtu bytových měřidel s informacemi o jednotlivých komponentách inteligentních sítí. Jsou zde také rozebrány výhody dálkového odečtu. Druhá kapitola je věnována popisu nejpoužívanějších standardů – Wireless M-Bus a ZigBee. V další části diplomové práce jsou informace o použitém přijímači a modulacích se zaměřením na úhlové – FM a FSK. V této části je také zahrnut popis I/Q dat. Poté je dopodrobna rozebrán přijímaný signál a šifrování AES, kterým jsou data šifrovaná. Konec práce je věnován vytvořenému softwaru.

1 PRINCIP DÁLKOVÉHO ODEČTU BYTOVÝCH MĚŘIDEL

V bytových domech se stále častěji uplatňují měřiče energií, které umožňují dálkový odečet naměřených hodnot. Díky této technologii je možné sledovat spotřebu jednotlivých domácností, aniž by byla potřeba zpřístupnit byt pro odečet. Měřiče a indikátory s možností dálkového odečtu výrazně usnadňují práci společnostem a majitelům bytů. Technik má možnost hodnotu odečítat během pochůzky, případně z projíždějícího dopravního prostředku. Uživatelům je také umožněno sledovat spotřebu jednotlivých bytů na webu. Domácnosti se tak mohou informovat o aktuální spotřebě a v případě potřeby svoji spotřebu optimalizovat.

V současné době se z důvodu nárůstu dálkově odečítaných bytových měřidel rozšířila mezi dodavateli komunikace pomocí standardu Wierless M-Bus. Rozšíření standardu umožňuje, aby zákazník nebyl vázán na produkty jednoho dodavatele, ale aby mohl kombinovat v jednom systému zařízení od různých dodavatelů s možností bezdrátové komunikace.[1]

1.1 Jednotlivé komponenty inteligentních sítí

Mezi základní komponenty inteligentních sítí patří: bytová měřidla neelektrických veličin, elektroměry, data koncentrátory, servery a zařízení, která informují uživatele o aktuální spotřebě, využívaného tarifu atd. Jednotlivé části inteligentních sítí jsou zobrazeny na obrázku 1.

Obrázek 1: Komponenty inteligentních sítí [2].

1.1.1 Vodoměry, plynoměry, kalorimetry

Vodoměry, plynoměry a kalorimetry komunikují s elektroměrem pomocí technologie Wierless M-BUS, ZigBee. Data se přenáší ve směru z měřidel do elektroměru. Pokud se jedná o pokročilejší měřidla, může probíhat i komunikace ve směru z elektroměru do měřidel [2].

1.1.2 Elektroměry

Primárně jsou elektroměry určeny k měření spotřeby elektrické energie, napětí v místě elektroměru a zaznamenávání dalších údajů souvisejících s danou přípojkou. Elektroměry mohou provádět změnu tarifu, spínání relé, odpojení zákazníka od sítě atd. Mezi další jejich funkci patří přijímání data z přiřazených bytových měřidel. Získané údaje spolu s vlastními daty mohou předávat do data koncentrátoru. Přenos může probíhat v rámci PLC komunikace. PLC modem je buď součástí příslušného elektroměru, nebo je doplněn. Mezi další způsoby komunikace patří GPRS, rádiový přenos, Ethernet, RS-485 apod [2].

1.1.3 Data koncentrátory

Data koncentrátory plní funkci brány mezi přenosem dat po rádiové nebo elektrické síti a jiným přenosem, nejčastěji typu TCP/IP. Data koncentrátor se nejčastěji umisťuje do trafostanice nebo společných prostorů bytového domu. Jeden data koncentrátor obsluhuje většinou přibližně 100 elektroměrů. Pokud se jedná o husté zástavby, může být počet obsluhovaných elektroměrů kolem 1000. Data koncentrátor se serverem komunikuje prostřednictvím Ethernetu, případně pomocí GPRS. Na obrázku 2 je data koncentrátor od firmy ModemTec [2].

Obrázek 2: Data koncentrátor od firmy ModemTec [4].

1.1.4 Servery

Servery představují poslední část řetězce. Zpracovávají data z data koncentrátoru a všechny nebo jen část dat poskytují klientským stanicím. Pracovníci v operátorském centru mohou přes servery posílat příkazy jednotlivým zařízením [2].

1.1.5 Monitory, televizory, mobilní telefony atd.

V této skupině jsou zařízení, která informují spotřebitele o aktuální spotřebě energie, ceně za spotřebovávanou energii, využívaném tarifu či jiné dostupné informaci. Zákazník tak má přehled o svých využívaných službách a může uzpůsobit spotřebu energie aktuální situaci [2].

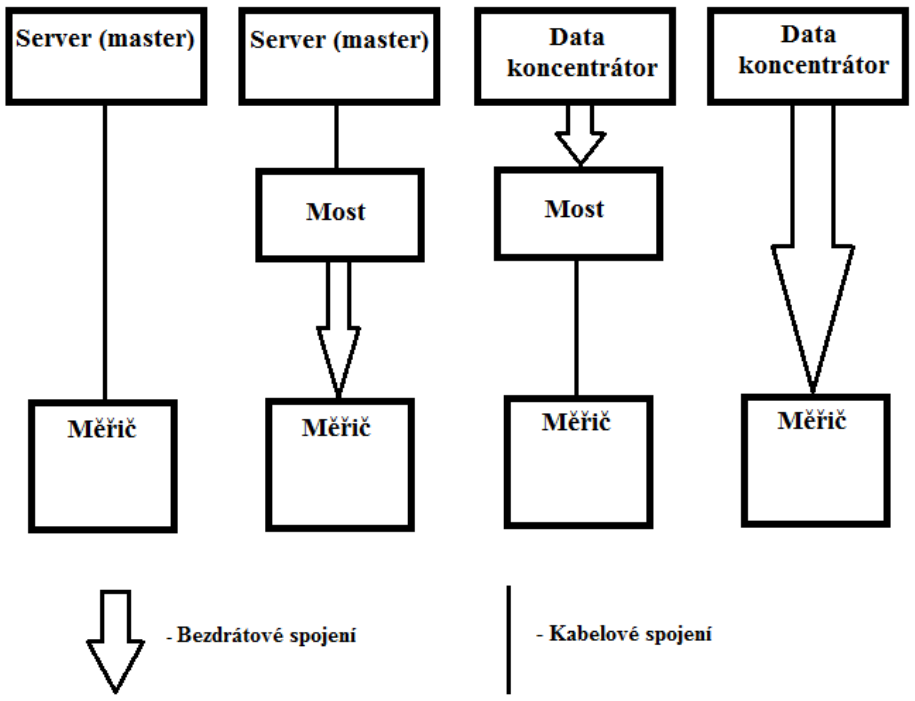
2 POPIS NEJPOUŽÍVANĚJŠÍCH STANDARDŮ

Mezi nejpoužívanější protokoly pro přenos dat v inteligentních sítích patří – Wireless M-Bus a ZigBee.

2.1 Wireless M-Bus

Počátky této bezdrátové komunikace jsou v drátové variantě M-Bus. Původně byl standard zaměřen pouze na provoz v pásmu 868 MHz, což umožňovalo dobrý kompromis mezi RF dosahem a velikostí antény. Později byly přidány další dvě pásma na frekvenci 169 MHz a 433 MHz [5].

Je možné kombinovat drátovou sběrnici M-Bus s bezdrátovou verzí prostřednictvím mostů (bridge). To je zobrazeno na obrázku 3 [6].



Obrázek 3: Možnosti využití sběrnice Wireless M-Bus a M-Bus [6].

2.1.1 Režimy rádiového přenosu

Existují 3 různé režimy rádiového přenosu, které se liší svoji přenosovou rychlostí. Označují se S, T a R. Každý režim se dělí na režim 1 a 2, což značí směr přenosu (jednosměrný nebo obousměrný). Režimy jsou popsány v tabulce 1 [6].

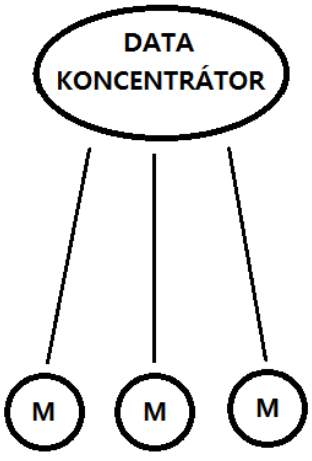
Tabulka 1: Tabulka režimů rádiového přenosu [6].

Přenosová rychlost	Označení jednocestné komunikace	Označení dvoucestné komunikace
4,8 kb/s	Neexistuje	R2
32,768 kb/s	S1/S1m	S2
100 kb/s	T1	T2

Komunikace v režimu T1 je vhodná k přenosu dat z bytových měřičů. Obousměrná komunikace T2 umožňuje mimo čtení stavů měřičů i například zpětné ovládání. Režim T1/T2 byl určený převážně pro systémy s častým přenosem, kde se data přenášejí několikrát za hodinu (případně častěji), nebo kde se přenáší nárazově větší objem dat. Větší přenosová rychlost se vyznačuje kratším komunikačním časem a menší spotřebou elektrické energie, proto je komunikační režim T vhodný pro měřiče napájené baterií. Pro systémy s méně častým přenosem lze použít pomalejší režim. Režim S dobře slouží v systémech, kde stačí přenášet malý objem dat například jednou denně. Režim R2 je zejména vhodný pro případy, kde se přenáší jen velmi malé množství dat na větší vzdálenost. U režimu R2 lze ručně zvolit 1 z 10 komunikačních kanálů, u režimu S2 a T2 se kanál volí automaticky [6].

2.1.2 Princip komunikace

Komunikace má hvězdicovitou strukturu, takže několik měřičů/snímačů přenáší svá data do datového koncentrátoru. Datový koncentrátor pracuje jako server (master) a stále jen naslouchá a čeká na navázání komunikace měřící jednotkou, která pracuje jako klient (slave). Pokud je nastavená obousměrná komunikace, přechází měřič do přijímacího režimu pouze na krátkou dobu a v tomto okamžiku může koncentrátor vysílat měřící jednotce data. Struktura komunikace je naznačena na obrázku 4 [6].



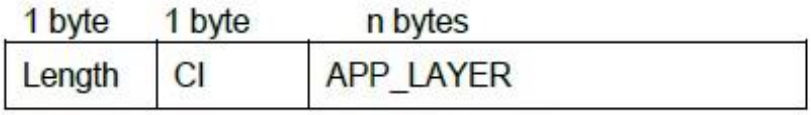
Obrázek 4: Struktura komunikace.

Každý měřič má svoji přidělenou adresu a využívá ji při příjmu i vysílání. Každý datový koncentrátor obsahuje tabulku adres měřičů, se kterými může komunikovat.

Tabulka se vytváří automaticky při registraci jednotky do sítě. Pokud koncentrátor neobsahuje tabulku adres, komunikuje se všemi měřiči v dosahu. [6]

2.1.3 Formát paketů

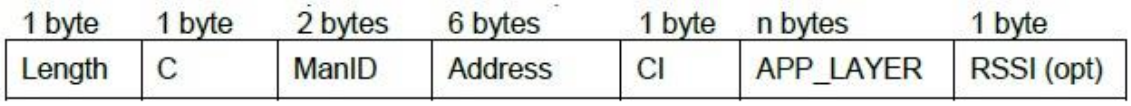
Aplikační vrstvu vytváří nadřazené jednotky (např. koncentrátor), které odesílají zprávy do RF modemu v podobě ukázané na obrázku 5 [6]:



Obrázek 5: Struktura zprávy nadřazené jednotky [6].

Komunikační modul, který pracuje jako modem, přidá automaticky následující pole (viz obrázek 6) :

- řídicí pole C,
- označení výrobce ManID,
- unikátní komunikační adresy,
- ještě se přidá na závěr zprávy informace o síle přijímaného signálu RSSI [6].



Obrázek 6: Kompletní podoba zprávy [6].

Paket se v této podobě zašifruje a může se přenášet. Pokud se provádí pouze komunikace mezi dvěma Wireless M-Bus modemy, je povolen i režim, kdy se nezasílá adresa a informace o měřící jednotce, viz obrázek 7 [6].

Obrázek 7: Zjednodušená forma paketu [6].

2.2 ZigBee

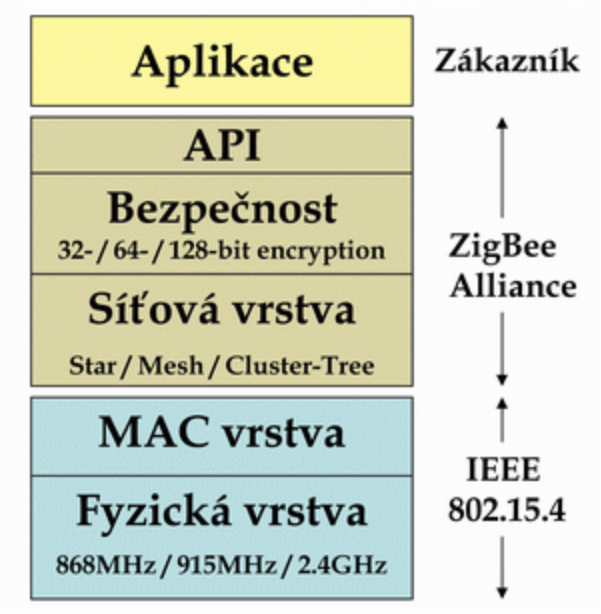
ZigBee je jednoduchý bezdrátový komunikační standard, který díky nízkým nárokům na hardware a nízké spotřebě nachází převážně uplatnění u bateriově napájených senzorů či měřidel. Přenos dat probíhá ve frekvenčním pásmu 2,4 GHz na vzdálenost

stovek metrů. Komunikační technologie ZigBee se snaží vyplnit mezeru mezi WIFI a Blutetooth. V porovnání s Bluetooth poskytuje ZigBee delší dosah komunikace při nižší spotřebě energie. Naproti tomu disponuje nižší přenosovou rychlostí, která pozitivně působí na odolnost proti rušení [7].

2.2.1 Struktura komunikačního standardu

ZigBee se popisuje OSI modelem, který lze rozdělit do základních bloků (viz obrázek 8):

- zákazník – definuje zákaznickou aplikaci v aplikační vrstvě OSI modelu,
- ZigBee Aliance – definuje vyšší vrstvy OSI modelu,
- IEEE 802.15.4 – definuje fyzickou a linkovou vrstvu OSI modelu [7].



Obrázek 8: OSI model ZigBee [7].

IEEE 802.15.4

Standard IEEE 802.15.4 definuje fyzickou a linkovou vrstvu (MAC vrstvu) ZigBee. Fyzická vrstva definuje způsob bezdrátové komunikace, které bylo přiděleno několik rádiových pás. Linková vrstva (MAC vrstva) určuje samotný komunikační protokol mezi jednotlivými zařízeními, který se skládá z rámců [7].

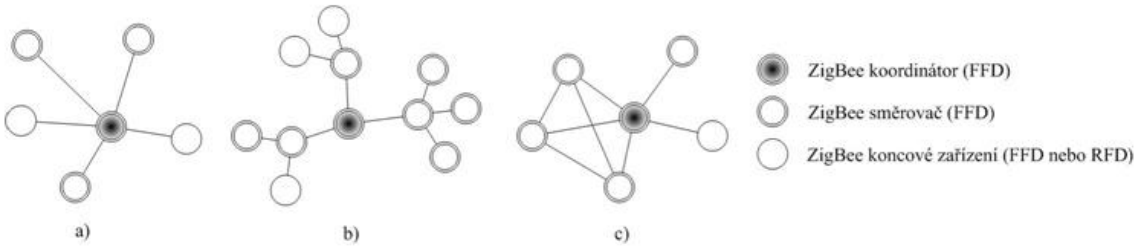
ZigBee Aliance

Nad vrstvami standardu IEEE 802.15.4 jsou definovány 2 vrstvy – síťová vrstva (NWK) a aplikační vrstva (APL). Síťová vrstva provádí připojování a odpojování k síti, zabezpečení a směrování datových rámců. Aplikační vrstva protokolu je složena z pomocné aplikační podvrstvy (APS), objektů ZigBee (ZDO) a uživatelských

aplikačních objektů. Vrstva APS podle poskytovaných služeb a požadavků umožňuje párování zařízení. Objekt ZigBee určuje roli zařízení v síti (např. ZigBee koordinátor) a spravuje poskytované služby [7].

2.2.2 Topologie sítě

Pro adresaci jednotlivých zařízení standard využívá binární adresovací kódy, které jsou dlouhé buď (64 bitů) nebo zkrácené (16 bitů). Každá vytvořená síť je jednoznačně určena pomocí 16bitového identifikátoru PAN ID (Personal Area Network ID), který se používá v případě, že je potřeba rozlišit překrývající se sítě. Z hlediska topologie jsou definovány tři typy sítí, které jsou vidět na obrázku 9. V topologii typu hvězda je řízením pověřen PAN koordinátor a ostatní zařízení pracují jako koncová zařízení [7]. Koncová zařízení komunikují v této topologii přímo s koordinátorem. Dalšími typy jsou síť a strom. V těchto topologiích spouští komunikaci a udává parametry sítě koordinátor. Na obrázku 9 jsou ukázány topologie sítě ZigBee různých typů [8].



Obrázek 9: Topologie sítě ZigBee typu a) hvězda, b) strom a c) síť [8].

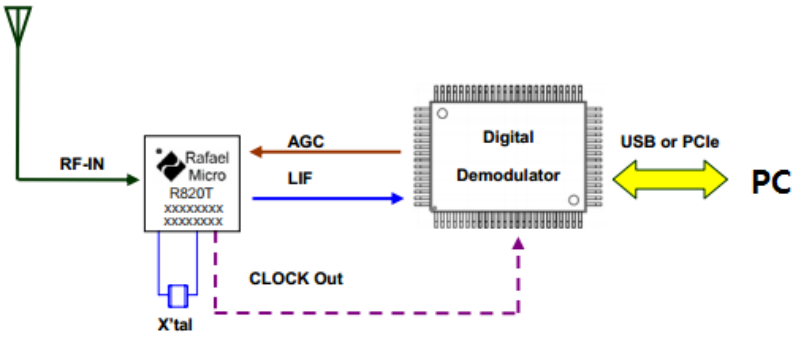
3 DVB-T TUNER S DAB+, FM A SDR S TUNEREM R820T

Tento TV tuner dokáže mimo klasického DVB-T zpracovat i FM a DAB/DAB+ signál. Tento tuner lze také využít k postavení svého vlastního softwarově definovaného rádia. Na obrázku 10 je ukázán použitý tuner a na obrázku 11 je zobrazeno blokové schéma přijímače [9].

Parametry a funkce tuneru:

- DVB-T tuner do USB,
- čip RTL2832U a R820T,
- DVB-T, DAB, FM,
- anténní konektor IEC 75 Ω,
- připojení do USB – 5 V, 500 mA,
- DVB-T 6/7/8 MHz od 50 MHz do 860 MHz,
- rozměry cca 90 x 30 x 17 mm [9].

Obrázek 10: Ukázka použitého tuneru [9].



Obrázek 11: Blokové schéma přijímače [10].

3.1 RTL2832U

DVB-T tunery založené na obvodu RTL2832U mohou být použity jako levné SDR, protože umožňují surový přenos I/Q vzorků. Obvod RTL2832U je vysoce výkonný DVB-T COFDM demodulátor. Podporuje 2K nebo 8K mód s šířkou pásma 6, 7 nebo 8 MHz. Modulační parametry, jako jsou například kódový poměr a ochranný interval, se detekují automaticky. RTL2832U podporuje tunery na IF (Intermediate Frequency – 36,125 MHz), low-IF (4,57 MHz) nebo Zero-IF používající 28,8 MHz krystal. Obvod podporuje FM/DAB/DAB + Radio. Obsahuje vysoce stabilní A/D převodník. Převodník má rozlišení 8 bitů a maximální rychlost vzorkování je 3,2 MS/s [11] [12].

3.1.1 Software pro RTL2832U

Pro obvod RTL2832 jsou dostupné knihovny a několik nástrojů příkazového řádku (rtl_fm, rtl_sdr, rtl_test a rtl_tcp). Každý nástroj má své vlastní vstupní parametry, které lze měnit. Pomocí parametrů se ladí frekvence, nastavuje rychlost vzorkování, zesílení atd. Tyto nástroje příkazového řádku používají knihovny k testování RTL2832 a k vykonávání základních funkcí pro přenos dat z a do zařízení.

V diplomové práci byl využit nástroj rtl_sdr, pomocí kterého bylo zařízení otestováno, zda přijímá správně IQ vzorky. Rtl_sdr má vstupní parametry:

- f: požadovaná frekvence [Hz],
- s: vzorkovací frekvence (defaultně: 204800 Hz),
- d: index zařízení (defaultně: 0),
- g: zesílení (defaultně: 0 pro automatické),
- p: pmm_error (defaultně: 0),
- b: velikost výstupního bloku (defaultně 16*16384),
- n: počet čtených vzorků (defaultně: 0 - nekonečno),
- S: synchronní/asynchronní výstup (defaultně: asynchronní),

název výstupního souboru [11].

Příkaz pro získání surových dat má následující podobu:

rtl_sdr.exe –f 868.830e6 -d 0 -n 0 surova_I_Q_data.txt .

3.2 R820T

R820T představuje vysoce výkonný TV tuner s nízkou spotřebou elektrické energie. V praxi se využívá například v Set top boxech, televizních PC kartách nebo TV přijímačích do osobních počítačů připojitelných přes USB port. Základní parametry a vlastnosti tuneru:

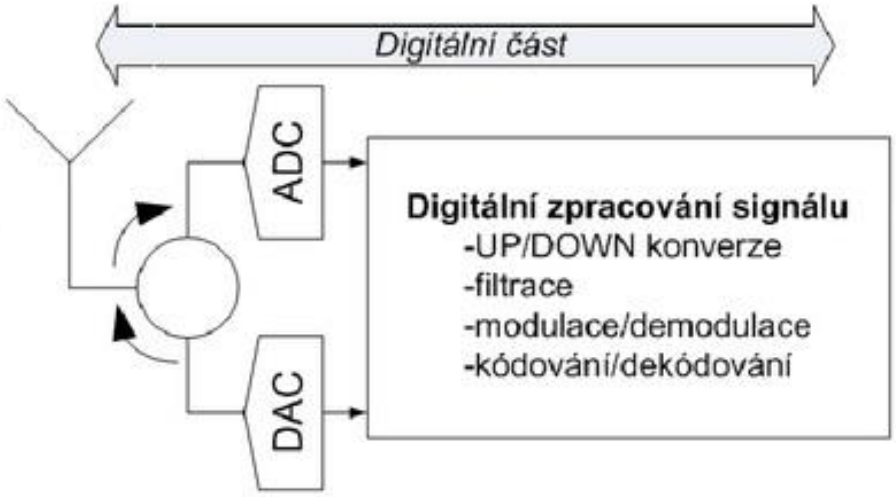
- frekvenční rozsah: 42 – 1002 MHz,
- šumové číslo: 3,5 dB,

- napájecí napětí: 3,3 V,
- spotřeba: <178 mA,
- maximální vstupní výkon: +10dBm,
- podporuje tyto TV digitální standardy: DVB-T, ATSC, DTMB a ISDB-T,
- dvouvodičové I2C rozhraní [10].

3.3 Softwarově definované rádio

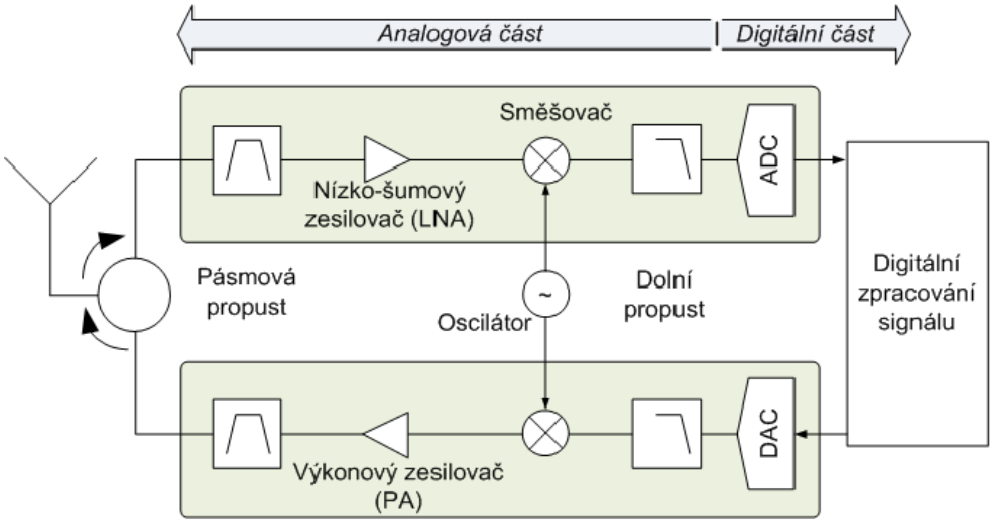
Softwarově definované rádio představuje rádiový komunikační systém, ve kterém jsou hardwarové prvky (například filtry, zesilovače, směšovače, modulátory/demodulátory atd.) vytvořené softwarem na počítači nebo softwarem v embedded systému. Tato skutečnost je velkou výhodou, protože umožňuje snadné přeladění přes velký rozsah kmitočtů na základě požadavků daného komunikačního kanálu (typ modulace, šířka pásma atd.), a to pouze změnou softwaru. Není tedy nutné měnit hardwarovou konfiguraci [13].

Obecné schéma ideálního SR (Softwarového rádia) lze vidět na obrázku 12. V přijímači je rádiový signál nejprve převeden do digitální podoby pomocí AD převodníku a poté je číslicově zpracován. Ve vysílači se nejprve digitální signál převede pomocí DA převodníku na analogový a následně se přivede na anténu [13].



Obrázek 12: Obecné schéma ideálního SR [13].

V současnosti je digitalizace RF signálu omezena technologickými limity AD převodníků, které jsou realizovatelné pro kmitočtový rozsah do několika stovek megahertzů. V současné době je proto zatím architektura ideálního SR nereálná a v praxi mluvíme o zjednodušené architektuře, kterou je SDR (softwarově definované rádio). Rozdíl oproti ideálnímu SR je v posunutí digitálního zpracování signálu do základního pásma nebo na nízkou mezifrekvenci, čímž se zjednoduší požadavky na AD převodník. Na obrázku 13 je zobrazeno schéma SDR [13].

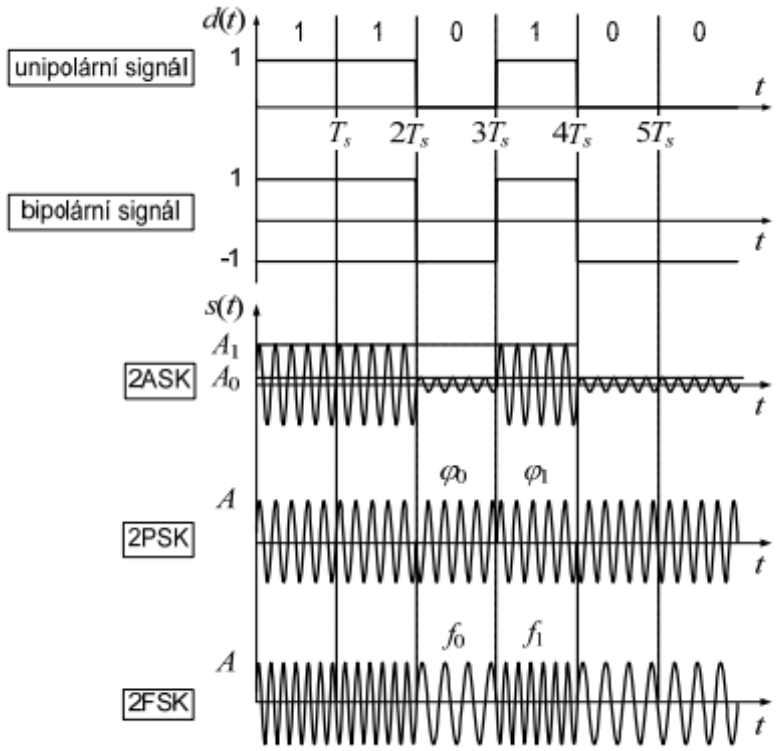


Obrázek 13: Schéma SDR [13].

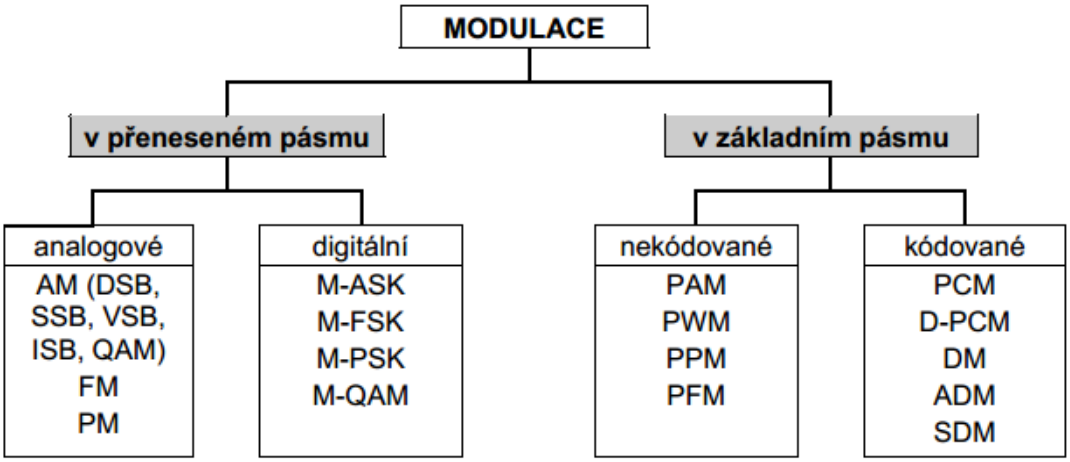
4 MODULACE

Modulace představuje proces, ve kterém je ovlivňována nějaká vlastnost nosné signálem modulačním. Jestliže je modulačním signálem signál spojitý (analogový), pak se jedná o analogovou modulaci, nebo může být číslicový (digitální), potom mluvíme o digitální modulaci. V případě digitálních modulací nosná vlna přechází mezi několika diskrétními stavy ovlivňovaného parametru, proto se tyto modulace označují klíčování. Klíčování může být obecně M-stavové, kde $M = 2^m$ různých signálů vyjadřuje m-tice bitů. Základní typy digitálních modulací jsou na obrázku 14 [14].

Modulační signál ovlivňuje amplitudu, kmitočet nebo fázi. Potom mluvíme o modulaci amplitudové (AM), kmitočtové (FM) nebo fázové (PM). U číslicových modulací rozeznáváme také tři základní druhy klíčování (viz Obrázek 14) – amplitudová (ASK), kmitočtová (FSK) a fázová (PSK). Existují i modulace v základním pásmu, které nevyužívají nosnou vysokofrekvenční vlnu. Základní rozdělení modulací je na obrázku 15 [14].



Obrázek 14: Průběhy základních typů digitálních modulací [14].



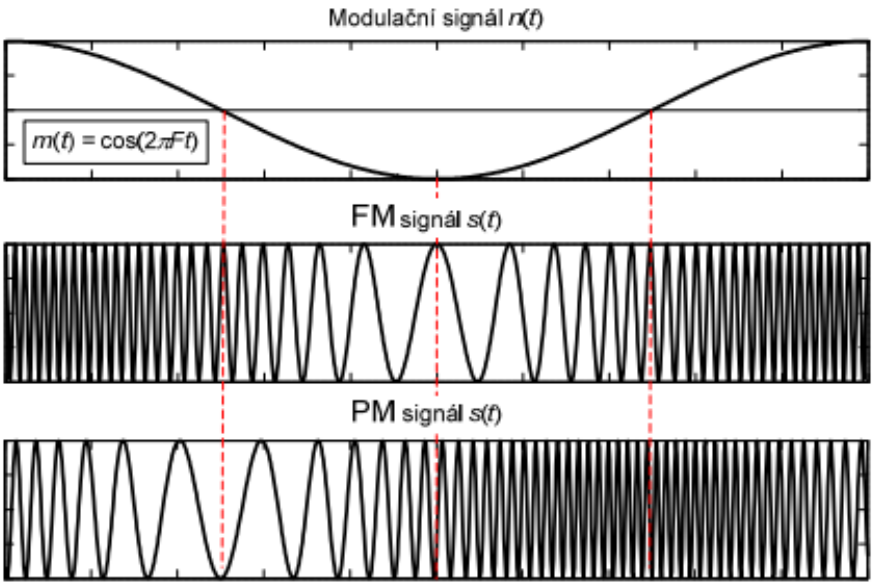
Obrázek 15: Základní typy modulací [14].

Teoretické požadavky na ideální modulační schéma:

- malá pravděpodobnost chyby,
- velká datová rychlost,
- malý vysílaný výkon,
- velká odolnost vůči interferencím,
- malá šířka pásma,
- malá složitost/výpočetní náročnost [14].

4.1 Úhlové modulace

Informace o modulačním signálu je přenášena ve změně fáze resp. kmitočtu nosné. Modulace se označují pojmem úhlové modulace, protože fáze Θ a úhlový kmitočet ω jsou vázány vztahem $\omega = d\Theta / dt$, a v obou případech dochází ke změně fáze (fázového úhlu) nosné vlny. U těchto modulací nedochází ke změnám amplitudy. Vztah modulací FM a PM při modulaci harmonickým signálem je vidět na obrázku 16 [14].



Obrázek 16: Vztah modulací FM a PM při modulaci harmonickým signálem [14].

4.1.1 Kmitočtová modulace – FM

U kmitočtové modulace je přímo úměrná změna frekvence nosné vlny vzhledem k modulačnímu signálu. Okamžitý kmitočet modulovaného signálu můžeme stanovit z rovnice [14]:

$$f_i(t) = f_c + \frac{1}{2\pi}\left[\frac{d\theta(t)}{dt}\right] = f_c + \frac{1}{2\pi} a_F m(t), \tag{1}$$

kde a_F je kmitočtová citlivost modulátoru, f_c je kmitočet nosné, $\theta(t)$ je fázový úhel a m(t) je modulační signál. Fázový úhel se vypočítá ze vztahu [14]:

$$\theta(t) = 2\pi \int_{-\infty}^{t} f(\alpha) d\alpha = a_F \int_{-\infty}^{t} m(\alpha) d\alpha . \tag{2}$$

Maximum odchylky kmitočtu modulovaného signálu od kmitočtu nosné se nazývá kmitočtový zdvih, pro který platí vztah [14]:

$$\Delta f = \max\left[f_i(t) - f_c\right] = \max\left\{\frac{1}{2\pi}\left[\frac{d\theta(t)}{dt}\right]\right\}. \tag{3}$$

Pro určení charakteru modulace se užívá index modulace, který se určí podle vztahu [14]:

$$\beta_F = \frac{\Delta f}{F_{\max}}, \tag{4}$$

kde F_{max} představuje maximální frekvenci ve spektru modulačního signálu. Pokud je $\beta_F << 1$, jedná se o úzkopásmovou FM – NBFM a pro širokopásmovou FM – WBFM platí $\beta_F >> 1$ [14].

4.2 Binární FSK – 2FSK (BFSK)

Modulace 2FSK mění kmitočet nosné vlny skokově mezi dvěma hodnotami, které se nazývají signalizační kmitočty. Modulaci lze vyjádřit vztahem [14]:

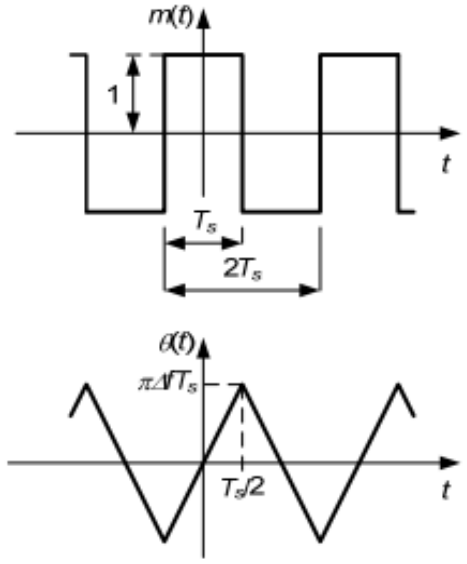
$$s_{FSK}(t) = \sum_{n=-\infty}^{\infty} A_c \cos\left[\omega_c t + \Delta\omega_n t\right] p(t - nT_s), \tag{5}$$

kde A_c je amplituda nosné, $\Delta\omega_n$ = {$2\pi\Delta$f;-$2\pi\Delta$f }, ω_c je úhlový kmitočet nosné a Δf je kmitočtový zdvih. Signalizační kmitočty jsou vyjádřeny $\omega_0 = \omega_c - 2\pi\Delta f$ a $\omega_1 = \omega_c + 2\pi\Delta f$. V intervalu $0 \geq t \geq T_s$ lze modulaci 2FSK popsat zjednodušeně [14]:

$$s_{FSK}(t) = A_c \cos(\omega_0 t) \text{ pro prvek 0,} \tag{6}$$

$$s_{FSK}(t) = A_c \cos(\omega_1 t) \text{ pro prvek 1.} \tag{7}$$

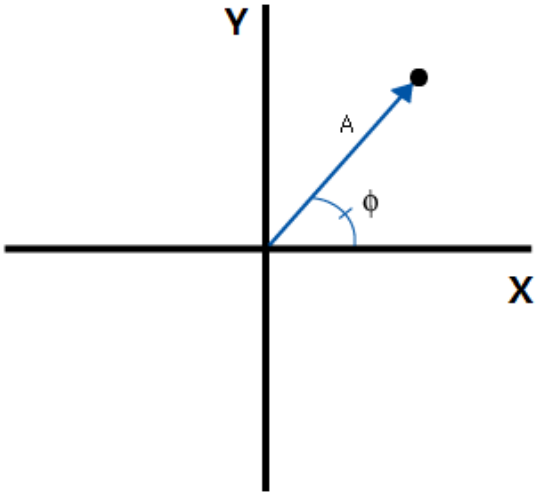
Na obrázku 17 je ukázán modulační signál a odpovídající hodnoty fáze. Jestliže bude modulační funkce m(t) dosahovat hodnot ±1, bude $a_F = 2\pi\Delta f$ a fáze se bude lineárně pohybovat mezi hodnotami $\pm 2\pi\Delta f T_S/2 = \pm\pi\Delta f T_S$ [14].



Obrázek 17: Modulační signál FSK a průběh hodnoty fáze [14].

4.3 I/Q data

I/Q data v podstatě znázorňují změny v amplitudě a fázi sinusoidy, kde I vyjadřuje soufázovou složku a Q představuje kvadraturní složku. Okamžité stavy sinusové vlny lze reprezentovat vektorem v komplexní rovině pomocí amplitudy a fáze v polárním souřadnicovém systému, viz obrázek 18. Vzdálenost od počátku až po černý bod představuje amplitudu sinusové vlny. Úhel od vodorovné osy znázorňuje fázi [15].



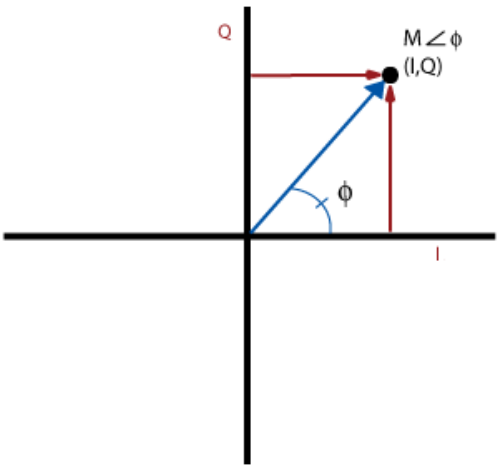
Obrázek 18: Polární reprezentace sinusové vlny, kde A značí amplitudu a ϕ fázi [15].

Pokud se amplituda sinusové vlny nemění, vzdálenost od počátku až do černého bodu se nemění. Fáze bodu se mění podle aktuálního stavu sinusové vlny. Ve skutečnosti se I/Q data zobrazují pomocí kartézského souřadného systému. Z polárního souřadného systému se provádí převod do kartézského souřadného systému pomocí trigonometrie. Tyto dvě reprezentace jsou ekvivalentní a obsahují stejné informace, jen v odlišných formách, jak je vidět na obrázku 19. Kartézský souřadný systém je zvolen kvůli jednodušší hardwardové implementaci. K obrázku 19 se vztahují následující dva vztahy:

$$I(t) = M(t)\cos(\phi(t)), \tag{8}$$

$$Q(t) = M(t)\sin(\phi(t)), \tag{9}$$

kde I(t) je vzdálenost I od počátku, M(t) je amplituda (vzdálenost od počátku do bodu M), Q(t) je vzdálenost Q od počátku a ϕ(t) je fáze [15].



Obrázek 19: I/Q zobrazení v polární formě, kde A značí amplitudu a ϕ fázi [15].

5 PŘIJÍMANÝ SIGNÁL

Protokol přijímaného signálu je Wireless M-Bus mód T1. Přenos v tomto módu probíhá pouze směrem od vysílače k přijímači. Přijímač neposílá žádnou odpověď. V tabulce 2 jsou uvedeny základní parametry signálu převzaté z [16].

Tabulka 2: Základní parametry přijímaného signálu.

Parametr	**Min**	**Typ**	**Max**	**Jednotky**
Střední frekvence	868,90	868,95	869	MHz
FSK odchylka	+/- 40	+/- 50	+/- 80	kHz
Bitová rychlost	90	100	110	Kb/s
Délka preambule (včetně synchronizačních bitů)	48			bitů
Délka postambule	2		8	bitů

Celý datový paket je kódován pomocí kódování 3 z 6. Jedná se o kódování s konstantní vahou, při kterém jsou každé 4 bity vyslaných dat kódovány na šesti bitová slova. Nejdříve se přenáší horní bajt, pouze u polí výrobce, v mém případě spotřeba, se přenáší první spodní bajt a poté horní. Paket obsahuje preambuli se synchronizačním bajtem a postambuli. Preambule se synchronizačním bajtem a postambula nejsou kódovány. Preambule (hlavička + synchronizační bajt) má podobu: n*(01) 0000111101, kde n > 19. Bitová sekvence 01010101 a synchronizace 00001111 nemůže v přenosu nikdy nastat a je použita k identifikaci začátku paketu. Tvar postambule záleží na úplně posledním bitu rámce v poli CRC. Pokud je poslední bit v CRC roven 0, minimální postambule má tvar 01, v ostatních případech je ve formátu 10. Posledních 32 bajtů paketu (bez CRC bajtů) je šifrovaných pomocí šifry AES módu CBC [16].

5.1 Formát paketu

Přenášený paket se skládá ze tří bloků. První blok tvoří linkovou vrstvu (tabulka 3) a ostatní bloky tvoří aplikační vrstvu (tabulka 4 a tabulka 5). Údaje v tabulkách jsou převzaty z literatury [16]. Aplikační vrstva je šifrovaná.

Tabulka 3: První blok vysílaného paketu.

Název pole	L	C	M	A	CRC
Počet bajtů	1	1	2	6	2

Tabulka 4: Druhý blok vysílaného paketu.

Název pole	CI	Data	CRC
Počet bajtů	1	15 (pokud je to poslední blok, tak je délka ((L-9 modulo 16)	2

Tabulka 5: Třetí blok vysílaného paketu.

Název pole	Data	CRC
Počet bajtů	16 (pokud je to poslední blok, tak je délka ((L-9 modulo 16)	2

5.1.1 Linková vrstva

Linková vrstva obsahuje následující pole:

- **Pole – L (Length Field):**

  První pole udává počet bajtů celého paketu. Do délky se nepočítá samotné pole L a CRC pole. Pokud ((L-9) modulo 16) není rovno 0, tak poslední blok obsahuje ((L-9) modulo 16) bajtů + 2 bajty CRC, v ostatních případech následují 2 CRC bajty po každých 16 bajtech [16].

- **Pole – C (Control Field):**

  Druhý bajt udává typ použité zprávy. Podle normy IEC 60870-5-2 se používají hodnoty:

  - 44_h: používaná v režimu T1, informuje o tom, že se neodesílá odpověď vysílači,
  - 46_h: tato hodnota udává, že je měřič v instalačním režimu,
  - 48_h: používá se pouze v režimu T2 [16].

- **Pole – M (Manufacturer ID):**

  Pole obsahuje 2 bajty, které nám udávají identifikační číslo výrobce [16].

- **Pole – A (Address):**

  Adresa je dána 6 bajty a musí být vždy unikátní. Obsahuje jeden bajt s informací o typu měřidla, mezi nejběžnější patří:

  - 01_h: olej,
  - 02_h: elektřina,
  - 03_h: plyn,
  - 04_h: teplo,
  - 06_h: teplá voda,
  - 07_h: studená voda [17].

- **Pole – CRCx (Cyclic Redundancy Check):**

  CRC je vypočteno z informací předchozího bloku podle následujícího polynomu: $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$ [16].

5.1.2 Aplikační vrstva

Aplikační vrstva vychází z normy EN1434-3. Vrstva obsahuje pole CI (Control Information) a rámec s proměnou strukturou dat, ve kterém se může přenášet libovolný počet hodnot, omezený velikostí datového pole. Tabulka 6 obsahuje formát aplikační vrstvu signálu v této diplomové práci. Veškerá data za datovou hlavičkou (Data header) jsou šifrovaná [16].

Tabulka 6: Aplikační vrstva.

Název pole	CI	Data header	AES verify	Dif	Vif	Spotřeba	Dif	Vif	Čas, datum	AES verify
Počet bajtů	1	4	2	1	1	4	1	1	4	1

- **Pole – CI (Control Information):**

  Je umístěno na začátku druhého bloku a informuje o typu protokolu a zprávě, která bude následovat:

  - 51_h: poslaná data z přijímače k vysílači jsou bez pevné hlavičky,
  - 72_h: aplikační vrstva s úplnou hlavičkou,
  - 78_h: aplikační vrstva bez hlavičky,
  - $7A_h$: aplikační vrstva s krátkou hlavičkou,
  - 82_h: pro použití s technologií CENELEC TC 205,
  - $A0_h$ – $B7_h$: aplikační vrstva specifikovaná výrobcem [16].

- **Pole – Data Header:**

  Hlavička dat má podobu podle hodnoty v CI. V tabulce 7 je uvedena hlavička pro CI = $7A_h$ [17].

Tabulka 7: Data header pro CI = $7A_h$ [17].

Název pole	Access number	Status	Signature
Počet bajtů	1	1	2

- **Pole – AES verify:**

  Slouží k ověření dešifrování. Při správném dešifrování dat je hodnota obou bajtů $2F_h$.

- **Pole – DIF (Data Information Field)**

  Informace, které obsahuje pole DIF, jsou vypsány v tabulce 8 [17].

Tabulka 8: Obsah pole DIF [17].

Číslo bitu	7	6	5,4	3,2,1,0
Název informace	Extension bit	LSB of storage number	Function field	Data field: Length and coding of data

Data field: udává, jak jsou data z vysílače interpretována. Pro typ integer vypadá datové pole následovně:

- 0000_b – žádná data,
- 0001_b – 8 bitový integer,
- 0010_b – 16 bitový integer,
- 0011_b – 24 bitový integer,
- 0100_b – 32 bitový integer,
- 0110_b – 48 bitový integer,
- 0111_b – 64 bitový integer [17].

Function field: přenáší informaci o tom, jaká data jsou přijímána:

- 00_b – okamžitá hodnota,
- 01_b – maximální hodnota,
- 10_b – minimální hodnota,
- 11_b – hodnota během chybového stavu [17].

LSB of storage number: nepoužívá se v komunikaci Wireless M-Bus.

Extension bit: signalizuje, zda bude následovat další DIF bajt:

- 0_b – nebude následovat žádný DIF bajt,
- 1_b – bude následovat DIF bajt [17].

- **Pole – VIF (Value Information Bit):**

  VIF udává informaci o přenášené hodnotě. Tabulka 9 obsahuje strukturu bajtu VIF [17].

Tabulka 9: Obsah pole VIF [17].

Číslo bitu	7	6,5,4,3,2,1,0
Název informace	Extension bit	Unit and multiplier (value)

Unit and multiplier (value): předává jednotku a násobitel přenášené hodnoty.

V tabulce 10 jsou vybrané základní jednotky a násobitele [17].

Tabulka 10: Tabulka vybraných jednotek a násobitelů přenášených hodnot [17].

Kód	**Popis**	**Násobitel a jednotky**
E000 0nnn	Energie	$10^{(nnn-3)}$ Wh
E000 1nnn	Energie	$10^{(nnn)}$ J
E001 0nnn	Objem	$10^{(nnn-6)}$ m^3
E001 1nnn	Hmotnost	$10^{(nnn-3)}$ kg
E010 00nn	Čas	nn = 00 sekundy nn = 01 minuty nn = 10 hodiny nn = 11 dny
E010 1nnn	Energie	$10^{(nnn-3)}$ W
E101 10nn	Teplota	$10^{(nn-3)}$ °C

6 ŠIFROVÁNÍ AES

Počátky šifrování AES se vztahují do roku 1997, kdy se hledal nástupce šifrování DES, který se stával náchylnější na útoky k prolomení. V roce 2002 byl schválen šifrovací standard AES (Advanced Encryption Standard). Jedná se o symetrický kryptosystém, který používá jeden klíč k zašifrování i dešifrování dat. Délky šifrovacích klíčů mohou být 128, 192 nebo 256 bitů. Mezi hlavní nevýhodu symetrického šifrování patří používání sdíleného klíče a mezi hlavní výhodu rychlost [18] [19].

AES je blokový šifrovací algoritmus používán na data s pevně danou délkou – 128 bitů. Jestliže jsou zpracovávaná data delší než 128 bitů, musí se rozdělit do více bloků. Naopak kratší datový blok se musí doplnit na odpovídající délku. Jednotlivé bloky se zapisují jako dvojice hexadecimálních symbolů uspořádaných do matice o velikosti 4x4. Šifrování AES využívá čtyři základní operace, které jsou využity v algoritmu:

- SubBytes,
- ShiftRows,
- MixColumns,
- AddRoundKey.

U blokových šifrovacích algoritmů se uvádí režim provozu, který určuje jak je aplikována šifra na jednotlivá data při velikosti dat větších než jeden blok [18].

Nejjednodušší mód je ECB (Electronic Codebook). Šifra se aplikuje vždy na jednotlivé bloky tak, jak jdou po sobě. Použití ECB se nedoporučuje, protože stejné bloky nezašifrovaného textu jsou zašifrovány vždy stejně [18].

Nedostatek ECB odstraňuje mód CBC (Cipher Block Chaining). Funkce CBC spočívá v tom, že se před zašifrováním odpovídající blok dat sečte pomocí funkce XOR s předcházejícím zašifrovaným blokem. Jednotlivé bloky jsou na sobě závislé a ke správnému dešifrování určitého bloku se musí dešifrovat i všechny předchozí. K dešifrování prvního bloku se používá inicializační vektor (IV) [18].

Další režim se nazývá CFB (Cipher FeedBack), který se podobá CBC. CFB nepracuje s celými bloky dat najednou, ale funguje jako proudová šifra. Dosahuje rychlejšího šifrování než výše zmíněné módy [20].

Mód OFB (Output Feedback) je režim podobný jako CFB [20].

CTR (Counter) režim nepotřebuje pevnou šířku bloků 128 bitů [20].

6.1 Šifrování

Data připravená k šifrování jsou uspořádaná do matice.

SubBytes

Všechny bajty jsou substituovány podle substituční tabulky pro přímou substituci. Každý bajt s indexem XY v hexadecimálním zápise je nahrazen bajtem v řádku X a sloupci Y [21].

ShiftRows

Pole v této matici jsou přeházena tak, že v řádku 1 se posunou všechny bajty o 1 pozici vlevo, v řádku 2 o 2 pozice vlevo a v řádku 3 o 3 pozice vlevo. V řádku 0 posun neprobíhá a zůstane stejný. Výsledkem je pak matice [21]:

$$\begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix} \gg \begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{11} & S_{12} & S_{13} & S_{10} \\ S_{22} & S_{23} & S_{20} & S_{21} \\ S_{33} & S_{30} & S_{31} & S_{32} \end{bmatrix}. \tag{10}$$

MixColumns

Operace MixColumns je použita na každý sloupec. Všechny sloupce jsou chápány jako polynomy třetího stupně v tělese GF(2^8). K získání výsledné matice se použije násobení sloupce polynomu s polynomem $a(x) = 03x^3 + 01x^2 + 01x + 02$ modulo $m(x) = x^4 + 1$ [21].

$$\begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix} \bullet \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} S_{00}' & S_{01}' & S_{02}' & S_{03}' \\ S_{10}' & S_{11}' & S_{12}' & S_{13}' \\ S_{20}' & S_{21}' & S_{22}' & S_{23}' \\ S_{30}' & S_{31}' & S_{32}' & S_{33}' \end{bmatrix}. \tag{11}$$

AddRoundKey

Jednotlivé bajty matice se pomocí XOR sečtou s jednotlivými bajty rundovního klíče [21]:

$$\begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix} \oplus \begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} S_{00}' & S_{01}' & S_{02}' & S_{03}' \\ S_{10}' & S_{11}' & S_{12}' & S_{13}' \\ S_{20}' & S_{21}' & S_{22}' & S_{23}' \\ S_{30}' & S_{31}' & S_{32}' & S_{33}' \end{bmatrix}. \tag{12}$$

6.2 Dešifrování

Stejně jako u šifrování jsou i při dešifrování data uspořádaná do matice.

InvShiftRows

V jednotlivých řádcích matice se provede přehození jednotlivých bajtů tak, že se v prvním řádku posunou bajty o jednu pozici vpravo, ve druhém řádku o 2 pozice

vpravo a ve třetím řádku o tři pozice vpravo [21]:

$$\begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix} \gg \begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{13} & S_{10} & S_{11} & S_{12} \\ S_{22} & S_{23} & S_{20} & S_{21} \\ S_{31} & S_{32} & S_{33} & S_{30} \end{bmatrix}. \quad (13)$$

InvSubBytes

Všechny bajty jsou substituovány podle substituční tabulky pro inverzní substituci. Každý bajt s indexem XY v hexadecimálním zápise je nahrazen bajtem v řádku X a sloupci Y [21].

AddRoundKey

Jednotlivé bajty matice se pomocí XOR sečtou s jednotlivými bajty rundovního klíče (viz AddRoundKey u šifrování na straně 25) [21].

InvMixColumns

Tato operace se provádí stejným postupem jako MixColumns, ale polynom a(x) se nahradí inverzním polynomem $a^{-1}(x) = 0Bx^3 + 0Dx^2 + 09x + 0E$ [21].

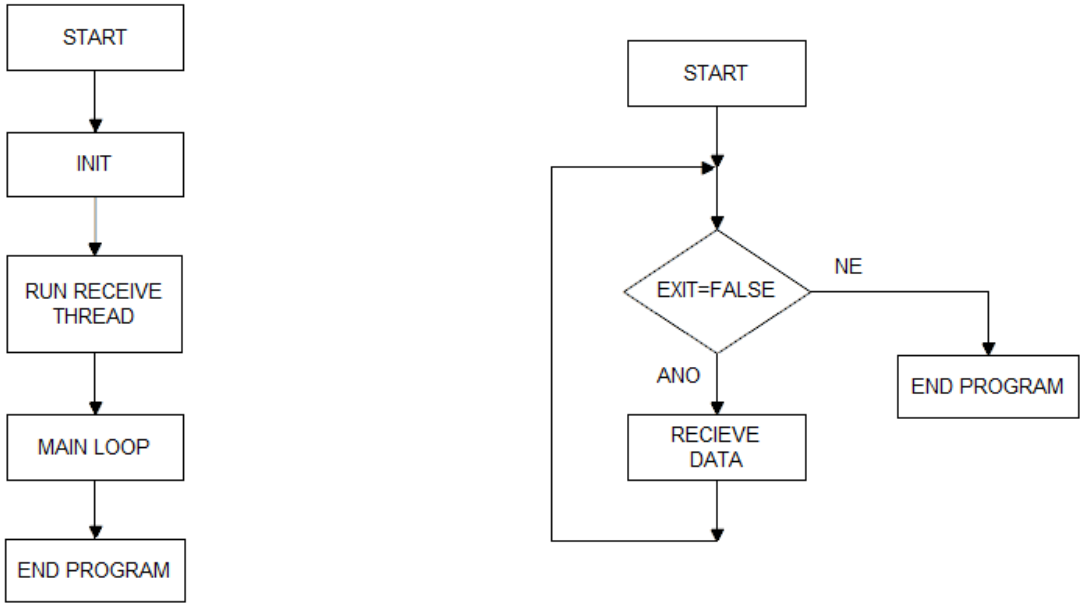
$$\begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix} \bullet \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} = \begin{bmatrix} S_{00}' & S_{01}' & S_{02}' & S_{03}' \\ S_{10}' & S_{11}' & S_{12}' & S_{13}' \\ S_{20}' & S_{21}' & S_{22}' & S_{23}' \\ S_{30}' & S_{31}' & S_{32}' & S_{33}' \end{bmatrix}. \quad (14)$$

7 SOFTWARE PRO PŘÍJEM A ZPRACOVÁNÍ DAT

Software pro příjem a zpracování dat byl napsán v programovacím jazyku C++ ve vývojovém prostředí Microsoft Visual Studio 2013.

7.1 Popis Softwaru

Celý program běží ve dvou vláknech. Jedno vlákno se stará o příjem dat a druhé o zpracování dat. Blokové schéma softwaru je znázorněno na obrázku 20 (vlevo – vlákno pro zpracování dat, vpravo – pro příjem dat). Ve vývojových diagramech jsou použity stejné názvy funkcí jako v samotném softwaru.



Obrázek 20: Blokové schéma softwaru.

7.1.1 Inicializace

Na začátku programu je provedena inicializace, která slouží k počátečnímu nastavení přijímače a k alokaci paměti pro datové buffery. Přijímač je nejdříve vyhledán a načten. Poté se provede nastavení zesílení přijímače, zapnutí vnitřní automatické kontroly zesílení, nastavení střední frekvence, nastavení vzorkovací rychlosti a vyprázdnění bufferu přijímače. Všechny příkazy pro ovládání tuneru byly získány z jeho knihoven. Nakonec inicializace je vytvořen mutex k druhému vláknu programu.

```
dataBuffer = (uint8_t *)malloc(DEFAULT_BUF_LENGTH * sizeof(uint8_t));
      //alokace paměti

dataBuffer2 = (uint8_t *)malloc(DEFAULT_BUF_LENGTH * sizeof(uint8_t));
      //alokace paměti

r = rtlsdr_open(&receiver, (uint32_t)dev_index);        //otevření
                                                           přijímače

rtlsdr_set_tuner_gain(receiver, 115);           //nastavení zesílení

r = rtlsdr_set_agc_mode(receiver, 1);           //zapnutí vnitřní
                                          automatické kontroly zesílení

rtlsdr_set_center_freq(receiver, ADSB_FREQ);    //nastavení střední
                                                           frekvence

rtlsdr_set_sample_rate(receiver, ADSB_RATE);    //nastavení vzorkovací
                                                           frekvence

rtlsdr_reset_buffer(receiver);                  //vyprázdnění bufferu

signalDataReady = CreateMutex(NULL, FALSE, NULL);   //vytvoření mutexu
```

7.1.2 Spuštění vlákna pro příjem dat

Funkce slouží k vytvoření nového vlákna pro neustálý příjem dat. K vláknu jsou potřebné další dvě funkce, které slouží k uzamknutí (lock) a uvolnění (unlock) mutexu. Mutex se musí uzamykat, aby nedocházelo k přepisování zpracovávaných dat nově přijatými.

Po vytvoření nového vlákna začne příjem surových dat a postupně se plní buffer. Po naplnění bufferu je volána funkce (rtlsdr_callback), ve které se nejdříve mutex uzamkne, následně dojde ke zkopírování přijatých dat a nakonec je mutex uvolněn. Poté začne znovu příjem nových dat a postup se opakuje.

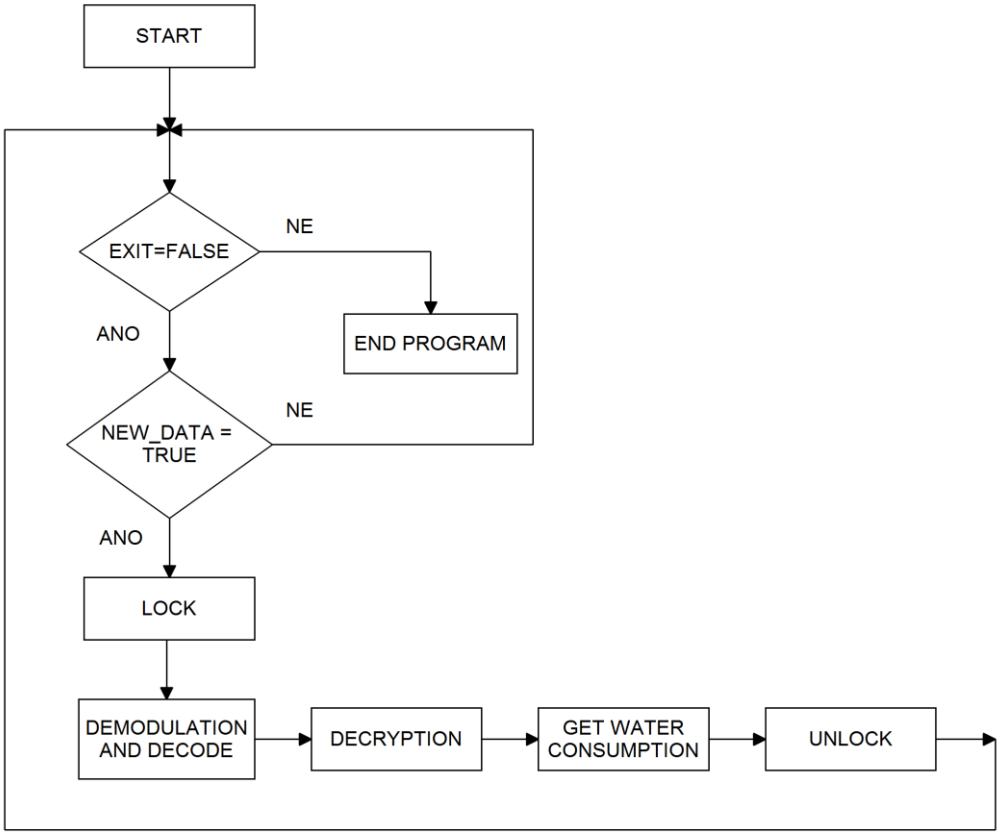
```
void lock()  //čekání na uvolněný mutex a následné uzamknutí
{
      WaitForSingleObject(signalDataReady, INFINITY);
}

void unlock()     //uvolnění mutexu
{
      ReleaseMutex(signalDataReady);
}

void rtlsdr_callback(unsigned char *buf, uint32_t len, void *ctx)
{
      lock();                          //uzamknutí mutexu
      dataLength = len;                //délka bufferu
      memcpy(dataBuffer, buf, len);    //zkopírování buffer
      canWriteNewData = TRUE;
      unlock();                        //uvolnění mutexu
}
```

Po uvolnění mutexu začne zpracování dat ve druhém vlákně.

7.1.3 Hlavní smyčka

V hlavní smyčce se zpracovávají data ve třech krocích. Nejdříve je provedena demodulace a dekódování dat. Následně se provede dešifrování zašifrované části zprávy a nakonec se získá spotřeba.

Při vstupu do hlavní smyčky se čeká na přijetí nových dat. Jakmile jsou nové vzorky přijaty, uzamkne se mutex a začnou se data zpracovávat. Po zpracování dat dojde znovu k uvolnění mutexu, aby mohlo dojít k přepsání již zpracovaných dat nově přijatými. Blokový diagram hlavní smyčky je zobrazen na obrázku 21.



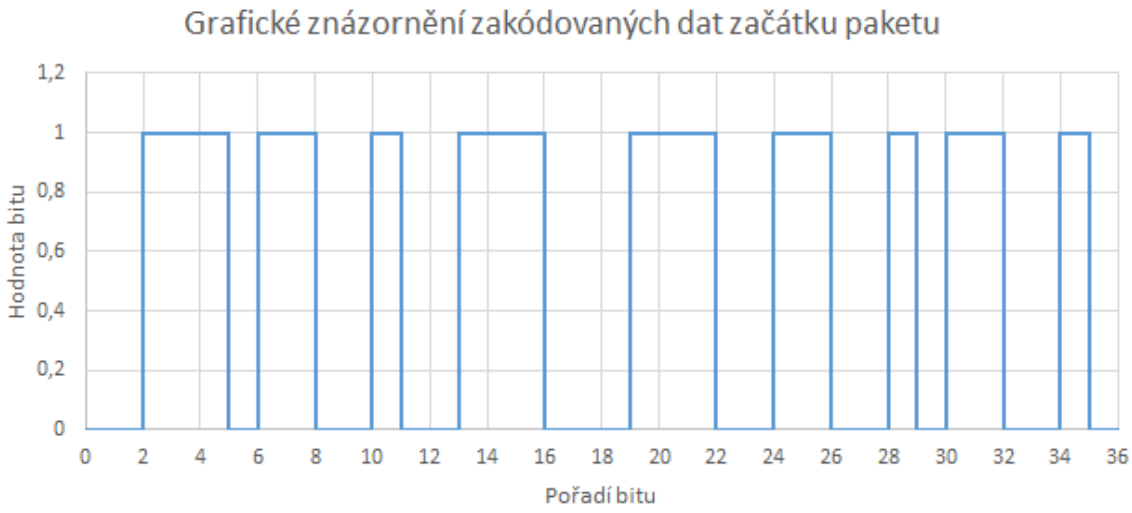
Obrázek 21: Blokový diagram hlavní smyčky.

DEMODULOVÁNÍ A DEKÓDOVÁNÍ DAT

Po přijetí nových dat dojde nejprve k jejich demodulování a dekódování. Prochází se celá délka přijatého datového bufferu a vybírají se z něho I a Q vzorky. Po každém získání vzorku I a Q se provede součet |I| + |Q|, který musí splňovat podmínku |I| + |Q| > 90. Touto podmínkou se odfiltrují vzorky, které nenesou žádnou informaci. Pokud je podmínka splněna, vypočítá se úhel mezi oběma složkami a ten se porovná s předchozím získaným úhlem. Porovnáním se určí směr pohybu po IQ diagramu a tím i hodnota navzorkovaného bitu. Při pohybu po směru hodinových ručiček je hodnota bitu rovna 1 a pro pohyb proti směru hodinových ručiček je naopak hodnota 0. Vzorkovací

frekvence (2 MHz) je několikanásobně vyšší než rychlost dat, proto se vyhodnotí po každé změně směru pohybu skutečný počet bitů, který je menší než získaný po vzorkování. Když se přijme preambule a synchronizace, začnou se jednotlivé bity ukládat do pole. Od tohoto okamžiku jsou všechna data zakódovaná a postupně se pro každých přijatých 6 bitů provádí dekódování. Dekódovaná data jsou ukládána a sestavuje se výsledný paket. Nejprve je uložena do pole část zprávy, která není zašifrovaná a poté do druhého pole zašifrovaná část.

Na obrázku 22 je znázorněn grafický průběh zakódovaných bitů začátku zprávy.



Obrázek 22: Grafický průběh prvních 36 zakódovaných bitů.

Dekódování se provádí vždy po 6 bitech, proto je potřeba tuto zprávu rozdělit. Výsledný tvar bude: 001110_b 110010_b 011100_b 011100_b 110010_b 110010_b. Upravená data se podle tabulky 11 dešifrují. Po dekódování bude tato vybraná část zprávy vypadat: $2E_h$ 44_h EE_h. Tímto způsobem se dekóduje celý paket.

Tabulka 11: Tabulka kódování 3 z 6 [16].

Binární tvar	Decimální tvar	Zakódovaný tvar
0000	0	010110
0001	1	001101
0010	2	001110
0011	3	001011
0100	4	011100
0101	5	011001
0110	6	011010
0111	7	010011
1000	8	101100
1001	9	100101
1010	10	100110
1011	11	100011
1100	12	110100
1101	13	110001
1110	14	110010
1111	15	101001

DEŠIFROVÁNÍ DAT

Po přijetí celého paketu se dešifrují šifrovaná data. K tomuto účelu je použita knihovna získaná z OpenSSL. Ve funkci pro dešifrování se nejprve vytvoří nové dešifrování. Poté se provede inicializace, ve které se určí dešifrovací klíč, inicializační vektor a jakým typem šifrování AES se budou data dešifrovat. Dešifrovací klíč i inicializační vektor jsou uloženy v poli. Po této inicializaci je zahájeno samotné dešifrování a získaná data jsou uložena do pole pro další zpracování. Nakonec se z výsledného pole otestují první dva bajty, zda se rovnají. Po správném dešifrování by se měly rovnat $2F_h$. Z dešifrovaných dat je poté potřeba získat údaj o spotřebě vody [22].

```
EVP_CIPHER_CTX *ctx;

ctx = EVP_CIPHER_CTX_new()              //start dešifrování

EVP_DecryptInit_ex(ctx,EVP_aes_128_cbc(),NULL,key,iv) //inicializace

EVP_DecryptUpdate(ctx, outbuf, &len, (const unsigned
char*)encryptedData, encryptedDataLength)    //dešifrování
```

```
if (outbuf[0] == outbuf[1])         //AES verify(2F,2F)
{
    waterConsumption(outbuf);        //funkce pro určení spotřeby vody
}

EVP_CIPHER_CTX_free(ctx);        //ukončení dešifrování
```

ZÍSKÁNÍ SPOTŘEBY VODY

Poslední fází zpracování dat je určení spotřeby vody. Hodnota spotřeby vody se nachází na pátém až osmém bajtu dešifrované zprávy. Nejdříve musí hodnota z paketu vyčíst. Přenáší se v opačném pořadí (tzn. první bajt je LSB), proto se musí jednotlivé bajty správně přeskládat. Následně se data převedou z hexadecimálního tvaru do decimálního. Dále se určí násobitel, kterým se tato spotřeba bude násobit k získání skutečné hodnoty. Násobitel se získá z pole VIF, které má hodnotu 13_h = 0001 0011_b. Podle tabulky 9 se určí násobitel: $10^{0011b-6} = 10^{-3}$.

7.2 Změna komunikačního protokolu

Pokud by nastala změna komunikačního protokolu, bylo by potřeba přistoupit k úpravám softwaru. Program pracuje s paketem rozděleným na dvě části. První část tvoří nezašifrovaná data a druhá část jsou data zašifrovaná. Obě části mají pevně danou délku. Nejdříve jsou ukládána do pole data nezašifrované části zprávy. Následně je do nového pole zapsán zbytek dat.

Pokud by došlo například ke změně délky protokolu, program by nefungoval správně. Pro správné fungování by bylo nutné přepsat nové délky do programu a zkontrolovat velikosti jednotlivých polí pro ukládání dat, zda budou dostatečně velká. Žádné další úpravy by software nevyžadoval a zmíněné úpravy by se dělaly ve funkci pro demodulování a dekódování dat.

Jestliže by došlo ke změně inicializačního vektoru nebo dešifrovacího klíče, stačilo by v programu upravit pole, ve kterém jsou tyto údaje uloženy. Jiný zásah do programu by nebyl nutný. Při změně typu šifrování by se v softwaru musely změnit příkazy pro dešifrování. Všechny tyto úpravy by se prováděly ve funkci pro dešifrování.

Další změnou v komunikačním protokolu, může být změna kódování. V současném softwaru se data zpracovávají po 6 bitech. Jestliže by se data po změně dekódování také dekódovaly po 6 bitech, stačilo by v programu pouze změnit dekódovací část, ve které se zakódovaná data porovnávají a podle porovnání se určí jejich skutečná dekódovaná hodnota. V případě, že by docházelo k dekódování po jiném počtu bitů než po šesti, software by si vyžadoval větší úpravy.

8 GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ (GUI)

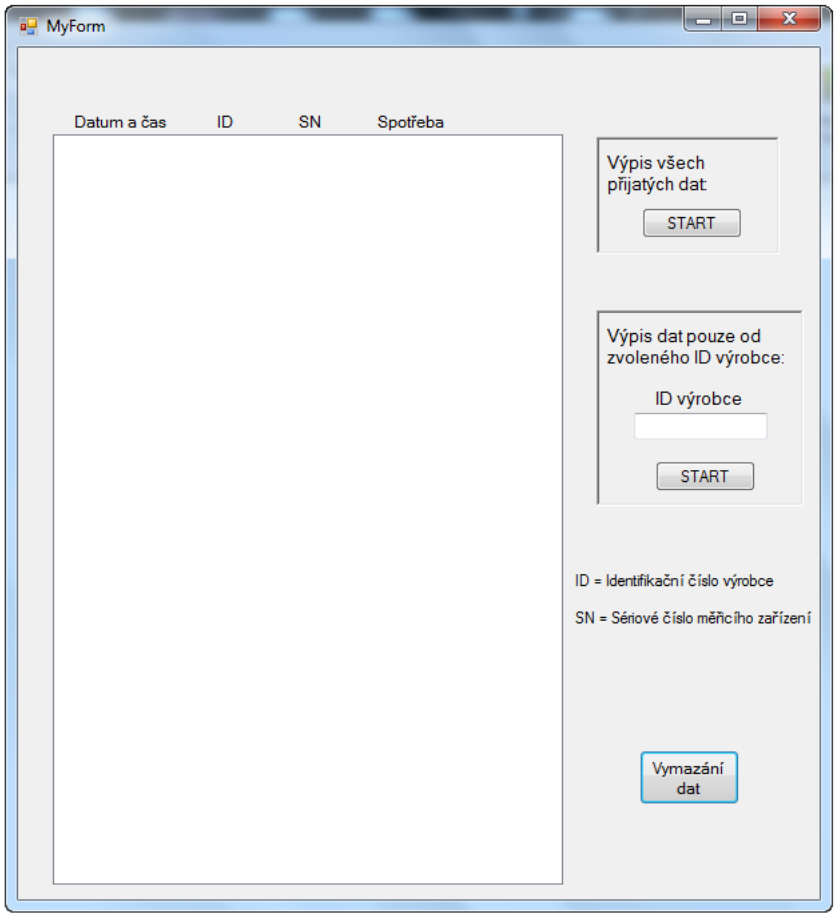
Tato kapitola obsahuje analýzu funkcí vytvořeného softwaru a popis vytvořeného grafického uživatelského rozhraní.

8.1 Analýza funkcí softwaru

Software bude vytvořen ve formě grafického uživatelského rozhraní pomocí vývojového prostředí Microsoft Visual Studio 2013 v jazyku C++. Zhotovený program by měl zobrazovat následující údaje: čas přijetí paketu, ID výrobce, sériové číslo měřicího zařízení a spotřebu. Dále by součástí softwaru měla být funkce, která uživateli umožní zvolit si příjem dat pouze od zvoleného vysílače podle ID výrobce. K tomuto účelu by mělo grafické rozhraní obsahovat textbox, do kterého uživatel zadá ID výrobce v hexadecimálním tvaru. Po potvrzení tohoto údaje tlačítkem by se měly zobrazovat pouze data ze zvoleného vysílače. Součástí aplikace by mělo být i tlačítko, které umožní smazat přijatá data.

8.2 Popis GUI

Grafické uživatelské rozhraní je vidět na obrázku 23. Přijatá data se vypisují do komponenty s název TextBox. Může probíhat výpis buď všech přijatých dat, nebo výpis dat pouze od zvoleného ID výrobce. K tomuto účelu jsou v aplikaci dvě tlačítka a jeden TextBox pro zadání ID výrobce. Pro výpis všech dat stačí pouze stisknout vrchní tlačítko. K vypisování pouze vybraných dat je nejdříve potřeba zadat ID výrobce a poté stisknout prostřední tlačítko. Zastavení zobrazování dat se provádí opětovným stiskem tlačítka, kterým bylo spuštěno. TextBox s přijatými daty lze vymazat pomocí spodního tlačítka.



Obrázek 23: Grafické uživatelské rozhraní.

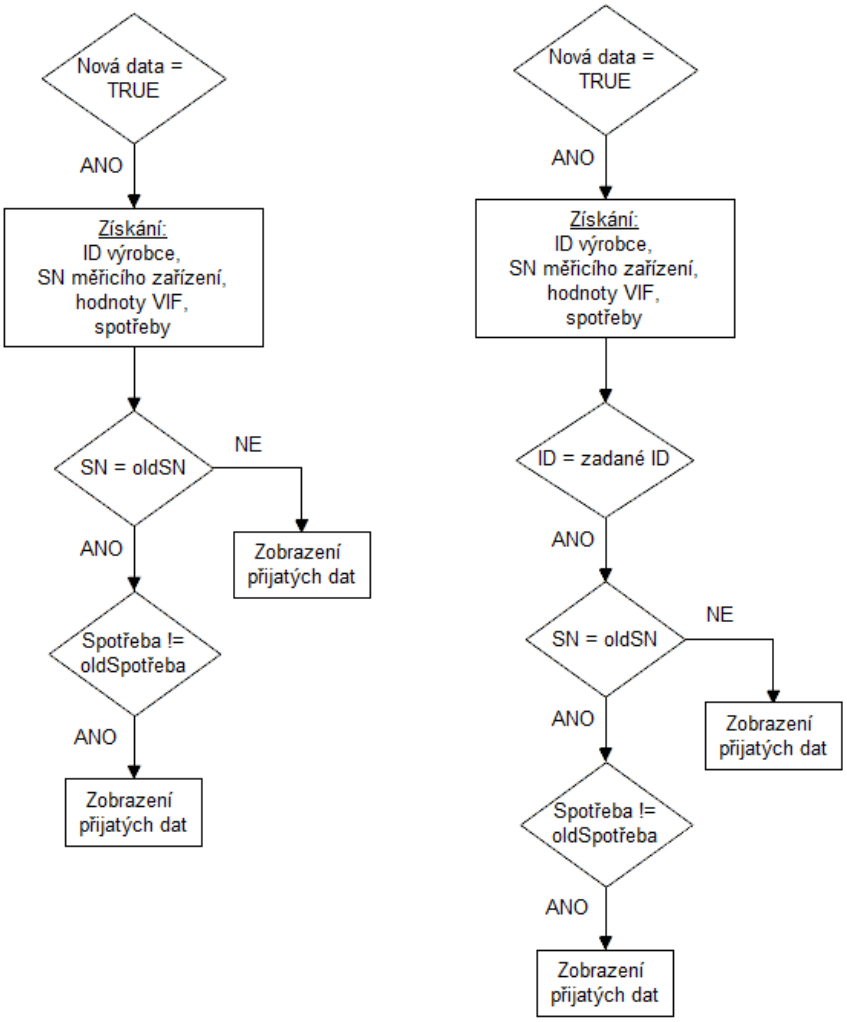
Po spuštění programu je vytvořeno nové vlákno, kde se spustí vytvořený software popsaný v kapitole 7. Základem aplikace je komponenta Timer, který je nastaven na 35 ms. Každých 35 ms je volána funkce, ve které se provádí zobrazování přijatých dat. Na začátku funkce se čeká na přijetí nových dat. Po přijetí dat probíhá jejich zpracování na základě toho, zda jsou vypisována všechna data, nebo pouze data od zvoleného ID výrobce. Na obrázku 24 jsou vidět blokové diagramy pro obě možnosti vypisování dat (vlevo – výpis všech dat, vpravo- výpis dat od zadaného ID výrobce).

- **Výpis všech dat**

  Nejdříve se získají tyto hodnoty: ID výrobce, sériové číslo měřicího zařízeni, hodnota pole VIF a spotřeba. Z hodnoty pole VIF je získán násobitel, kterým se násobí spotřeba. Poté se rozhoduje, zda je přijaté sériové číslo stejné jako poslední přijaté. Pokud není stejné jako poslední, dojde k zobrazení přijatých dat. Jestliže je stejné jako poslední přijaté, dojde k vypsání dat pouze v případě, že se změnila hodnota spotřeby od posledního přijatého paketu. Data jsou zobrazena se správnou jednotkou, podle hodnoty VIF.

- **Výpis všech dat**

  Stejně jako v předchozím případě nejprve dojde k získání těchto hodnot: ID výrobce, sériové číslo měřicího zařízení, hodnota pole VIF a spotřeba. Z hodnoty pole VIF je získána násobící konstanta. Následuje podmínka, ve které se porovnává přijaté ID výrobce se zadaným. Pokud se nerovnají, nedojde k výpisu přijatých dat. V případě, že jsou si ID výrobce rovna, program pokračuje stejným způsobem jako při zobrazování všech přijatých dat. Nejdříve je porovnáno přijaté sériové číslo s předchozím přijatým a poté je porovnána přijatá spotřeba s poslední přijatou.



Obrázek 24: Blokové diagramy pro výpis dat.

ZÁVĚR

V diplomové práci jsou popsány principy dálkového odečtu měřičů energií v domácnostech a jsou zde informace o jednotlivých komponentech inteligentních sítí pro odečet bytových měřidel. V práci je také zmínka o analogových a digitálních modulacích se zaměřením na frekvenční modulaci. Dále jsou zde informace o obvodech RTL2832U a R820T, které jsou součástí zvoleného přijímače. Součástí práce je detailní popis přijímaného signálu a šifrování AES, kterým je přijímaný signál šifrován. Závěrečná část práce je věnovaná popisu vytvořeného softwaru.

Pro praktickou část byl jako vysílač byl použit Double dongle RF M-bus & ZigBee od firmy ModemTec pracující na frekvenci 868 MHz. Každých 500 ms je vysílán paket, ve kterém se inkrementuje počítadlo spotřeby s každou vyslanou zprávou o 1. Tento paket je po příjmu DVB-T tunerem podroben demodulaci, dekódování a dešifrování. Software pro zmíněné zpracování signálu byl napsán v jazyku C++. Nejdříve byly ukládány přijaté surové vzorky do textového souboru pro zjištění, jakou velikost mají vzorky nesoucí informaci. Poté byla provedena demodulace a demodulovaná data byla ukládána do textového souboru pro jejich analýzu. Z těchto uložených dat bylo porovnáním se známým zadaným paketem zjištěno, kolik přijatých bitů odpovídá skutečnému počtu vyslaných bitů. Následně se tyto data dekódovala a zašifrovaná dešifrovala. K dešifrování dat byla použita knihovna z OpenSSL. Poté co byl software odladěn a přijatá data byla správně zpracována, vytvořilo se grafické uživatelské rozhraní, které obsahuje přehledný výpis přijatých dat. Aplikace umožňuje zobrazení všech přijatých dat nebo pouze dat od zvoleného výrobce. Pro každý přijatý paket se zobrazuje datum a čas přijetí, ID výrobce, sériové číslo měřicího zařízení a spotřeba. Vytvořená aplikace není přenositelná mezi systémem Windows a Linux, protože obsahuje příkazy použitelné pouze pro Windows.

Software je funkční a příjem dat z výše uvedeného vysílače probíhá bez větších problémů. Jediný zaznamenaný problém je občasná ztráta paketů, která se vyskytuje nepravidelně. Stává se například to, že se přijme a správně zpracuje šest po sobě jdoucích paketů, ale sedmý ne. Nebyla zjištěna přesná příčina této chyby, ale při příjmu dat může docházet k nepřesnostem v počtech přijatých bitů a z toho důvodu může vznikat rozdíl při přiřazení skutečného počtu bitů k těmto přijatým. Tato skutečnost znemožní správné dekódování dat.

LITERATURA

[1] PERGL, J. Nejnovější trendy v měření spotřeby vody a tepla. [online]. 31.10.2013 [cit. 2014-12-13]. Dostupné z: http://www.nazeleno.cz/bytove-domy/nejnovejsi-trendy-v-mereni-spotreby-vody-a-tepla.aspx.

[2] FRANEK, L. *Data koncentrátor pro chytré sítě*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřící techniky, 2012. 114 s. Diplomová práce. Vedoucí práce: Ing. Pavel Kučera, Ph.D.

[3] BLAŽEK, J. Měření vyrobené i spotřebované energie. [online]. 21.2.2014 [cit. 2014-12-13]. Dostupné z: http://automatizace.hw.cz/mereni-vyrobene-i-spotrebovane-energie .

[4] MODEMTEC, MT34A – ver. DK [online]. [cit. 2014-12-16]. Dostupné z: http://www.modemtec.cz/cz/produkty/datakoncentratory/mt34a-ver-dk#soubory-ke-stažení .

[5] Wireless M-Bus protocol software. [online]. [cit. 2014-12-13]. Dostupné z: http://www.ti.com/tool/wmbus#descriptionArea .

[6] VOJÁČEK, A. Sběrnice Wireless M-BUS – jde to i bezdrátově…. [online]. 13.2.2010 [cit. 2014-12-13]. Dostupné z: http://automatizace.hw.cz/sbernice-wireless-mbus-jde-i-bezdratove .

[7] ZigBee – novinka na poli bezdrátové komunikace [online]. 8.6.2005 [cit. 2014-12-13]. Dostupné z: http://www.hw.cz/navrh-obvodu/rozhrani/zigbee-novinka-na-poli-bezdratove-komunikace.html .

[8] KOTON, J, ČÍKA, P, KŘIVÁNEK, V. Standard nízkorychlostní bezdrátové komunikace ZigBee [online]. 18.4.2006 [cit. 2014-12-13]. Dostupné z: http://access.feld.cvut.cz/view.php?cisloclanku=2006032001 .

[9] DVB-T USB TUNER S DAB+ FM a SDR s tunerem R820T. [online]. [cit. 2014-12-13]. Dostupné z: http://harektrade.cz/eshop/dvb-t-pc-karty/70-dvb-t-usb-tuner-s-dab-fm-a-sdr-s-tunerem-r820t.html .

[10] RAFAEL MICROELECTRONICS, R820T High Performance Low Power Advanced Digital TW Silicon Tuner Datasheet [online]. 2011 [cit. 2014-12-16]. Dostupné z: http://superkuh.com/gnuradio/R820T_datasheet-Non_R-20111130_unlocked.pdf .

[11] Osmocom. OsmoSDR: RTL-SDR. [online]. 2014 [cit. 2014-12-13]. Dostupné z: http://sdr.osmocom.org/trac/wiki/rtl-sdr .

[12] RTL2832U. [online]. [cit. 2014-12-13]. Dostupné z: http://www.realtek.com.tw/products/productsView.aspx?Langid=1&PFid=35&Level=4&Conn=3&ProdID=257 .

[13] DVB-T USB TUNER S DAB+ FM a SDR s tunerem R820T. [online]. [cit. 2014-12-13]. Dostupné z: http://harektrade.cz/eshop/dvb-t-pc-karty/70-dvb-t-usb-tuner-s-dab-fm-a-sdr-s-tunerem-r820t.html .

[14] PROKEŠ, A. *Rádiové komunikační systémy*. Vyd. 1. V Brně: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2013. 164 s. ISBN 978-xxxxxx.

[15] What is I/Q Data?. [online]. [cit. 2014-12-13]. Dostupné z: http://www.ni.com/white-paper/4805/en/ .

[16] DRAFT prEN 13757-4. [online]. [cit. 2015-5-11]. Dostupné z: http://oldfjarrvarme.unc.se/download/1309/fj .

[17] Modes of AES. [online]. [cit. 2015-5-12]. Dostupné z: http://www.heliontech.com/aes_modes_basic.htm .

[18] VALÁŠEK, M. Symetrické šifrování AES/Rijndael v .NET [online]. 16.4.2007 [cit. 2015-5-12]. Dostupné z: http://www.aspnet.cz/Articles/147-symetricke-sifrovani-aes-rijndael-v-net.aspx .

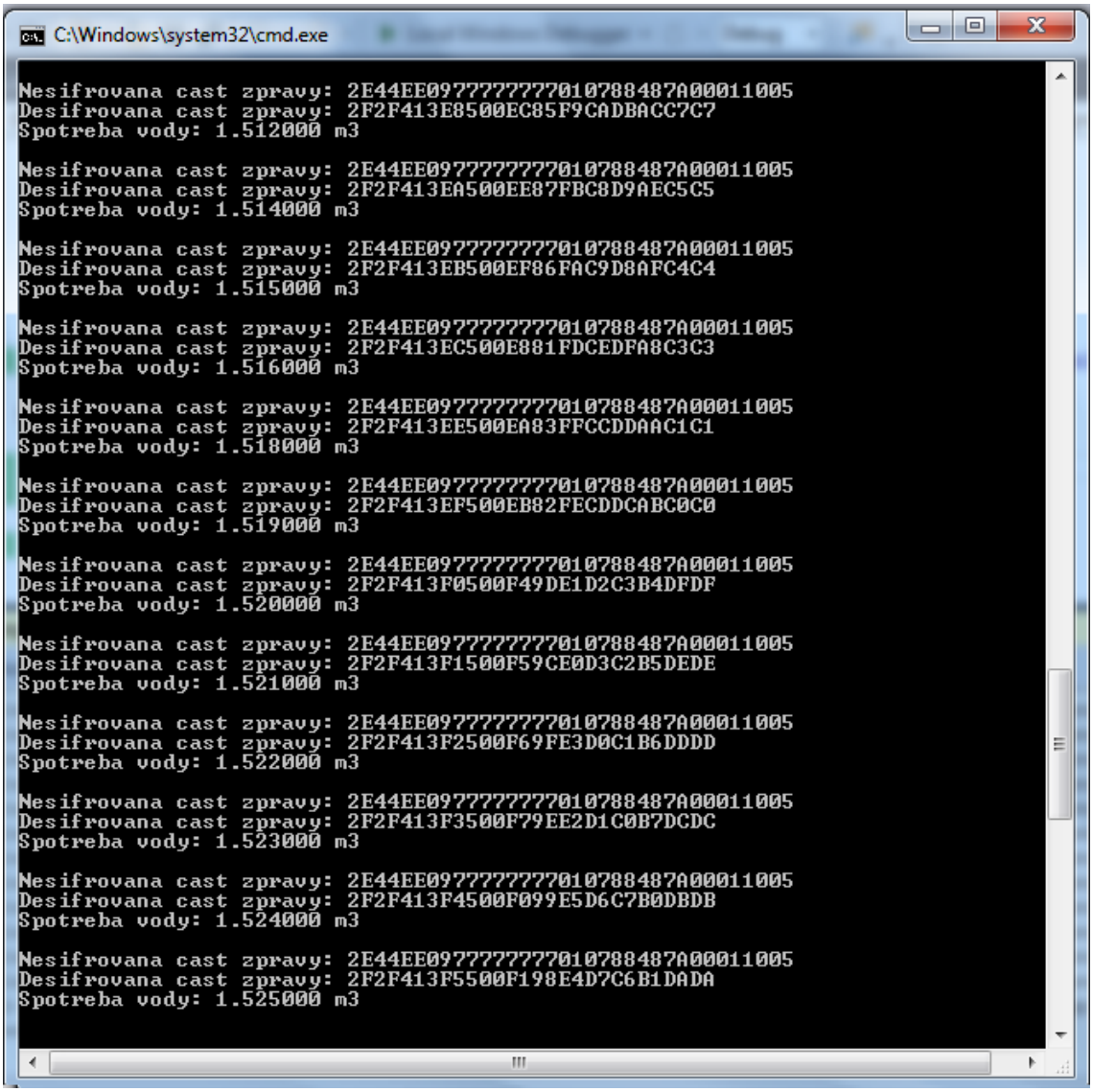
[19] Advanced Encryption Standard (AES). [online]. [cit. 2015-5-19]. Dostupné z: http://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard .

[20] Dedicated Application Layer (M-Bus). [online]. [cit. 2015-5-11]. Dostupné z: http://www.m-bus.com/files/w4b21021.pdf .

[21] KOCÁB, M. *Aplikace pro bezpečné ukládání dat do paměti mobilních zařízení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 42 s. Bakalářská práce. Vedoucí práce: Ing. Tomáš Mácha.

[22] OpenSSL. [online]. [cit. 2015-5-19]. Dostupné z: https://www.openssl.org/ .

A PŘIJATÁ DEKÓDOVANÁ A DEŠIFROVANÁ DATA



B ZOBRAZENÍ PŘIJATÝCH DAT