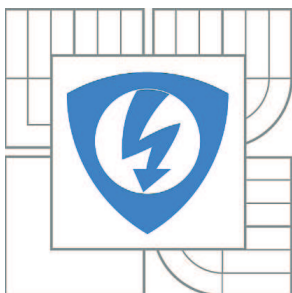


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

FM VYSÍLAČ APRS TELEMETRICKÝCH DAT V PÁSMU 435 MHZ

FM TRANSMITTER OF APRS TELEMETRY IN 435 MHZ BAND

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

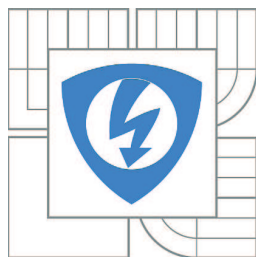
Bc. MARTIN KLÍMA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ URBANEC, Ph.D.

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Martin Klíma

ID: 78591

Ročník: 2

Akademický rok: 2010/2011

NÁZEV TÉMATU:

FM vysílač APRS telemetrických dat v pásmu 435 MHz

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte s obvodem CC1010 FM vysílač s PLL pro FSK a AFSK modulaci v pásmu 435MHz, dále periferní obvody, které umožní sbírat data z A/D převodníku, vytvářet datové rámce a pod. Seznamte se s použitým APRS protokolem. Sestavte schéma zapojení telemetrického modulu.

Vytvořte podklady pro DPS a konstrukci realizujte s maximálním využitím SMD součástek. Napište základní softwarové vybavení pro obsluhu vysílače a potřebných periférií. Ověřte komunikaci s GPS modulem protokolem NMEA.

Realizujte kompletní telemetrický modul včetně potřebného softwarového vybavení pro APRS protokol vysílaný AFSK modulací rychlostí 1200Bd a FSK modulací rychlostí 9600Bd. Realizujte vhodné soft. vybavení pro PC zajišťující kompletní konfiguraci telemetrického modulu.

DOPORUČENÁ LITERATURA:

[1] FREJLACH, K. Paket rádio. České Budějovice: PVT - reprografická sekce, 1994. ISBN 80-900046-3-6

[2] APRS Protocol Specification, Version 1.0 [online]. APRS Working Group, 2000 - [cit.7.5.2008]. Dostupné na [www: http://www.tapr.org/tapr/html/Faprswg.html](http://www.tapr.org/tapr/html/Faprswg.html)

Termín zadání: 7.2.2011

Termín odevzdání: 20.5.2011

Vedoucí práce: Ing. Tomáš Urbanec, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

FM vysílač APRS telemetrických dat je zařízení schopné sbírat telemetrická data z prostředí, ve kterém se nachází, a prostřednictvím radiového kanálu je vyslat uživateli. Je schopno tvořit datové rámce, které umožní formátování a spolehlivý přenos dat. Toto zapojení pracuje v pásmu 435 MHz, používá FSK a AFSK modulaci. Diplomová práce popisuje základní informace o použitých protokolech, zapojení vlastního přístroje a jeho programové vybavení.

KLÍČOVÁ SLOVA

FM vysílač, APRS protokol, NMEA, GPS, telemetrie, telemetrický modul.

ABSTRACT

FM transmitter APRS of telemetric data is device able to measure telemetric values from its ambient. It is able to send this data to user on earth by radio frequency channel. This device can create data frames, with proper formatting allowing sending data. It operates in 435 MHz band, uses FSK and AFSK modulation. This work describes basic information of used protocols, electronic connection of this transmitter and software equipment.

KEYWORDS

FM transmitter, APRS protocol, NMEA, GPS, telemetry, telemetry station.

KLÍMA, M. *FM vysílač APRS telemetrických dat v pásmu 435 MHz*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2011. 50 s. Diplomová práce. Vedoucí práce: Ing. Tomáš Urbanec, PhD.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma FM vysílač APRS telemetrických dat v pásmu 435 MHz jsem vypracoval samostatně pod vedením vedoucího semestrální 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 jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu semestrální práce Ing. Tomáši Urbancovi, PhD. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Děkuji též panu Bc. Tomáši Novotnému za cenné rady při tvorbě programového vybavení a svým blízkým za trpělivost kterou, mi prokázali.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	1
1 Komunikační protokoly	2
1.1 Přenos dat obecně	2
1.2 Komunikační protokol AX.25 – fyzická vrstva	3
1.3 Komunikační protokol AX.25 – spojová vrstva	4
1.4 Protokol APRS	6
1.4.1 Sestava dat typu Position – určení polohy	6
1.4.2 Sestava telemetrického hlášení APRS	7
1.4.3 Parametrické jméno zprávy	8
1.4.4 Jednotky / označení zprávy	8
1.4.5 Rovnice koeficientů zprávy	8
1.4.6 Bitový význam / Projektové jméno zprávy	8
1.5 Protokol NMEA	9
1.5.1 Věta RMC	10
1.5.2 Věta GGA	10
1.5.3 Věta VTG	11
1.5.4 Věta GSA	11
1.5.5 Věta GSV	11
2 Obvody použité v konstrukci	13
2.1 RF transceiver CC1010	13
2.2 Praktické možnosti obvodu CC1010 pro aplikaci protokolu AX.25	14
2.3 GPS modul Leadtek LR 9552	15
3 Návrh zapojení FM vysílače	16
3.1 Vysokofrekvenční část	16
3.2 Digitální část	17
3.3 Napájecí zdroj	17
3.4 Zapojení konektorů vysílače	18
3.5 Deska plošných spojů	22

4	Návrh přenosu dat	24
4.1	Obecný popis hardware vysílače pro přenos APRS hlášení	24
4.2	Obecný popis software vysílače pro přenos APRS hlášení	25
5	Návrh softwarového vybavení	28
5.1	Obsluha periférií, inicializační funkce	29
5.1.1	Sériové linky	29
5.1.2	A/D převodník	29
5.1.3	Časovače	29
5.1.4	RF vysílač	30
5.1.5	Obsluhy přerušení	31
5.2	Příprava dat	32
5.2.1	Měření analogových hodnot	32
5.2.2	NMEA parser	32
5.3	Zpracování dat – aplikace APRS a AX.25 protokolu	33
5.3.1	Tvorba rámce AX.25	34
5.3.2	Tvorba adresního a řídicího pole	36
5.3.3	Bit stuffing, NRZI kódování	37
5.3.4	Výpočet CRC kódu	38
5.3.5	Skrambler	39
5.3.6	Vysílání vytvářených dat	40
5.3.7	Hlavní programová smyčka	41
5.4	Konfigurace modulu	44
5.4.1	Programové zajištění ze strany vysílače	44
5.4.2	Programové zajištění ze strany PC	45
6	Použité programovací prostředí	47
7	Dosažené výsledky	48
8	Závěr	49
	Literatura	50

ÚVOD

FM vysílač APRS telemetrických dat slouží k vysílání digitálních dat obsahujících telemetrické informace. Jedná se o přístroj schopný sebrat či přijmout množství informací, vhodným způsobem je upravit a radiovým kanálem vyslat k jednomu či více uživatelům. FM vysílač APRS telemetrických dat pracuje v pásmu 435 MHz.

Obsahem této diplomové práce je návrh FM vysílače založeného na obvodu CC1010 pro FSK a AFSK modulaci v pásmu 435 MHz, dále zapojení umožňující sběr dat z A/D převodníku. V konstrukci se počítá s komunikací s GPS modulem pro zjištění zeměpisné polohy a se sběrem dat z různých snímačů jako je čidlo tlaku nebo teploty. V neposlední řadě má být modul vysílače schopný podat zprávu o stavu svého napájecího zdroje a dalších veličinách, protože provoz modulu je zcela autonomní bez možnosti zásahu obsluhy. Dále práce obsahuje seznámení s použitým protokolem APRS, který umožňuje přenos telemetrických dat. Konečně cílem práce je navrhnout schéma zapojení telemetrického modulu, obrazec desky plošných spojů a kompletní softwarové vybavení k zajištění vysílání hlášení dle protokolu APRS. Závěrem práce je umožnit konfiguraci základních parametrů vysílače a navrhnout vhodné SW vybavení.

1 KOMUNIKAČNÍ PROTOKOLY

Vysílač APRS telemetrických dat má za úkol naměřit potřebné analogové hodnoty, odečíst další digitální data z připojených periférií, jako je přijímač GPS a tyto data korektně odvysílat. Pro činnost telemetrického vysílače je nutné použít přenosový protokol splňující požadavky dané zvyklostmi, jež jsou osvědčené, zejména z důvodu kompatibility s ostatními zařízeními. Pro práci vysílače je dobrým zvykem používat protokol APRS, stojící jako nadstavba nad protokolem AX.25. Nejdříve je třeba říci, jaké mají tyto protokoly vlastnosti a možnosti použití. Zároveň je třeba si uvědomit, v jakých vrstvách OSI modelu pracují. Proto je nejdříve uveden popis protokolu AX.25, nad kterým je potom použit protokol APRS popsáný dále.

1.1 Přenos dat obecně

Přenos dat je problematika rozsáhlá a velmi široká. Obsahuje veškeré přenosové prostředky od nejnižší úrovně fyzikálních jevů až po úroveň nejvyšší, kdy se zabýváme kódováním a šifrováním dat. Takto obsáhlá oblast je rozdělena pod regulací organizace ISO (International Standards Organization) v rámci mezinárodního standardu OSI (Open System Interconnection) na sedm vrstev, viz. Obrázek 1, převzatý z [8]. V těchto vrstvách se definuje způsob, jakým budou data zpracovávána a podle potřeby doplňována dalšími informacemi. Ve své podstatě je OSI model teoretický a značně obecný. Proto se v komunikačních systémech v praxi využívají jen některé z definovaných vrstev, nebo mohou některé vrstvy splývat. Při popisu modelu OSI postupujeme od nejnižší úrovně výše.



Obr. 1: Referenční model ISO / OSI

Základním kamenem přenosového systému je prostředí, kterým se šíří fyzikální veličina reprezentující informaci. Toto prostředí je běžně látkového charakteru. Tradičně se pro přenos informace používá elektrický obvod uzavřený přes kovové vodiče tvořící *vedení*. Takové vedení nazýváme metalické. Dále je možno využít šíření elektromagnetických vln ve volném prostoru, resp. atmosféře, popřípadě ve zvláštním druhu vedení. Pro tuto práci je nejzajímavější přenos radiových elektromagnetických vln volným (atmosférickým) prostředím, protože umožňuje mobilitu komunikujících účastníků či zařízení na značné vzdálenosti bez nákladů na spojovací médium. Při úvaze nad vlastnostmi tohoto prostředí vystoupí do popředí nevýhody dané jeho sdílením. Jedná se o rušení nebo únik signálu či jiné rušivé vlivy působící na přenášený signál. Vzhledem k těmto vlastnostem je třeba hodnotit kvalitu přenosu, což lze za pomoci

kritérií jako *poměr signál / šum* (SNR – Signal to Noise Ratio) či *chybovost* (BER - Bit Error Rate). Těmito parametry se definuje první vrstva v modelu OSI – *fyzická vrstva*.

Dalším hlediskem při zajišťování přenosu je otázka zabezpečení proti chybám vzniklým při přenosu Data, která přenášíme, je možné a vhodné zabezpečit proti ztrátě informace pomocí komunikačních protokolů. Tyto protokoly stanoví způsob, jakým jsou data upravena zpravidla do bloků a doplněna informacemi o jejich délce, pořadí, času vytvoření a podobně. Data obsažená v blocích se nazývají *rámec*; přitom nad rámci lze provést zabezpečení, které umožní při příjmu poškozeného rámce chybu identifikovat nebo opravit. Takové datové zabezpečení definuje *spojová vrstva* v OSI modelu.

Nalezení vhodné cesty v síti a směrování dat zabezpečuje v OSI modelu *síťová vrstva*, ve které se k rámcům dat přidávají další informace potřebné pro směrování. Těmi jsou zejména adresa příjemce a adresa odesílatele zprávy, adresy průchozích míst v síti. Takto vytvořený blok dat se nazývá *paket*. Během přenosu paketu sítí se do něj postupně přidávají informace umožňující zejména zabezpečení a adresování.

Mimo tyto tři tzv. základní vrstvy, které obsahuje každý systém, zahrnuje OSI model ještě vrstvy vyšší. Těmi jsou vrstva transportní, vrstva relační, vrstva prezentační a vrstva aplikační. Tyto nejsou nutné k přenosu dat v síti a nejsou využity pro potřeby popisovaného telemetrického modulu, tím pádem nejsou popisovány. Podrobnější popis viz. [1].

1.2 Komunikační protokol AX.25 – fyzická vrstva

Fyzická vrstva specifikuje požadavky na spoj z elektrického a mechanického hlediska. Zároveň jsou zde specifikovány požadavky na modulace a demodulace. Vzhledem k tomu, že protokol AX.25 je poměrně málo definován, je možné použít různé modulace a různé přenosové rychlosti. Pro radioamatérskou praxi je ale zvykem používat protokol AX.25 pro provoz paketrádiové sítě, kde se nejčastěji používají dvoustavové modulace FSK (Frequency Shift Keying – kmitočtové klíčování) či AFSK (Audio Frequency Shift Keying – kmitočtové klíčování v akustickém pásmu). Modulace FSK používá kmitočtový skok nosné vlny, kdežto modulace AFSK využívá kmitočtový skok akustického signálu, tj. v základním pásmu, kterým je modulována nosná. Jeden ze dvou možných kmitočtů vždy reprezentuje logickou úroveň přenášených dat. Na přijímací straně se provádí FM demodulace, po které je nutno demodulovanému signálu přiřadit logické úrovně odpovídající modulačnímu signálu.

Pro komunikaci v paketrádiové síti je třeba vybavení. V podstatě stačí použít běžný komunikační transceiver a modem pro příslušný druh provozu nebo inteligentní modem, tzv. TNC (Terminal Node Controller). Na rozdíl od běžného modemu poskytuje TNC na výstupu přímo uživatelská data, protože obsahuje mikropočítač, který přímo provádí dekódování dat z přijatých rámců. Připojení modemu či TNC je na běžný výstup na reproduktor či sluchátka transceiveru a mikrofonní vstup. Právě zde je slabé místo z hlediska modulační rychlosti. Díky tomuto omezení a dovolené šířce pásma je v nižších radioamatérských pásmech (29 – 144 MHz s FM modulací) vymezený segment s nízkou modulační rychlostí 1200 Bd a modulací AFSK. Šířka pásma jednoho kanálu zpravidla odpovídá šířce pásma fonického kanálu, tedy 12,5 kHz, resp. dnes se používá pro fonii i kanálový krok 5 kHz.

Ve vyšších radioamatérských pásmech (pro nás zajímavých 430 MHz) se používá také vyšší modulační rychlost 9600 Bd. Ovšem, jak bylo naznačeno výše, zde se nepoužívají transceivery s připojením modemu přes audio rozhraní, nýbrž transceivery přímo konstruované pro tento druh provozu. Digitální modulace je zavedena přímo do FM modulátoru – tzn. používá se modulace FSK. Datový modulační signál je přivedený přímo do smyčky fázového závěsu, která je tímto signálem řízena. Podobně se z fázového závěsu odebírá digitální signál při demodulaci. Toto uspořádání v zásadě vylučuje zkreslení digitálního signálu v nízkofrekvenčních obvodech transceiveru a tím omezuje vznik ISI – mezisymbolových interferencí. Pro spojení mezi nody a uzly BBS se používá modulační rychlost i vyšší než 9600 Bd.

Spojení s terminálem či osobním počítačem je realizováno datovým modemem či TNC. Obě tato zařízení plní základní funkci, tj. přiřazují při vysílání logické hodnotě napětíovou úroveň a při příjmu naopak. Mimo to se modulační signál upravuje dolní propustí, případně jiným filtrem. V pásmu 430 MHz se používá gaussovský filtr představující přijatelný kompromis mezi zabránou šířkou pásma a možností zpětného obnovení signálu. Jeden z nejrozšířenějších modemů zajišťující toto zpracování je modem podle G3RUH. TNC na rozdíl od modemu provádí také datové kódování a dekódování, případně data dále zpracovává. Znamená to, že na výstupu TNC jsou dostupná uživatelská data, nikoliv jen demodulované logické úrovně jako v případě modemu. Data na výstupu TNC jsou na úrovni *spojové vrstvy*. Zpracování na úrovni *fyzické vrstvy* náleží TNC a transceiveru. V případě použití modemu se terminál či počítač podílí na kódování a dekódování dat, tedy ve *fyzické vrstvě*. Jinou možností je použít pro spojení tzv. datový transceiver slučující TNC a speciální transceiver do jednoho bloku. Datový transceiver komunikuje s radiovou sítí na úrovni *fyzické vrstvy*; na datovém rozhraní pak na úrovni *spojové vrstvy*.

Podobně jako v případě použitých modulací a modulačních rychlostí nespecifikuje protokol AX.25 ani datové kódování. Jistým standardem se však stalo kódování NRZI pro použití v pásmech 144 MHz a nižších. V pásmech 430 MHz a výše se používá NRZI kódování pro pozemní spoje, pro družicové spoje potom kódování Manchester. Pro potřeby této práce se počítá s použitím kódování NRZI.

NRZI kódování (Non Return to Zero Inverted) je ze skupiny NRZ kódů. Princip kódování NRZI spočívá v tom, že se mění logický stav na opačný s každým příchodem nuly, přitom při příchodu logické jedničky zůstává stávající stav zachován. Přestože má tento kód zásadní nevýhodu v nenulové střední hodnotě, řeší se tato nevýhoda protokolem AX.25 vkládáním nuly vždy po pěti jedničkách. Toto se zároveň používá k rámcové synchronizaci.

1.3 Komunikační protokol AX.25 – spojová vrstva

Protokol AX.25 je postaven na protokolu CCITT X.25 a popisuje spojovou vrstvu pro amatérské vysílání. Zajišťuje spolehlivý přenos dat, přičemž přenášená data mohou mít libovolnou strukturu. Data nemusí tedy být nutně osmibitová, není vyžadovaný bajtový tvar, pouze je třeba je doplnit na počet bitů dělitelný osmi. Díky tomu je možno jednoduše implementovat protokol APRS do informačních bloků protokolu AX.25. Protokol AX.25 tedy zajišťuje zabezpečení přenosu a protokol APRS zajišťuje formátování telemetrických dat.

Protokol AX.25 používá dvě skupiny rámců:

- číslované – informační (I) rámce slouží k přenosu dat a řídicí (S) rámce slouží k přenosu informací o spojení v jeho průběhu,
- nečíslované – rámce (U) a (UI) slouží k navazování a ukončování spojení nebo při hrubé chybě přenosu a konverzačním režimu stanic.

Struktura rámců je dána jednoznačně. Obsahují pole s daným počtem bitů, mimo to je pevně stanoven obsah prvního a posledního pole flag rámce na hodnotu 7F(hex) neboli 01111110(b), která se nesmí nikde jinde v rámci vyskytovat. Blíže viz. Tab. 1 a 2. Mimo užitečných dat se v rámcích pomocná data pro zajištění synchronizace přijímané zprávy, označení pořadí bloků a další jako např. cyklický součet pro detekci chybných rámců.

Tab. 1: Skladba U- a S-rámců.

flag	adresa	ctrl	FCS	flag
7Eh	112..560b.	8b.	16b.	7Eh

Tab. 2: Skladba I- a UI-rámců.

flag	Adresa	ctrl	PID	Info	FCS	flag
7Eh	112..560b.	8b.	8b.	N*8 bitů	16b.	7Eh

Význam jednotlivých polí rámce

Pole flag

Pole flag je dlouhé minimálně jeden bajt a obsahuje pevně danou hodnotu sloužící k synchronizaci rámců, jak již bylo naznačeno výše. Vzhledem k potřebě ustálení parametrů vysílače a potřebě synchronizovat přijímač je třeba vyslat typicky deset a více flagů. Pole flag obsahuje šest logických “1”, a tím je jediným blokem, který nemůže obsahovat bit-stuffing sloužící k zamezení výskytů flagů. Mimo pole flag musí být v každém bloku po každých pěti jedničkových bitech vložena nula. Příjemce tento nulový bit musí očekávat a z přijatých dat ho vyjmout.

Pole adresa

Pole adresa obsahuje adresu příjemce a adresu odesílatele. Adresy se skládají z velkých písmen a číslic ASCII kódu kromě bajtu SSID. Délka adresního pole je volitelná. K rozšíření adresního pole slouží nejnižší bit. Je-li tento bit nula, počítá se s dalším bajtem adresy. Z toho vyplývá, že je v každém bajtu adresního pole adresa vynásobena dvěma právě proto, aby se uvolnil poslední bit pro zmíněnou indikaci.

Pole PID

Pole PID je určené k identifikaci použitého protokolu v I a UI blocích. PID má význam typu protokolu síťové vrstvy. Pro využití v telemetrickém modulu, kde se

síťová vrstva nevyužívá je vhodné nastavení pole PID = síťová vrstva není implementována, tedy PID = F0(hex).

Pole info

Pole info slouží k přenosu zpráv, tedy vlastních dat s omezením délky na 256 bajtů. Existují tři typy bloků - I, UI a FRMR. FRMR bloky slouží k zajištění bezchybného přenosu dat v oboustranném spojení. V jednostranném spojení tento blok nemá smysl. Při jednostranném spojení lze dobře použít I a UI bloky. Pro standardizovaný příjem dat je vhodné přenášet data v UI bloku pomocí APRS rámce.

Pole FCS

Pole kontrolního součtu FCS je číslo dlouhé 16 bitů, sloužící k zabezpečení bezchybného přenosu dat. Pokud se vyskytne při příjmu chyba v součtu FCS, znamená to ztrátu bloku dat, protože v jednostranném spojení nelze žádat o opakování přenosu. Příjemce tedy o tento blok dat přijde.

Sled přenášených bitů

V protokolu AX.25 platí, že se vysílá nejdříve nejméně významný bit (LSB) s jednou výjimkou, v případě pole FCS se vysílá nejdříve nejvíce významný bit (MSB). Protokol AX.25 je v radioamatérské praxi velice rozšířený. Zde uvedený popis shrnuje nejdůležitější vlastnosti protokolu vzhledem k potřebám této práce. Podrobný rozbor lze nalézt v literatuře [1].

1.4 Protokol APRS

Protokol APRS je protokol používaný pro přenos textů, parametrů, souřadnic a dalších veličin a dat. Používá se společně s protokolem AX.25, který zajišťuje zabezpečení přenosu v plném rozsahu, tj. obsahuje adresy, synchronizuje rámce a zabezpečuje přenos proti chybám. APRS protokol je nadstavba, která je umísťována do informačních rámců protokolu AX.25. Vzhledem k charakteru přenosu, kdy není třeba číslovat rámce, ani není možné použít řízení přenosu, budou použity UI rámce. Identifikace APRS dat je dána určeným parametrem. Podrobnější informace lze nalézt v [2]. Vysílač je uvažovaný k vysílání dat typů telemetrické hlášení a určení polohy s časovým údajem.

1.4.1 Sestava dat typu Position – určení polohy

Rámec určení polohy s časovým údajem, viz. Tab. 3, obsahuje identifikaci ASCII znakem „/“ nebo „@“. Dále následuje časový údaj ve formátu HMS, údaje o zeměpisné šířce, délce, volitelném rozšíření a pole komentáře. Pole zeměpisné délky a šířky je rozděleno znakem „/“ a zakončeno znakem „-“, pokud je poloha známa, jinak jsou použity znaky „\“, resp. „.“.

Rámec určení polohy umožňuje v poli APRS data extension použít rozšiřující informace, mezi které patří i kombinace údajů kurz / rychlost, kterou požadujeme vysílat, pokud je známa. Kurz je udaný ve stupních, rychlost v uzlech. Podobně protokol umožňuje do pole komentář umístit údaj o nadmořské výšce ve stopách, který

je také požadován.

Tab. 3: Rámec určení polohy s časovým údajem.

	@	Čas hhmmsssh	Zem. šířka ddmm.hhN	Symbol ID /	Zem. délka dddmm.hhE	Symbol -	Kurz / rychlost CSE/SPD	Komentář /A=aaaaaa
Bajty	1	7	8	1	9	1	7	0-36

1.4.2 Sestava telemetrického hlášení APRS

Protokol APRS je do jisté míry univerzální; pro použití v této práci je použito tzv. telemetrické hlášení, které odpovídá požadavkům na přenos veličin, včetně jejich číselné hodnoty a jednotky. Rámec telemetrického hlášení nazývaný též rámec dálkového měření slouží pro přenos číselných hodnot. Identifikace varianty telemetrické hlášení je provedena na začátku každého rámce bloku sestavy rámců prvním bajtem s hodnotou ASCII „T“.

Blok rámců je základní jednotkou, která nese číselnou hodnotu zároveň s pomocnými informacemi jako je např. označení jednotek. Dále rámec obsahuje řadu čísel se třemi hodnotami, zpravidla tři digitální čísla určující pořadí, nebo tři znaky „MIC“. Po těchto znacích se očekává přijetí analogových a digitálních hodnot reprezentovaných neoznačenými pěti skupinami analogových a digitálních dat v rozsahu 0-255. Analogové hodnoty jsou kódovány v ASCII – každá po třech znacích a oddělovací čárce. Digitální hodnota je také v ASCII – znaky „0“ nebo „1“. Označení dat se zajišťuje ostatními rámci. Po přijetí těchto pěti analogových hodnot následuje osmibitová digitální hodnota, za kterou může být umístěn komentář. Struktura rámce je naznačena v Tab. 4.

Tab. 4: Rámec telemetrického hlášení.

	T	Označení bloku # xxx ,	Analogová hodnota 1 aaa ,	Analogová hodnota 2 aaa ,	Analogová hodnota 3 aaa ,	Analogová hodnota 4 aaa ,	Analogová hodnota 5 aaa ,	Digitální hodnota bbbbbbbb	K
B	1	5	4	4	4	4	4	8	n

Sestava telemetrických dat

Parametry přenášené systémem může libovolně definovat uživatel, přičemž se přidržuje pravidel APRS protokolu. Tyto parametry jsou obecné, nejsou specifikované předem. Z důvodu, aby příjemce zprávy věděl, o jaká data jde, je třeba nejprve přenést pomocná data, která definují jednotky přenášených dat. K tomu slouží sada čtyř rámců:

- Parametrické jméno zprávy
- Jednotky / označení zprávy
- Rovnice koeficientů zprávy
- Bitový význam / Projektové jméno zprávy

1.4.3 Parametrické jméno zprávy

Parametrické jméno zprávy je rámec obsahující názvy pěti analogových a osmi digitálních kanálů jedné zprávy dálkového měření. Struktura rámce viz. Tab. 5.

Tab. 5: Parametrické jméno zprávy.

	PARM.	A1 N	A2 , N	A3 , N	A4 , N	A5 , N	B1 , N	B2 , N	B3 , N	B4 , N	B5 , N	B6 , N	B7 , N	B8 , N
Bajty	5	1-7	1-7	1-6	1-6	1-5	1-6	1-5	1-4	1-4	1-4	1-3	1-3	1-3

1.4.4 Jednotky / označení zprávy

Jednotky/označení zprávy je rámec obsahující jednotky pěti analogových a osmi digitálních hodnot v jedné zprávě. Struktura rámce viz. Tab. 6.

Tab. 6: Rámec zprávy nesoucí jednotky.

	UNIT.	A1 U	A2 , U	A3 , U	A4 , U	A5 , U	B1 , L	B2 , L	B3 , L	B4 , L	B5 , L	B6 , L	B7 , L	B8 , L
Bajty	5	1-7	1-7	1-6	1-6	1-5	1-6	1-5	1-4	1-4	1-4	1-3	1-3	1-3

1.4.5 Rovnice koeficientů zprávy

Rovnice koeficientů zprávy je rámec obsahující po třech koeficientech kvadratické rovnice pro všechny analogové kanály. Tímto je možné výslednou hodnotu korigovat na hodnotu číselně odpovídající běžně používané stupnici. V rovnici koeficientů zprávy (1) není nutné použít všechny koeficienty a, b, c . Bajty n jsou ASCII hodnoty. Struktura rámce viz. Tab. 7.

$$u = a \cdot v^2 + b \cdot v + c \quad (1)$$

Tab. 7: Rámec rovnice koeficientů zprávy.

	EQNS.	A1			A2			A3			A4			A5		
		a	,b	,c	,a	,b	,c	,a	,b	,c	,a	,b	,c	,a	,b	,c
Bajt	5	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n

1.4.6 Bitový význam / Projektové jméno zprávy

Bitový význam / Projektové jméno zprávy je rámec obsahující dva typy informací. Osm bitů tohoto bloku určuje, který digitální kanál má smysl číst. Zápis je v bitové ASCII formě s odpovídajícími nulami a jedničkami. Druhá část zprávy určuje jméno projektu spojeného s telemetrickým modulem. Struktura rámce viz. Tab. 8.

Tab. 8: Rámec bitového významu a projektového jména zprávy.

	BITS.	B1 X	B2 x	B3 x	B4 x	B5 x	B6 x	B7 x	B8 x	Jméno projektu
Bajty	5	1	1	1	1	1	1	1	1	0-23

Pro aplikaci měření telemetrických dat je možno přímo použít tento protokol. Protokol APRS umožňuje přenést hodnoty, označení veličiny a jednotky. Kromě toho lze přenášet korekční koeficienty až do druhého řádu v rámci *Koeficienty zprávy*. Rámec *Bitový význam a jméno zprávy* lze využít pro označení meteorologického balónu, zároveň jej lze použít k označení počtu měřených veličin. Podrobněji viz. [4].

Protokol APRS mimo jiné umožňuje specifikovat SSID, které umožní rozlišit druh a tím i symbol vysílací stanice. Lze je vysílat třemi různými způsoby. První z nich využívá příslušné pozice v adresním poli AX.25. Toto SSID je uloženo jako poslední ASCII znak za zdrojovou adresou a implicitně je nulové. Další možností je vložit kód symbolu do informační části protokolu. Zajímavé jsou např. varianty /> (Automobil), /O (Balón), /b (Jízdní kolo), které v první variantě, tedy umístěním do SSID bajtu adresního pole zdrojové adresy nabývají binárních hodnot 9, 11 a 4.

Díky umístění protokolu APRS do Info bloku protokolu AX.25 vznikl protokol použitelný k přenosu telemetrických dat, který je možno provozovat na běžných zařízeních pro provoz paket rádia.

1.5 Protokol NMEA

Protokol NMEA je standardním protokolem k vysílání dat nejen GPS přijímačů. NMEA výstup je typu EIA422A, ale je možné ho považovat za kompatibilní s RS-232, který se používá zcela běžně např. jako rozhraní osobních počítačů. NMEA používá parametry přenosu:

- rychlost 4800 b/s
- formát 8 datových bitů, bez parity, 1 stop bit.

Všechny věty NMEA 0183 jsou kódovány v ASCII, což zjednodušuje komunikaci. Formát NMEA vět je specifický; začátek každé věty začíná znakem \$ a končí znaky <CR><LF>. Obsah věty je rozdělen znakem čárka, znamená to také, že každému údaji ve větě přenášenému patří oddělovací čárka, která musí být uvedena, i když parametr samotný ve větě není obsažen. Jen tak je zabezpečena detekce žádaných přijímaných dat. Závěrem věty může být uvedený kontrolní součet. Za počátečním znakem \$ následuje adresní pole ve formátu acccc, kde aa je ID zařízení, v případě GPS přijímače má hodnotu GP. Na pozicích ccc se nachází formátovací sekvence neboli jméno zprávy. Podrobnosti viz. [6]. Formát nejdůležitějších zpráv je uveden níže.

1.5.1 Věta RMC

Formát věty:

\$GPRMC,hhmmss.ss,A,III.II,a,yyyy.yy,a,x.x,x.x,ddmmyy,x.x,a*hh

Význam věty a jejího obsahu:

RMC = Doporučená minimální GPS/TRANSIT data

- 1 = UTC zachycení polohy**
- 2 = Status dat (V=varování přijímače)**
- 3 = Zeměpisná šířka**
- 4 = N nebo S**
- 5 = Zeměpisná délka**
- 6 = E nebo W**
- 7 = Rychlost nad zemí v uzlech**
- 8 = Skutečný kurz ve stupních**
- 9 = UT datum**
- 10 = Úhel magnetické variace**
- 11 = E nebo W**
- 12 = Kontrolní součet**

Ve větě RMC jsou soustředěna nejdůležitější data jak o poloze uzamknutí přijímače, časovém údaji uzamknutí, tak také o kurzu a rychlosti pohybu. Jeví se tedy jako postačující zdroj informace, ovšem neobsahuje údaj o nadmořské výšce. Ten je obsažen ve větě GGA, viz. níže.

1.5.2 Věta GGA

Formát věty:

\$GPGGA,hhmmss.ss,III.II,a,yyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh

Význam věty a jejího obsahu:

GGA = GPS data zachycení

- 1 = UTC zachycení polohy**
- 2 = Zeměpisná šířka**
- 3 = N nebo S**
- 4 = Zeměpisná délka**
- 5 = E nebo W**
- 6 = GPS indikátor kvality (0=neplatné; 1=GPS zachycení; 2=Dif. GPS zachycení)**
- 7 = Počet použitých satelitů**
- 8 = Horizontální znepřesnění polohy**
- 9 = Výška nad mořem (geoidem)**
- 10 = M (jednotka výšky)**
- 11 = Odstup geoidu (rozdíl mezi elipsoidem WGS-84 a střední výškou moře)**
- 12 = M (jednotka odstupu)**
- 13 = Doba od posledního updatu od diferenční stanice**
- 14 = ID diferenční stanice**
- 15 = Kontrolní součet**

Věta GGA obsahuje vedle doplňujících informací jako jsou počet použitých satelitů či stáří posledního updatu z diferenční stanice především časový údaj a údaj o poslední zjištěné poloze. Tato věta bude využita jako hlavní zdroj informace pro

vysílač, konkrétně jako zdroj časového údaje, zeměpisné polohy a nadmořské výšky. Dále je potenciálně zajímavá následující věta VTG, ovšem nebude použita – údaje o kurzu a rychlosti budou získána z věty RMC.

1.5.3 Věta VTG

Formát věty:

\$GPVTG,t,T,,,s.ss,N,s.ss,K*hh

Význam věty a jejího obsahu:

VTG = Aktuální kurz a rychlost nad zemí

- 1 = Kurz**
- 2 = Pevný text 'T' indikující, že kurz je vůči skutečnému severu**
- 3 = nepoužito**
- 4 = nepoužito**
- 5 = Rychlost nad zemí v uzlech**
- 6 = Pevný text 'N' indikující jednotku uzly**
- 7 = Rychlost nad zemí v km/h**
- 8 = Pevný text 'K' indikující jednotku km/h**
- 9 = Kontrolní součet**

Věta VTG – Obsahuje údaj o rychlosti pohybu nad zemí v nejpoužívanějších jednotkách – tedy informace o aktuálním pohybu.

1.5.4 Věta GSA

Formát věty:

\$GPGSA,A,3,19,28,14,18,27,22,31,39,,,,,1.7,1.0,1.3*35

Význam věty a jejího obsahu:

GSA = mód GPS přijímače, satelity použité k navigaci a DOP hodnoty.

- 1 = Mód:**
 - M=Manuální, přepnutý do módu 2D nebo 3D**
 - A=Automatický, 3D/2D**
- 2 = Mód:**
 - 1=Nezachyceno**
 - 2=2D**
 - 3=3D**
- 3-14 = ID použitých satelitů (nula v nepoužitých polích)**
- 15 = PDOP**
- 16 = HDOP**
- 17 = VDOP**

Obsah věty GSA má význam informace o stavu přijímače; zejména informace o módu zachycení přijímače je významná – určuje, zda je GPS přijímač schopen určit třetí rozměr – nadmořskou výšku, rovněž podle hodnot DOP lze usuzovat na kvalitu údajů.

1.5.5 Věta GSV

Formát věty:

\$GPGSV,4,1,13,02,02,213,,03,-3,000,,11,00,121,,14,13,172,05*67

Význam věty a jejího obsahu:

GSV = počet viditelných satelitů, hodnoty PRN, elevace, azimut a hodnoty SNR.

- 1 = Celkový počet zpráv**
- 2 = Číslo zprávy**
- 3 = Celkový počet viditelných satelitů**
- 4 = Hodnota PRN**
- 5 = Elevace ve stupních, 90 maximum**
- 6 = Azimut, úhel ke skutečnému severu, 000 - 359**
- 7 = SNR, 00-99 dB (nula pokud není použit)**
- 8-11 = Informace o druhém satelitu, stejná pole jako 4-7**
- 12-15 = Informace o třetím satelitu, stejná pole jako 4-7**
- 16-19 = Informace o čtvrtém satelitu, stejná pole jako 4-7**

Tato věta podává informace o poloze až čtyř viditelných družic a o kvalitě signálu z nich přijímaného.

Zejména požadované či vhodné informace jsou uzamknutí pozice, aktuální poloha a nadmořská výška, kurz a rychlost pohybu a dále doplňující informace o stavu přijímače. Tyto informace nejsou nutně požadovány, ale například počet „viditelných“ družic a kvalita jejich signálu může být vhodným doplněním informací základních. Podporované věty obecně doporučené v NMEA specifikaci [6] jsou GGA, GLL, RMC a VTG.

2 OBVODY POUŽITÉ V KONSTRUKCI

2.1 RF transceiver CC1010

Obvod CHIPCON CC1010 je integrovaný FM transceiver s řídicím mikropočítačem na jednom čipu. Je ideální součástí pro konstrukci malých transceiverů v pásmu 300 – 1000 MHz. Vyžaduje pouze minimum pasivních součástek, přitom je velmi dobře použitelný. Blíže viz. [4].

FM vysílač je umístěn společně na čipu s mikroprocesorem a jeho pomocnými obvody. FM vysílač je konfigurován a kontrolován pomocí speciálních SFR registrů. Přes speciální SFR registry je také prováděn zápis a čtení dat pro vysílání a pro příjem. Čtení nebo zápisu dat se žádá pomocí přerušení nebo testovacích bitů. Tento obvod má značně variabilní vlastnosti a možnosti použití, blíže viz. [3].

Řídicí část obvodu, mikrokontrolér s jádrem 8051 je 2,5krát rychlejší než běžné obvody s těmito jádery. Obvod také umožňuje výběr krystalového oscilátoru mezi hlavním a pomocným, který slouží pro blok reálného času a má mnohem menší pracovní kmitočet. Tím je možné šetřit napájecí zdroje. V tomto režimu ale není možné přenášet data mezi mikroprocesorem a FM transceiverem. Pomocné obvody lze použít stejně jako u ostatních obvodů.

Parametry RF transceiveru:

- pracovní kmitočet 300MHz až 1000MHz
- velmi malá proudová spotřeba (RX: 9.1mA)
- vysoká citlivost (typická hodnota -107 dBm při 2.4 kBaud)
- programovatelný výstupní výkon (od -20 do +10 dBm)
- FSK modulace s horní přenosovou rychlostí do 76.8kBaud
- velmi málo externích součástek
- rychlý PLL fázový závěs
- RSSI – indikátor intenzity přijímaného signálu (možnost vzorkování ADC)
- samostatný anténní vstup pro přijímač a samostatný anténní výstup pro vysílač
- Manchester nebo NRZ hardware kódování a dekodování
- vyhovuje s EN 300 220 a FCC CFR47 části 15

Parametry mikrokontroléru 8051:

- typicky 2.5x výkonnější, než klasický mikrokontrolér 8051
- 8 bitové jádro s odděleným prostorem pro program a pro data (Harvardská architektura)

- 32kB flash, 2048+128 Byte SRAM
- interní registry organizované do 4 bank po 8 registrech
- instrukční sada podporuje přímé i nepřímé a registrové adresování
- datová paměť je rozdělena do interní a externí RAM (obě jsou na čipu, pouze externí je dostupná jen několika instrukcemi)
- instrukční čas trvá 4 hodinové cykly
- dvojité data ukazatel
- periférie jsou řízeny přes speciální SFR registry umístěné v interní RAM
- 4x čítače
- vstupně/výstupní porty s celkem 26 piny
- 3 kanálový 10bitový ADC (max. 23 kSamples/s)
- 2x PWM
- Watchdog
- SPI (sériový komunikační interface)
- RTC (obvod reálného času, který může generovat přerušení 1 až 127 sekund)
- DES (56 bitový šifrovací klíč v rozsahu 1 až 256 bajtů)
- napájení v rozsahu 2.7V až 3.6V
- 64 vývodové pouzdro TQFP
- IDLE a SLEEP mód (režim nečinnosti a spánku)

2.2 Praktické možnosti obvodu CC1010 pro aplikaci protokolu AX.25

Protokol APRS umístěný v informační části protokolu AX.25 lze požadovat za přenášená data a dále se není třeba v této části kapitoly jeho existencí zabývat. Velikost paměti RAM je 2048 bajtů a je tedy možnost uložit vytvářený protokol do paměti RAM, která je součástí čipu CC1010. Vytvořený kompletní protokol, včetně výpočtu cyklického zabezpečení, je možné začít vysílat s pouhým vyčítáním z paměti a minimálním programovým zatížením mikroprocesoru.

Lze říci, že obvod CC1010 je schopný generovat a zpracovat rámce protokolu AX.25 a protokolu APRS. Obvodem je možné, při vhodném návrhu měřících prvků, měřit požadované meteorologické veličiny.

2.3 GPS modul Leadtek LR 9552

GPS modul Leadtek LR 9552 je kompaktní modul s integrovanou plošnou anténou. Zde uvedený přehled technických dat je jen výčet nejdůležitějších parametrů; podrobnosti lze nalézt v [5]. Základní podmínka pro použití tohoto modulu je kompatibilita napájení a komunikačního rozhraní. Modul podporuje napájení proudem o napětí 3,3 V, které je v konstrukci použito s maximálním proudovým odběrem cca 55mA, se kterým je počítáno. Tento modul obsahuje sériové rozhraní kompatibilní s RS-232 v logice TTL, které umožní přímé propojení GPS modulu s vysílačem bez žádného přizpůsobování.

Tento modul obsahuje dvě sériové linky; jednu pro průběžnou komunikaci tj. nastavení a výstup dat z GPS přijímače, druhou pak pouze jako vstupní pro příjem rozdílové korekce. Tato korekce je poskytována organizací RTCM (The Radio Technical Commission for Maritime Services – nezisková organizace pro námořní radiové služby).

Pro určení režimu funkce budou důležitým parametrem časy potřebné k uzamčení pozice (Time To First Fix – TTFF). Z hlediska úspory energie bude zvolen kompromis mezi časy TTFF a hloubkou uspávání GPS modulu.

Základní technická data:

Čipová sada SiRFstarIII jednočip (GSC 3f)

Hodnota vysílací frekvence L1, 1575.42 MHz (C/A code 1.023 MHz chip rate)

Počet kanálů 20

Citlivost -159 dBm

Přesnost

Pozice 10 metrů, 2D RMS

5 metrů 2D RMS, WAAS korekce

<5metrů(50%)

Rychlost 0.1 metrů/sekundu

Čas synchronizace s časem GPS 1 mikrosekunda

Datum Default WGS-84 (volitelné pro ostatní Datum)

Time to First Fix (TTFF) - čas prvního uzamčení pozice
(při přímém přístupu k obloze a při nepohyblivém stavu)

Reakvizice 0.1 sec., průměr

Snap start 1 sec., průměr

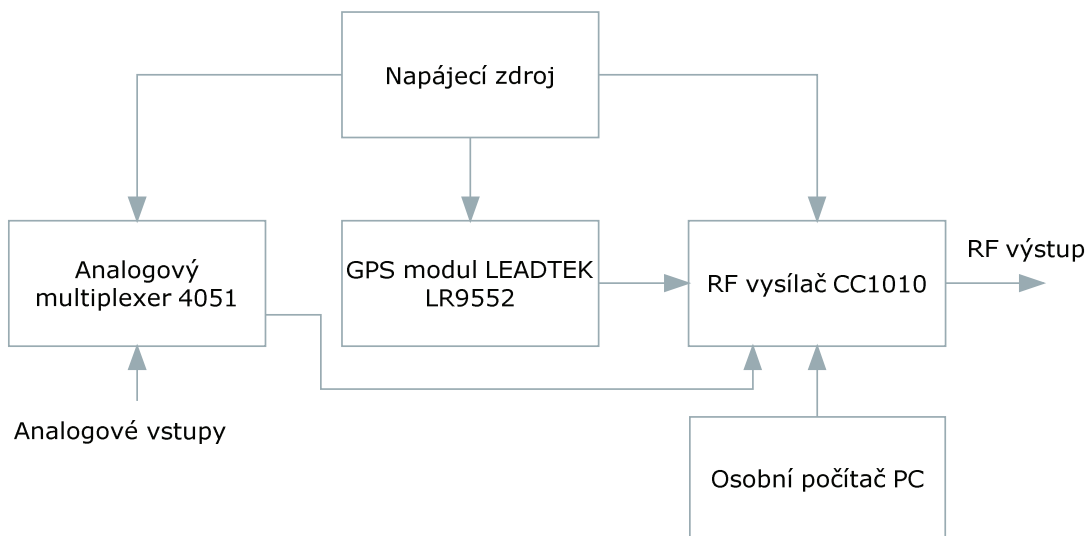
Hot start 1 sec., průměrná typická TTFF

Warm start 38 sec., průměrná typická TTFF

Cold start 42 sec., průměrná typická TTFF

3 NÁVRH ZAPOJENÍ FM VYSÍLAČE

Návrh zapojení vychází ze zapojení doporučeného výrobcem a přebírá z něj vysokofrekvenční část. Ostatní zapojení pro komunikaci s okolím, zejména s GPS modulem, s terminálem a zapojením pro sběr dat, je zcela původní, konstruovaná podle požadavků na funkci zařízení – telemetrického modulu. Blokové schéma vysílače je na Obr. 2. Obvodové schéma zapojení stanice je pak na Obr. 3. Zapojení elektronické části odpovídá požadavkům na funkci, kterou zajišťuje obvod CC1010 prostřednictvím softwarového vybavení.



Obr. 2: Blokové schéma vysílače telemetrických dat

3.1 Vysokofrekvenční část

Jak již bylo uvedeno, zapojení vysokofrekvenční části, včetně zapojení krystalových oscilátorů a napájecích vstupů, je s úpravami převzato z výrobce doporučeného zapojení, resp. tabulky 28 v katalogovém listu obvodu CC1010 [3]. Nebyl důvod měnit hodnoty součástek VF obvodů, protože zapojení pracuje v kmitočtovém pásmu, pro které výrobce tyto hodnoty přímo udává, tj. 433MHz. Použita bude pouze část vysílače, popis přijímače se tedy omezí. Obvody krystalů pro VF část a mikroprocesor 8051 také nebylo nutné upravovat. Z technických důvodů byl kapacitní trimr CT1 nahrazen kondenzátorem s pevnou hodnotou, která se nalezne pomocí měření. V prostředí s proměnlivým klimatem mají pevné kondenzátory podstatně lepší vlastnosti než trimry, proto je tato změna přínosem pro vlastnosti zapojení. Napájecí obvody a blokové kondenzátory jsou ponechány beze změn.

K přizpůsobení RF vstupu přijímače slouží LC článek C31 / L32, kde cívka L32 je použita pro BIAS portu. C41, L41 a C42 přizpůsobují vysílač k 50-Ohmové zátěži. Obvod CC1010 obsahuje pole přizpůsobovacích kondenzátorů, které se k portu připojují pomocí SFR RX_MATCH, resp. TX_MATCH. Optimální hodnota přizpůsobení se nalezne krokováním TX_MATCH a měřením výstupního výkonu. Cívka VCO L101 je jedinou externí součástí tohoto bloku.

Mezi přizpůsobovací článek a antenní konektor je zapojen přídavný filtr C8, L1, C9, omezující harmonické kmitočty vysílače. Doporučené hodnoty součástek filtru jsou uvedeny v [3], avšak závisí na realizaci a jsou zvoleny pro optimální výsledky. Jako antenní konektor je použit SMD samec typu U.FL, který má výborné elektrické vlastnosti a je miniaturní.

3.2 Digitální část

Digitální část mikrokontroléru je využita podle požadavků na funkci zcela původním způsobem. Pro komunikaci s okolím jsou vyvedeny dvě sériové linky v úrovních TTL. Linka UART0 slouží pro komunikaci s připojeným GPS modulem a druhá linka UART1 slouží ke konfiguraci telemetrického modulu pomocí osobního počítače - terminálu. Na konektor je zároveň vyvedeno spínané napájecí napětí pro GPS modul a permanentní záložní napájení paměti GPS modulu. Toto zapojení je zvoleno z důvodu úspory energie, protože umožňuje vypnout GPS modul v době, kdy není potřeba. Jeho paměť je přitom zálohována a nehrozí tedy ztráta množství energie při dlouhé inicializaci GPS modulu po jeho opětovném zapnutí. Tato úprava zapojení zůstává zachována z důvodu univerzálnosti i přes to, že použitý GPS modul vyžaduje pouze jediné napájecí napětí a záložní baterii již obsahuje – nepřipojuje se tedy externí. Do úsporných režimů se přepíná softwarově pomocí sériové linky.

Na vstup A/D převodníku je připojen osmikanálový multiplexer 74HC4051, umožňující sledování až osmi analogových veličin zároveň. Vstupy jsou vyvedeny přímo na konektor SV3; výstup multiplexoru je pak přiveden na vstup AD0 obvodu CC1010. Ovládání multiplexoru pomocí řídicích vstupů A, B a C, je realizováno pomocí portu P2.4 – P2.2 obvodu CC1010. Vstup Inhibit multiplexoru je nevyužit a je uzemněn. Napájení multiplexeru je ze společného stabilizovaného zdroje a je trvalé. Toto řešení nemá nepříznivý vliv na spotřebu; spotřeba multiplexeru je zanedbatelná. Oproti zapojení výrobce, je vypuštěn zdroj referenčního napětí LM61 pro A/D převodník a filtrační RC článek z důvodu zjednodušení [4]. Proti výrobce doporučenému zapojení jsou vypuštěny rezistory na jednotlivých branách sloužící jako pull_up. Ponechány jsou pouze na programovacích vstupech, kde jsou při programování potřebné.

Taktovací kmitočet a referenční kmitočet pro obvod fázového závěsu je zajištěn krystalovým oscilátorem X1 s hodnotou 14,7456 MHz. Krystalový oscilátor X2 s hodnotou 32,768 kHz je používán obvodem reálného času a je možné nakonfigurovat ho jako zdroj taktovacího kmitočtu pro jádro 8051. Blok RF přijímače/vysílače je mimo jedné externí indukčnosti umístěn na čipu CC1010. Externí cívka určuje pracovní kmitočtový segment. Připojením k čipu CC1010 je spojena s vnitřním VCO oscilátorem.

3.3 Napájecí zdroj

Napájecí zdroj je realizován lineárním stabilizátorem napětí 3,3 V LE33, filtračními kondenzátory a blokovacími kondenzátory umístěnými u příslušných obvodů. Stabilizátor upravuje napájecí napětí z akumulátoru stanice.

Spotřeba obvodu CC1010 za provozu v aktivním módu, se skládá ze spotřeb jednotlivých periférií a jádra. Typická spotřeba jádra na pracovním kmitočtu 14,7456 MHz je 14,8 mA. Spotřeba RF transceiveru v režimu vysílání je při maximálním výkonu 10 dBm typicky 26,6 mA. Významného snížení spotřeby lze dosáhnout dvěma způsoby, případně jejich kombinací. Prvním z nich je přepnutí do některého z úsporných režimů, které obvod nabízí, nebo přepnutí na nižší hodinový kmitočet, 32,768 kHz od obvodu hodin reálného času pomocí SFR X32CON. Způsob přepnutí do úsporných režimů má dvě nevýhody – buď se velmi omezí možnosti funkce obvodu, nebo se nedosáhne většího snížení spotřeby. Přepnutí do použitelného módu Idle sníží spotřebu na asi 12,8 mA, což nemá význam. Pro snížení spotřeby, při zachování funkčnosti, byť s velmi omezeným výkonem, je vhodné přepnout na nízký hodinový kmitočet od hodin reálného času. Při kmitočtu oscilátoru 32,768 kHz klesne spotřeba obvodu v aktivním módu na 1,3 mA, což je zlomek původní hodnoty, přitom může běžet program. Vzhledem k tomu, že připojený GPS přijímač má spotřebu mnohem horší, nemá smysl příliš komplikovat situaci a nejlepší je přepínat oscilátor mezi vysokým a nízkým kmitočtem. Dosáhne se tak dobré spotřeby při dobré funkčnosti.

3.4 Zapojení konektorů vysílače

Konektor SV1 je určen ke komunikaci a napájení GPS modulu. Obsahuje tedy sériovou linku s úrovněmi LVTTTL, stálé a spínané napájecí napětí a zem. Je zapojený dle Tab. 9.

Tab. 9: Zapojení konektoru SV1

Číslo vývodu	Název signálu
1	VCC trvalé
2	VCC spínané
3	GND
4	TXD1 P2.1
5	RXD1 P2.0

Konektor SV2 slouží ke komunikaci s terminálem či osobním počítačem za účelem konfigurace provozních parametrů a obsahuje sériovou linku s úrovněmi LVTTTL. Zapojený je následovně:

Tab. 10: Zapojení konektoru SV2

Číslo vývodu	Název signálu
1	GND
2	TXD0 P3.1
3	RXD0 P3.0

Na konektor SV3 je vyvedeno osm vstupů multiplexeru A/D převodníku, zem a napájecí napětí baterie. Analogové vstupy nejsou uspořádány postupně, ale podle následujícího pořadí.

Tab. 11: Zapojení konektoru SV3

Číslo vývodu	Název signálu
1	Vbat
2	GND
3	In6
4	In4
5	In7
6	In5
7	In3
8	In2
9	In1
10	In0

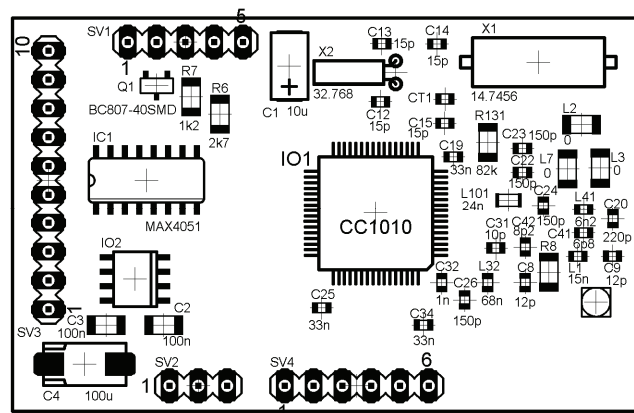
Konektor SV4 slouží k programování obvodu CC1010, má 6 pinů a obsahuje tyto signály:

Tab. 12: Zapojení konektoru SV4

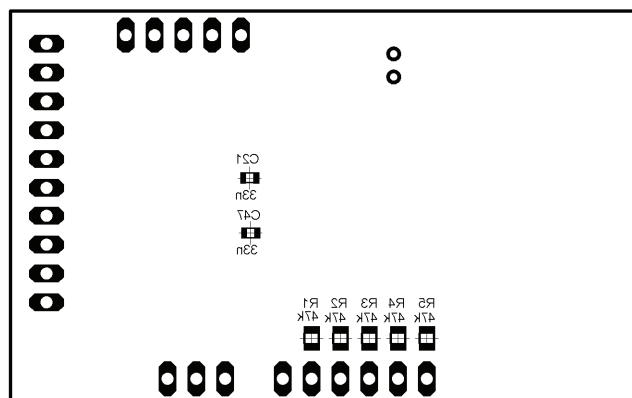
Číslo vývodu	Název signálu
1	GND
2	MISO P0.2
3	MOSI P0.1
4	SCK P0.0
5	/PROG
6	/RESET

3.5 Deska plošných spojů

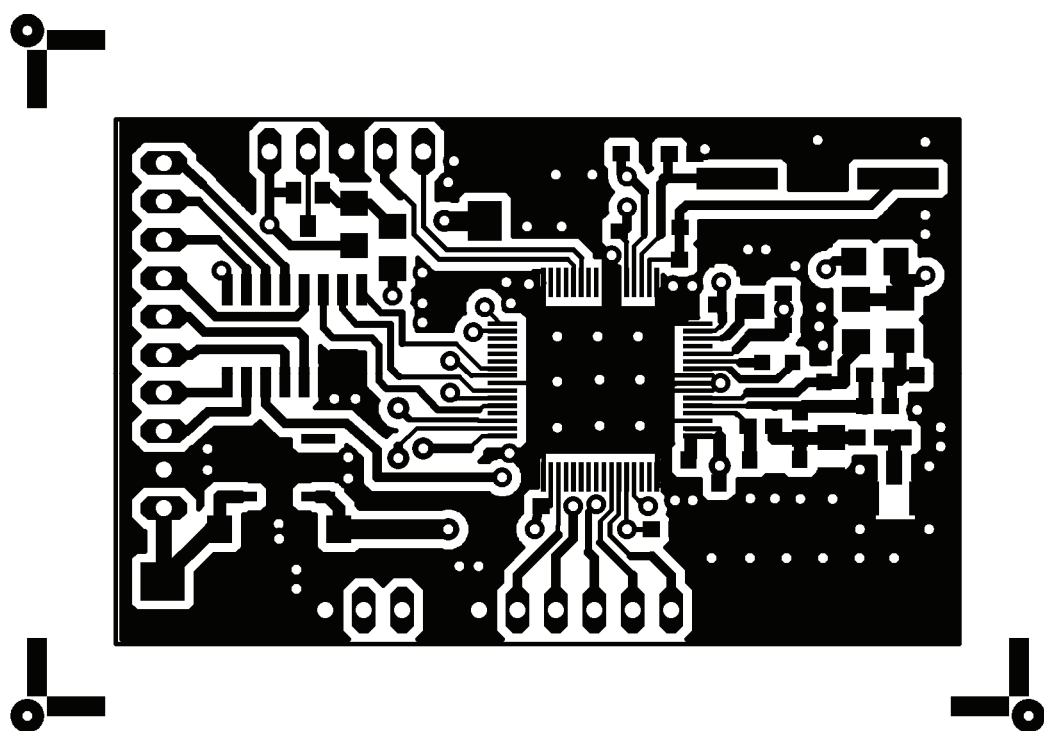
Deska plošných spojů je navržena s prioritou jednostranné montáže SMT a minimálních rozměrů. Obsahuje jednak VF část zakončenou anténním konektorem U.FL, jednak část napájecí a periferní. Na desce jsou přehledně umístěny konektory s různou délkou a klíčem, zamezující nesprávnému propojení, viz. Obr. 4 a 5. Při návrhu byl brán zřetel na logické rozmístění součástí s ohledem na optimální funkci. Napájecí i zemní spoje byly vedeny tak, aby poskytovaly kvalitní propojení a maximálně velkou souvislou zemní plochu na spodní straně desky, zejména pod VF částí, viz Obr. 6 a 7. Dobré propojení zemního spoje podporuje množství prokovených otvorů.



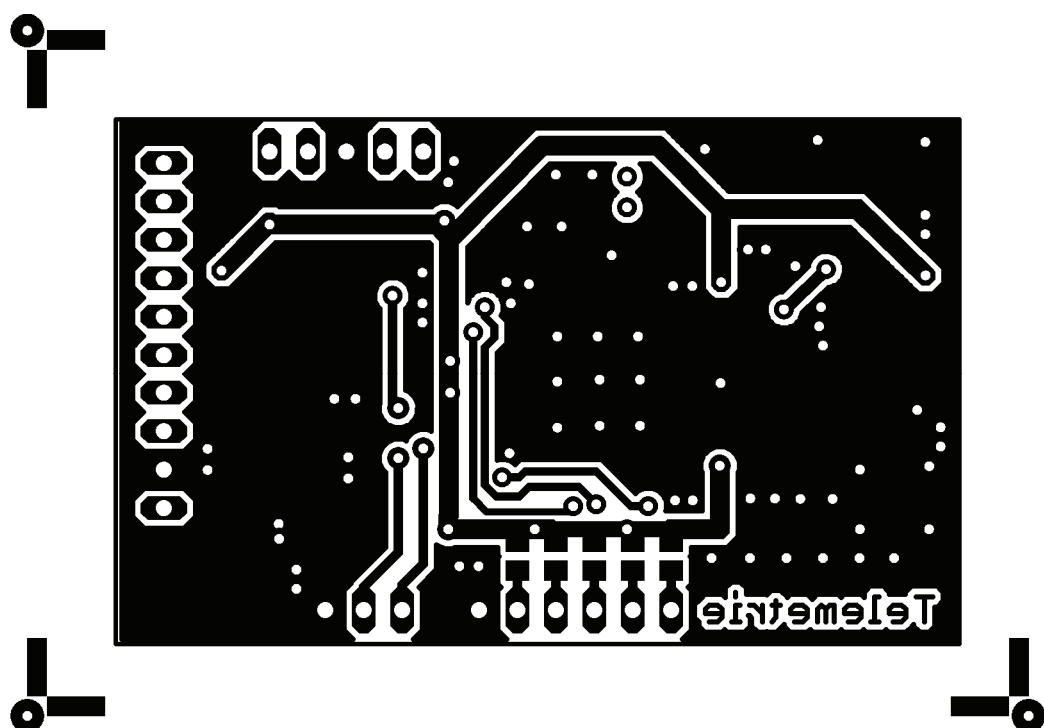
Obr. 4: Osazovací plán strany TOP, M 1,5:1



Obr. 5: Osazovací plán strany BOTTOM, M 1,5:1



Obr. 6: Deska ze strany TOP, M 2:1



Obr. 7: Deska ze strany BOTTOM, M 2:1

4 NÁVRH PŘENOSU DAT

4.1 Obecný popis hardware vysílače pro přenos APRS hlášení

Telemetrický vysílač je radiový vysílač určený pro konkrétní použití. Pracuje v digitálním módu, kdy přenáší digitální data vytvořená mikroprocesorem a upravená tak, aby jimi bylo možno modulovat vysílač. Data určená k vysílání jsou získávána různým způsobem. Jsou buď naměřena např. A/D převodníkem a převedena do digitální podoby, nebo mohou být odečtena přímo jako digitální hodnoty, např. z digitálních vstupů. V neposlední řadě mohou být tato data vyčtena z externě připojeného zařízení, poskytujícího zajímavé informace. Příkladem takového zařízení může být GPS přijímač, který je zpravidla vybaven sériovou linkou s TTL výstupem. Pro všechny činnosti spojené se získáním užitečných dat musí být vysílač patřičně hardwarově vybaven. Konstrukce popsaná v předešlých kapitolách těmto požadavkům odpovídá. K vysílači vybaveném GPS přijímačem je potřebné připojit napájecí zdroj a vhodnou anténu. Vhodným napájecím zdrojem může být např. NiCd baterie o napětí 4,8 V, nebo jiná vhodná baterie dodávající energii na provoz. Kapacita baterie závisí na požadované době provozu a má přímý dopad na hmotnost kompletního vysílače. Tím je také dán požadavek na nosnost meteorologického balónu.

Vhodná anténa pro používané pásmo 435 MHz by měla vyhovět způsobu použití vysílače. Vzhledem k předpokládanému použití konstrukce na meteorologickém balónu lze předpokládat nekontrolovatelný pohyb vysílače, což znamená nutnost použití všesměrové antény. Zároveň je možné uvažovat šíření vln dominantně přímou vlnou, protože se předpokládá pohyb v nízkých vrstvách atmosféry. Je možné použít různé druhy antén, např. spirálové nebo smyčkové, ale vzhledem k váze a jednoduchosti je nejvhodnější např. prutová čtvrtvlnná anténa. Tato anténa poskytuje optimální výsledky [4]. Čtvrtvlnná anténa je monopól tvořený jedním ramenem a protiváhou. Protiváha je jednoduše tvořena samotným vysílačem, případně lze anténu připojit pomocí kabelu a vyrobit složitěji. Monopól je v praxi vyroben z drátu, jehož délku L je možné spočítat podle vztahu (2) z literatury [3].

$$L = 7125 / f, \quad (2)$$

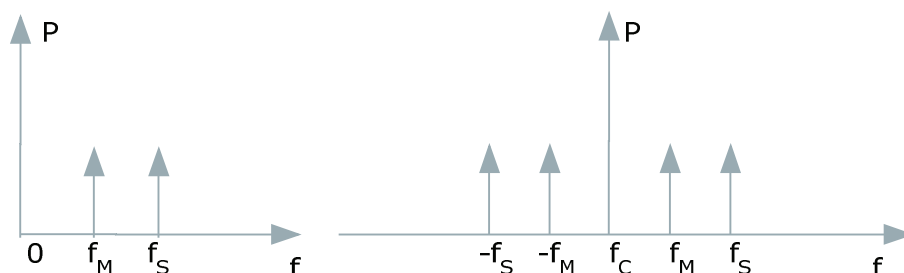
kde L je délka antenního vodiče v [cm] a f je pracovní kmitočet vysílače v [MHz]. Tento vztah již zahrnuje korekce o činitel zkrácení a další vlivy. Pro následující výpočet délky zářiče – ramene monopólu je vybrán pracovní kmitočet 433,500 MHz. Po dosazení do vztahu (2) vychází délka zářiče

$$L = 7125 / 433,500 = 16,4 \text{ cm.}$$

V pásmu 435 MHz, ve kterém má být popisovaný vysílač provozován se používají dva druhy modulací, s nimiž jsou svázány přenosové rychlosti, které se obvykle používají. V síti Paket rádia se používá, jak již bylo popsáno v úvodu, modulace FSK s modulační rychlostí zpravidla 9600 Bd pro použití na vytíženějších částech sítě, nebo modulace AFSK s modulační rychlostí 1200 Bd. Jak již bylo zmíněno v úvodních kapitolách, modulace FSK je použitelná pouze pro provoz v síti, jejíž stanice umožňují tzv. digitální provoz. Mají specificky řešenou detekční část a umožňují přenášet data na

vyšších modulačních rychlostech. Pro použití obyčejných fonických stanic je potřebné použít modulaci, která je kompatibilní s nízkofrekvenční částí běžných stanic. Nemůže proto obsahovat stejnosměrnou složku a její kmitočtové spektrum musí odpovídat fonickému provozu. Takovým podmínkám vyhovuje modulace AFSK.

AFSK modulace pracuje v akustickém pásmu na principu střídání dvou nízkofrekvenčních tónů. Každý tón má význam jednoho symbolu. Použitá varianta pracuje s tóny 1200 Hz pro značku (mark) a 2200 Hz pro mezeru (space), jak je zřejmé z obrázku 8.



Obr. 8: Spektra modulačního signálu a nosné vlny modulované AFSK

Použitý hardware neumí tuto modulaci přímo generovat a je potřebné ji vyřešit softwarově.

4.2 Obecný popis software vysílače pro přenos APRS hlášení

Sběr dat

Software vysílače má za úkol kompletně obsluhovat všechny jeho periferie, zajišťovat sběr dat a jejich zpracování. Musí vytvořit rámec dat podle zvoleného protokolu a tato data musí umět odvysílat radiovým kanálem. K měření analogových veličin je použit A/D převodník vybavený multiplexerem, zvyšujícím počet jeho vstupů. Po převedení analogových veličin na čísla jsou tato data dále zpracovávána. Vzhledem k tomu, že je použit externí multiplexer, je potřebné jej vhodně adresovat, což je zajištěno připojením adresních vstupů k portu obvodu CC1010. Dále je k vysílači připojen GPS modul pomocí sériové linky. V obvodu je proto použit UART, ze kterého jsou v přerušení ukládána přijatá data k pozdějšímu zpracování.

Kódování dat.

Software vysílače zpracovává sebraná data z okolí podle protokolu AX.25, v jehož informačním poli je umístěno hlášení vytvořené podle protokolu APRS, jak již bylo naznačeno v úvodních kapitolách. Kódování dat je možné provádět několika způsoby. Je možné použít pevný, v programové paměti předem nachystaný rámec, a ten pak pouze doplnit o naměřená a jinak získaná uživatelská data. Potom je nutno provést nad tímto souborem dat kontrolní součet a vygenerovat obsah pole FCS. Následně je nutno tento blok dat odvysílat.

Jiný způsob spočívá ve vytvoření všech dat dynamicky s použitím tabulkových údajů, tento způsob poskytuje větší pružnost tvorby hlášení. Použitý způsob kódování je ještě dokonalejší a namísto neměnné sestavy dat umožňuje online nakonfigurovat tvar vysílaného hlášení.

Vysílání dat

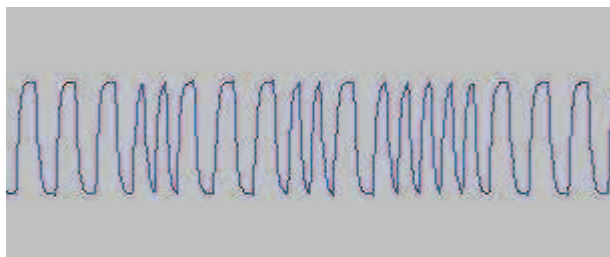
Na tvorbu protokolu úzce navazuje další úloha softwaru vysílače a tou je odvysílání vytvořených dat. První možnost je odvysílat předem vytvořená data z paměti, ať už jsou statická a pouze doplněná o uživatelská data, nebo se jedná o data vytvořená dynamicky. Zásadní je skutečnost, že se v paměti musí vytvořit a obsadit nejméně tak velký prostor, jako je délka kompletní sestavené zprávy. Toto zvyšuje nároky na paměť a při delších hlášení, nebo při sérii delších hlášení mohou být nároky na paměť příliš vysoké. Použitý způsob vysílání dat je přímo provázán s tvorbou protokolu a je její součástí. Data pro protokol jsou tedy připravena pouze na uživatelské úrovni – tedy zpravidla uložena do řetězcových proměnných - a jsou k dispozici potřebné tabulkové hodnoty. Všechna tato data se za běhu bit po bitu zakódují a ihned odvysílají. Tento způsob značně zkracuje dobu přípravy dat a jejich odvysílání, protože všechny potřebné operace se provádí souběžně. Zároveň se šetří paměť, protože není nutné samostatně ukládat celou zakódovanou zprávu.

Softwarová modulace AFSK

Před rozpracováním níže popsaného postupu byla použita jednoduchá metoda, která měla za úkol ověřit, zda je vůbec možné generovat v potřebné rychlosti přerušení, kdy je třeba relativně složitě přistupovat do paměti pro potřebná data, provádět některé kontroly a zapisovat do SFR registru. Použitá zkušební metoda používá dva tóny, které jsou soudělné, 1200 Hz pro značku a 2400 Hz pro mezeru. Tyto byly generovány s úspěchem. Metoda pracuje s jediným časovačem, který běží stále stejně rychle. Frekvence přerušování je dvojnásobkem nejvyššího generovaného kmitočtu. V přerušení se podle aktuálního vysílaného symbolu střídá hodnota modulace. Vzhledem k tomu, že kmitočty tónů i symbolové rychlosti jsou soudělné, stačí počkat na jedno přerušení pro mezeru, nebo na dvě přerušení pro značku. V každém z přerušení se pouze invertuje modulační proměnná, resp. přímo registr. Nevýhodou této metody je, že kmitočty tónů 1200 Hz a 2400 Hz nejsou běžně používané, nicméně tato varianta existuje.

AFSK signál lze generovat i jiným způsobem, a to s použitím dvou časovačů. Je použit první časovač generující takt 1200 Bd. Ovšem druhý časovač generující tóny negeneruje fázová kvanta, nýbrž rovnou půl periody tónu, kdy se v jeho přerušení obrací fáze tónu. Zde je ovšem problém s přesným dodržení fáze tónu, který lze rychle určit dle aktuální hodnoty časovače tónu odečtené v okamžiku přerušení taktovacího časovače. Poté je provedeno přímé nastavení počáteční fáze nového tónu vyčtením předem vypočtené hodnoty z tabulky. Tato hodnota odpovídá poměrné fázi tónu předešlého.

Metoda generování AFSK použitá v této konstrukci je založena na principu rozdělení periody modulačního signálu do jistého počtu fázových dílů. Používá dva časovače. Jeden časovač slouží k odměřování fázových dílů modulačního signálu a druhý určuje dobu trvání symbolu. Fázových dílů je tolik, aby fázová chyba při změně modulačního tónu byla co nejmenší. V praxi se ukázalo, že již rozdělení do 16 dílů umožňuje generovat dostatečně kvalitní modulační signál.

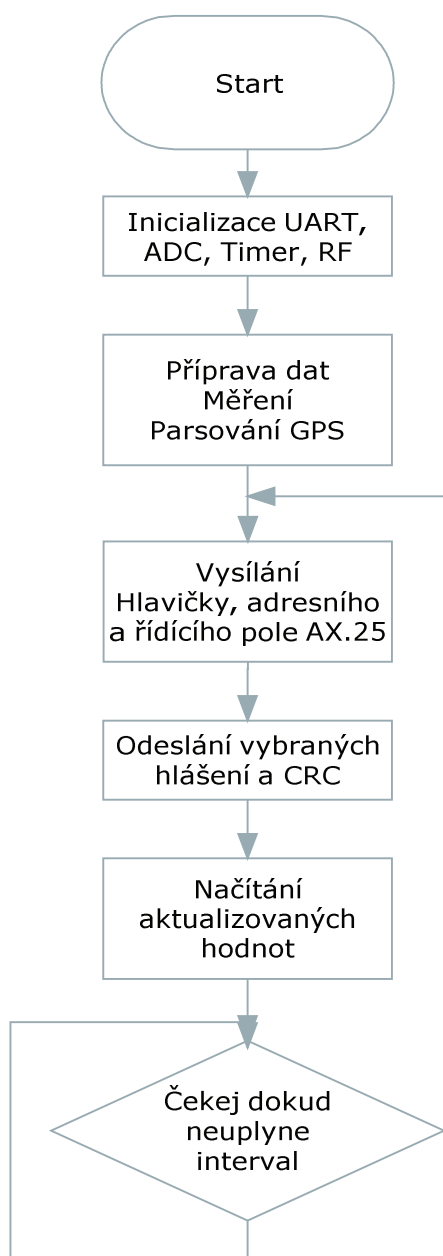


Obr. 9: Ukázka AFSK signálu s tóny 1200 Hz a 2200 Hz

Tato metoda má ještě jednu výhodu v konstrukcích, kde není přímo integrovaný digitální vysílač a kde se používá externě připojená běžná radiostanice. Je totiž možné i s poměrně levným mikrokontrolerem bez D/A převodníku poměrně dobře aproximovat sinusový průběh hardwarově, pomocí odporové sítě připojené k portu [7]. Signál vytvořený tímto způsobem může být použitý pro modulaci radiostanice přes mikrofonní vstup. Ovšem toto řešení je vhodné pouze pro stacionární vysílače, kde nevadí podstatně větší rozměry a hmotnost.

5 NÁVRH SOFTWAREVÉHO VYBAVENÍ

Kompletní softwarové vybavení slouží k zajištění celkové funkce vysílače, tedy zejména k zakódování naměřených a přijatých hodnot a jejich odvysílání, jak vyplývá z vývojového diagramu na Obr. 10. Z periférií jsou použity zejména: obě sériové linky, A/D převodník, výstupní port a vysílač, k němuž se přistupuje přes speciální funkční registry podobně jako k ostatním perifériím. Popis registrů je dostupný v katalogovém listu výrobce obvodu [3].



Obr. 10: Vývojový diagram hlavní programové funkce

5.1 Obsluha periférií, inicializační funkce

5.1.1 Sériové linky

Pro inicializaci sériových linek vysílače je třeba vyslovit požadavky nastavení. Sériová linka 0 na portu 3 je použita pro spojení s počítačem PC za účelem nastavování parametrů vysílače; linka 1 na portu 2 pak pro spojení s GPS modulem, ze kterého přijímá NMEA data. Je vhodné použít standardní, běžně používaná nastavení, což pro sériovou linku 0 znamená rychlost např. 9600 nebo 4800 Bd, formát 8N1. Pro sériovou linku 1 je pak rychlost 4800Bd, a formát 8N1, které jsou dány požadavkem ze strany GPS modulu.

Funkce inicializující obě použité sériové linky má za úkol inicializovat proměnné potřebné ke správné funkci dalších částí SW, především Message Handleru, který přijaté znaky z portů dále interpretuje. Provádí se inicializace bufferu a jeho ukazatelů na začátek a konec bufferu. Dále se obsadí porty obvodu tak, aby bylo možné připojit vnitřní rozhraní, nastaví se sfr registry SCON, PCON pro režim provozu a režim práce časovače 1 pro nastavení přenosové rychlosti. Sériové kanály nemají vlastní generátor hodin, a tak používají časovač 1 pro tuto funkci. Nakonec se povolí přerušení od sériového kanálu. Tyto akce zajišťují příslušná makra.

5.1.2 A/D převodník

Nastavení A/D převodníku odpovídá jeho hardwarovému zapojení. Je využit jeho první vstup AIN0, který je přiveden na analogový multiplexer. Volba jednoho z osmi vstupů se tudíž řídí nastavením tohoto multiplexoru připojeného na port 2 obvodu CC1010. Inicializace tohoto HW se provádí ve funkci `ADCInit`, která volá makra pro nastavení přerušení od A/D převodníku. Toto přerušení je sdílené s jinou periférií, musí se proto nastavit v několika krocích. Dále se nastavují sfr `ADCON` a `ADCON2` s požadovanými parametry. Ty definují režim A/D převodníku. Převodník je dynamicky zapínán a vypínán podle potřeby provádět měření analogových hodnot. Pokud je zapnut, běží volně s přerušením. Jako reference je zvoleno poměrně přesné a stabilní napájecí napětí VDD. Předdělička je nastavena tak, aby rychlost převodu byla v povolených mezích dle katalogového listu [3]. S ohledem na použití externího multiplexoru se před voláním této inicializace definují příslušné piny portu 2 jako výstupní. Toto zajišťují tři makra `PORTDIRBIT(2, N, POUT)`; , kde N je číslo pinu nastavovaného jako výstupní a nabývá hodnot 2, 3 a 4. Tím se zajišťuje možnost přepínání kanálů převodníku.

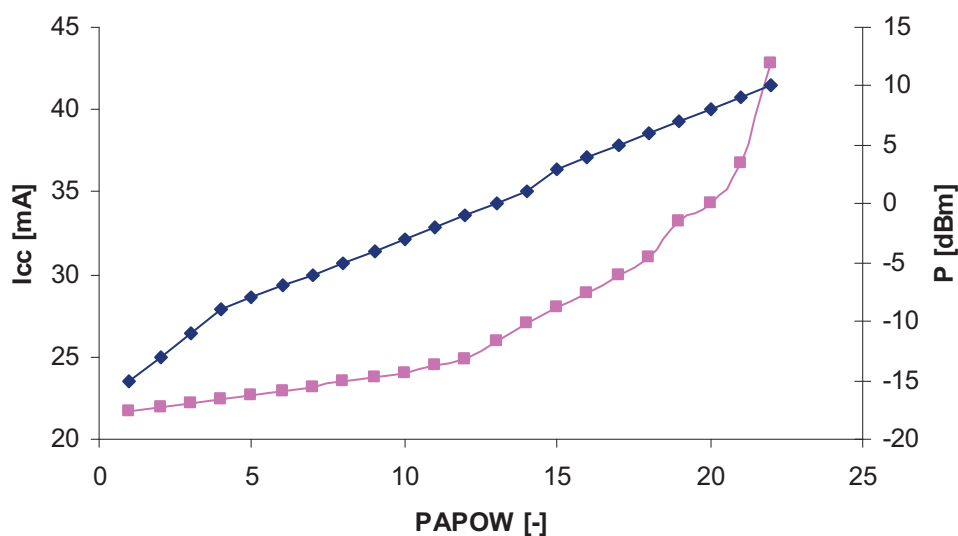
5.1.3 Časovače

Použité časovače se nastavují funkcí `TimerInit`, prostřednictvím zápisu potřebných hodnot do příslušných SFR registrů. Časovač `timer0` je použit ke generování tónu a k definování čekacích smyček použitých v programu. Časovač `timer1` pak ke generování přenosové rychlosti pro generování modulace AFSK v době jejího vysílání. Mimo tuto dobu slouží jako generátor hodin pro sériové linky obvodu, což je dané konstrukcí obvodu CC1010, který v tomto zachovává funkci původního jádra 8051. Tyto časovače jsou nastaveny následovně. Časovač `timer0` je nastaven do 8-mi bitového módu s automatickou předvolbou, v registru `TCON` s předděličkou o hodnotě 12

zvolenou zapsáním slova v registru CKCON. Nakonec je nastavené povolení přerušení časovače timer0 IE a vysoká priorita jeho přerušení. Obdobně je nastaven i časoveč timer1.

5.1.4 RF vysílač

Samostatnou inicializační funkci si zasloužilo nastavení vlastního vysílače, které probíhá v několika krocích. Nejprve je potřeba definovat pole RF_SETTINGS, které obsahuje kompletní sestavu konfiguračních dat. Ve třech bajtech je uloženo nastavení modemu a dvou alternativních frekvencí FREQ A a FREQ B, ve dvou bajtech je uloženo nastavení FSEP, neboli rozdíl dvou kmitočtů rovný dvojnásobku požadovaného frekvenčního zdvihu. Dále následují jednobajtové parametry PLL_RX a PLL_TX specifikující předděličku, CURRENT_RX a CURRENT_TX nastavující vhodné napájecí proudy VF části, dále parametry FREQEND, PA_POW, MATCH a PRESCALER, které jsou s výjimkou výstupního výkonu PA_POW převzaty z katalogových údajů. Závislost výkonu a spotřeby obvodu na nastavení je patrné z obrázku 11. Další z potřebných kroků je vytvoření pole proměnných RF_CALDATA, které používá následně volaná funkce halRFCalib(&RF_SETTINGS, &RF_CALDATA);. Po této proceduře je možné vysílač zapnout příkazem halRFSetRXTxOff(RF_TX, &RF_SETTINGS, &RF_CALDATA); a v návratové hodnotě funkce zjistit zachycení či nezachycení fázového závěsu.



Obr. 11: Závislost spotřeby obvodu CC1010 a výstupního výkonu na nastavení

Problematika provozu v proměnlivých klimatických podmínkách vyžaduje opakovanou kalibraci VCO, protože jeho určujícími součástkami jsou teplotně (cívka) nebo napěťově (ladicí obvod) závislé obvody. Kalibrace VCO se má provést vždy, když se napájecí napětí změní o více než 0,5 V, nebo když se teplota změní o více než 40°C. Kalibrace je provedena při každém zapnutí vysílače za provozu automaticky.

5.1.5 Obsluhy přerušení

Sériové linky

Obsluha přerušení pro první sériovou linku, která je určena ke komunikaci s PC, je krátká, ukládá se v ní přijatý znak do fronty `inbuf` k následnému zpracování. Ve funkci je nejprve zjištěno, zda přišlo přerušení od přijímače či do vysílače. Pokud přišlo od přijímače, uloží se obsah přijímacího bufferu do fronty `inbuf`. Poté se nuluje příznak přerušení pomocí `makra`. Obsluha přerušení při vysílání pak pracuje s výstupním bufferem `outbuf`, ze kterého odvysílá všechny nové znaky, dokud není splněna podmínka, že již není k dispozici žádný nový znak.

Obsluha přerušení druhé sériové linky je v zásadě podobná, část pro příjem znaku má vlastní buffer, pouze je určena pro jiný účel – je určena pouze pro příjem dat z GPS přijímače, nemusí tedy obsahovat obsluhu přerušení při vysílání.

Časovače

Časovač `timer0` slouží ke generování tónu pro AFSK modulaci. Princip funkce spočívá v rozdělení jedné periody tónu na množství dílčích kroků, ve kterých se aktualizuje hodnota časovače podle aktuálního požadavku na tón. A zároveň se aktualizuje modulační signál pro vysílač. Je definováno pole hodnot `sine[]` a index `sine_index`, který stanoví, v jaké části periody se nyní program nachází. Pokud je splněna podmínka, že stavová proměnná signalizuje stav vysílání, pak se v každém kroku této periody posunuje index tohoto pole, čímž se zajišťuje nastavení správné hodnoty periody jako modulačního signálu. Tento modulační signál se zapisuje do vysílacího bufferu `RFBUF RF` vysílače jako modulační hodnota. Tóny pro symboly MARK nebo SPACE jsou určeny pevně hodnotami 1200 Hz resp. 2200 Hz. Omezená přesnost nastavení dovolila nastavit tóny na skutečnou hodnotu 1195 Hz, resp. 2194 Hz. Pokud se nevysílá, je nastaven nejdelší čas časovače, který je poté využíván jako zdroj zpoždění pro funkci `Delay`, využitou dále v programu.

Obsluha přerušení časovače `Timer1` slouží během vysílání pouze k určení symbolové doby; udává tedy takt, ve kterém se vysílají modulační symboly MARK nebo SPACE podle aktuálně připravených dat k vysílání. Hodnota časovače je daná pevně a zaručuje modulační rychlost 1200Bd pro AFSK modulaci. Tuto rychlost se podařilo nastavit pomocí měření skutečné rychlosti na hodnotu 1181Hz, což představuje chybu -1.6% a plně postačuje ke správné detekci v přijímači, jak bylo ověřeno zkouškou.

A/D převodník

Obsluha přerušení od A/D převodníku sdílí stejný přerušovací kanál s jinou periferií, obsluha tedy testuje dva příznaky a oba nuluje. V této obsluze se periodicky měří všech 8 vstupů A/D převodníku a naměřené hodnoty se ukládají do pole hodnot `adc[]` o velikosti 8 bajtů. V každém průchodu přerušením se o jedničku zvyšuje pomocná hodnota `admux_temp`, která po úpravě posunutím slouží k přímé adresaci externího multiplexeru.

5.2 Příprava dat

Před započítím tvorby zprávy je třeba připravit proměnné a řetězce pro jednotlivé funkce programu.

K Inicializaci některých proměnných slouží funkce `MsgInit`. Dočasné doplňkové řetězcové proměnné pro hlášení polohy jsou zde naplněny potřebnými znaky. Jedná se o proměnné k hlášení rychlosti, azimutu a nadmořské výšky. Dále jsou zde inicializovány stavové a řídicí proměnné pro vysílání jednotlivých zpráv a předvyplněna adresní pole na korektní výchozí hodnotu.

5.2.1 Měření analogových hodnot

Konstrukce umožňuje připojit až 8 vstupních napětí v rozsahu 0 – VCC, tj. 3,3 V. V každém pracovním cyklu vysílače dojde ke sběru hodnot z A/D převodníku, který je vhodně nakonfigurován. Vzhledem k tomu, že použitý MCU má jen tři analogové vstupy, byl vybrán jeden, který je multiplexován analogovým multiplexerem 74HC4051. Tento se adresuje pomocí tří linek P2.2 – P2.4.

Funkce měření analogových hodnot je realizovaná v přerušení od A/D převodníku, jak je popsáno výše. Hodnotu naměřenou A/D převodníkem lze získat pomocí funkce `ADCGet` volanou s parametrem čísla kanálu, která jako návratovou hodnotu vrací hodnotu odpovídající pozici v pracovním poli naměřených hodnot v přerušení.

Funkce `Serial_Processes` je volaná vždy, když program čeká na nové vysílání zprávy. V tomto mezikase se za pomoci této funkce zpracovávají fronty přijatých dat z GPS přijímače nebo z počítače PC. Funkce `Serial_Processes` kontroluje, jestli přišel nový znak do fronty `inbuf`, resp. `inbuf2` pro obě sériové linky. Pokud ano, pak tento přijatý znak předá funkci `MsgHandler`, která jej již zpracuje. Pokud není ve frontách žádný nezpracovaný znak, funkce končí.

5.2.2 NMEA parser

Funkce třídění GPS dat filtruje požadované věty NMEA protokolu a z těchto ukládá potřebné informace. Funkce je psaná jako stavový automat, který cyklicky aktualizuje data požadovaného typu. Typy žádaných dat lze pružně měnit díky blokové struktuře kódu. Konkrétní data požadovaná pro vysílání jsou zejména čas UTC, zeměpisná poloha, nadmořská výška a parametry rychlost pohybu a kurz (azimut). Vyhodnocuje se také počet používaných (ne počet viditelných) družic.

NMEA parser má za úkol uložit do globálně definovaných řetězcových polí adresovaných globálně definovanými ukazateli jednotlivá slova příslušné NMEA věty. Postup činnosti je v zásadě ten, že se po zavolání parseru s parametrem `NULL` inicializují lokální proměnné a konstanty. Konstanta `commas` určuje počet čárek ve větě, což je spolehlivý způsob, jak určit, kterou část věty právě program zpracovává. Pouze čárky jsou totiž jako oddělovače určené přímo specifikací NMEA [12]. Proměnná `sentence_type` určuje typ věty; nulová hodnota značí zatím nezjištěnou větu.

Každá z částí věty, tedy každé slovo může mít definovány různé proměnné závislé na tom, zda se jedná o řetězec, či jen o jediný znak. Jediný znak stačí uložit do jediné proměnné typu `char`, kdežto řetězec vyžaduje vytvořit pole vhodné délky a ukazatel do

něj. Pro každé slovo je použita lokální proměnná `index` určující vzdálenost v řetězci.

Parser kontroluje výskyt důležitých znaků, jako je \$ značící začátek NMEA věty, nebo čárky v ní obsažené. Znakem NULL se inicializuje. Pokud přijde znak \$, je zřejmé, že přišel začátek nové věty neznámého typu. Nastaví se tedy proměnná `commas` na nulu a typ zprávy na FALSE. Tím se připraví rozhodnutí o typu zprávy. Pokud přijde znak čárka, inkrementuje se proměnná `commas`, čímž se dává najevo, že věta postoupila do dalšího slova.

Když se parser pohybuje v hlavičce věty, což značí nulový počet čárek, rozhodne podle výskytu znaků obsažených v hlavičce o typu věty a nastaví stavovou proměnnou na tento typ, pokud je žádaný.

Hlavní částí parseru jsou určeny k uložení žádaných slov do příslušných temp proměnných, což je zajištěno jednoduchým způsobem. Otestuje se, zda je platný daný typ NMEA věty a podle počtu čárek v ní se v konstrukci `switch - case` určí cílová řetězcová proměnná. Do té se uloží přijatý znak na pozici danou lokálním indexem, který se ihned inkrementuje. Tímto způsobem se naplní všechny žádané dočasné řetězcové proměnné. Tyto proměnné, dříve inicializované funkcí `MsgInit` se po tomto naplnění čerstvými daty mohou použít funkcí `MsgPrepare`, která připraví aktuální soubor dat určený k okamžitému odvysílání.

Řetězce pro hlášení polohy APRS hlášením `Position` se připravují funkcí `MsgPrepare`, která má za úkol nakopírovat z dočasných proměnných aktualizované texty do řetězcových proměnných určených pro přímé odeslání. Tato funkce pomocí sady cyklů `for` překopíruje jednu řetězcovou proměnnou po druhé, přičemž pevně stanoví délku a tvar jednotlivých slov a každé zakončí znakem NULL. Po této přípravě je možné přistoupit k odvysílání žádaných hlášení.

5.3 Zpracování dat – aplikace APRS a AX.25 protokolu

Hlavní činností dále popsaných částí programu je výběr ze žádaných veličin, vstupů a jejich formátování podle protokolu a následné kódování dat pro vysílání a jejich odvysílání.

Základní princip tvorby datového rámce spočívá v získání potřebných dat, která budou vysílána, a jejich doplnění potřebnými daty tvořícími rámec. Formát rámce protokolu AX.25 byl uveden výše, stejně jako formát jednotlivých rámců protokolu APRS, který je přenášen v informačním poli UI rámce protokolu AX.25. UI rámec je nečíslovaný informační rámec, používaný pro jednosměrný přenos uživatelských dat. Z programátorského hlediska existují dva možné přístupy, jak připravit data k vysílání. První možnost, která předpokládá velký výpočetní výkon a relativně nízkou přenosovou rychlost, počítá s tím, že je možné data sbírat až během vysílání. Předpokládá to dostupnost těchto dat v každou chvíli, kdy může nastat požadavek na vyslání datového rámce. Tento princip je možné použít tehdy, když jsou všechna potřebná data k dispozici, což je v tomto případě beze zbytku splněno, jelikož jsou všechna potřebná data připravena předem v čekací smyčce programu mezi intervaly vysílání. Druhá možnost nejprve sebere všechna data potřebná ke složení rámce, tato potřebným způsobem upraví do patřičného formátu a teprve potom takto připravený rámec odvysílá

bajt po bajtu, resp. bit po bitu. Lze říci, že tento přístup je náročnější na paměť, protože se do ní vždycky uloží nejméně jednou celý rámec, který je třeba odvysílat. Pro větší délky rámců nebo jejich větší počet to může být omezujícím faktorem, navíc složitost softwarového vybavení se tímto postupem zvyšuje a znepráhledňuje zejména část kódování dat.

Data potřebná k sestavení zprávy se skládají z povinných částí daných specifikací protokolu AX.25, jenž poskytuje prostor pro přenesení hlášení dle protokolu APRS a samotného užitečného obsahu. Sestavení hlášení dle protokolu APRS se řídí jeho specifikací. V tomto případě je využito hlášení polohy a telemetrické hlášení, včetně doplňkových informací, jako jsou názvy, jednotky a koeficienty přenášených měřených hodnot. Podrobnosti k protokolům AX.25 a APRS lze nalézt v úvodní části práce nebo přesněji a podrobněji v literatuře [1] a [2].

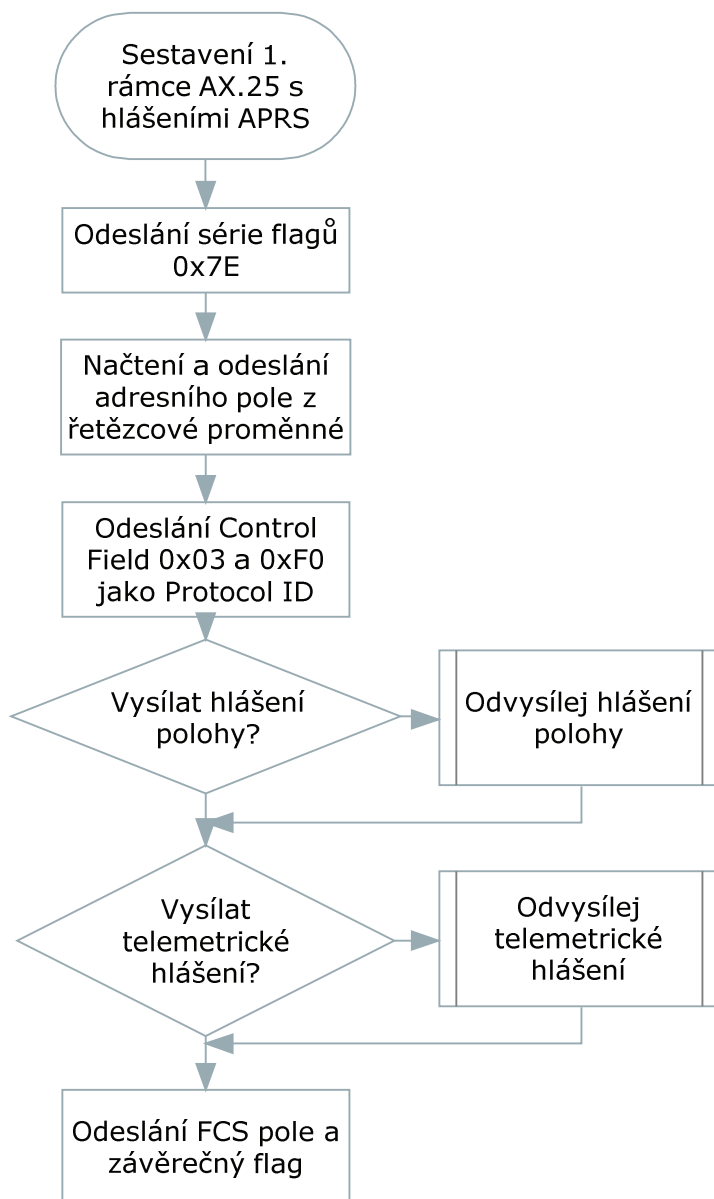
Jednotlivé části zprávy je třeba patřičně zakódovat, přičemž platí, že každá z částí se zpracovává odlišným způsobem. Všechna uživatelská data jsou přenášena jako ASCII; pouze synchronizační pole flag, kontrolní součet CRC v poli FCS a celé adresní pole se přenášení binárně.

Z hlediska zpracování dat lze určit některé operace, které budou prováděny pouze na některých datech, nebo naopak na celém rámci. Podle toho je uspořádána tvorba protokolu AX.25 do několika fází, které se svým způsobem vzájemně prolínají. Operace, které se provádějí pouze nad uživatelskými daty jsou bit stuffing a výpočet CRC. Nad polem FCS obsahujícím CRC se provádí jen bit stuffing. Po opatření protokolu poli flag je nad všemi daty provedeno kódování linkovým kódem NRZI. Tyto operace jsou programem vykonávány souběžně, nad každým bitem. Poslední operace provedená s daty před vysláním je závislá na použité modulaci. V případě použití modulace FSK je třeba provést scrambling z důvodu úpravy spektra vysílaného signálu. V případě použití modulace AFSK je nutné data odvysílat zvláštní funkcí, která softwarově nahrazuje AFSK modulátor a je popsána ve zvláštní kapitole.

5.3.1 Tvorba rámce AX.25

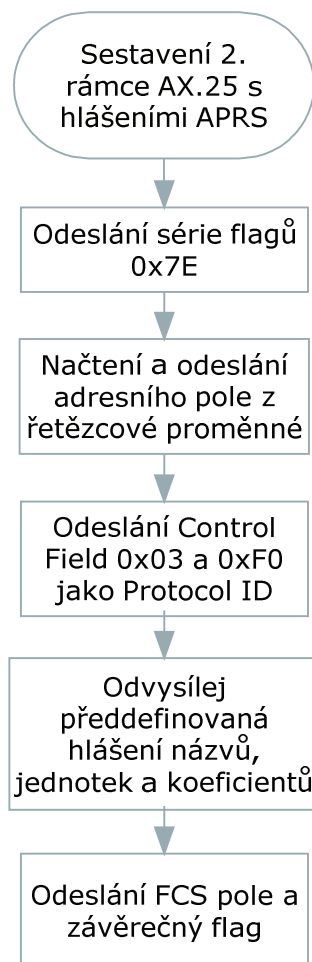
Celá tvorba rámce AX.25 realizovaná programem začíná funkcí `ax25sendHeader`, která inicializuje registr pro výpočet CRC na začátku pole a poté vyšle pole znaků flag s hodnotou 7E hexa. Nad těmi se neprovádí bit-stuffing, ani nejsou zahrnuta do výpočtu CRC. Za tyto data jsou dále umístěny uživatelsky volitelné adresy, tj. adresa cílová (destination) a zdrojové adresy, (source) včetně SSID, které může reprezentovat vhodný symbol. Následují pole s pevně definovaným obsahem určující druh rámce UI (hodnota 0x03) a ID použitého protokolu (0xf0). Od této chvíle se již plní informační pole protokolu AX.25 paketem APRS obsahujícím konkrétní uživatelská data. V případě APRS paketu hlášení polohy je vložen identifikační znak „!“ za nějž jsou uložena v souladu se specifikací upravená textová data přijatá z připojeného GPS přijímače. Vzhledem k tomu, že protokol umožňuje použít různá rozšíření, je za tyto pole s jejich nutnými oddělovači vloženo rozšiřující textové pole kurz/rychlost, ve kterém je uveden aktuální kurz a rychlost pohybu vysílače. Ve zbývajícím prostoru, které zaujímá pole komentáře je opět vložena textová informace o nadmořské výšce získaná jako všechny zde uvedené užitečné informace parsováním NMEA vět z GPS přijímače. Další částí je vložení volitelného komentáře. Následuje již jen vložení FCS pole, tedy CRC kódu cyklického zabezpečení, a uzavření rámce AX.25

polem flag, případně odeslání dalšího paketu APRS v rámci informačního pole protokolu AX.25, které může být až 256 znaků dlouhé [1].



Obr. 12: Vývojový diagram sestavení prvního rámce AX.25 s hlášením polohy a telemetrickým hlášením

Z důvodu, potřeby odvíjet značně větší počet dat, než je kapacita jednoho informačního pole rámce AX.25 je vysílání rozděleno do dvou částí. V první části dle Obr. 12 je odvíjena nejčastěji žádaná kombinace hlášení polohy a telemetrického hlášení podle toho, jak jsou nastaveny jejich časové intervaly. Během jednoho průchodu se může odvíjet libovolné z hlášení, nebo obě naráz. V případě, že je třeba odvíjet i druhou část s názvy, jednotkami a koeficienty telemetrické zprávy, viz. Obr. 13, je vytvořen a odeslán druhý rámec AX.25 s požadovaným obsahem.



Obr. 13: Vývojový diagram sestavení druhého rámce AX.25

5.3.2 Tvorba adresního a řídicího pole

Pravidla pro tvorbu adresních polí jsou přesně stanovená. Adresní pole je v tomto případě sestaveno ze čtyř adres, kde každá je dlouhá 7 bajtů. V prvních šesti je umístěn vlastní volací znak, v posledním bajtu se pak přenáší SSID. Význam SSID je možné najít v příkladech naznačených v úvodní kapitole nebo přímo ve specifikaci protokolu APRS [2]. Vzhledem k tomu, že nejnižší bit je v adresním poli brán jako signalizace konce adresního pole, musí se tento bit uvolnit posunutím původních řetězců o jeden bit doleva. Nejnižší bit posledního bajtu adresního pole se potom nastaví na jedničku. Ihned za adresním polem je umístěno ovládací pole s hodnotou 0x03 a označuje UI rámec. Následuje Protocol ID identifikátor s hodnotou 0xF0 značící, že není implementována třetí vrstva.

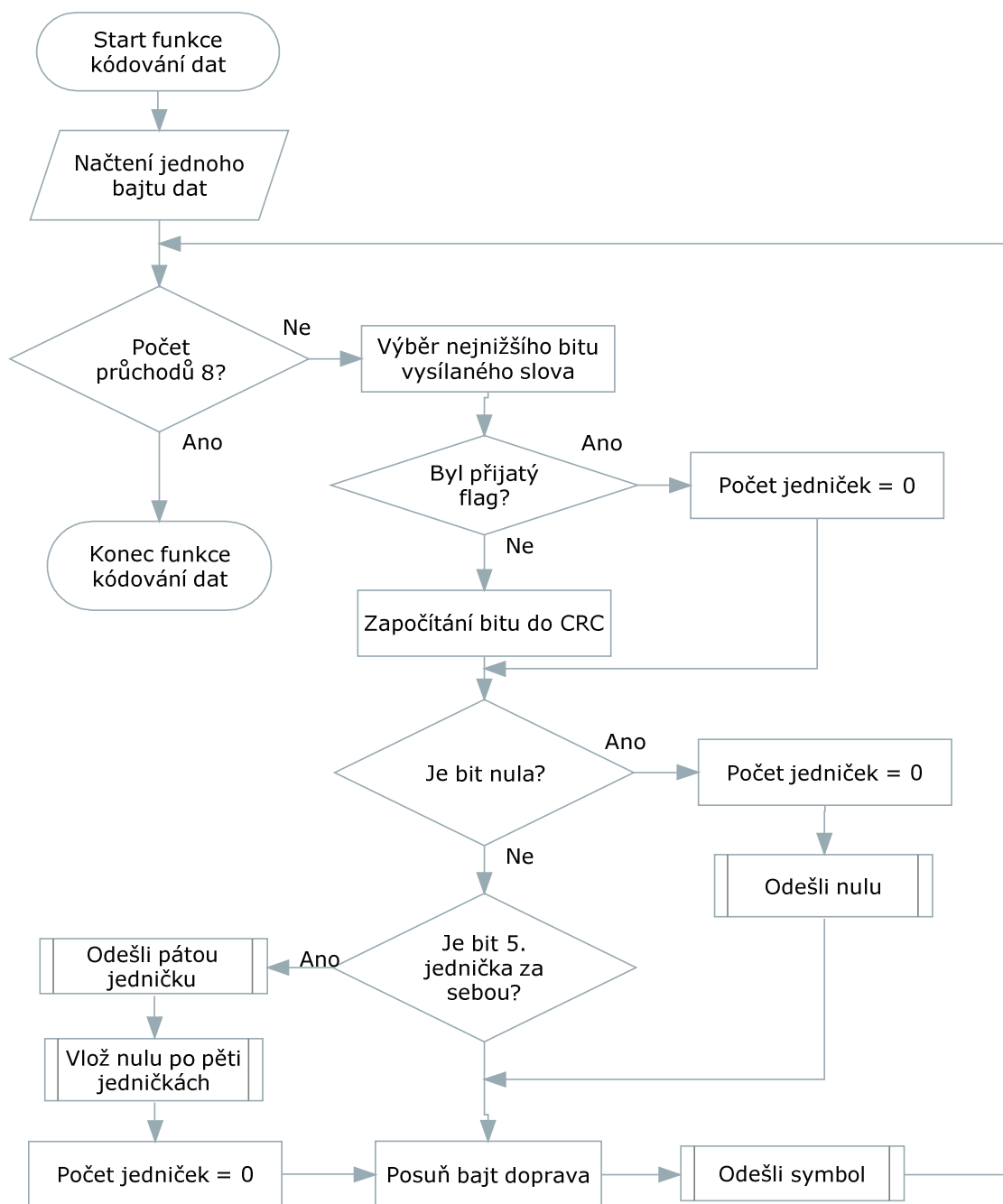
Dále již následuje info pole obsazené hlášením APRS. Rámec AX.25 je následně uzavřen odvysíláním nižšího a vyššího bajtu CRC kódu v poli FCS a závěrečný flag. Tím je kompletní zpráva uzavřena.

5.3.3 Bit stuffing, NRZI kódování

Každý bajt rámce AX.25 je tvořen univerzální funkcí `ax25sendByte`, které je předáván aktuálně připravovaný bajt `dat`, nad kterým má být vždy provedeno potřebné kódování NRZI a jen nad některými daty výpočet CRC či bit – stuffing.

Funkce `ax25sendByte` projde v cyklu `for` daným bajtem vždy osmkrát, pro každý jeho bit. Nejdříve je vymaskován nejnižší bit. Pak je provedena kontrola jestli se nejedná o flag. Pokud ano, je vynulováno počítadlo jedniček pro bit-stuffing, protože flagy se nestuffují. Pokud nejde o flag, je bit započítán do CRC. Nejlépe tuto funkci vystihuje vývojový diagram na Obr. 14.

Následuje kódování NRZI. Aktuální bit je kontrolován, jestli je jeho hodnota nula nebo jedna. Pokud je nula, je vynulováno počítadlo jedniček pro bit-stuffing a mění se modulační tón. Pokud je hodnota jedna, je práce složitější o kontrolu, jestli jde o pátou jedničku za sebou. Pokud ano, je třeba v rámci bit-stuffingu tento bit odvysílat a hned za něj vložit nulu, čehož je dosaženo podržením současného tónu a poté jeho změnou. Zároveň je vynulováno počítadlo jedniček. Pokud se nejedná o pátou jedničku v řadě, není třeba se bit-stuffingem zabývat a zkrátka je po jednu symbolovou dobu ponechán stávající tón. Před dalším průchodem smyčkou je vysílaný bajt posunut doprava o bit, aby se v dalším kroku zpracoval další bit v pořadí.



Obr. 14: Vývojový diagram NRZI kódování, bit-stuffing a vysílání

5.3.4 Výpočet CRC kódu

Obsah FCS polí protokolu AX.25, neboli kontrolní součet CRC se počítá ve funkci `ax25crcBit`. Funkce provádí exkluzivní součet předaného bitu dat s nejnižším bitem pracovního registru. Po té je pracovní registr posunut o pozici doprava. Dále je provedena kontrola, zda je výsledek exkluzivního součtu hodnota jedna. Pokud ano, posune se pracovní registr ještě jednou doprava. Bajty pole FCS jsou jediné, které jsou vysílány v pořadí, ve kterém byly vypočteny.

5.3.5 Skrambler

Skrambler je funkce realizující skramblovací funkci charakterizovanou polynomem, který je v našem případě

$$p(x) = x^0 + x^{12} + x^{17} \quad (2)$$

a používá se z důvodu úpravy spektra radiového signálu za účelem zmenšení výkonové úrovně spektrálních čar zasahujících do sousedních kanálů.

Popsaná a použitá funkce je proudového charakteru a provádí skramblování nad již vytvořenými daty. Vzhledem k tomu, že data uložená v paměti jsou orientována bajtově a skramblování je bitově orientovaná operace, je třeba použít dvě vnořené programové smyčky, kdy jedna má za úkol cyklicky adresovat každý jednotlivý bajt dat od začátku pole až do jeho konce, případně s přesahem nutným ke správné funkci. Druhá smyčka slouží k adresaci každého jednotlivého bitu v aktuálně adresovaném bajtu a provádí samotnou operaci skramblování.

Samotná funkce `void scramble(void)` na začátku deklaruje proměnné `i`, `j` a `bit_ptr` zastávající funkci indexů, resp. ukazatele na bit. Dále deklaruje booleovskou proměnnou `bool send_bit`, do které se ukládá mezivýsledek skramblování. Vlastní funkce, jak již bylo předesláno, obsahuje hlavní cyklus `for` pracující po celé délce pole `ax25_buf` a jeho indexem je „bajtový“ index `i` počínající od nuly. V něm je vložený druhý cyklus `for` s indexem `j`, který je omezen na hodnoty 0 – 7. Oba indexy jsou využity ve skramblovací funkci a vždy se inkrementují o jedničku. Vnořený `for` inicializuje mimo vlastní „bitový“ index `j` také bitový ukazatel `bit_ptr`, který se s každým průběhem posouvá doprava.

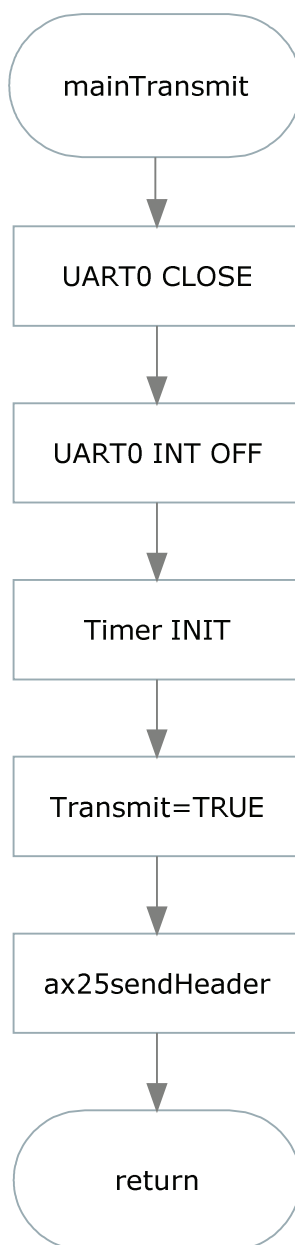
Vlastní skramblování bit po bitu se provádí uložením výsledku exkluzivního součtu hodnot

```
(ax25_buf [i] & bit_ptr),  
((ax25_buf [i + (j + 12)/8]) & (0x80 >> (j + 12)%8)),  
((ax25_buf [i + (j + 17)/8]) & (0x80 >> (j + 17)%8)),
```

do pomocné proměnné `send_bit`. Hodnoty výrazů jsou z hlediska aktuálního skramblovaného bitu bit na nulté pozici, dvanácté a sedmnácté pozici. První z výrazů má hodnotu bitu aktuálně vymaskovaného ukazatelem z aktuálního bajtu. Druhý výraz má hodnotu bitu vymaskovaného bitovým ukazatelem posunutým na zbytek po dělení modulo osmi bitového indexu + 12 v bajtu určeném zbytkem po dělení modulo osmi bitového indexu + 12 přičteném k bajtovému indexu `i`. Podobně je určen i sedmnáctý bit vzhledem k aktuálnímu skramblovanému bitu. Nakonec se pomocná proměnná `send_bit` uloží do aktuální pozice pomocí přičtení jedničky přiřazovacím příkazem `ax25_buf[i] = ax25_buf[i] | bit_ptr`; nebo vynásobením nulou získanou jako bitový doplněk bitového ukazatele `bit_ptr` přiřazovacím příkazem `ax25_buf[i] = ax25_buf[i] & ~bit_ptr`;

5.3.6 Vysílání vytvářených dat

Vysílání probíhá kontinuálně souběžně s tím, jak jsou data počítána. Vysílání započne zavoláním funkce `mainTransmit`, která provede uzavření portů, inicializaci hardwaru a poté přistoupí k odvysílání hlavičky, jak je zřejmé z obrázku 15.

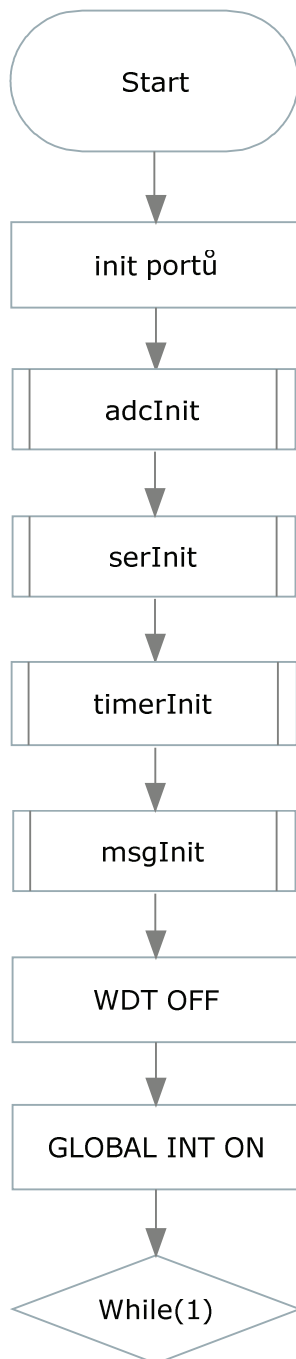


Obr. 15: Vývojový diagram `mainTransmit`

Dále provede pro modulaci FSK volání funkce `scramble()`, která provede skramblování výsledných dat. Poté následuje volání inicializace vysílače `rf_init()`, nastavení do bajtového módu makrem `RF_SET_BYTEMODE()`, vynulování `RFBUF` a zapnutí koncového stupně. Funkce pak skočí do cyklu vysílání. V tomto cyklu se plní `RFBUF` pomocí makra `RF_WAIT_AND_SEND_BYTE(ax25_buf[i])`.

5.3.7 Hlavní programová smyčka

Hlavní funkce `main` provádí počáteční inicializaci proměnných, periférií a poté končí v hlavní smyčce programu, kde již provádí potřebné činnosti, viz. Obr 16.

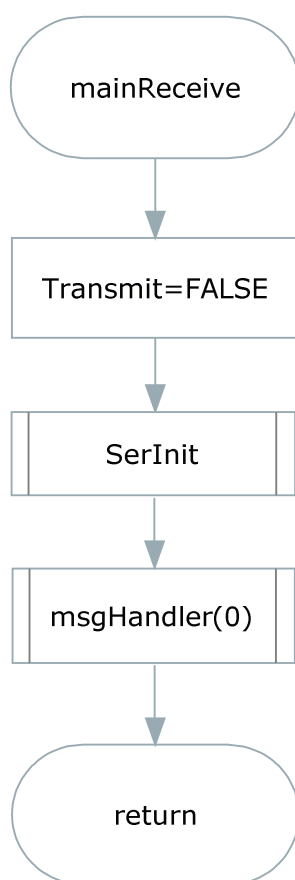


Obr. 16: Vývojový diagram hlavní funkce

Provede se tedy inicializace potřebných proměnných pro běh hlavní smyčky. Jako první akce s hardwarem se nastaví optimální režim časování externí paměti a napájení Programové paměti pomocí maker `MEM_NO_WAIT_STATES()`; a `FLASH_SET_POWER_MODE()`; nastavující optimální výkon a spotřebu, kdy druhé

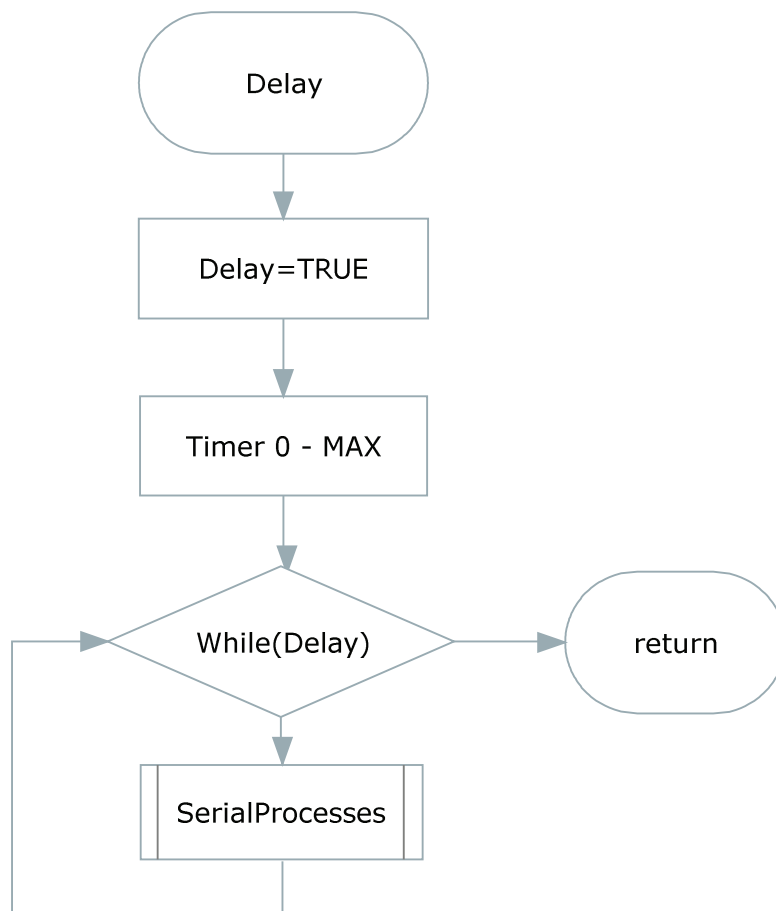
z maker svým parametrem vypíná napájení FLASH paměti mezi čteními. Dále se nastaví porty 2 a 3 podle požadavku na připojení k okolí. Následně se vypne watchdog zapsáním předepsaných bajtů do registru WDT ve smyčce while, aby byla jistota, že se provede. Následuje volání inicializačních funkcí ADCInit, TimerInit a MsgInit, které provedou vlastní inicializaci periférií a inicializaci proměnných pro tvorbu zprávy. Nakonec je provedeno globální povolení přerušení a již se přechází do hlavní smyčky.

Hlavní smyčka plní základní funkce, jako je zavolání přípravy zprávy MsgPrepare, zavolání vyslání zprávy mainTransmit a poté rozhodnutí o vysílání konkrétních hlášení podle toho, jak jsou stanoveny intervaly vysílání jednotlivých hlášení. Po odvysílání požadovaných hlášení se volá funkce mainReceive, která ukončí režim vysílání, provede potřebná nastavení hardwaru a nezbytnou re-inicializaci handlerů pro zpracování vět přijatých ze sériových linek, viz. Obr. 17.



Obr. 17: Vývojový diagram mainReceive

Vysílané zprávy mají různou délku podle toho, jaká hlášení obsahují. Je možné zároveň odvysílat jak hlášení polohy, tak i telemetrické hlášení. Vzhledem k tomu, že se k telemetrickému hlášení pojí další hlášení, jako názvy, jednotky a koeficienty jednotlivých hodnot, je provedeno opatření, kdy se krátká verze zprávy odvysílá najednou v jednom rámci. Dlouhá verze s názvy, jednotkami a koeficienty se potom odvysílá nadvakrát; délka infopole v rámci AX.25 je totiž omezena. Názvy, jednotky a koeficienty se tedy odvysílají samostatně ve svém rámci, který téměř naplní.



Obr. 18: Vývojový diagram funkce Delay

V poslední části hlavní smyčky je umístěna rozhodovací logika, která určuje časování hlášení. Každý typ hlášení má individuálně nastavitelnou periodu vysílání, omezenou shora na 255 vteřin. Dobu, která předchází vysílání, program stráví ve smyčce, která zpracovává přijaté znaky ze sériových linek. Slouží k tomu funkce Delay, která používá časovač 0 k určení časové jednotky, viz. Obr. 18. Po přerušení od časovače se z této funkce vyskočí a lze ji zavolat znovu pro delší zpoždění. Jako základní časová jednotka je použita jedna vteřina. Každou vteřinu se v cyklu `for` provede kontrola, jestli nadešel čas k odvysílání některého z hlášení. Pokud ano, je v dalších kontrolách zjištěno která hlášení se budou vysílat a podle toho jsou nastaveny jejich stavové proměnné – příkazy k odvysílání. Zároveň jsou vynulována počítadla intervalu pro tato hlášení, aby se po průchodu vysílací částí začala inkrementovat znovu. Intervaly, které jsou požadovány, jsou v porovnávacích proměnných.

Funkce této konstrukce je taková, že ve výchozím stavu jsou inicializovány proměnné, například takto: počítadlo `cntTelem` na nulu, příkaz `cmdTelem` na `FALSE` a interval `intTelem` na např. 10s. Každou vteřinu projde program kontrolou a po té inkrementuje všechna počítadla. Až při kontrole zjistí, že kterékoliv počítadlo dosáhlo hodnoty intervalu, tedy při shodě `cntTelem` a `intTelem`, vynuluje počítadlo `cntTelem` a nastaví příkaz `cmdTelem` na `TRUE`. Na závěr se ukončí časovací cyklus.

Poté je provedeno odvysílání daného hlášení a ihned je nastaven příkaz `cmdTelem` na `FALSE`. V tomto cyklu smyčka pokračuje až do vypnutí napájení.

5.4 Konfigurace modulu

Konfiguraci vysílače zajišťuje příslušná část softwaru samotného vysílače a protistrana, kterou je osobní počítač PC. Konfigurace umožňuje uživatelsky nastavit kompletní adresní pole rámce AX.25 o čtyřech adresách s volitelným SSID. Dále umožňuje specifikovat, které typy zpráv se mají vysílat a jak často – je nastavitelný interval mezi jejich vysíláním. Dále je možné vložit kompletní specifikaci telemetrického hlášení, tedy jeho názvy, jednotky a koeficienty. Nakonec je možné doplnit komentář, který se má vysílat v nevyužitém prostoru pole komentář v hlášení polohy. V tomto poli je již umístěn údaj o nadmořské výšce.

Konfigurační věta se do vysílače odešle po sériové lince z příslušné aplikace pro PC, která je přiložena, nebo po předchozím sestavení i z libovolného terminálového programu. Ten je třeba nastavit na parametry přenosu 4800Bd, formát 8N1.

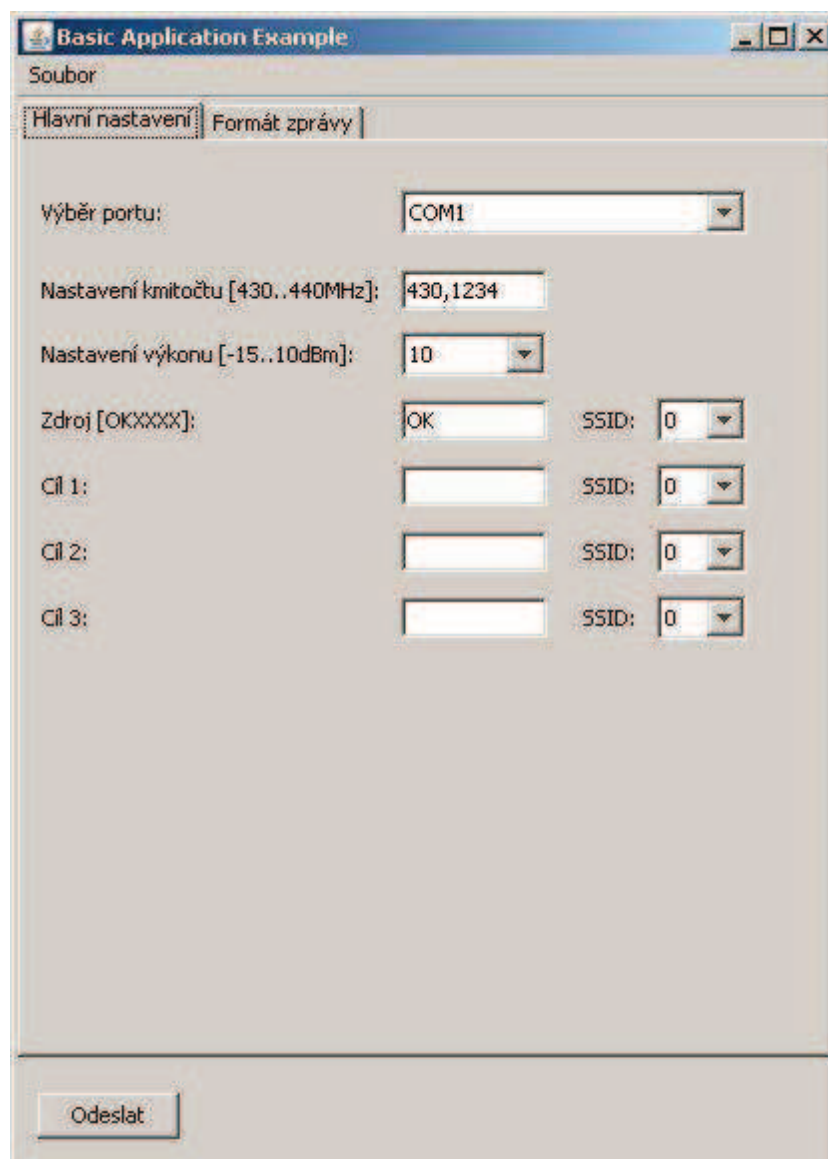
5.4.1 Programové zajištění ze strany vysílače

Funkce `MsgHandler2` zajišťující konfiguraci na straně vysílače je podobná parseru již použitému pro výběr užitečných dat z NMEA věty GPS přijímače, zastává však jinou úlohu. Převážná část konfigurační zprávy jsou ASCII texty, pouze několik hodnot je přeneseno binárně. Začátek věty je uveden sekvencí osmi znaků `$` a konec věty sekvencí osmi znaků `#`. Obsah konfigurační věty je pak seřazen podle pořadí důležitosti a způsobu přenosu. Nejprve jsou přeneseny tři bajty obsahující nastavení vysílací frekvence. Následuje čtvrtý bajt určující požadovanou úroveň výkonu. Dále jsou přeneseny tři bajty určující intervaly vysílání jednotlivých hlášení. Následují řetězce kompletního adresního pole přesně podle specifikace AX.25, textový komentář, hlášení parametrů, hlášení jednotek a hlášení koeficientů, kde každá řetězcová proměnná je ukončená znakem `NULL`.

Funkce `MsgHandler2` používá několik počítadel, která umožňují orientaci v konfigurační větě. Sleduje se počet uvozujících znaků hlavičky a po přijetí platné hlavičky jsou data ukládána do cílových proměnných. První tři bajty jsou uloženy do struktury nastavení vysílače na pozici frekvence, čtvrtý bajt na pozici výkonu vysílače. Dále jsou aktualizovány intervaly vysílání jednotlivých hlášení. Po uložení těchto bajtových hodnot se třídí přijaté řetězce do příslušných cílových řetězců na základě podmínek, které sledují počet přijatých bajtů a počet přijatých ukončovacích znaků `NULL`. Po každém přijatém znaku `NULL` se tento znak zapíše i na poslední pozici cílové řetězcové proměnné, protože je využíván při čtení řetězců během vysílání. Funkce `ax25sendString` je koncipovaná tak, aby ukončila činnost pokud narazí na znak `NULL`. Poté, co se přijmou všechna potřebná nastavení, se konfigurace uzavře.

5.4.2 Programové zajištění ze strany PC

K zajištění konfigurace je možno použít terminálový program a odeslat potřebná data ze souboru, nebo aplikaci SerialManager k tomu účelu vytvořenou. Aplikace SerialManager je grafická a obsahuje dvě záložky, viz. Obr. 19. V první záložce Hlavní nastavení je možnost výběru portu COM, ke kterému je vysílač připojen. V roletovém menu se objeví všechny dostupné porty COM na konkrétním počítači. Dále je zde pole nastavení kmitočtu a roletová nabídka s nastavením výkonu v dBm. Následují čtyři textová pole pro specifikaci cílové adresy a zdrojových adres. Ke každé adrese je možno přiřadit SSID.



Obr. 19: Ukázka aplikace SerialManager

Ve druhé záložce Formát zprávy je možno především specifikovat obsah textového komentáře a textový obsah názvů, jednotek a koeficientů telemetrické zprávy. Je také možno nastavit intervaly vysílání jednotlivých hlášení. Po vyplnění voleb podle požadavků se tlačítkem Odeslat v levém dolním rohu aplikace odešle konfigurační věta

do vysílače

Aplikace vyžaduje standardní počítač PC s alespoň jedním sériovým portem a s knihovnami nakopírovanými do složek podle popisu příloženého u programu.

Aplikace funguje ve své podstatě tak, že v grafickém prostředí je možné specifikovat veškeré potřebné parametry. Jednotlivé grafické prvky představují objekty, které obsahují zadané informace a je možné je získat pro další práci v aplikaci. Po té, co uživatel klikne na tlačítko Odeslat, je inicializována třída z formulářů z uživatelského rozhraní a je spuštěna třída generující konfigurační větu. Tato věta je následně odeslána za pomoci knihovny RxTx přes sériový port. Sériová linka má nastaveny parametry: rychlost 4800 Bd a formát 8N1.

Konfigurační věta má následující formát. Věta začíná osmi znaky \$ pro otevření komunikace. Následují tři bajty binárního 24-bitového slova, které nastavuje frekvenci vysílače. Hodnota tohoto slova je vypočtena odečtením konstanty 8512 od požadované frekvence zarovnané na čtyři desetinná místa. Dále je poslán bajt nastavení výkonu vysílače. Hodnota vychází z tabulky uvedené v [3]. Následují tři bajty, jejichž hodnota odpovídá časovému intervalu vysílání hlášení polohy, telemetrického hlášení a sestavy názvů, jednotek a koeficientů ve vteřinách. Následuje textový komentář dlouhý až 26 znaků ukončený znakem null, kompletní adresní pole ukončené znakem null a řetězec názvů, jednotek a koeficientů podle protokolu APRS vždy ukončené znakem null. Komunikace osobního počítače s vysílačem se po té uzavře vysláním osmi znaků #.

6 POUŽITÉ PROGRAMOVACÍ PROSTŘEDÍ

Kompilátor použitý k práci je SDCC volně šiřitelný pod licencí GNU/GPL, dostupný na ... , ovšem použití knihoven dodaných výrobcem čipu CC1010 bylo třeba upravit, aby byly kompatibilní bez výhrad i s tímto kompilátorem. K tomuto účelu slouží skript napsaný v PERLu od autora Patrika Spiesse[]. Tento skript byl využit, i přesto že SDCC obsahuje soubor compiler.h, který maže rozdíly mezi kompilátory způsobené nestandardními zápisy. Část potíží s nestandardními zápisy byla patrně způsobena vazbou tvůrců maker a knihovních funkcí na kompilátor Keil, který nebyl použit z důvodu omezení volné verze.

Některé funkce byly odladovány na PC s použitím kompilátoru GCC, který je volně šiřitelný pod licencí GNU/GPL. Po přesunu těchto kódů na platformu CC1010 / SDCC byly provedeny jen minimální úpravy z důvodu kompatibility.

Jako editor zdrojových textů byl použit osvědčený program PSPad. K flashování čipu CC1010 byl použit software Chipcon Flash programmer volaný z PSPadu jako externí program.

Programovací kabel použitý k programování byl zapojen identicky s kabelem popsáným v [4].

Jako vývojové prostředí k vytvoření konfigurační aplikace SerialManager bylo použito NetBeans IDE ve verzi 6.9.1. V tomto prostředí byla aplikace vytvořena pomocí integrovaného nástroje pro tvorbu grafického uživatelského rozhraní. Aplikace SerialManager byla naprogramována v jazyce Java. Hlavní výhodou tohoto objektově orientovaného jazyka je nezávislost na platformě. Samotný program je překládán do pseudokódu, který je interpretován pomocí JVM (java virtual machine). Podrobněji např. v [9]

Grafická podoba je vytvořena pomocí knihovny Swing, která poskytuje aplikační rozhraní pro tvorbu grafického uživatelského rozhraní. V průběhu práce bylo čerpáno z dokumentace dostupné v [10] a z tutorialů dostupných v [11].

7 DOSAŽENÉ VÝSLEDKY

Použití vysílače s modulací AFSK

Modulace AFSK není obvodem CC1010 podporována přímo, lze ji však naprogramovat s jistým omezením přesnosti generovaných tónů, modulační rychlosti a přesnosti fáze v přechodech mezi tóny. Pro konkrétní použití byla stanovena modulační rychlost 1200 Bd a modulační tóny 2200 Hz pro mezeru a 1200 Hz pro značku. Tyto tóny jsou generovány pomocí jednoho 8b čítače T/C 0 s vhodnou předděličkou střídavě, podle aktuálních dat a hodinového kmitočtu generovaného dalším časovačem T/C 1. Během vysílání AFSK je zachovávána fáze jednotlivých modulačních tónů tak, aby na ni bylo možno navázat s minimální chybou. Tato dosahuje v nejméně příznivém případě cca 22,5°, což je dáno dělením jedné periody tónu na 16 fázových jednotek. Vždy po načítání 8 fázových jednotek se změní fáze tónu o 180°, což se řeší v přerušení od T/C generujícího tón. Zde je omezení v přesnosti generovaných tónů co do kmitočtu a co do přesnosti návaznosti fáze. Důvodem je protichůdnost potřeb nastavení časovače, který by měl přetekat co možná nejčastěji pro minimální fázový krok a zároveň by měl mít co nejvyšší hodnotu pro maximální přesnost kmitočtu tónu. Popsané řešení je sice kompromisní, ale i přesto jej použitý obvod spolehlivě generuje a popsané přesnosti stačí na to, aby byla zpráva korektně přijata přijímačem a správně dekodována.

```
FRAME TYPE           : Unnumbered Frame
Flag                 : 7e
Destination Address  : APRS    60
Source Address       : OK2CPV  76
Repeater 1 Address   : RELAY   60
Repeater 2 Address   : WIDE2   60
Control Field Type    : UI      Unnumbered Information
P/F                  : 0
p@190853z0000.00
N/00000.00EO0.0/
012043T#103,005,
005,005,005,005,
00000000,005,190
853
FCS                  : dd6b    OK
Flag                 : 7e
```

Vzorek přijatého hlášení zde uvedený ukazuje situaci, kdy byla odvysílána dvě hlášení v jednom rámci AX.25. První částí je hlášení polohy, kde je zřejmý čas záznamu. Druhá část je telemetrické hlášení s analogovými hodnotami.

8 ZÁVĚR

Práce se zabývá přehledem informací o potřebných protokolech pro provoz FM vysílače telemetrických dat. Z přehledu je patrné, že vlastnosti použitého obvodu jsou dostačující pro daný druh provozu a je možné tento vysílač realizovat s možným omezením daným nižším výpočetním výkonem. Dále práce navrhuje zapojení telemetrického modulu s ohledem na požadavky na něj kladené. Uvádí schéma zapojení a návrh desky plošných spojů umožňující funkci v plném rozsahu požadavků. Dále práce uvádí kompletní softwarové vybavení vysílače včetně všech použitých programových nástrojů použitých při vývoji. Při návrhu byl brán ohled na specifické prostředí provozu, a tím i omezené možnosti napájecího zdroje.

Práce měla za úkol vytvořit vysílač APRS telemetrických dat v pásmu 435MHz, tedy v rozsahu 430 – 440MHz. Dosažené pásmo přeladění je přibližně od 430 do 446MHz, což je nejvyšší uspokojivě vyzkoušené pásmo. Vysílač byl testován ve spojení s čítačem do 1GHz, PMR ruční stanicí Motorola XTR446 a profesionální ruční stanicí Motorola P 040 ve spojení s PC a softwarem SkySweeper. V práci se podařilo realizovat modulaci AFSK, s modulační rychlostí 1200 Bd a tóny 1200Hz pro značku a 2200Hz pro mezeru. Vysílač obsahuje osm analogových vstupů, umožňujících připojit libovolná čidla, či jiné obvody s výstupním napětím v rozsahu od 0 V do 3,3 V. Dále je na desce plošných spojů osazen tranzistorový spínač k zapínání GPS přijímače. Vzhledem k časovým relacím mezi uvažovanou periodou vysílání a dobou zachycení GPS přijímače je GPS přijímač připojen na stálé napětí. Vysílač je schopen vysílání nakonfigurovaných APRS hlášení, zejména hlášení polohy včetně volitelného komentáře a telemetrické hlášení s názvy, jednotkami a koeficienty naměřených hodnot.

Ke konfiguraci vysílače je vytvořena aplikace SerialManager pro osobní počítač, umožňující nastavení základních parametrů vysílače a formát a intervaly vysílání uvedených hlášení.

LITERATURA

- [1] FREJLACH, K. *Paket rádio*. České Budějovice: PVT – reprografická sekce, 1994. ISBN 80-900046-3-6
- [2] Ian Wade, G3NRW [APRS Protocol Specification Version 1.0.1](#) [online]. APRS working group, 2000 - [cit. 2.4. 2009]. Dostupné na [www: <http://www.tapr.org/aprs_working_group.html>](http://www.tapr.org/aprs_working_group.html).
- [3] CC1010 Single Chip Very Low Power Transceiver with 8051-Compatible Microcontroller. Texas Instruments, 13.4.2009 - [cit. 2.5. 2009]. Dostupné na [www: <http://focus.ti.com/docs/prod/folders/print/cc1010.html>](http://focus.ti.com/docs/prod/folders/print/cc1010.html).
- [4] Kaštánek, M. *APRS telemetrická stanice s obvodem fy. CHIPCON*. Brno: Bakalářská práce ÚREL FEKT VUT Brno, 2006.
- [5] Leadtek SMD GPS Module Specifications Sheet. Leadtek Research Inc., 25.8.2008 – [cit. 3.12.2009]. Dostupné na ftp://ftp.leadtek.com.tw/gps/9552/LR9552_v0.6_042808.pdf
- [6] CC1010IDE, software, dostupné na [www: <www.ti.com/litv/zip/swrc033>](http://www.ti.com/litv/zip/swrc033)
- [7] Gary Dion, N4TXI The WhereAVR APRS tracker. 18.7.2005 – [cit. 2.4.2011]. Dostupné na <http://www.garydion.com/projects/whereavr/>
- [8] Počítačové sítě - Model ISO/OSI, site.the.cz, [cit. 5.10.2011]. Dostupné na <http://site.the.cz/index.php?id=4>
- [9] Java (programovací jazyk), Wikipedie, 13.3.2011 – [cit. 2.5.2011]. Dostupné na http://cs.wikipedia.org/wiki/Java_%28programovac%C3%AD_jazyk%29
- [10] Java 2 Platform SE v.1.4.2, API Specification, Oracle, 2010 – [cit. 2.5.2011]. Dostupné na <http://download.oracle.com/javase/1.4.2/docs/api/>
- [11] The Java Tutorials, Lesson: Using Swing Components, Oracle 2011 – [cit. 2.5.2011]. Dostupné na <http://download.oracle.com/javase/tutorial/uiswing/components/index.html>
- [12] NMEA 0183 Version 6.1 , Peter Barnett, Sept. 15, 1997 – [cit. 5.5.2011]. Dostupné na <http://www.kh-gps.de/nmea.faq>