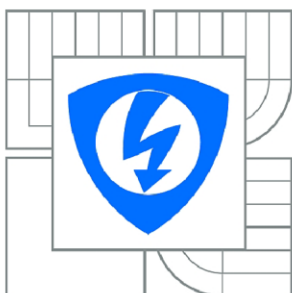




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

AUTONOMNÍ OVLADAČ ANTÉN V AZIMUTU A ELEVACI

ANTENNAS CONTROLLER IN AZIMUTH AND ELEVATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

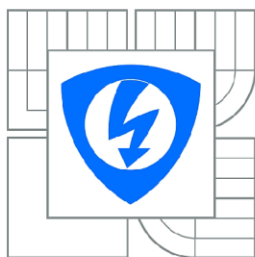
ANTONÍN LYKO

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. MIROSLAV KASAL, CSc.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Antonín Lyko

ID: 134546

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Autonomní ovladač antén v azimutu a elevaci

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte způsoby elektrického ovládání anténních rotátorů YAESU a EGIS. Seznamte se s jednočipovými mikropočítači. Navrhněte ideové uspořádání ovladače antén, který může být ovládán z řídicího počítače pomocí rozhraní USB. Návrh rozšířte o možnost samostatného ovládání tlačítky. Navrhněte desku plošného spoje a napište program pro mikropočítač.

Ovladač realizujte. Odlad'te program a funkci systému ověřte.

DOPORUČENÁ LITERATURA:

[1] POŠ, J. Mikrokontroléry PIC12C5XX. Praha: BEN – technická literatura, 2002.

[2] HRBÁČEK, J. Komunikace mikrokontroléru s okolím. Praha: BEN – technická literatura, 2002.

[3] LONG, H. The LVB Tracker. Dostupné na WWW:
<http://www.g6lvb.com/Articles/LVBTracker/index.htm>

Termín zadání: 11.2.2013

Termín odevzdání: 31.5.2013

Vedoucí práce: prof. Ing. Miroslav Kasal, CSc.

Konzultanti bakalářské práce:

prof. Dr. Ing. Zbyněk Raida

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Dokument popisuje realizaci rozhraní k řízení anténních rotátorů pomocí počítače. Jsou vysvětleny způsoby externího ovládní anténních rotátorů, současná řešení této problematiky a navrhnutá nová možná řešení. Závěr práce uvádí samotnou realizaci i s nepřesnostmi při návrhu a jejich vhodným řešením.

KLÍČOVÁ SLOVA

Řadič rotátorů, externí řízení, rozhraní.

ABSTRACT

Document describes the realization of an interface to control antenna rotators using a computer. It explain ways of external control antenna rotators, the current solution to this problem and suggested a new possible solutions. The conclusion presents itself with the implementation of imperfections in the design and appropriate solution.

KEYWORDS

Rotators tracker, external control, interface.

LYKO, A. *Autonomní ovladač antén v azimutu a elevaci*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 46 s., 5 s. příloh. Bakalářská práce. Vedoucí práce: prof. ing. Miroslav Kasal, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Autonomní ovladač antén v azimutu a elevaci jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce prof. ing. Miroslavu Kasalovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....

(podpis autora)

OBSAH

Seznam obrázků	VII
Seznam tabulek	VIII
Úvod	1
1 Anténní rotátory	2
1.1 Anténní rotátory YAESU.....	2
1.2 Anténní Rotátory EGIS.....	4
2 Las Vegas Boulevard Tracker	6
2.1 Uživatelské Rozhraní LVBT	6
2.2 Základní Periferie LVBT	8
2.3 Schéma Zapojení LVBT	12
3 Specifikace rozhraní RS232C	13
3.1 Základní Parametry Asynchronního Přenosu Dat	13
4 Modifikace LVBT	16
4.1 Rozhraní RS232C - komunikace s rotátory Egis	16
4.2 Modifikace DPS LVBT	17
4.3 Modifikace Firmwaru LVBT	20
5 Realizace	24
5.1 DPS Modifikovaného LVBT	24
5.2 Modifikace Firmwaru LVBT	25
5.3 Zprovoznění Modifikovaného LVBT	27
6 Závěr	31
Literatura	32
Seznam příloh	33
Návrh zařízení	34

SEZNAM OBRÁZKŮ

Obr. 1. 1:	Konektor externího řízení řídicí jednotky rotátoru Yaesu G-5500.....	2
Obr. 1. 2:	Propojení počítače pomocí sériového portu, rozhraní GS-232A a řídicí jednotky rotátoru Yaesu G-5500.....	3
Obr. 1. 3:	Konektor externího řízení rotátorů Egis.	4
Obr. 2. 1:	Zapojení LVBT v celkovém uspořádání externího řízení rotátorů Yaesu.	6
Obr. 2. 2:	Nastavení parametrů pro sériovou komunikaci s LVBT prostřednictvím softwaru Terminal v1.9b.....	7
Obr. 2. 3:	Schéma zapojení rozhraní pro ovládání rotátorů Yaesu a mikrokontroléru PIC16F876	8
Obr. 2. 4:	Schéma připojení ovládacích tlačítek a LCD displeje k řídicímu mikrokontroléru.	9
Obr. 2. 5:	Schéma zapojení rozhraní LVBT pro komunikaci s počítačem.	10
Obr. 2. 6:	Schéma zapojení napájecí části LVBT.	11
Obr. 2. 7:	Modul DLP-USB232M.....	11
Obr. 2. 8:	Celkové schéma zapojení LVBT	12
Obr. 3. 1:	Struktura protokolu 8N1 pro sériový asynchronní přenos dat.	14
Obr. 3. 2:	Propojovací kabely v 7 drátovém propojení s úplným řízením toku a 3 drátovým propojením-null modem.	14
Obr. 3. 3:	Převodník napět'ových úrovní MAX232.	15
Obr. 4. 1:	Paralelní připojení dvou zařízení na sériovou linku.	16
Obr. 4. 2:	Schéma zapojení rozhraní RS232C pro rotátory Egis	17
Obr. 4. 3:	Schéma zapojení napájení MLVBT.....	18
Obr. 4. 4:	Schéma zapojení modifikovaného LVBT k řízení rotátorů Egis.....	19
Obr. 5. 1:	Klasické rozložení pinů tranzistoru BC547 v pouzdře TO92.....	25
Obr. 5. 2:	Nastavení konfiguračního slova pro aplikaci modifikovaného firmwaru LVBT.	27

SEZNAM TABULEK

Tab. 1. 1:	Funkce jednotlivých pinů vstupu pro externí řízení rotátorů Yaesu G-5500.	3
Tab. 1. 2:	Standardní příkazy GS-232A.....	4
Tab. 1. 3:	Základní příkazy podporované řídicími jednotkami rotátorů Egis.....	5
Tab. 2. 1:	Podporované Easycomm příkazy LVBT	7
Tab. 2. 2:	Podporovaná rozšířená sada příkazů GS-232A LVBT.....	7

ÚVOD

Anténní rotátory se obecně využívají k natáčení konkrétních zařízení (antény, kamery, solární panely atd.) požadovaným směrem ke zdroji signálu. Jejich ovládání zprostředkovává externí řídicí jednotka.

Hlavní motivací k vytvoření komunikačního rozhraní mezi počítačem a řídicí jednotkou rotátoru je možnost automatizovaného měření či pouze ovládání.

1 ANTÉNNÍ ROTÁTORY

Anténními rotátory obecně rozumíme servomotory, s možností otáčení v jedné, nebo dvou na sebe kolmých rovinách. Azimutu a elevace. Rotátory se využívají k natočení konkrétních zařízení (antény, kamery, solární panely atd.) požadovaným směrem ke zdroji signálu.

1.1 Anténní rotátory YAESU

1.1.1 Řídící jednotka rotátorů Yaesu

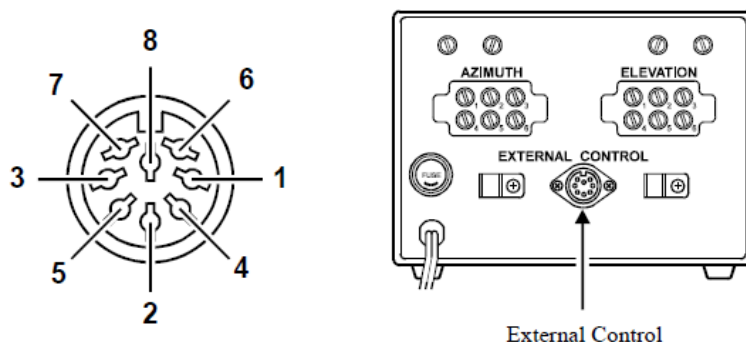
Všechny typy rotátorů značky Yaesu jsou ovládány skrze svou externí řídicí jednotku. Ta je vybavena ovládacími tlačítky, ručkovým displejem znázorňujícím aktuální úhel natočení rotátoru a vstupním konektorem určeného k externímu řízení. Tyto řídicí jednotky se pro jednotlivé řady a typy rotátorů liší a s nimi i vstupní a výstupní konektory.

1.1.2 Externí ovládání rotátorů Yaesu

Z řad rotorů Yaesu se vyznačuje natáčením v rovině azimutu i elevace typ G-5500. Ostatní rotátory nabízejí natáčení pouze v jedné z rovin.

Vhodnou kombinací jednotlivých typů rotátoru lze získat celek, schopný natáčení v obou rovinách. V takovém případě jsou zapotřebí dvě řídicí jednotky a k nim dva vstupy externího řízení.

Pro jednoduchost je vysvětlen princip na řídicí jednotce rotátoru G-5500. Vstup externího řízení zobrazuje Obr. 1.1 a funkci jednotlivých pinů vysvětluje Tab. 1.1.



Obr. 1. 1: Konektor externího řízení řídicí jednotky rotátoru Yaesu G-5500 [1].

Připojením jednotlivých řídicích pinů ke svorce s nulovým napěťovým potenciálem je řídicí jednotce předána instrukce o požadované operaci a ta ji poté vykonává.

Zpětnou vazbu s informací o aktuální poloze rotátoru poskytují piny 6 a 1 ve formě stejnosměrného napětí do 5V, které je lineárně závislé na úhlu natočení rotátoru. Je použit klasický 8 pinový DIN konektor (samice).

Tab. 1. 1: Funkce jednotlivých pinů vstupu pro externí řízení rotátorů Yaesu G-5500[1].

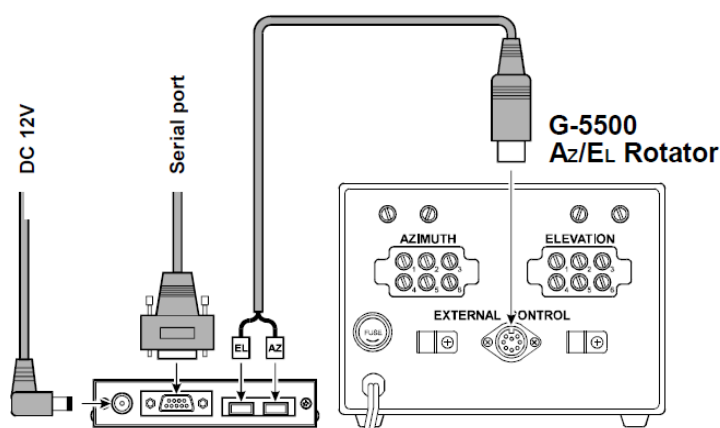
Pin	Funkce
6	Poskytuje 2 až 4,5VDC odpovídající 0° až 450°
1	Poskytuje 2 až 4,5VDC odpovídající 0° až 180°
4	Připojení k pinu 8 - otáčení vlevo
2	Připojení k pinu 8 - otáčení vpravo
5	Připojení k pinu 8 - otáčení dolů
3	Připojení k pinu 8 - otáčení nahoru
7	Poskytuje 13 až 6 VDC do 200mA
8	Poskytuje 0VDC

(pozn.: V oficiálním datasheetu výrobce je chyba a funkce pinů 5 a 3 jsou zaměněny [2])

1.1.3 Rozhraní GS-232A

Pro řízení rotátoru z počítače pomocí sériového rozhraní RS232C vyvinula firma Yaesu zařízení nazvané GS-232A Computer Control Interface. Tato přídatná jednotka funguje jako rozhraní mezi sériovým portem počítače a vstupem externího řízení řídicí jednotky rotátoru. Jelikož existuje více druhů řídicích jednotek a jejich vstupních konektorů, jsou vyráběny speciální propojovací kabely, které zajistí kompatibilitu mezi výstupem GS-232A a vstupem řídicí jednotky.

Způsob propojení počítače pomocí sériového portu skrze jednotku GS-232A s řídicí jednotkou rotátoru je naznačen na Obr. 1.2.



Obr. 1. 2: Propojení počítače pomocí sériového portu, rozhraní GS-232A a řídicí jednotky rotátoru Yaesu G-5500 [3].

Přídavnou jednotku GS-232A je definována určitá sada řídicích příkazů (viz. Tab. 1.2), díky které je uživatel schopen provádět operace ovládání polohy a kalibrace rotátoru prostřednictvím sériového linky.

Tab. 1. 2: Standardní příkazy GS-232A [3].

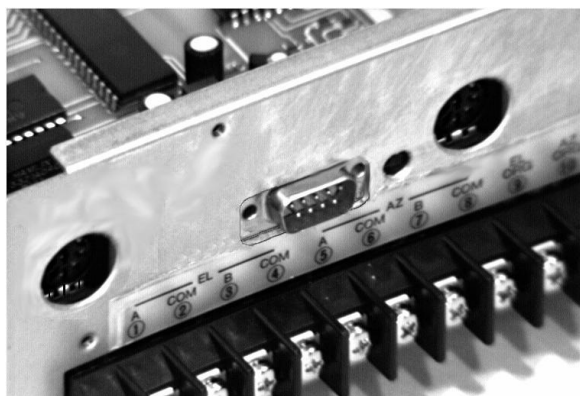
Příkaz	Funkce
C	Vrací azimut
C2	Vrací azimut a elevaci
Maaa	Nastaví azimut na úhel <aaa> ve stupních
S	Zastaví rotátor
Waaa	Nastaví azimut na úhel <aaa> ve stupních
Waaa eee	Nastaví azimut a elevaci na úhel <aaa> a <eee> ve stupních

1.2 Anténní Rotátory EGIS

1.2.1 Řídící jednotka rotátorů Egis

Rotátory Egis jsou obdobně jako rotátory Yaesu ovládány svou řídicí jednotkou. Ta je vybavena tlačítky a displejem pro přímou komunikaci s obsluhou.

Tyto jednotky se mírně liší v závislosti na zvolené řadě rotátoru. K externímu řízení počítačem však využívají všechny rozhraní RS-232C (viz. Obr. 1.3)



Obr. 1. 3: Konektor externího řízení rotátorů Egis[4].

1.2.2 Externí ovládání rotátorů Egis

Sériový přenos dat se řídí protokolem 8N1, jenž definuje použití 8 datových bitů, žádnou paritu a jeden stop bit. Podporované rychlosti jsou od 50 do 9900 Bd. Implicitně nastavená hodnota je 1200 Bd. Jednotka podporuje hardwarový handshaking. Parametry komunikace (rychlost a handshaking) je možné nastavit pomocí programovacího módu PRG 23.

Forma řídicích příkazů (viz. Tab. 1.3) je pevně definována a musí být dodržena. Všechny instrukce musí začínat symboly “&*” a končit netisknutelnými CR znaky (Carriage Return), kterým odpovídají v ASCII tabulce hexadecimální hodnoty 0D a 0A. Mezi tyto atributy se píše jednotlivé povely (viz. Tab. 1.3).

Příklad syntaxe:

&* (mezera) AZ=182.05 (CR)
&* (mezera) EL=? (CR)

Tab. 1. 3: Základní příkazy podporované řídicími jednotkami rotátorů Egis [5].

Příkaz	funkce
AZ=000.00	Nastav azimut na úhel 000.00
EL=000.00	Nastav elevaci na úhel 000.00
AZ=?	Vrací hodnotu aktuálního azimutu
EL=?	Vrací hodnotu aktuální elevace
P=000	Nasměruje rotátor na satelit číslo 000
SATLIST=?	Vrací kompletní tabulku satelitů z paměti

Jednotka obsahuje konektor CANNON typu samec (viz Obr. 1.3). S počítačem, který má rovněž tento konektor, se propojuje pomocí nulového modemu.

2 LAS VEGAS BOULEVARD TRACKER

LVBT je zařízení, určené k externímu řízení rotátorů Yaesu prostřednictvím sériové linky nebo USB portu, s licenci typu Open Source. Tzn., že schéma zapojení i zdrojový kód řídicího mikrokontroléru jsou volně dostupné s možností vlastních úprav.

LVBT plní funkci rozhraní GS-232A, jak ukazuje Obr. 2.1, a podporuje také další sady řídicích příkazů jiných komerčních výrobců. Tím zajišťuje kompatibilitu s již existujícími obslužnými softwary (např. Wisp, Nova atd.).



Obr. 2. 1: Zapojení LVBT v celkovém uspořádání externího řízení rotátorů Yaesu [2].

2.1 Uživatelské Rozhraní LVBT

LVBT je vybaven dvouřádkovým LCD displejem na kterém je zobrazována aktuální poloha rotátoru ve stupních v rovinách azimutu i elevace a zároveň požadovaný úhel natočení v obou rovinách. Dále obsahuje čtyři tlačítka k rotaci v daných směrech (nahoru, dolů, vpravo, vlevo). Pro komunikaci s počítačem je možné využít buďto sériovou linku nebo USB port.

2.1.1 Ovládání pomocí USB/RS232C

Součástí LVBT je modul DLP-USB232M od firmy FTDI, jenž plní funkci rozhraní USB-UART. Po připojení k počítači pomocí USB, pracuje vytvořené spojení jako virtuální sériový port. Pro správnou funkci modulu je nutné mít nainstalovány ovladače.

Posílání dat přes sériový port počítače může být realizováno pomocí libovolného programového terminálu. K odzkoušení funkce byl použit software Terminal v1.9b. Jednotlivé parametry komunikace zobrazuje Obr. 2.2 a jsou následující

Použita bitová rychlost: 9600 Bd.

Počet datových bitů: 8.

Kontrola parity: není.

Stop bit: jeden

Řízení datového toku handshaking: není.

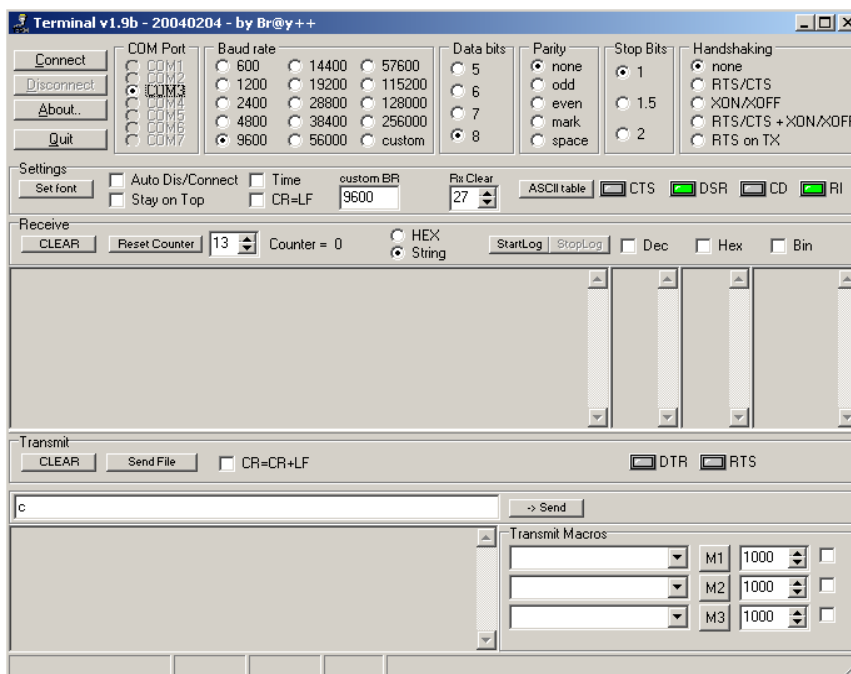
LVBT podporuje standardní sadu příkazů GS-232A (viz. Tab. 1.2), rozšířenou sadu příkazů GS-232A (viz. Tab. 2.2) a Easycomm příkazy (viz. Tab. 2.1).

Tab. 2. 1: Podporované Easycomm příkazy LVBT [2].

Příkaz	Funkce
Azaaa.a	Nastavit azimut na <aaa.a>
Eleee.e	Nastavit elevaci na <eee.e>

Tab. 2. 2: Podporovaná rozšířená sada příkazů GS-232A LVBT [2].

Příkaz	Funkce
FAE	Nastavit 360° kalibraci azimutu
FAF	Nastavit 450° kalibraci azimutu
FAOooo	Nastavit offset azimutu na <ooo>
FAS	Nastavit 0° kalibraci azimutu
FEE	Nastavit 180° kalibraci elevace
FEN	Nastavit 90° kalibraci elevace
FEOooo	Nastavit offset elevace na <ooo>
FES	Nastavit 0° kalibraci elevace
FN	Nastavit severně zastavující rotátor
FS	Nastavit jižně zastavující rotátor
FW	Zapsat kalibrační data do EEPROM paměti



Obr. 2. 2: Nastavení parametrů pro sériovou komunikaci s LVBT prostřednictvím softwaru Terminal v1.9b.

2.2 Základní Periferie LVBT

Podkapitola 2.2 nabízí rozbor základních periférií LVBT, z nichž některé je nutno modifikovat, za účelem vytvoření komunikačního rozhraní RS232C k obsluze rotátorů EGIS.

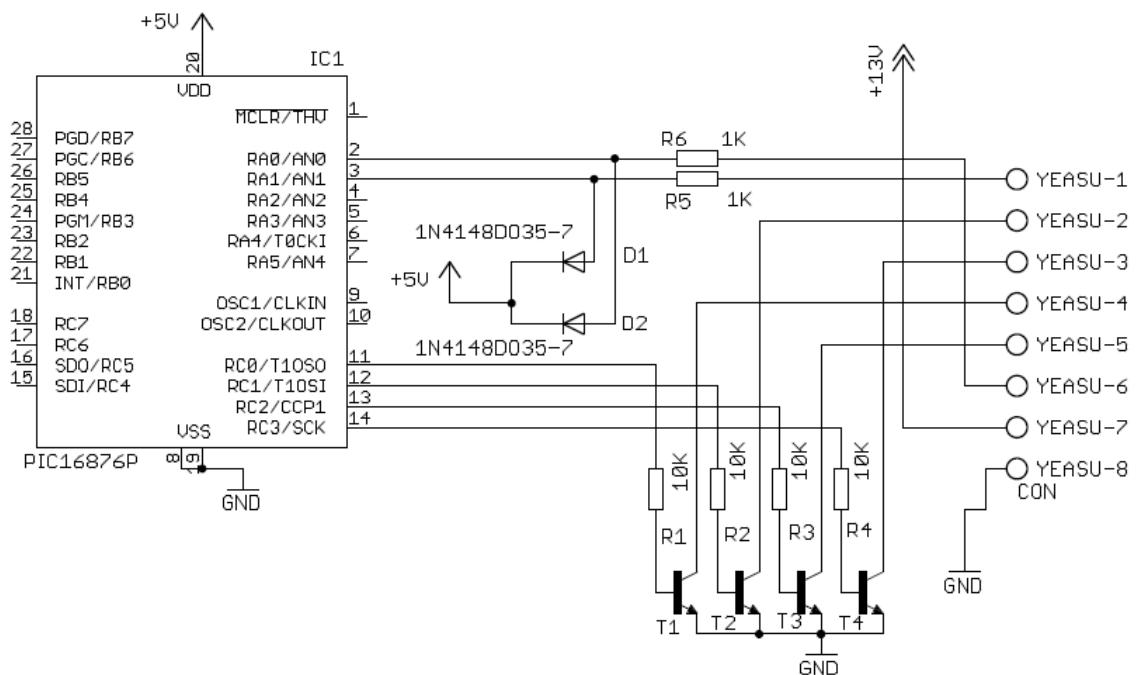
2.2.1 Řídící mikrokontrolér PIC16F876

Jádrem autonomního ovladače antén je 8 bitový mikropočítač PIC16F876 od firmy Microchip, který disponuje 8K x 14 slov flash programovou pamětí. Vyrábí se v pouzdech DIP a SOIC s 28 piny.

Hlavními perifériemi, které mikropočítač pro tuto aplikaci využívá, jsou dostatečný počet vstupně/výstupních portů, A/D převodník, paměť EEPROM a modul USART.

2.2.2 Rozhraní externího ovládání rotátorů Yaesu

Řídícím mikropočítačem PIC16F876 jsou skrze porty RC0, RC1, RC2 a RC3 spínány tranzistory BC547 přes rezistory R1,R2,R3 a R4 s odpory 10KΩ. Výrobce řídicích jednotek Yaesu neudává proud, který protéká při sepnutí řídicích pinů se zemním vodičem, a napětí mezi těmito piny v rozpojeném stavu. Jelikož byl LVB Tracker úspěšně testován, je zřejmé, že tyto hodnoty musí být menší, než mezní hodnoty tranzistorů BC547. Tzn., že maximální spínaný proud je do 100mA a maximální napětí mezi piny je do 45V. Způsob zapojení ukazuje Obr. 2.3.



Obr. 2.3: Schéma zapojení rozhraní pro ovládání rotátorů Yaesu a mikrokontroléru PIC16F876 [2]

Na pinech 1 a 6 konektoru externího řízení řídicí jednotky Yaesu jsou stejnosměrná napětí udávající úhel natočení v rovinách azimutu a elevace. Tyto signály jsou přivedeny prostřednictvím rezistorů R6 a R5 o hodnotě odporu 1KΩ na A/D převodník skrze porty mikrokontroléru RA0 a RA1. Závislost napětí na úhlu natočení je lineární, avšak pro správnou interpretaci rozsahu pohybu rotátoru je nutné ověřit mezní hodnoty napětí udávané výrobcem.

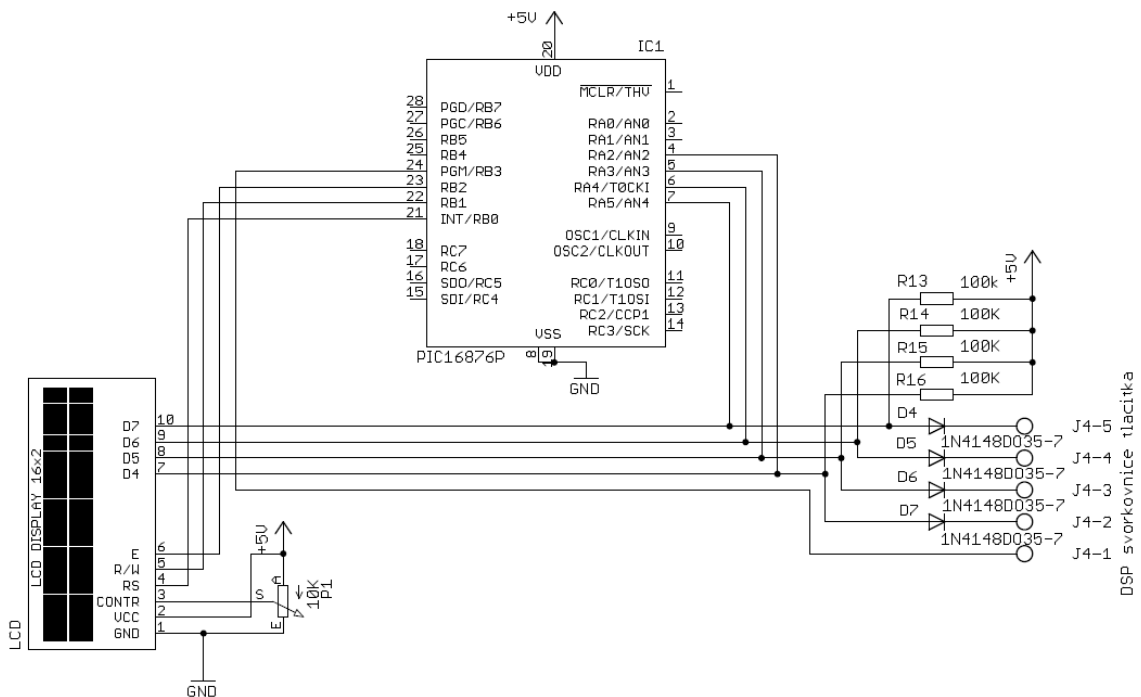
Kalibrace se provádí postupným natočením rotátoru do obou krajních poloh a uložením jim odpovídajícím velikostem napětí do paměti mikrokontroléru. Ze známého rozsahu natočení rotátoru (90°, 180°, 360°, 450°), jenž je zadán obsluhou pomocí kalibračních příkazů, je vypočítáno potřebné zpětnovazební napětí pro natočení o jeden stupeň.

Tyto kalibrační hodnoty jsou uloženy do paměti typu EEPROM, která nepotřebuje pro uchování dat napájecí napětí.

2.2.3 Řízení LCD displeje a tlačítek

Datové piny LCD displeje jakož i tlačítka jsou připojeny k portům RA2, RA3, RA4 a RA5. Ke spolehlivé obsluze pouze jedné periferie v daném okamžiku je využito přepínání brány RA mezi vstupním a výstupním režimem a adresování zařízení pomocí pinů RB2 a RB3, jak ukazuje Obr. 2.4.

Napětí na svorce displeje nazvané E (enable), plní funkci hodinového signálu, kdy při sestupné hraně přečte řadič LCD data ze sběrnice a vykoná příslušnou operaci.



Obr. 2. 4: Schéma připojení ovládacích tlačítek a LCD displeje k řídicímu mikrokontroléru [2].

Napětí společného vodiče všech tlačítek je řízeno softwarově prostřednictvím pinu RB3, který je aktivní v nízké úrovni. Při čtení tlačítek jsou uvedeny odpovídající porty RA do vstupního režimu. V klidové poloze je na vstupech brány RA logická jednička

a při stlačení tlačítka je na konkrétním pinu RA logická nula.

Potenciometrem připojeným k LCD displeji je nastavován kontrast.

2.2.4 USB a RS232C rozhraní pro komunikaci s počítačem

Modul DLP-USB232M (viz Obr. 2.7) od firmy FTDI, jenž je součástí LVBT, pracuje jako rozhraní USB-UART. Převádí tedy USB komunikaci na asynchronní sériový přenos dat s napěťovými úrovněmi v TTL logice.

Pro správnou funkci převodníku DLP-USB232M je zapotřebí instalovat ovladače dvakrát, přesně podle výzvy operačního systému (realizováno na os Win XP). Připojené zařízení poté pracuje jako virtuální sériový port. V případě, že je tomuto portu přiděleno vysoké pořadové číslo COM, je vhodné jej pro spolehlivou funkci komunikace v operačním systému snížit.

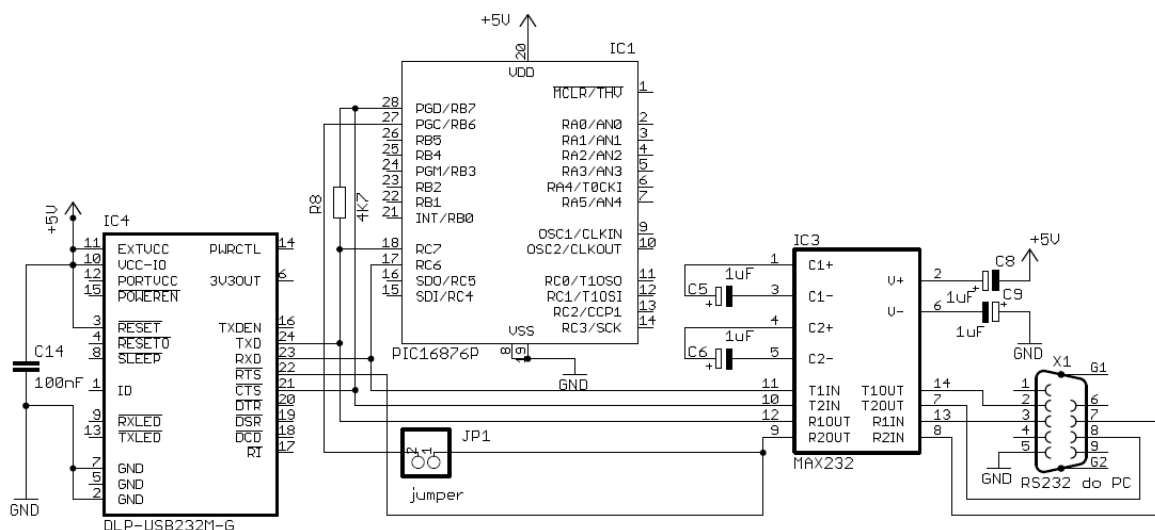
K následnému přenosu dat prostřednictvím USB portu počítače již stačí libovolný programový terminál (viz. Obr. 2.2).

Při využití rozhraní RS232C je zapotřebí zprostředkovat převod napěťových úrovní mezi jednotlivými částmi komunikační dráhy. Modul USART, který je součástí mikrokontroléru PIC16F876, pracuje v napěťových hladinách TTL logiky, tedy do 5V. Napětí použita na sériové lince jsou až $\pm 15V$ s opačnou reprezentací logických nul a jedniček než je tomu u TTL logiky.

Potřebnou konverzi mezi těmito formáty zajistí integrovaný obvod MAX232.

Jelikož jsou obě rozhraní, jak USB tak RS232C, připojeny k jedinému modulu USART, je nutné využívat vždy pouze jedno toto rozhraní a k němu odpovídající modul. Připojením všech tří modulů na jedinou sběrnici může docházet ke kolizím a nefunkčnosti veškeré komunikace.

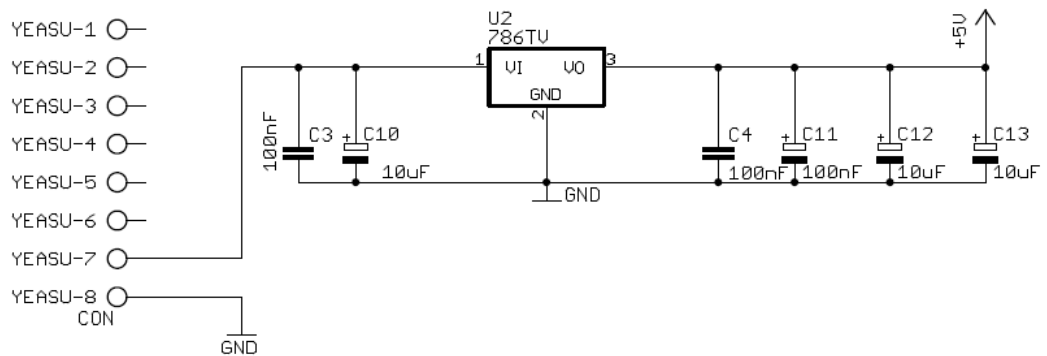
Schéma zapojení těchto periférií LVBT ukazuje Obr. 2.5.



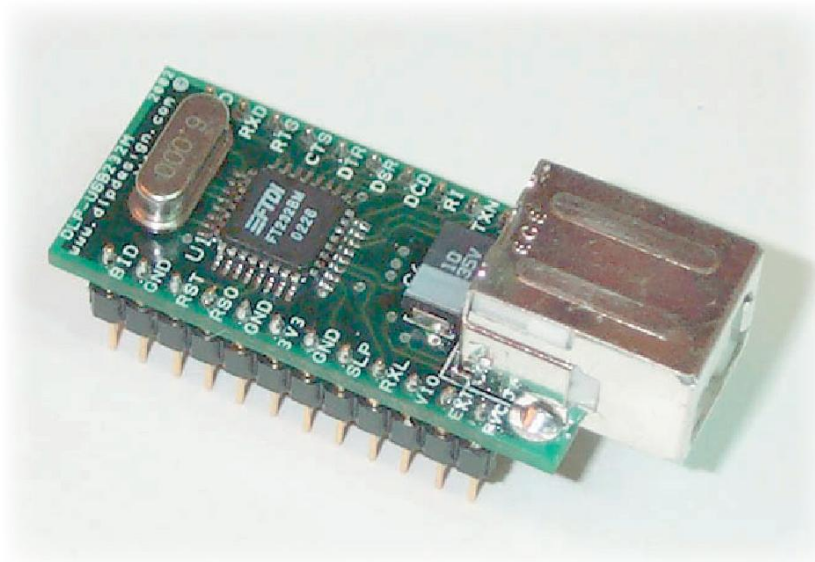
Obr. 2. 5: Schéma zapojení rozhraní LVBT pro komunikaci s počítačem [2].

2.2.5 Napájení LVBT

LVBT je napájen prostřednictvím řídicí jednotky rotátorů Yeasu, která na pinu 7 DIN konektoru pro externí řízení poskytuje stejnosměrné napětí 13V. Pomocí obvodu 7805 je napájecí napětí stabilizováno na hodnotě 5V, kterými jsou poté napájeny jednotlivé periferie LVBT.



Obr. 2. 6: Schéma zapojení napájecí části LVBT [2].



Obr. 2. 7: Modul DLP-USB232M [6].

3 SPECIFIKACE ROZHRANÍ RS232C

3.1 Základní Parametry Asynchronního Přenosu Dat

Sériový přenos dat se vyznačuje odesláním jednotlivých bitů pomocí jediného vodiče v čase za sebou.

Asynchronní komunikace je charakteristická absencí tzv. synchronizačních pulsů (využívajících samostatný vodič), které v čase určují konkrétní polohu jednotlivých bitů.

Synchronizace je u tohoto způsobu přenosu dat dosaženo prostřednictvím předem definované struktury, v jaké jsou bity přenášeny. Musí zde být tedy stanoven určitý postup nebo protokol, v jakém jsou data posílána, aby došlo na straně přijímače k jejich správnému vyhodnocení.

3.1.1 Struktura asynchronní sériové komunikace

Základem asynchronní sériové komunikace je známá bitová rychlost. Ta definuje délku trvání jednoho bitu. Pro úspěšný přenos dat mezi zařízeními je potřeba, aby obě zařízení pracovala se stejnou bitovou rychlostí.

V klidovém stavu, kdy nejsou posílána žádná data, se nachází sběrnice v logické jedničce.

Počátek přenosu dat je zahájen tzv. start bitem (logická nula), který slouží k synchronizaci přijímače.

Poté jsou již odesílány data po jednotlivých bitech od LSB (bit s nejmenší vahou 1) po MSB (bit s největší vahou, ta závisí na počtu použitých bitů) v různé délce v závislosti na použitém protokolu. Nejčastěji se používá 7 nebo 8 datových bitů. Lze se však setkat i s délkami 5 až 9 datových bitů. Na Obr. 3.1 je použito 8 datových bitů.

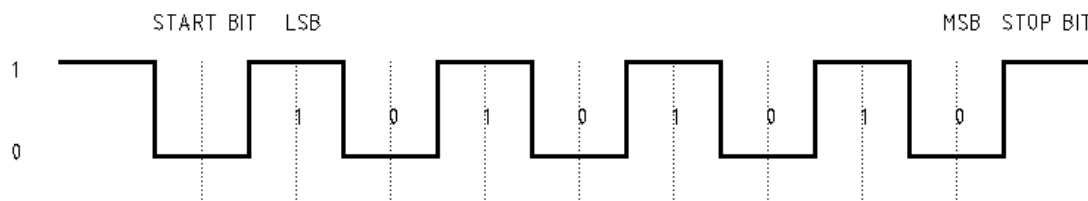
Za datovými bity může následovat paritní bit, který slouží k identifikaci možné chyby v přenesených datech na straně přijímače. Paritu rozlišujeme sudou nebo lichou. Podle jejího druhu je paritní bit doplněn buďto jedničkou nebo nulou tak, aby byl celkový počet jedniček v jednom rámci (rámec je ohraničen start a stop bitem) sudý nebo lichý. Základním předpokladem pro úspěšnou detekci chyb je shoda vysílače i přijímače v použití a druhu parity.

Celý jeden rámec je ukončen tzv. stop bitem nebo bity, jejíž logická úroveň je jedna. Podle daného protokolu je použit buď jeden, jeden a půl nebo dva stop bity. Po tomto okamžiku je sběrnice opět v klidovém stavu a přijímač již může očekávat další start bit.

Jelikož může tento způsob komunikace probíhat pouze jedním směrem, je nutné vést mezi dvěma zařízeními vždy dva vodiče pro oba směry přenosu dat.

Obr. 3.1 zobrazuje strukturu protokolu 8N1 jenž definuje použití 8 datových bitů, žádnou paritu a jeden stop bit. Datový byte má binární hodnotu 1010101b. V dekadické

soustavě ji odpovídá číslo 85, pro nějž je v ASCII tabulce definován znak “U”.



Obr. 3. 1: Struktura protokolu 8N1 pro sériový asynchronní přenos dat.

3.1.2 Zapojení konektorů a použití kabelů

Rozhraní RS232C vymezuje použití 25 nebo 9 pinového konektoru CANNON. Jednotlivé piny jsou v závislosti na druhu konektoru (samec, samice, navzájem zrcadlově číslování) očíslovány a je jim přidělena určitá funkce.

Pro 9 pinový konektor CANNON plní nejdůležitější funkci piny 3 (vysílací), 2 (přijímací) a 5 (nulový potenciál napětí).

Dvě zařízení, komunikující přes sériovou linku, musí být propojena takovým způsobem, aby byl vysílací signál z jednoho zařízení převeden na přijímací pin druhého a naopak. Toho je dosaženo pomocí propojovacích kabelů a způsobu zapojení konektorů.

Je zvykem u konektoru CANNON typu samice provést křížení vysílacích a přijímacích signálů mezi zařízením a vnitřní stranou konektoru. U typu samec se křížení neprovádí a oba signály jsou vedeny přímo.

Jelikož musí na přenosové dráze dojít k lichému počtu (nejlépe 1) křížení vysílacích a přijímací signálů a zapojení konektorů je předvídatelné, lze stanovit druhy propojovacích kabelů a situace jejich použití.

Propojení dvou zařízení pouze se samčími nebo samičými konektory se provádí pomocí nulového modemu, což je ve své podstatě křížený kabel (viz. Obr. 3.2). Samčí se samičím konektorem se propojují přímým kabelem.



Obr. 3. 2: Propojovací kabely v 7 drátovém propojením s úplným řízením toku a 3 drátovým propojením-null modem [7].

3.1.3 Napětové úrovně, převodník MAX232

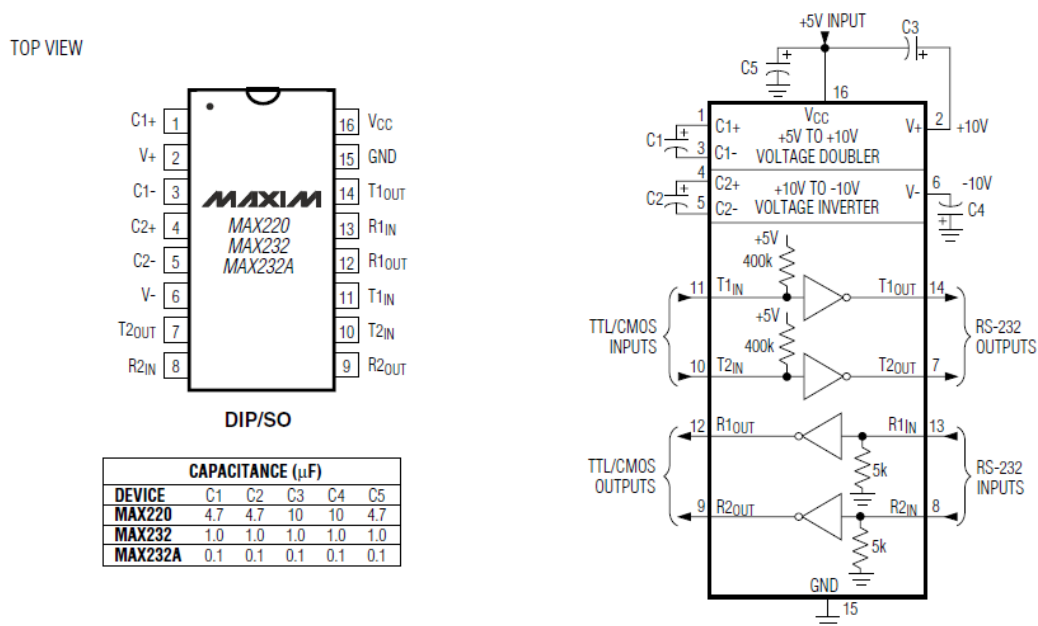
Sériová linka byla navržena pro přenos dat v průmyslovém prostředí na velké vzdálenosti. Neklade tedy důraz na objem přenesených dat za určitý časový interval, ale na spolehlivost.

Proto byly zvoleny velké rozestupy jednotlivých napětových úrovní reprezentujících logickou nulu a jedničku. Tyto úrovně mohou být $\pm 5V$, $\pm 10V$ nebo

$\pm 15V$. Logickou jedničku zde reprezentují záporná napětí a logickou nulu kladná.

Zařízení pracující na bázi TTL logiky pracují na napěťových úrovních přibližně 0 až 5V, kdy logickou nulu představuje nižší napěťová hladina a logickou jedničku vyšší. Ve všech případech existuje určité zakázané pásmo, ve kterém nemá úroveň napětí žádnou logickou hodnotu.

MAX232 je integrovaný obvod navržený pro převod napěťových úrovní TTL/CMOS logiky na napěťové úrovně sériové linky. Přitom je napájen pouze 5V. Vyšší napětí si dokáže vyprodukovat samostatně pomocí externích kondenzátorů a spínaných nábojových měničů napětí. Způsob zapojení a funkci jednotlivých pinů ukazuje Obr. 3.3.



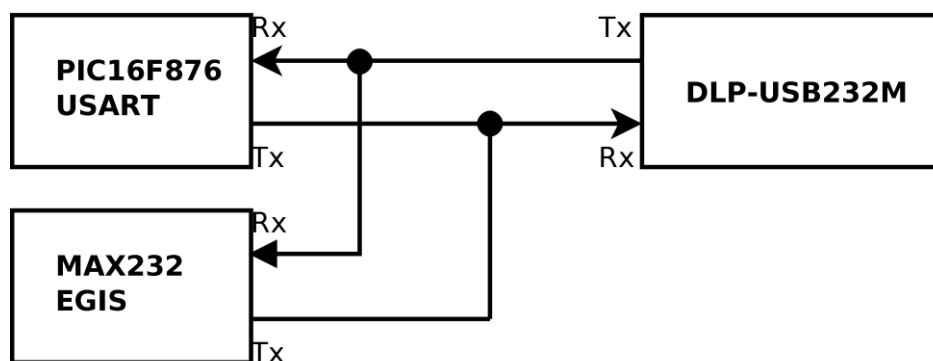
Obr. 3. 3: Převodník napěťových úrovní MAX232 [7].

4 MODIFIKACE LVBT

4.1 Rozhraní RS232C - komunikace s rotátory Egis

Cílem modifikace je zajistit tok dat (instrukcí) z počítače prostřednictvím USB portu k řídicí jednotce rotátorů EGIS a využít tak integrovaný převodník USB/UART. Toho může být dosaženo několika způsoby, jenž jsou popsány níže, v pořadí podle rozsáhlosti potřebných modifikací.

- 1) Díky rozdílné instrukční sadě rotátorů Yaesu a Egis, dochází při paralelním zapojení dvou zařízení na jedinou sériovou linku (viz. Obr. 4.1) k selekci řídicích příkazů samotnými rotátory. Všechna tři zařízení (modul USART mikrokontroléru reprezentující rozhraní Yaesu, převodník napěťových úrovní MAX232 a modul DLP-USB232M) jsou připojena na jedinou sériovou linku tak, aby se navzájem neovlivňovala.



Obr. 4. 1: Paralelní připojení dvou zařízení na sériovou linku.

- 2) Vytvoření softwarově generovaného sériového portu na vstupně výstupních pinech mikrokontroléru PIC16F876 k přeposílání řídicích příkazů určených rotátorům Egis tímto rozhraním.
- 3) Použití mikrokontroléru se dvěma moduly USART.

4.1.1 Připojení tří zařízení na jednu sériovou linku

Experimentálně bylo odzkoušeno připojení modulu USART, DLP-USB232M a integrovaného obvodu MAX232 k jediné sériové lince tak, aby se zařízení v navzájem připojených vysílacích větvích neovlivňovala.

Vysílací vodiče jednotlivých modulů byly odděleny diodami.

Z pokusu vyplývá, že je toto zapojení nefunkční.

4.1.2 Softwarově generovaný sériový port

Většina řešení je soustředěna do modifikace firmwaru mikrokontroléru, kde je kladen důraz na přesné načasování okamžiků čtení a zápisu dat na vstupně/výstupní porty. Jelikož se toto řešení stalo funkčně spolehlivým, další možnosti nebyly realizovány.

4.2 Modifikace DPS LVBT

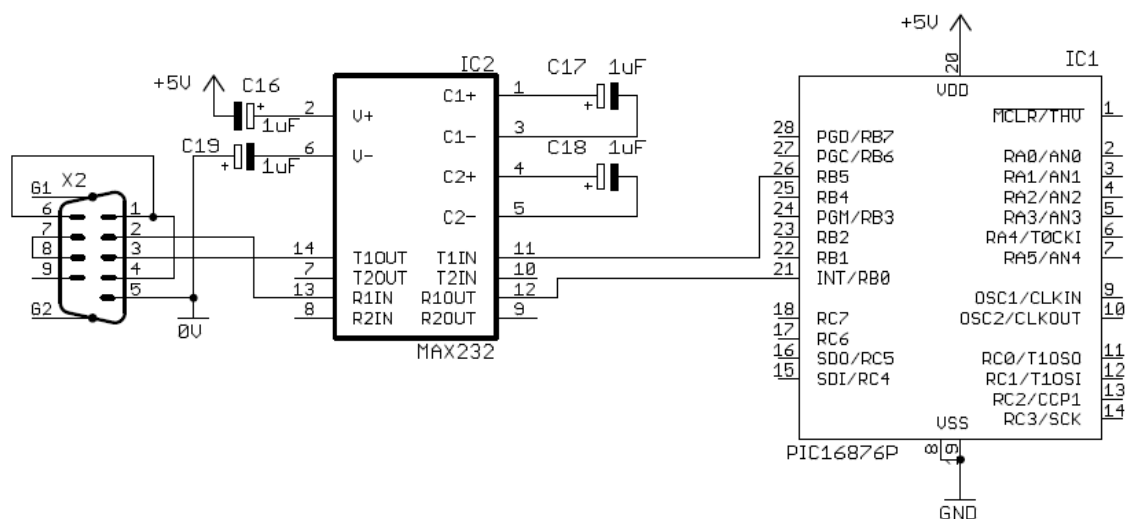
4.2.1 Vytvoření rozhraní RS232C

Řídicí jednotky rotátorů Egis jsou vybaveny konektory typu samec. Jelikož zde existuje předpoklad externího řízení přímo z počítače (také samčí konektor), je v datasheetu výrobce uveden požadavek k použití nulového modemu. Z důvodů možného opětovného použití toho propojovacího kabelu byl k řízení rotátorů Egis vybrán konektor typu samec.

Za vysílací svorku byl zvolen dosud nepoužívaný pin 5 portu B. K přijímacím účelům byl vybrán pin 0 portu B, z důvodu jeho alternativní funkce, kterou je externí přerušení probíhajícího programu a vykonání následné obsluhy při sestupné nebo náběžné hraně signálu na tomto pinu. Toho je využito k počáteční synchronizaci při příchodu start bitu.

Jelikož byl původně v LVBT pin 0 portu B využíván k řízení LCD displeje, byl nyní nahrazen portem RB4. Firmware LVBT byl vhodně modifikován tak, aby akceptoval konstrukční změny.

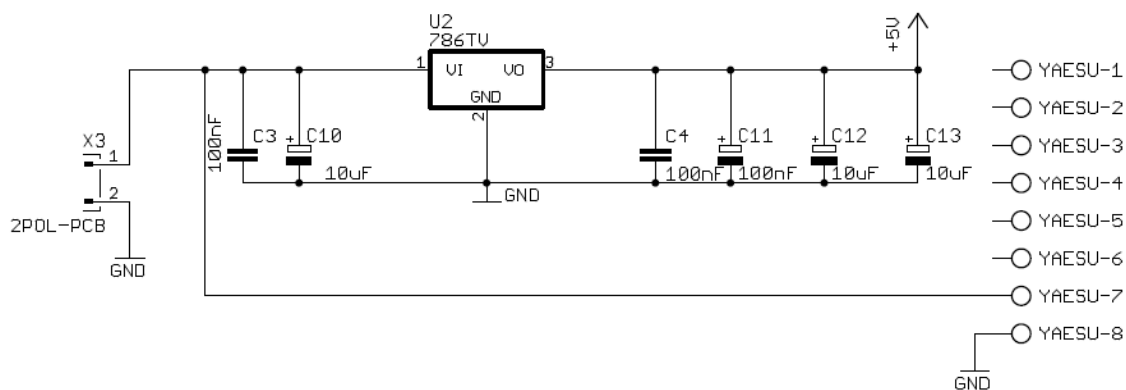
Na vnitřní straně konektoru CANNON typu samec nedochází ke křížení vysílacích a přijímacích signálů (viz. Obr 4.2).



Obr. 4. 2: Schéma zapojení rozhraní RS232C pro rotátory Egis

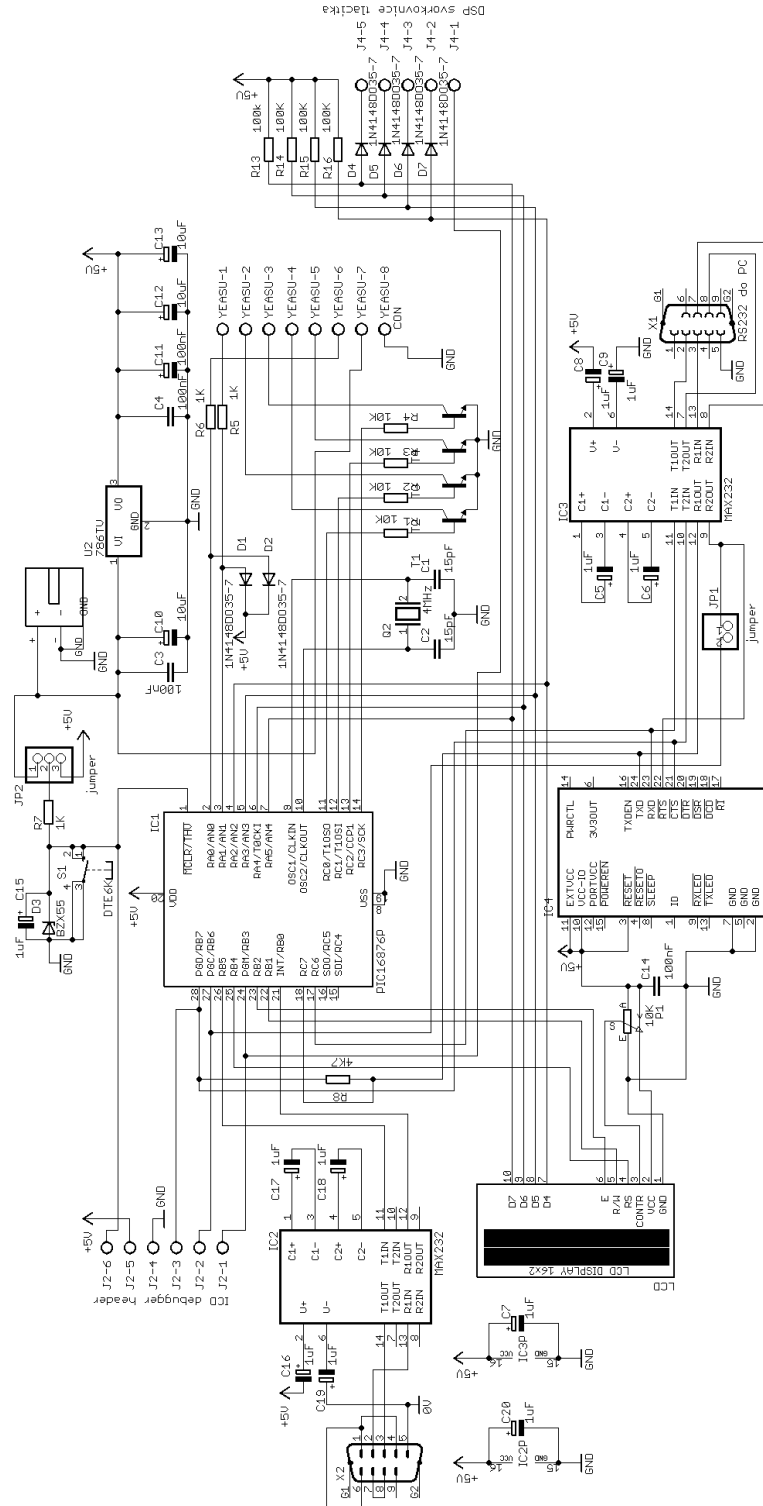
4.2.2 Modifikace napájecí části

Původní LVBT byl napájen z řídicí jednotky rotátoru Yaesu. Z důvodu možného řízení samostatného rotátoru Egis byl přidán do panelu zařízení konektor pro externí napájení (viz. Obr. 4.3). Ten je připojen paralelně ke zdroji 13V. Proto musí být používán pouze jeden z těchto zdrojů. Pokud je ovladač antén připojen k řídicí jednotce rotátorů Yaesu, nesmí být použit zdroj externího napájení. V opačném případě musí být použit.



Obr. 4. 3: Schéma zapojení napájení MLVBT

4.2.3 Celkové zapojení modifikovaného LVBT pro rotátory Egis



Obr. 4. 4: Schéma zapojení modifikovaného LVBT k řízení rotátorů Egis.

4.3 Modifikace Firmwaru LVBT

Firmware LVBT byl psán v programovacím jazyce C. V jeho modifikaci byla změněna makra definující vstupně výstupní porty mikrokontroléru podle upraveného schéma zapojení.

Dále byly přidány funkce detekující instrukce určené rotátorům Egis, funkce řízení softwarově generovaného sériového portu a zpožďovací funkce použité při časování okamžiků čtení a zápisu dat na softwarově řízený sériový port pro pevně danou bitovou rychlost 9600Bd.

4.3.1 Modifikace maker definující porty mikrokontroléru

Definice maker se provádí standardní direktivou #define.

```
#define LCD_RS PORTB.4
#define TxD PORTB.5
#define RxD PORTB.0
```

V ukázce kódu je nadefinováno makro LCD_RS k portu RB4, jenž je používáno k rozlišení zápisu instrukce nebo dat do LCD displeje. Makra TxD a RxD definují piny použité k přenosu dat.

4.3.2 Základní nastavení portů mikrokontroléru pro softwarově řízený sériový port

Port RB5 je nastaven do výstupního režimu a RB0 do vstupního. Přerušení z externího zdroje je povoleno při sestupné hraně signálu.

```
void main(void)
{
...
  TRISB=0b.0100.0001;
  PORTB=0b.1010.1000;
  // UART EGIS
  INTEDG=0;           //přerušení na sestupnou hranu
  INTE=1;            //přerušení zapnuto
...
}
```

4.3.3 Funkce generující sériový port

K dodržení přesně definovaných časových okamžiků vhodných ke čtení nebo zápisu dat na porty mikrokontroléru jsou využívány programové zpožďovací funkce delay.

Délka jednoho bitu je dána bitovou rychlostí, která je pevně zvolena na 9600 Bd. Jednotka Baud definuje počet změn stavů přenosového média za jednu sekundu. Pro sériovou komunikaci prostřednictvím RS232C platí, že 1bit/s=1Bd. Obecně to platit nemusí, neboť při některých způsobech komunikace lze během jedné změny signálu přenést větší množství dat než jeden bit [7].

Pro délku časového intervalu jednoho bitu platí vztah:

$$t_{bit} \equiv \frac{1}{f_{bit}} = \frac{1}{9600} = 104,16\mu s, \quad (4.1)$$

Kde t_{bit} je délka trvání jednoho bitu a f_{bit} je bitová rychlost.

Požadované časové zpoždění je realizováno prostřednictvím funkce Delay se vstupní proměnou BAUD. Tělo funkce a její parametry ukazuje následující příklad.

```
#define DELAY_CONSTANT 13 //Software delay constant: 13 for 4MHz
#define BAUD 1

static void Delay(unsigned char uc) /* software delay uc * 100us */
{
    while (uc)
    {
        char uc2;
        uc--;
        for (uc2=0;uc2<DELAY_CONSTANT;uc2++)
        {}
    }
}
```

Do části programu, v níž jsou testovány jednotlivé typy řídicích příkazů, byla přidána podmínka detekce instrukcí patřících rotátorům Egis. Je zde využita nutnost psaní hesla před každým příkazem ve formátu "&*". Pokud přijde znak "&", je volána funkce TransmitToEgis, která zajistí přeposlání obdrženého příkazu přes softwarově generovaný sériový port. Tato podmínka je realizována uvnitř přepínače switch, jehož vstupní proměnou je první přijatý symbol uložený v poli _acLine. Do tohoto pole jsou ukládány všechny příchozí znaky. Přidaná část kódu vypadá následně.

```
switch (c)
{
    ...
    case '&':
    {
        TransmitToEgis();
    }
    ...
}
```

Funkce TransmitToEgis zajišťuje odesílání jednotlivých příkazů do řídicí jednotky rotátorů Egis. Tělo funkce ukazuje následující příklad.

```
static void TransmitToEgis(void)
{
    U8 u8Pos=0;
    ParseWhite(&u8Pos); // ignore white space
    while(u8Pos<=_u8LinePos)
    {
        SendCharToEgis(_acLine[u8Pos]);
        u8Pos++;
    }
    SendCharToEgis('\n');
}
```

Funkce SendCharToEgis odesílá jednotlivé byty od LSB po MSB skrze pin RB5 do řídicí jednotky rotátorů Egis jak ukazuje tento příklad.

```
static void SendCharToEgis(char c)
{
    char x=0;
    TxD=0; //start bit
    Delay(BAUD);
    while(x<8)
    {
        if((c & (1<<x)) !=0) {TxD=1;}
        else{TxD=0;}
        x++;
        Delay(BAUD);
    }
    TxD=1; //stopbit
    Delay(BAUD);
}
```

V situaci, kdy rotátor vysílá data ovladači antén, je vyvoláno přerušení, v němž je provedena potřebná obsluha a přeposlání dat skrze modul USART do počítače. Jsou zde použity dvě základní funkce. ReadCharFromEgis a SendCharToPc. Obsluhu přerušení zobrazuje následný příklad zdrojového kódu.

```

interrupt serverX(void)
{
    int_save_registers // W, STATUS (and PCLATH)
    char sv_FSR; sv_FSR = FSR; // if required
    ...
    if(INTF && INTE)
    {
        RxDPointer=0;
        INTF=0;
        ReadCharFromEgis();
        SendCharToPc();
    }

    FSR = sv_FSR; // if required
    int_restore_registers // W, STATUS (and PCLATH)
}

```

Funkce ReadCharFromEgis načítá jednotlivé znaky a ty ukládá do globálního pole RxDBuffer. Pokud jde za sebou více znaků najednou, jsou postupně ukládány do tohoto pole. Zápis této funkce znázorňuje následující ukázka kódu.

```

static void ReadCharFromEgis(void)
{
    char c=0, x=0;
    while(RxD==0)
    {
        while(x<8)
        {
            Delay(BAUD);
            if(RxD==1){c+=(1<<x);}
            x++;
        }
        RxDBuffer[RxDPointer]=c;
        RxDPointer++;
        Delay(2*BAUD);
    }
}

```

Funkce SendCharToPc posílá přijaté znaky od rotátoru Egis z pole RxDBuffer do počítače pomocí modulu USART. Její zápis je znázorněn níže.

```

static void SendCharToPc(void)
{
    char x=0;
    while(1)
    {
        if(TXIF)
        {
            TXREG=RxDBuffer[x];
            x++;
            if(x==RxDPointer){break;}
        }
    }
}

```

5 REALIZACE

5.1 DPS Modifikovaného LVBT

K vytvoření návrhů DPS byl použit software Eagle 6.3.0 Light Edition.

5.1.1 Prototyp I.

Výstup k vytvoření předlohy byl vygenerován pomocí CAM procesoru ve formátu postskript. Převod do pdf, který je vhodný pro tisk, byl proveden pomocí online konvertoru ps2pdf.com. Výsledná předloha však nebyla ve správném měřítku a došlo k celkovému zmenšení všech rozměrů. Tato chyba byla zprvu přisuzována online konvertoru. Dalším řešením byla odhalena chyba tisku.

5.1.2 Prototyp II.

Pro převod mezi formáty ps a pdf byl zvolen software Acrobat Distiller. Ten však způsoboval změnu tvaru měděných plošek ve vrstvě Pads. Zaznamenána byla přeměna u některých součástí s pady kruhového tvaru na čtvercové, s orientací vrcholy nahoru. Tím docházelo k nechtěným změnám v minimálních povolených odstupech (distance) mezi jednotlivými měděnými ploškami.

Nakonec byl pro převod z formátu vybrán software PDFCreator. U takto vzniklých předloh pro leptání stále docházelo k nechtěné změně měřítka. Takto byla detekována chyba nastavení tisku.

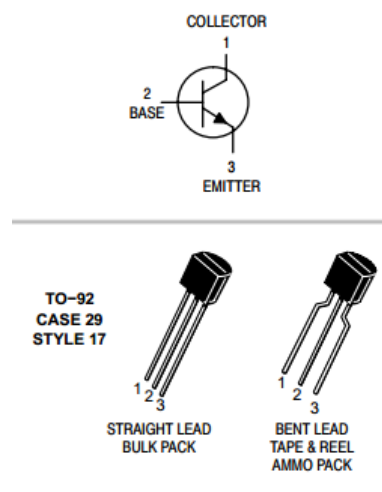
Při osazování již vyrobené DPS bylo zjištěno několik chyb.

- 1) Nekorektnost rozložení jednotlivých pinů tranzistorů u pouzdra TO92. Vývojové prostředí Eagle nabízí různou škálu uspořádání jednotlivých pinů u tohoto pouzdra, avšak podpora ze strany dodavatelů je omezená. V návrhu bylo použito rozložení s bází na kraji pouzdra. Většina tuzemských dodavatelů požadovaný tranzistor s takovýmto uspořádáním nenabízí. Klasické rozložení jednotlivých pinů zobrazuje Obr. 5.1.
- 2) Použití špatného druhu konektoru CANNON při návrhu. Místo samčího konektoru byl použit konektor typu samice. Ty však mají číslování pinů vůči sobě zrcadlově.

5.1.3 Prototyp III.

U prototypu III byly odstraněny všechny předchozí chyby návrhu.

Další vhodnou úpravou bylo přivedení všech měděných cest k pájeným pinům na dolní straně desky nikoliv horní, hlavně pak u konektorů, svorkovnic a patič. To sice vedlo ke zvětšení počtu propojek mezi vrstvami, ale zároveň ke zkvalitnění zapájení.



Obr. 5. 1: Klasické rozložení pinů tranzistoru BC547 v pouzdře TO92 [9].

5.2 Modifikace Firmwaru LVBT

Jako vývojové prostředí k návrhu řídicího programu mikrokontroléru byl použit software MPLAB IDE v8.88 od firmy Microchip. Jelikož původní kód LVBT využívá knihovnu math24f.h (floating point library), tedy knihovnu, která zajišťuje základní matematické operace s 24 bitovými čísly datového typu float, je použit kompilátor, který tuto knihovnu obsahuje. Knudsen Data CC5X.

5.2.1 Změna direktivy kódu

Původní firmware LVBT byl přeložen pomocí kompilátoru Knudsen Data Standard CC5X v3.1J, jenž je zpoplatněn.

Pro kompilaci modifikace LVBT byl využit volně dostupný kopilátor Knudsen Data CC5X jenž nepodporuje veškerou původní direktivu. Proto byl kód vhodně upraven pro překlad.

Mezi nepodporovanou direktivu patří zápis ve tvaru:

```
s16/=_fAzMul;
```

Úprava vypadá následně:

```
s16=s16/_fAzMul;
```

Dále byla upravena direktiva definující odesílání a přijímání jednotlivých znaků přes sériovou linku RS232C pro rotátory Egis. Zápis funkcí se tím vizuálně ztížil. Použitý tvar funkcí SendCharToEgis a ReadCharFromEgis je znázorněn níže.

```

static void SendCharToEgis(char c)
{
    TxD=0; //start bit
    Delay(BAUD);
    if((1&c)!=0){TxD=1;}
    else{TxD=0;}
    Delay(BAUD);
    if((2&c)!=0){TxD=1;}
    else{TxD=0;}
    Delay(BAUD);
    if((4&c)!=0){TxD=1;}
    else{TxD=0;}
    Delay(BAUD);
    if((8&c)!=0){TxD=1;}
    else{TxD=0;}
    Delay(BAUD);
    if((16&c)!=0){TxD=1;}
    else{TxD=0;}
    Delay(BAUD);
    if((32&c)!=0){TxD=1;}
    else{TxD=0;}
    Delay(BAUD);
    if((64&c)!=0){TxD=1;}
    else{TxD=0;}
    Delay(BAUD);
    if((128&c)!=0){TxD=1;}
    else{TxD=0;}
    Delay(BAUD);
    TxD=1; //stopbit
    Delay(BAUD);
}

static void ReadCharFromEgis(void)
{
    char c;
    while(RxD==0)
    {
        c=0;
        Delay(BAUD);
        if(RxD==1){c+=1;}
        Delay(BAUD);
        if(RxD==1){c+=2;}
        Delay(BAUD);
        if(RxD==1){c+=4;}
        Delay(BAUD);
        if(RxD==1){c+=8;}
        Delay(BAUD);
        if(RxD==1){c+=16;}
        Delay(BAUD);
        if(RxD==1){c+=32;}
        Delay(BAUD);
        if(RxD==1){c+=64;}
        Delay(BAUD);
        if(RxD==1){c+=128;}
        RxDBuffer[RxDPointer]=c;
        RxDPointer++;
        Delay(BAUD);
    }
}

```

5.2.2 Optimalizace paměti

Paměť programu je rozdělena na 4 stránky. Programátor může pomocí direktivy `#pragma codepage` definovat do které ze čtyř stránek paměti (číslováno 0,1,2,3) má být následující kód zapsán. Příklad takového zápisu ukazuje následující příklad.

```
#pragma codepage 2
```

Vlivem předchozích změn direktivy nebylo původní rozdělení LVBT do paměťových bloků optimální a muselo být upraveno. Výsledné využití paměti ukazuje částečný výpis z výstupu kompilátoru.

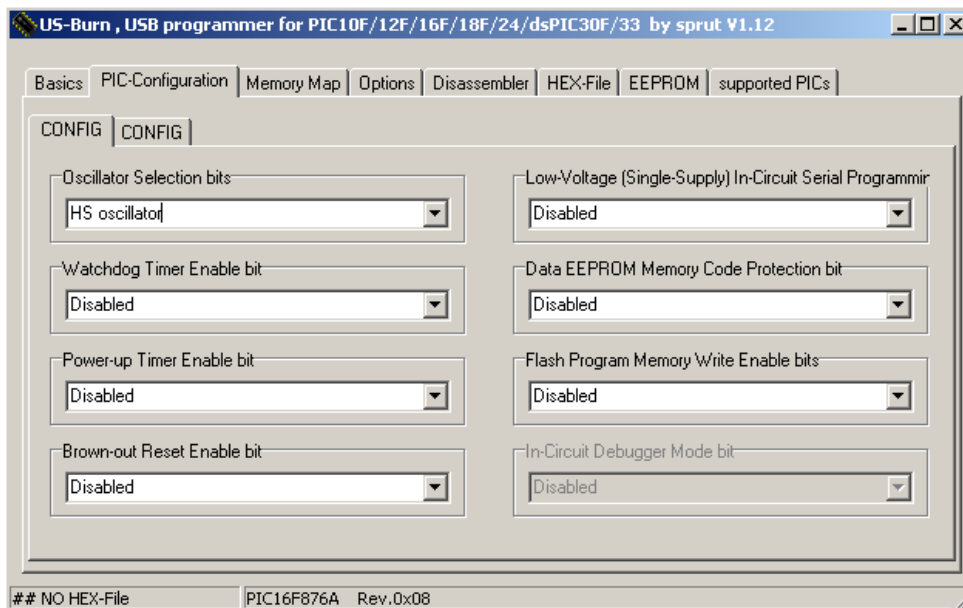
```
Codepage 0 has 1847 word(s) : 90 %
Codepage 1 has 1194 word(s) : 58 %
Codepage 2 has 1855 word(s) : 90 %
Codepage 3 has 1360 word(s) : 66 %
Total of 6256 code words (76 %)
* Estimated CODE SIZE of full optimization: 4873 code words (-22 %)
```

5.3 Zprovoznění Modifikovaného LVBT

Zprovoznění modifikace LVBT obnášelo další úpravy firmwaru za účelem zajištění spolehlivé funkce.

5.3.1 Naprogramování mikrokontroléru PIC16F876

K naprogramování mikrokontroléru byl použit programátor PIC-Brenner 8 ovládaný pomocí softwaru USBurn112. Jelikož kód neobsahuje definici konfiguračního slova (configuration word) musí být toto nastavení provedeno manuálně (USBurn112) před nahráním programu do mikrokontroléru. Požadované nastavení konfiguračního slova pro tuto aplikaci ukazuje Obr. 5.2.



Obr. 5. 2: Nastavení konfiguračního slova pro aplikaci modifikovaného firmwaru LVBT.

5.3.2 Inicializace LCD displeje

Při zprovoznění vykazoval MLVBT nevhodnou inicializaci LCD displeje. Po zapnutí byl řadič displeje nastaven pouze pro ovládání jediného řádku. Po následném resetu mikrokontroléru PIC16F876 proběhla inicializace v pořádku a řadič LCD displeje již obsluhoval oba řádky.

Špatná inicializace LCD displeje při startu byla způsobena různým náběhem napájecího napětí u jednotlivých komponent LVBT. V okamžiku, kdy mikrokontrolér již vysílal instrukce řadiči LCD displeje o jeho nastavení, řadič nebyl připraven tyto instrukce přijímat.

Tato chyba byla odstraněna vytvořením 100ms zpožděním před začátkem samotné inicializace LCD displeje. Toto řešení zobrazuje následující příklad zdrojového kódu.

```
...
Delay(250);
LCDInit();
```

5.3.3 Sériová komunikace pro rotátory Egis

K odladění správné funkce rozhraní RS232C pro komunikaci s rotátory Egis, bylo využito připojení tohoto portu k jednomu terminálu počítače za současného připojení USB k druhému terminálu. Vzájemným přeposíláním dat mezi terminály bylo dosaženo zpětné vazby potřebné k detekování chybné funkce rozhraní.

I přes přesné nastavení časových okamžiků čtení a zápisu dat ve vývojovém prostředí MPLAB IDE docházelo v reálném přenosu k nesprávné funkci.

K zajištění spolehlivosti při odesílání dat směrem k rotátoru, byly provedeny následující úpravy ve funkci TransmitToEgis, kterou zobrazuje ukázka zdrojového kódu níže.

- 1) vypnutí všech přerušení během komunikace,
- 2) úprava podmínky pro rozsah čtení z pole _acLine
- 3) přidání CR znaku na konci odesílaných dat.

```
static void TransmitToEgis(void)
{
    GIE=0;
    U8 u8Pos=0;
    ParseWhite(&u8Pos); // ignore white space
    while(u8Pos<_u8LinePos)
    {
        SendCharToEgis(_acLine[u8Pos]);
        u8Pos++;
    }
    SendCharToEgis('\r');
    SendCharToEgis('\n');
    GIE=1;
}
```

Potřebné zpoždění pro odesílání dat bylo změněno prostřednictvím makra DELAY_CONSTANT4, jak ukazuje kód níže.

```
#define DELAY_CONSTANT4 7
```

Pro tuto časovou konstantu vykazuje simulátor vývojového prostředí MPLAB IDE zpoždění 93 μ s. S uvažováním podmínkových instrukcí ve funkci SendCharToEgis je délka trvání jednoho bitu podle simulátoru 100 μ s.

Po těchto úpravách již vykazoval MLVBT spolehlivý přenos dat ve směru k rotátoru Egis.

Odladění příjmu dat MLVBT se za pomoci simulátoru nezdařilo. Proto byl vytvořen pomocný signál na vstupně výstupním portu mikrokontroléru, který určoval přesný okamžik čtení dat. Pomocí osciloskopu byla měřena přijímaná data vůči okamžikům čtení dat. Postupně bylo dosaženo přesného časování za pomoci měření.

Podoba výsledných funkcí je následující.

```
static void ReadCharFromEgis (void)
{
    char c;
    while (RxD==0)
    {
        c=0;
        PORTC.5=1;
        if (RxDPointer==0) {Delay2 (BAUD); }
        else {Delay2 (BAUD); }
        PORTC.5=0;
        if (RxD==1) {c+=1; }
        Delay2 (BAUD);
        PORTC.5=1;
        if (RxD==1) {c+=2; }
        Delay2 (BAUD);
        PORTC.5=0;
        if (RxD==1) {c+=4; }
        Delay2 (BAUD);
        PORTC.5=1;
        if (RxD==1) {c+=8; }
        Delay2 (BAUD);
        PORTC.5=0;
        if (RxD==1) {c+=16; }
        Delay2 (BAUD);
        PORTC.5=1;
        if (RxD==1) {c+=32; }
        Delay2 (BAUD);
        PORTC.5=0;
        if (RxD==1) {c+=64; }
        Delay2 (BAUD);
        PORTC.5=1;
        if (RxD==1) {c+=128; }
        RxDBuffer [RxDPointer]=c;
        RxDPointer++;
        Delay3 (BAUD);
    }
}
```

```

static void SendCharToPc(void)
{
    char x=0;
    while(x<RxDPPointer)
    {
        if(TXIF)
        {
            TXREG=RxDBuffer[x];
            RxDBuffer[x]=0;
            x++;
        }
    }
}

```

Vytvoření pomocného signálu na portu RC5 bylo poté odstraněno a nahrazeno instrukcí se stejným počtem strojových cyklů. Po překladu kompilátorem však došlo k chybě v časování čtecích okamžiků a funkce rozhraní nebyla spolehlivá. Z toho důvodu bylo od nahrazení pomocného signálu jinými instrukcemi upuštěno.

6 ZÁVĚR

Autonomní ovladač antén v azimutu a elevaci vychází z veřejně dostupného rozhraní Las Vegas Boulevard Trackeru určeného k řízení rotátorů Yaesu. Byla provedena modifikace zapojení a řídicího programu mikrokontroléru, jejímž výsledkem je vytvoření nového rozhraní RS232C pro komunikaci s rotátory Egis. Toto rozhraní je řízeno zcela programově bez použití komunikačního modulu USART. Mikrokontrolérem je prováděno usměrnění datového toku podle potřebného cílového umístění přijímaných instrukcí. Úpravami firmwaru a vytvořením komunikačního rozhraní RS232C na DPS LVBT bylo dosaženo efektivního využití integrovaného převodníku USB/UART DLP-USB232M.

Nově vzniklé rozhraní MLVBT je možné využít k automatizovanému ovládání rotátorů Yaesu a Egis prostřednictvím USB portu počítače.

Autonomní ovladač antén v azimutu a elevaci byl úspěšně realizován a testován.

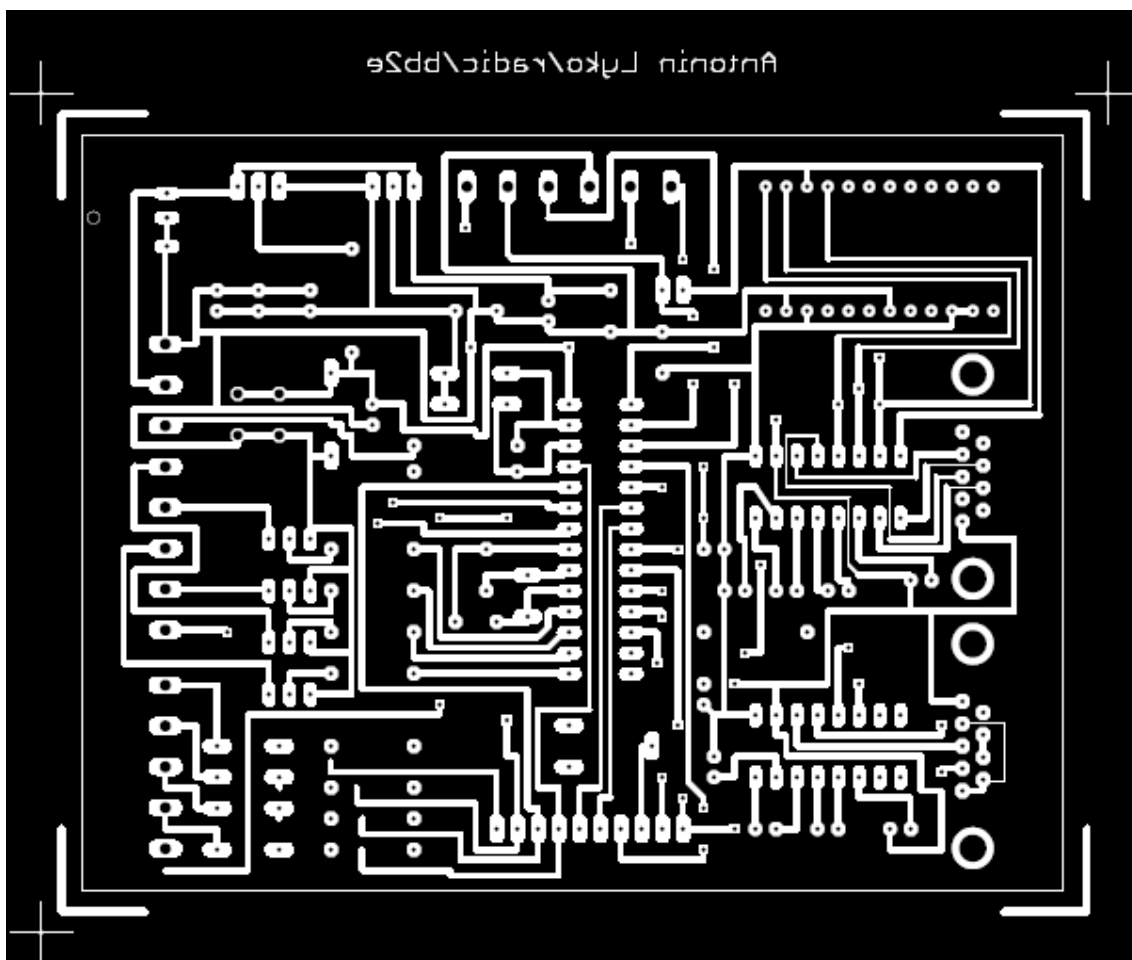
LITERATURA

- [1] YAESU. *G-5500 Operating Manual*. 1998. Dostupné z: yaesu.com
- [2] G6LVB. *The Las Vegas Boulevard Tracker* [online]. 23.11.2003 [cit. 2013-5-31]. Dostupné z: <http://www.g6lvb.com/Articles/LVBTracker/index.htm>
- [3] YAESU. *GS-232A Computer Interface Manual*. 2000. Dostupné z: yaesu.com
- [4] EGIS. [online]. 2003-2012 [cit. 2012-12-13]. Dostupné z: http://www.egis.org/aroto_us.html
- [5] EGIS. *Communication Interface PROFITRACKER*.
- [6] FTDI. *DLP-USB232M User's Manual* [online]. 2002 [cit. 2012-12-14]. Dostupné z: <http://www.ftdichip.com/>
- [7] HW.CZ. *Sériová linka RS-232l* [online]. 2005 [cit. 2012-12-14]. Dostupné z: <http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>
- [8] GME. *BC547A* [online]. 1990-2012 [cit. 2012-12-14]. Dostupné z: gme.cz

SEZNAM PŘÍLOH

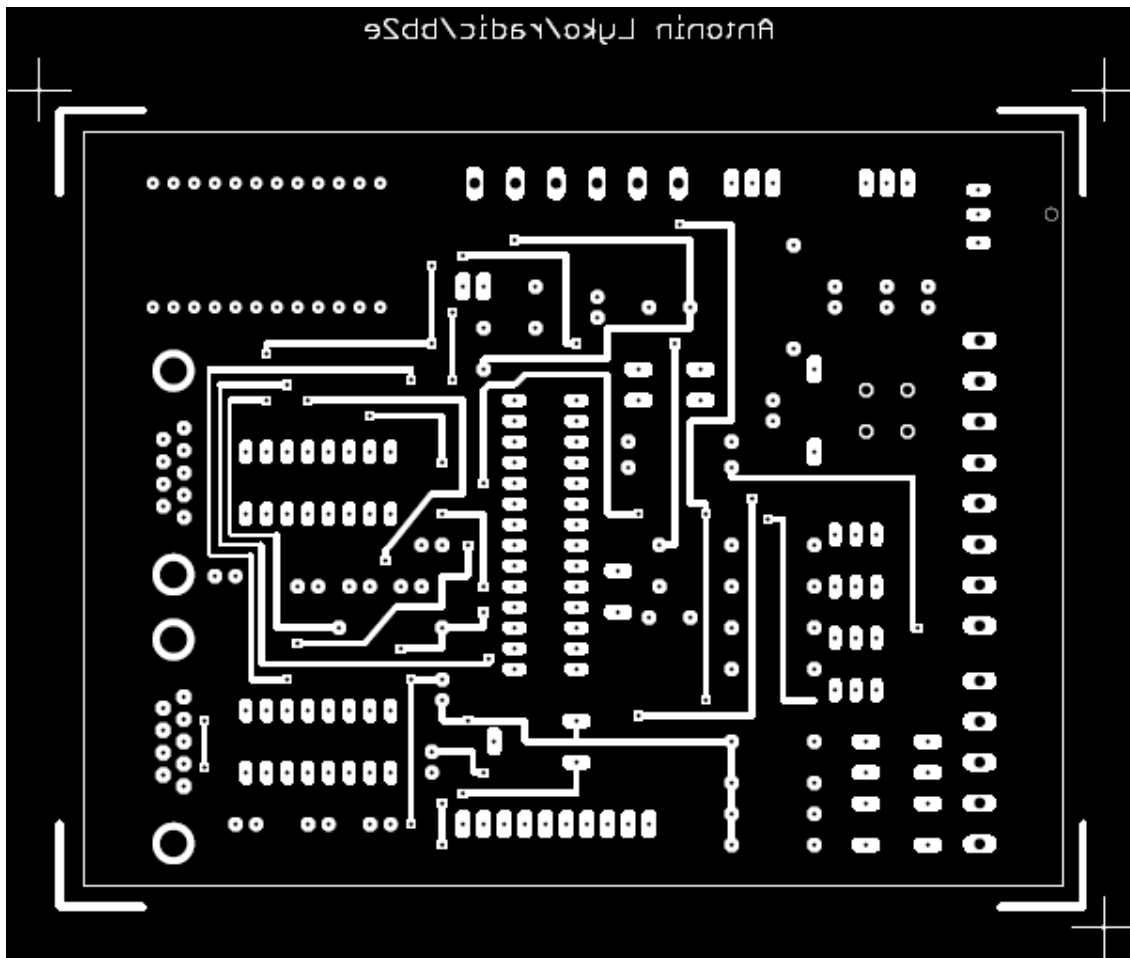
A	návrh zařízení	34
A.1	Obvodové zapojení	34
A.2	Deska plošného spoje – bottom (strana spojů)	35
A.3	Deska plošného spoje – top (strana součástek).....	36
B	seznam součástek	37

A.2 Deska plošného spoje – bottom (strana spojů)



Rozměry desky 120 x 93 [mm], měřítko M1:1

A.3 Deska plošného spoje – top (strana součástek)



Rozměry desky 120 x 93 [mm], měřítko M1:1

B SEZNAM SOUČÁSTEK

Označení	Hodnota	Pouzdro	Popis
C1	15pF	5mm	Keramický kondenzátor
C2	15pF	5mm	Keramický kondenzátor
C3	100nF	5mm	Keramický kondenzátor
C4	100nF	5mm	Keramický kondenzátor
C5	1uF	2,54mm	Elektrolytický kondenzátor
C6	1uF	2,54mm	Elektrolytický kondenzátor
C7	1uF	2,54mm	Elektrolytický kondenzátor
C8	1uF	2,54mm	Elektrolytický kondenzátor
C9	1uF	2,54mm	Elektrolytický kondenzátor
C10	10uF	2,54mm	Elektrolytický kondenzátor
C11	100nF	2,54mm	Elektrolytický kondenzátor
C12	10uF	2,54mm	Elektrolytický kondenzátor
C13	10uF	2,54mm	Elektrolytický kondenzátor
C14	100nF	5mm	Keramický kondenzátor
C15	1uF	2,54mm	Elektrolytický kondenzátor
C16	1uF	2,54mm	Elektrolytický kondenzátor
C17	1uF	2,54mm	Elektrolytický kondenzátor
C18	1uF	2,54mm	Elektrolytický kondenzátor
C19	1uF	2,54mm	Elektrolytický kondenzátor
C20	1uF	2,54mm	Elektrolytický kondenzátor
D1	1N4148	DO35	Dioda
D2	1N4148	DO35	Dioda
D3	BZX55	DO35	Zenerova dioda
D4	1N4148	DO35	Dioda
D5	1N4148	DO35	Dioda
D6	1N4148	DO35	Dioda
D7	1N4148	DO35	Dioda
IC1	PIC16F876	DIL-28	Mikrokontrolér
IC2	MAX232	DIL-16	Integrovaný obvod
IC3	MAX232	DIL-16	Integrovaný obvod
IC4	DLP-USB232M	DIL-24	
J2	6	5mm	svorkovnice
J4	5	5mm	svorkovnice
JP1	2	2,54mm	kolík do DPS
JP2	3	2,54mm	kolík do DPS

LCD	MC1602E-TRV	2,54mm	LCD1602, HD44780
P1	10K	5x10mm	Uhlíkový trimr
Q2	4MHz	5mm	Krystaly HC49U
R1	10K	10,14mm	Metalizovaný rezistor
R2	10K	10,14mm	Metalizovaný rezistor
R3	10K	10,14mm	Metalizovaný rezistor
R4	10K	10,14mm	Metalizovaný rezistor
R5	1K	12,7mm	Metalizovaný rezistor
R6	1K	12,7mm	Metalizovaný rezistor
R7	1K	12,7mm	Metalizovaný rezistor
R8	4K7	12,7mm	Metalizovaný rezistor
R13	100k	10,14mm	Metalizovaný rezistor
R14	100K	10,14mm	Metalizovaný rezistor
R15	100K	10,14mm	Metalizovaný rezistor
R16	100K	10,14mm	Metalizovaný rezistor
S1	DTE6K	5x5mm	Mikrospínače do DPS
T1	BC547	TO92	Bipolární tranzistor NPN
T2	BC547	TO92	Bipolární tranzistor NPN
T3	BC547	TO92	Bipolární tranzistor NPN
T4	BC547	TO92	Bipolární tranzistor NPN
U2	786TV	TO220	Lineární regulátor napětí
X1	9-Samice	CANNON	Konektor do DPS
X2	9-Samec	CANNON	Konektor do DPS
YEASU	7	5mm	svorkovnice