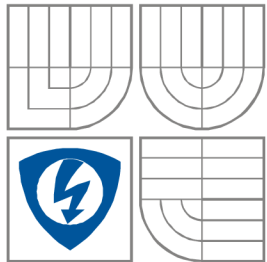


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

MĚŘENÍ SMĚROVÝCH CHARAKTERISTIK ANTÉN

MEASUREMENT OF THE ANTENNA DIRECTIONAL PATTERN

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN DIVÍN

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. ZDENĚK NOVÁČEK, CSc.

ABSTRAKT

Tato diplomová práce se zabývá tématem měření antén ve vzdálené zóně. Zvláště pak automatizaci tohoto měření, snímači z optické myši, magnetický n-kodér jako snímač polohy a ovládací jednotku tohoto pracoviště, která umí komunikovat s PC pomocí sběrnice USB. Je zde popsáno zhotovení ovládacího programu pro PC.

KLÍČOVÁ SLOVA

Měření, směrové charakteristiky antén, měření ve vzdálené zóně, snímač polohy, automatizované měření, snímač v počítačové myši, magnetický n-kodér.

ABSTRACT

This Master's Thesis is dealing with measuring antennas in far field. Especially with automation this measure, direction sensors made by optical mouse, Integrated Hall ICs for Linear and Off-Axis Rotary Motion Detection and remote unit this workplace, which can made communication with PC by USB. It describes the making control program for the PC.

KEYWORDS

Measuring, antenna directional pattern, antennas far field measuring, direction sensor, automatized working place, sensor in computer mouse, Integrated Hall ICs for Linear and Off-Axis Rotary Motion Detection.

DIVÍN, J. *Měření směrových charakteristik antén*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií., 2011. 56 s. 8 s. příloh. Diplomová práce. Vedoucí práce: doc. ing. Zdeněk Nováček, CSc.

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Měření směrových charakteristik antén jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu 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 20. května 2011

.....
podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce doc. Ing. Zdeňkovi Nováčkovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 20. května 2011

.....
podpis autora

OBSAH

Seznam obrázků	viii
Úvod	1
1 Měření antén ve vzdálené zóně	2
1.1 Metodika měření	2
1.2 Parametry antén	2
1.2.1 Úhlová šířka hlavního laloku	2
1.2.2 Činitel směrovosti	3
1.2.3 Zisk antény	3
2 Automatizované měření	4
2.1 Jednoduché automatizované pracoviště	4
2.2 Použití PC na sběr dat	4
2.3 Profesionální měřicí pracoviště (použito v této práci)	5
3 Snímač polohy	6
3.1 Optický snímač v myši – obecná definice	6
3.1.1 Optický snímač v myši – konkrétní použitý případ	7
3.1.2 Formát výstupních dat ze snímače polohy z optické myši	8
3.1.3 Zhodnocení optického snímače	8
3.2 Magnetický n-kodér	9
3.2.1 Použitý magnetický n-kodér	9
3.2.2 Formát výstupních dat z magnetického n-kodéru	9
4 Promax Prolink – 1B	12
4.1 Popis přístroje	12
4.2 Komunikační protokol	13
4.3 Nastavovací příkazy	13
4.4 Dotazovací příkazy	14
5 Konstrukční řešení automatizovaného pracoviště	15
5.1 Zapojení pracoviště	15
5.2 Původní konstrukční provedení ovladače pro optický snímač	16

5.3	Konstrukční provedení ovladače pro magnetický snímač	19
5.3.1	Schéma zapojení ovladače	19
5.3.2	Výkonová část.....	20
5.3.3	Ovládání relé.....	20
5.3.4	Indikace.....	20
5.3.5	Manuální ovládání	20
5.3.6	Napájení Ovladače.....	20
5.3.7	Mechanická konstrukce ovladače točny	21
5.4	Softwarové řešení ovladače pro optický snímač.....	22
5.4.1	Nastavení parametrů pro chod programu.....	22
5.4.2	Obsluha příjmu bajtu ze snímače polohy	24
5.4.3	Hlavní smyčka programu.....	25
5.4.4	Podprogram pro odeslání bajtu do snímače	25
5.5	Softwarové řešení ovladače pro magnetický n-kodér	26
5.5.1	Nastavení parametrů pro chod programu.....	26
5.5.2	Hlavní smyčka programu.....	26
5.5.3	Obsluha snímače	27
5.5.4	Obsluha příjmu znaku z PC a z měřicího přístroje	28
5.5.5	Funkce kalibrace	28
6	Ovládací program AMAP	29
6.1	Seznámení se s programem.....	29
6.2	Popis funkcí programu	29
6.3	Ovládání programu	30
6.3.1	Spuštění programu	30
6.3.2	Kalibrace a nastavování	30
6.3.3	Automatizované měření	30
6.3.4	Výstupy	31
6.3.5	Vystředování grafů.....	32
7	Programátorská příručka	33
7.1	Struktura NamerenyBod	33
7.2	Funkce timer_Tick	33
7.3	Funkce SendSerial	34
7.4	Funkce OnReceived	35

7.5	Funkce příjemChar	35
7.6	Funkce příjemPolohy	37
7.7	Funkce btStart_Click	37
7.8	Funkce Measure	38
7.9	Funkce pro nastavení frekvence na Prolink-1B	41
7.10	Funkce uložitToolStripMenuItem_Click	41
8	Závěr	43
	LITERATURA	44
	SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK	45
	Seznam příloh	46
	PŘÍLOHY	47

SEZNAM OBRÁZKŮ

Obr. 1.1:	Umístění měřené antény (převzato z [1]).....	2
Obr. 1.2:	Šířka hlavního laloku pro pokles o 3dB a) v polárních b) v kartézských souřadnicích (převzato z [1]).	3
Obr. 2.1:	Zapojení automatizovaného pracoviště pro případ s PC, ale bez snímače polohy.	4
Obr. 2.2:	Profesionální měřicí pracoviště (použito v této práci).	5
Obr. 3.1:	Ukázka snímače polohy v optické myši.....	6
Obr. 3.2:	Použitý snímač polohy.	7
Obr. 3.3:	Strana součástek a strana spojů použitého snímače.	7
Obr. 3.4:	Magnetické kodéry. a) lineární magnetický kodér. b) magnetický rotační n-kodér (převzato z [5]).....	9
Obr. 3.5:	Vzájemné poměry mezi výstupními signály magnetického n-kodéru (převzato z [5]).....	10
Obr. 3.6:	Strana spojů plošného spoje snímače polohy točny.....	10
Obr. 3.7:	Rozmístění součástek na straně spojů a straně součástek.	10
Obr. 3.8:	Detail snímače a jeho umístění na anténním rotátoru.	11
Obr. 4.1:	Promax Prolink 1B (převzato z [7]).....	12
Obr. 5.1:	Blokové schéma navrhovaného ovladače.	15
Obr. 5.2:	Schéma zapojení navrhovaného ovladače.	17
Obr. 5.3:	Zapojení ovladače točny - pohled ze strany spojů.	18
Obr. 5.4:	Zapojení ovladače točny - pohled ze strany součástek.	18
Obr. 5.5:	Obrázek plošného spoje – strana spojů.....	19
Obr. 5.6:	Čelní panel krabičky ovladače.	21
Obr. 5.7:	Zadní panel krabičky ovladače.	22
Obr. 5.8:	Pohled do krabičky ovladače.	22
Obr. 5.9:	Vývojový diagram programu mikroprocesoru - optický snímač.	24
Obr. 5.10:	Přenos ze snímače polohy do ovladače točny[4].	24
Obr. 5.11:	Přenos z ovladače točny do snímače polohy (převzato z [4]).....	25
Obr. 5.12:	Vývojový diagram programu mikroprocesoru magnetický n-kodér.	27
Obr. 6.1:	Vzhled programu AMAP verze 2.0.	29
Obr. 6.2:	Graf s naměřenými daty a graf s normovanou směrovou charakteristikou. 31	

Obr. 6.3: Normovaná směrová charakteristika v polárním grafu.....	31
Obr. 6.4: Vystředování grafu.....	32

ÚVOD

Tato práce se zabývá zautomatizováním měření antén ve vzdálené zóně. Snahou je zjednodušit zaznamenávání hodnot úhlu natočení měřené antény tak, aby byly zaznamenávány automaticky a nebylo nutno je zaznamenávat ručně. Dnes hojně používanou a nejefektivnější metodou automatizovaného měření je záznam hodnot z pracoviště do počítače, kde se již údaje pohodlně zobrazí v přehledných grafech a jdou dopočteny parametry antény.

V tomto textu je věnována pozornost samotné teorii měření antén ve vzdálené zóně. Především je rozebírána možnost využití dvou různých snímačů polohy natočení antény. A to optické myši a magnetického n-kodéru. U obou možností jsou i popsány realizace ovladače pro řízení otáčení točny, snímání úhlu natočení, komunikaci s měřícím přístrojem a komunikace s PC.

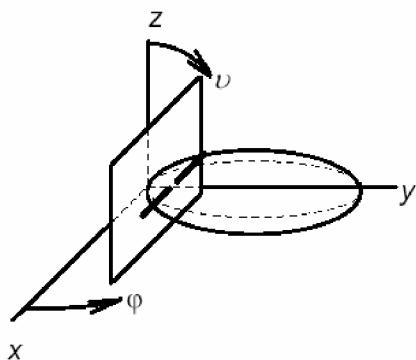
V závěru práce je detailně popsána konstrukce ovládací krabičky tohoto měřícího pracoviště a je zde popsáno sestavení ovládacího programu pro PC a jeho obsluha. Tento ovládací program je i detailně popsán v programátorské příručce.

1 MĚŘENÍ ANTÉN VE VZDÁLENÉ ZÓNĚ

V následujícím textu bude informativně vysvětlena problematika měření antén ve vzdálené zóně.

1.1 Metodika měření

Jeden z nejdůležitějších parametrů určující mnoho dalších vlastností antény je její směrová charakteristika. Měřením ji můžeme získat tak, že ve stálé vzdálenosti r od měřené antény, buzené generátorem se stálou úrovní, měříme velikost (modul) intenzity pole pro jednotlivé směry určené úhly ν a φ (lze vidět na obr. 1.1.) Tuto charakteristiku lze zobrazit v kartézských nebo polárních souřadnicích.



Obr. 1.1: Umístění měřené antény (převzato z [1]).

Při měření je lhostejné, zda se anténa měří jako vysílací nebo přijímací. Relativní směrové charakteristiky totiž nezávisí na účelu antény. Pro náš případ budeme uvažovat, že měřená anténa je přijímací.

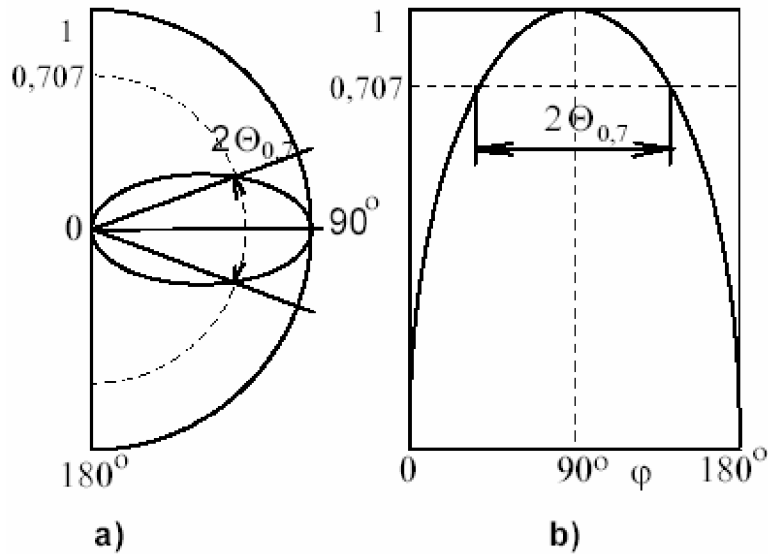
Měří-li se zkoumaná anténa jako přijímací, vytvoříme si potřebné pole zářením pomocné antény, napájené generátorem požadovaného kmitočtu a s dostatečným výkonem. Tato anténa musí vyzařovat vlnu s požadovanou polarizací. Na její směrové vyzařovací charakteristice již tak nezáleží, ale je potřeba aby intenzita pole v prostoru, kde se otáčí měřená anténa, byla co nejvíce homogenní.

Na vysílací straně se udržuje stálý proud a měří se indukované napětí v závislosti na natočení antény. Chyba výsledků může vzniknout hlavně odrazem od objektů v okolí nebo od země. Proto je důležité, aby se v okolí obou antén nevyskytovaly žádné předměty ani osoby. Ideální místo pro měření směrových charakteristik je volné prostranství.

1.2 Parametry antén

1.2.1 Úhlová šířka hlavního laloku

Úhlová šířka laloku je úhel $2\Theta_{0,7}$, který leží mezi body směrové charakteristiky, ve kterých je záření antény nižší o 3dB vůči maximu (viz. obr. 1.2). Číselné hodnoty šířky hlavních laloků pro roviny E a H jsou obecně různé.



Obr. 1.2: Šířka hlavního laloku pro pokles o 3dB a) v polárních b) v kartézských souřadnicích (převzato z [1]).

1.2.2 Činitel směrovosti

Činitel směrovosti D_{\max} udává, kolikrát větší výkon je vyzářen ve směru maxima záření ve srovnání s všesměrovým izotropním zářičem napájený stejným výkonem.

Přímé určení D_{\max} je celkem složité. Pokud ale má anténa výrazný hlavní lalok a nevelké laloky postranní tak se dá přibližně určit podle vztahu

$$D_{\max} \cong \frac{35000}{2\Theta_E \cdot 2\Theta_H}, [-, ^\circ, ^\circ] \quad (1.1)$$

kde $2\Theta_E$ je šířka laloku v rovině elektrického pole a $2\Theta_H$ je šířka laloku v rovině pole magnetického.

1.2.3 Zisk antény

Zisk antény G_{\max} je decibelovým vyjádřením D_{\max} . Udává se v jednotkách dBi z toho důvodu, aby bylo patrné, že je vztahen k izotropnímu zářiči.

2 AUTOMATIZOVANÉ MĚŘENÍ

V této kapitole budou uvedeny postupy, kterými lze zautomatizovat měření antén ve vzdáleném poli.

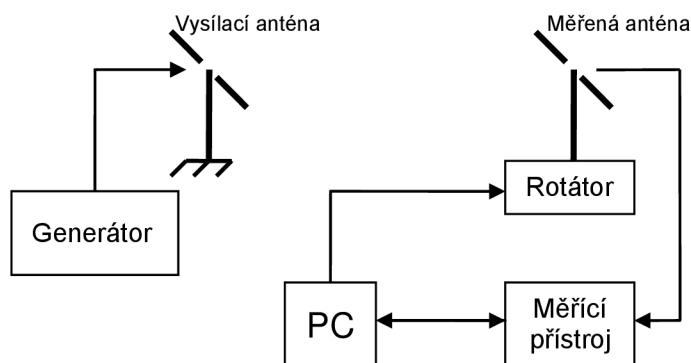
2.1 Jednoduché automatizované pracoviště

Automatizované pracoviště pro měření směrových charakteristik antény lze v nejjednodušším případě postavit na souřadnicovém zapisovači, kterému by osa x vzrůstala lineárně s časem a to přesně v době, kdy je rotátor antény v pohybu a na osu y by se zapisovala hodnota střídavého napětí indukovaná v anténě. Otočení antény by muselo být provedeno o přesně známý úhel. Nastavení této metody je komplikované, odečítání hodnot nepohodlné a přesnost získaných výsledků je nízká.

2.2 Použití PC na sběr dat

Zlepšením oproti odchozí metodě je výměna souřadnicového zapisovače za PC (viz. obr. 2.1), který dává rotátoru povel k otáčení. Zlepšením je také přidání mikrovoltmetru, který měří úroveň střídavého napětí indukovaného na anténě. Podmínkou fungování této metody je znalost o jaký známý úhel se za přesný známý čas otočí rotátor, z čehož lze vypočítat směr natočení k jednotlivým naměřeným bodům. Tuto metodu lze vylepšit použitím koncového vypínače, který po přednastavené hodnotě (např. 360°) zastaví rotátor i načítání hodnot z měřicího přístroje a následném dopočítání směru natočení.

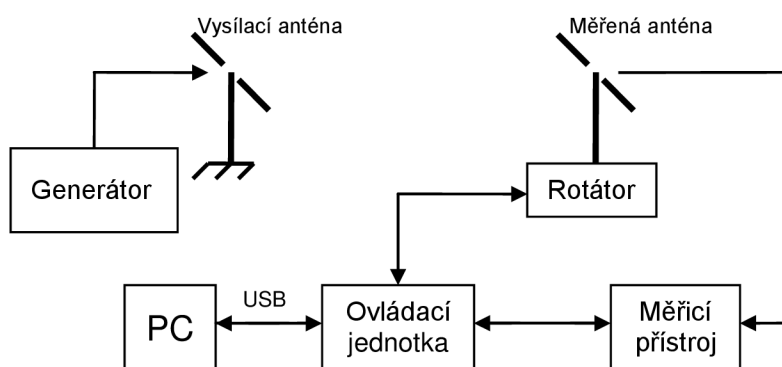
Tato metoda má výhodu počítačového zpracování, kdy je možno si v různých grafech zobrazit vyzařovací charakteristiku i jednoduše naprogramovat rutinní výpočty a zobrazit je ihned po doměření. Ukládání a další zpracování výsledků je tak již značně ulehčeno. K profesionálnímu použití chybí této metodě přesné odečítání směru natočení měřené antény.



Obr. 2.1: Zapojení automatizovaného pracoviště pro případ s PC, ale bez snímače polohy.

2.3 Profesionální měřicí pracoviště (použito v této práci)

Jestliže je potřeba přesnějších výsledků a měřicí pracoviště, na kterém i po více měřeních nebude potřeba znova nastavovat polohu točny, je nutno mít na točně antény snímání polohy. Po instalaci nové antény na točnu se provede kalibrace natočeného směru. Tento výchozí směr zůstane pro všechna další měření stejný a není již nutno jakkoliv manuálně zasahovat do měřicího pracoviště, jelikož jeho ovládání se provádí z PC. Tato možnost postavení automatizovaného pracoviště je blokově navrhována na Obr. 2.2.



Obr. 2.2: Profesionální měřicí pracoviště (použito v této práci).

Tato metoda má veškeré výhody plynoucí z použití PC a to vykreslování různých grafů z naměřených hodnot podle přání uživatele, provádění výpočtů, které uživateli dokážou ihned prozradit kvalitu měřené antény, export naměřených dat, tisk naměřených výsledků i ovládání pracoviště ze vzdáleného PC.

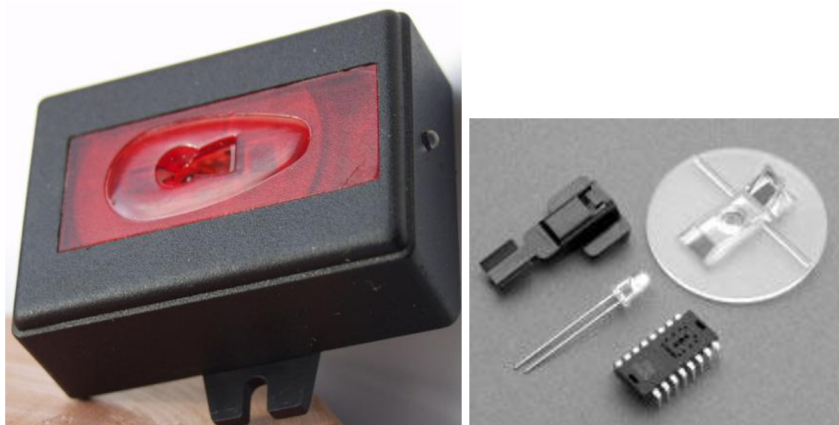
3 SNÍMAČ POLOHY

V této kapitole bude pojednáno o použitém typu snímače a bude popsán konkrétní použitý snímač polohy.

3.1 Optický snímač v myši – obecná definice

Největší výhodou tohoto snímače je snadná montáž a bezkontaktní snímání (viz. obr. 3.4). Tento snímač umí snímat a rozeznávat pohyb ve všech směrech (pro potřebu snímání natočení antény postačí pouze jeden směr). Jeho další výhodou je snadná dostupnost, jelikož optickou myš s konektorem PS/2 má většina počítačů (u novějších počítačů nahrazováno rozhraním USB). Citlivost tohoto snímače je plně dostačující (rozlišení na desetinu milimetru).

Vnitřní zapojení u optických myši bývá řešeno dvěma způsoby. První používá dvou vnitřních integrovaných obvodů, kde jeden je určen pro snímač, z něhož vycházejí pro každý směr dva signály shodné jako z optotranzistorů u kuličkové myši a druhý IO tyto signály spočítá a společně s údaji o stisknutých tlačítkách je převede na dané pakety a odešle protokolem PS/2 do počítače. Druhý, dnes častější způsob, používá jediný integrovaný obvod v myši, který zajistí vše od snímání polohy a zjišťování stavu stisknutých tlačítek, až po odeslání dat přes protokol PS/2.



Obr. 3.1: Ukázka snímače polohy v optické myši.

PS/2 myši používají synchronní obousměrný protokol master-slave, který umožňuje připojit pouze jediné slave zařízení (v našem případě myš). Důležitou vlastností je, že hodiny generuje vždy slave. Master (PC nebo v našem případě MCU) je ovšem nadřazen – pomocí clk signálu povoluje, zakazuje nebo přerušuje přenos dat. Hodinový kmitočet se pohybuje v rozmezí cca 10–16 kHz. Nepříjemností tohoto protokolu je, že čtení a zápis bitu na sběrnici není se sestupnou hranou hodin, ale přesně uprostřed L úrovně hodin. To odpovídá přibližně 15-25 μ s po sestupné hraně hodin.

Po připojení k napájení by myš měla provést "selftest" a v případě úspěchu vrátit hodnoty 0xAA (selftest ok) a 0x00 (mouse ID). ID 0x00 značí standardní 2/3 tlačítkovou myš. V rámci zpětné kompatibility se po resetování musí všechny myši nastavit do tohoto režimu. Různé rozšíření, jako třeba známé "kolečko", se pak zapínají speciální sekvencí příkazů [4]. Při samotné komunikaci posílá myš do počítače tříbitové pakety, kde prvním bajtu jsou

především příznaky ke stisknutým tlačítkům. Další dva bajty nesou údaje o tom, o kolik se myš od posledního odeslání dat posunula ve směru x a y v rozmezí hodnot -256 – 255.

3.1.1 Optický snímač v myši – konkrétní použitý případ

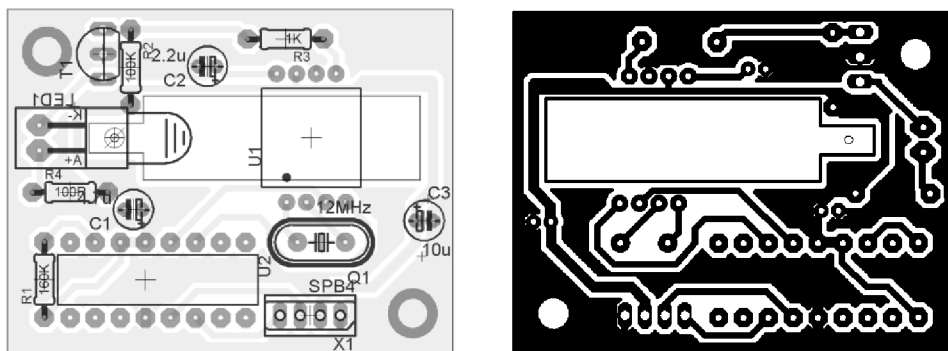
Jako snímač polohy byla použita starší optická myš „Genius NetScroll+ Eye Blue PS/2“. Tato myš měla vnitřní zapojení se dvěma integrovanými obvody a to optický snímač Agilent A2610 C0331 a obvod který zajišťuje komunikaci myši s PC.

Rozlišení tohoto snímače je 18 pulzů/mm . Tato hodnota nereagovala na změnu parametru rozlišení myši popsaného v [4]. Hodnota rozlišení myši byla u každé testované myši jiná, u novějších typů však bylo rozlišení kroku vždy vyšší. Pro účel této diplomové práce je nižší hodnota rozlišení dostatečná, jelikož zjednodušuje komunikaci, data z myši nemusí být tak často aktualizovaná a nedojde k “přetečení“ osmibitového čítače uvnitř myši.

Další vlastností tohoto snímače byl právě osmibitový čítač, na rozdíl od devítibitového, který uvádí [4]. U testovaných myši byly tyto čítače osmibitové a to v rozsahu +128 až -127.



Obr. 3.2: Použitý snímač polohy.



Obr. 3.3: Strana součástek a strana spojů použitého snímače.

Z uvedené počítačové myši byly použity součástky a byl navržen nový plošný spoj, který nemá zbytečně velké rozměry a vejde se do menší krabičky. Toho bylo dosaženo nepoužitím

tlačítek myši a funkce kolečka.

3.1.2 Formát výstupních dat ze snímače polohy z optické myši

Po resetu se všechny myši musí tvářit jako standardní 2/3-tlačítkové s ID $0x00$. Známé rozšíření o "kolečko" (tzv. "M\$ Intellimouse protocol" [4]) lze aktivovat pomocí speciální sekvence příkazů: po resetu myši zapíšeme postupně vzorkovací frekvence 200, 100 a 80. Pakliže myš podporuje toto rozšíření, tak při následném čtení ID vrátí hodnotu $0x03$ a také změní formát paketů, takže už není zpětně kompatibilní! Pokud toto rozšíření není podporováno, dojde pouze k několika změnám vzorkovací frekvence a ID je stále $0x00$.

Další známé rozšíření je "kolečko" + 2 tlačítka (tzv. "Explorer mouse"). To lze aktivovat opět zápisem sekvence příkazů: tentokrát vzorkovací frekvence 200, 200 a 80. Pokud je mód podporován, změní se ID na $0x04$.

Pro účely použití myši jako snímače polohy se spokojíme s původní standardní 2/3 tlačítkovou myší. Zde bude popsán formát paketů, kterými myš vrací informace o změně polohy, stisku tlačítek, atd. V následující tabulce je uveden význam bitů tříbajtového paketu, který myš vrací ve standardním módu (ID $0x00$).

-	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
bajt 0	y_owerrorlow	x_owerrorlow	dy.8	dx.8	1	btn_midle	btn_right	btn_left
bajt 1	dx.0 – dx.7							
bajt 2	dy.0 – dy.7							

Tabulka. 1: Datový paket odesílaný snímačem polohy.

První bajt obsahuje především bitové příznaky jednotlivých tlačítek btn_left, btn_right a btn_midle (HIGH pro stisknutí). Znaménková proměnná dx určuje relativní změnu polohy myši v ose x od posledního úspěšného přenosu paketu. Proměnná je tvořena devíti bity, takže je rozdělena - nižších 8 bitů je ve druhém bajtu a znaménkový bit je v prvním bajtu (dx.8). Hodnota dx se tedy může pohybovat v rozsahu -256 až +255. Obdobným způsobem je dělena i proměnná dy pro určení pohybu v ose y. Bity x_overflow a y_overflow jsou nastaveny, pokud dojde k přetečení dx/dy [4].

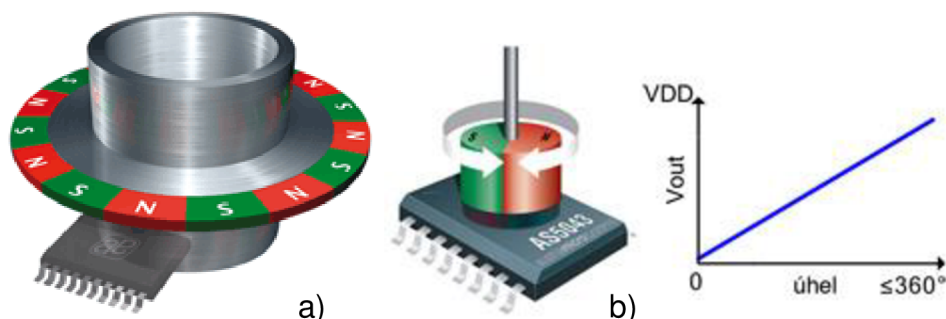
3.1.3 Zhodnocení optického snímače

Po několika měření byl vyvozen závěr, že snímač z počítačové myši není vhodný jako snímač polohy natočení točny. Přesnost tohoto snímače se sice zvýšila úpravou snímaného povrchu (střídání tenkých bílých a černých svislých proužků), ale přesto nedosahovala potřebné spolehlivosti. Při provedení pěti totožných měření za sebou, byla chyba tohoto snímače jeden až pět stupňů.

Neopakovatelnost výsledku měření při stejném natočení točny vyloučila tento snímač pro použití k účelu této diplomové práce. Jako další vhodný bezkontaktní snímač byl navrhnut magnetický n-kodér.

3.2 Magnetický n-kodér

Tyto snímače (na obr. 3.6) dokážou s vysokou přesností rozeznávat polohu i směr otáčení. Lineární magnetický kodér rozezná až 5120 poloh na jednu otáčku, což je rozlišení na $0,07^\circ$. Magnetický rotační kodér rozezná až 1024 poloh na jednu otočku a jeho rozlišení je tedy $0,35^\circ$.



Obr. 3.4: Magnetické kodéry. a) lineární magnetický kodér. b) magnetický rotační n-kodér (převzato z [5]).

3.2.1 Použitý magnetický n-kodér

Jako nejvhodnější snímač pro účely snímání polohy na dané točně byl určen lineární magnetický n-kodér **AS5304B** od firmy „Austriamicrosystems“.

Magnetický n-kodér AS5304 vyžaduje mnohopólový magnetický pásek nebo kruh s délkou magnetického pole 2 mm (magnetický pár 4 mm). Velikost magnetického pole mnohopólového magnetu by měla být od 5 do 60 mT na povrchu čipu [5].

Posunováním magnetického pásku nad polem Hallových sond, vzniká sinusový signál. Díky správnému umístění Hallových sond vzniká ještě jeden sinusový signál posunutý o 90° (COS). Počet těchto period je 40 na magnetický pár. Tyto signály jsou dále především tvarovány na obdélníkový signál a jsou k nim přidány signály AO (Analog Output) a Index, které jsou získány z obvodu automatického zesílení pro Hallové sondy. Vhodným zpracováním těchto výstupních signálů lze získat rozlišení až 160 pulzů na délku magnetického páru. Signál index pak vyše obdélníkový impuls při přechodu na další magnetický pár.

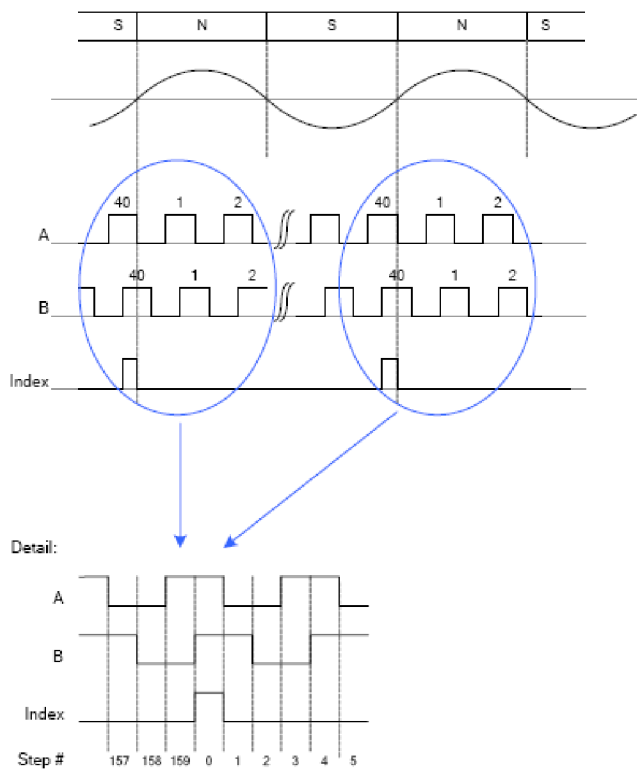
3.2.2 Formát výstupních dat z magnetického n-kodéru

Z magnetického n-kodéru AS5304 vystupují dva obdélníkové signály A a B navzájem posunutě o 90° . Díky tomu lze rozeznat směr pohybu a také máme možnost rozpoznání kroku o velikosti čtvrt délky periody obdélníkového signálu.

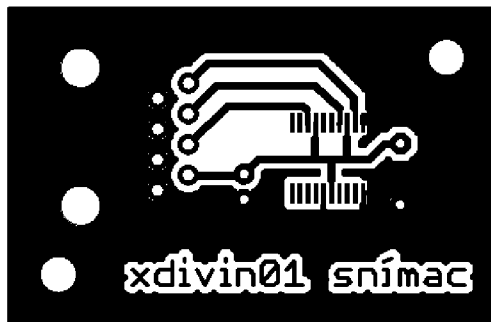
Dalším výstupem je signál Index, který je vyslán vždy při přechodu z jednoho magnetického páru na druhý, takže díky němu můžeme nalézt a zkorigovat chyby při načítání jednotlivých impulsů ze signálů A a B (viz Obr. 3.5).

Dalším výstupním signálem je signál AO, jehož výstupem je napětí odpovídající intenzitě magnetického pole. Toto napětí se nachází mezi $0,5\text{ V}$ a $4,3\text{ V}$, přičemž doporučená intenzita magnetického pole pro spolehlivou funkci n-kodéru se nachází při výstupních napětích AO mezi $1,2\text{ V}$ a $3,45\text{ V}$.

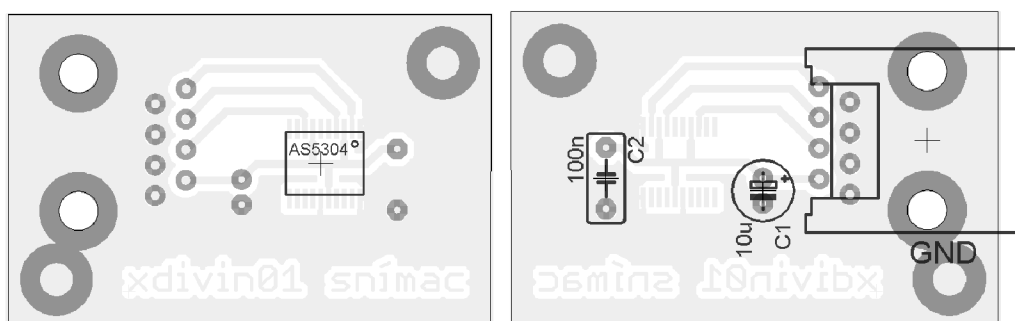
Výstupy A, B a Index jsou u n-kodéru AS504B typu otevřený drain. Další možností těchto výstupů je, že umí indikovat slabé nebo ztrátu magnetického pole a to při jejich nastavení Index = high, A = B = low.



Obr. 3.5: Vzájemné poměry mezi výstupními signály magnetického n-kodéru (převzato z [5])



Obr. 3.6: Strana spojů plošného spoje snímače polohy točny.



Obr. 3.7: Rozmístění součástek na straně spojů a straně součástek.

Snímač je umístěn v malé krabičce o rozměrech $45 \times 35 \times 15 \text{ mm}$. Pro zjednodušení přestavování měřicího pracoviště, nebyl výstupní kabel ze snímače připojen na pevně, ale přes konektor. Jako vhodný konektor byl zvolen telefonní konektor typu RJ45, a to především díky snadné dostupnosti kabelů pro tento typ konektoru. Výhodou tohoto řešení je i samostatné stínění každého signálového vodiče.



Obr. 3.8: Detail snímače a jeho umístění na anténním rotátoru.

4 PROMAX PROLINK – 1B

4.1 Popis přístroje

Prolink – 1B je přenosný měřič úrovní [7]. V pásmech TV a FM umožňuje měření analogových i digitálních signálů s vysokou přesností. Umožňuje měření poměru Video/Audio, ladění zvukové nosné, dálkové ovládání přes RS-232C a samočinné nastavení zaváděcí konfigurace.



Obr. 4.1: Promax Prolink 1B (převzato z [7]).

Měřič pokrývá frekvenční pásmo od 46 MHz do 870 MHz. Navíc se volbou **OPT-101-67** rozšíří jeho frekvenční pásmo od 5 MHz do 48 MHz, což lze využít pro měření zpětné trasy kabelové televize.

Úroveň signálu je možné číst z numerického displeje nebo ve sloupcovém grafu. Numerický displej je výhodnější pro přesné odečítání měřené hodnoty. Přístroj používá vnitřní poloautomatický atenuátor 30 dB.

Při měření zvukového kanálu si lze vybrat jeden ze tří druhů demodulátorů: **FM**, **AM** nebo „**Level sound**“, při kterém je přijímaná úroveň signálů indikována výškou tónu vestavěného reproduktoru. Tento doplněk dovoluje uživateli používat přístroj k nastavování maximální nebo minimální úrovně signálu aniž by musel sledovat údaj na displeji. Odezva detektoru je volitelná: **špičkový detektor** nebo **detektor střední hodnoty signálu**. Tím lze správně hodnotit úroveň zvukonosných s modulací AM nebo digitální.

Promax Prolink – 1B je vybaven sériovým rozhraním RS-232C, za pomoci kterého může být propojen s osobním počítačem. Komunikace je standardizována výrobcem a je založena na příkazech a odpovědích (více v kapitole o komunikačním protokolu). Měřič úrovní může být kalibrován i ovládán na dálku a měřená data mohou být zaznamenána připojenou tiskárnou. Přístroj umožňuje tisk v textové i grafické podobě, což je velmi vhodné pro porovnávání naměřených hodnot. Týká se to především změny úrovně video a audio signálů. Přístroj také umí vytisknout spektra jednotlivých kanálů.

Měřič úrovně je napájen akumulátorem, který poskytuje asi 3 hodiny provozu (30% zapnuto). Pro dobíjení lze použít dobíječ, který pracuje, pokud je přístroj napájen. Přístroj může být napájen autobaterií anebo externím síťovým adaptérem. Dobíjení je indikováno kontrolkou na předním panelu přístroje.

4.2 Komunikační protokol

Standardně je každou sekundu poslán na sériovou linku znak **XON** (*code 11h*). Tímto znakem přístroj sděluje, že je připraven přijímat příkazy. Kdykoliv se na Prolink – 1B obdrží znak * (*hvězdička*), znamená to začátek příkazu. Prolink – 1B přestane vysílat znaky **XON** a po přijetí celého příkazu, který je ukončen znakem **CR** (*carriage return, code 0Dh*), jej vyhodnotí a vykoná. Mezitím odešle znak **XOFF** (*code 13h*) identifikující zaneprázdněnost přístroje výpočtem. Pokud je přijatý příkaz přijat správně a přístroj jej dokázal rozeznat, odesílá potvrzující znak **ACK** (*acknowledge, code 06h*). V opačném případě jde o negativní odezvu **NAK** (*not acknowledge, code 15h*) informující připojené zařízení o špatném zadání příkazu. Za potvrzení **ACK** nebo **NAK** jsou posílány také ukončující znaky: **CR** (*carriage return, code 0Dh*) a **LF** (*line feed, code 0Ah*). Po správném přijetí příkazu, identifikaci a vykonání, Prolink – 1B posílá odezvu, která je ukončena ukončujícími znaky **CR + LF**. Po odezvě je opět vysílán znak **XON** v jednosekundovém intervalu.

Z programátorova pohledu vypadá celý proces následovně:

Čekání na příjem znaku XON

Odeslání kompletního příkazu a přijetí té samé odezvy bez znaku CR

Odeslání znaku CR

Přijetí znaku XOFF jako potvrzení přijatého příkazu a následného zpracování

Přijem ACK v případě, že vzdálený příkaz byl rozpoznán. V opačném případě, příjem znaku NAK, kdy příkaz nebyl rozpoznán.

Přijem znaků CR + LF k oddělení potvrzovacích znaků od odezvy.

Pokud byl v bodě 5 přijat znak ACK, tak je přijata odezva na příkaz ukončená znaky CR + LF

Po skončení příjmu je opět přijímán znak XON v jednosekundovém intervalu.

4.3 Nastavovací příkazy

Všechny nastavovací příkazy Promax Prolink – 1B začínají znakem * (*hvězdička*) [4]. Případná čísla jsou hexadecimální. V projektu je využíván příkaz ***Fxxxx**.

Příkaz ***Fxxxx** nastavuje laděnou frekvenci přeprogramováním nové hodnoty pro PLL. Hodnota musí být poslána v hexadecimálním tvaru ve formátu:

$xxxx = \text{hexadecimální hodnota z } [16 \cdot (f_{in} + 33,375)]$

kde f_{in} je vstupní frekvence v MHz. Více o psaní odborných textů na [7].

Například pro vstupní hodnotu 750 MHz bude příkaz vypadat takto: $16 \cdot (750 + 33,375) = 12534$ a hexadecimálně 30F6, takže příkaz bude ‘*F30F6’.

4.4 Dotazovací příkazy

Dotazovací příkazy jsou podobné nastavovacím s rozdílem návratové odpovědi. Odpovědí je výsledek měření s hodnotou v hexadecimálním tvaru. Ke znaku * (hvězdička) se přidává ? (otazník). V projektu je využíván příkaz ***?Am**.

Příkazem ***?Am** je přístrojem vrácena úroveň elektromagnetické indukce signálu. Parametr „m“ může nabývat hodnot:

1) měření průměrných hodnot,

6) měření špičkových hodnot,

8) zobrazení displeje (změřená hodnota, jednotka měřící veličiny a kanál nebo frekvence). Vrací napětí v rozsahu 0 až 4,095 V.

Příklad: odešle se příkaz ‘*?A6’+CR (CR = 0Dh)

Odezva: *A60237

Hodnota 0237 představuje dekadickou hodnotu 567 a indikuje úroveň v milivoltech (mV). Pro získání dB hodnoty se počítá:

$$Výsledek = 0,567 \text{ V} \cdot 23 + 15 = 28 \text{ dB}$$

Příkazem ***?&xx** je vrácena hodnota obsažená ve vnitřních proměnných přístroje. Tyto proměnné jsou adresované i vrácené jedním bajtem a jsou vždy hexadecimální. Tímto způsobem je umožněno číst hodnoty přímo z LCD displeje přístroje. Každý znak je adresovaný hexadecimální hodnotou od 20 do 2F zleva doprava. Tím jsou interpretovány všechny znaky 16 znakového LCD displeje. Převedením načtené hodnoty na decimální číslo je získáno číslo znaku z ASCII tabulky. Například pro načtení prvního znaku, indikující přetečení rozsahu přístroje bude příkaz vypadat následovně:

‘?&20’+CR+LF

Odpovědí přístroje bude ‘*yy’

Pokud bude yy rovno ‘<’ nebo ‘>’ (resp. 3Ch nebo 3Eh hexadecimálně) z ASCII tabulky znaků, jsou indikovány okraje měřícího rozsahu přístroje.

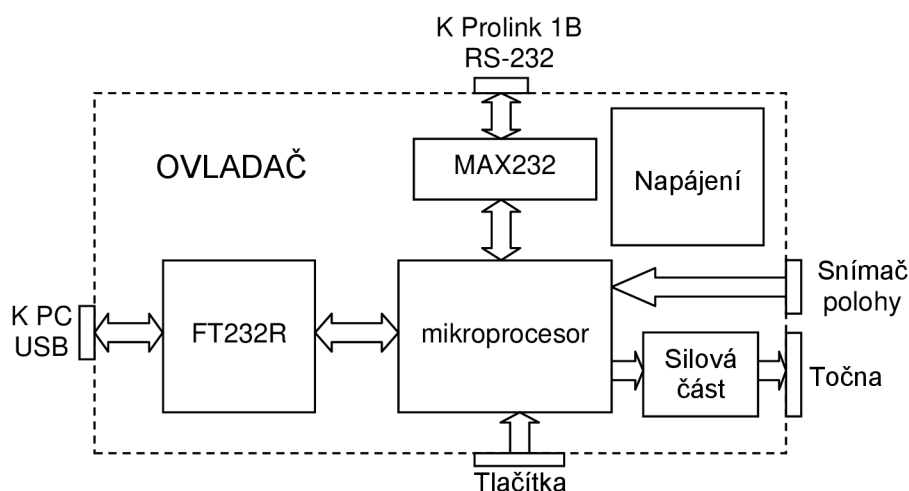
5 KONSTRUKČNÍ ŘEŠENÍ AUTOMATIZOVANÉHO PRACOVIŠTĚ

V této kapitole bude pojednáno o konstrukčním řešení automatizovaného pracoviště a je popsána funkce ovládací krabičky.

5.1 Zapojení pracoviště

Pro automatizované měření směrových charakteristik antén je v laboratoři sestaveno pracoviště podle uspořádání uvedeného na obr. 2.2. Měřená anténa přijímá vlnění vybuzeané pomocnou anténou, která má pevnou polohu a je buzena generátorem. Úroveň buzení antény lze nastavit přímo na generátoru. Měřená anténa je upevněna na točny a při měření se otáčí kolem svislé osy. Otáčení točny a snímání úrovně přijímaného signálu je zajištěno ovladačem. Ovladač je řízen vlastním mikroprocesorem ATmega162-16AU který řídí otáčení točny a zaznamenává v sobě údaje o momentálním úhlu natočení točny. Řízen je buď lokálně pomocí tlačítek na krabičce ovladače, nebo programem z osobního počítače, který mimo řízení točny také zaznamenává všechna naměřená data a vynáší je do grafu v kartézských souřadnicích.

Aby automatizované pracoviště vyhovělo požadavkům moderní výpočetní techniky a zpracování dat (například notebooky už nebývají vybaveny sériovým rozhraním), tak musí komunikace z PC být po sběrnici USB. Ovladač musí v takovém případě umět převést Protokol USB na logiku vhodnou pro sériovou linku mikroprocesoru, vhodný je například obvod FT232R firmy FTDI, který je použit i v této práci. Důležité je také mít programové vybavení v PC na takové úrovni, aby bylo kompatibilní s dnes rozšířenými operačními systémy a to především MS Windows XP, Vista, a Windows 7. Pro přesné snímání polohy byl použit magnetický n-kodér.



Obr. 5.1: Blokové schéma navrhovaného ovladače.

Mikroprocesor v této aplikaci obsluhuje komunikaci s PC přes sériovou linku, dále obsluhuje komunikaci s měřicím přístrojem, na kterou využívá druhou sériovou linku. Z tohoto důvodu byl vybrán mikroprocesor, který má dvě sériové linky. Mikroprocesor ještě navíc snímá stav tlačítek, která slouží k manuálnímu ovládní otáčení točny a dovolí kalibrovat na nulovou polohu přímo na krabičce ovladače. Ovšem hlavní funkce mikroprocesoru spočívá v komunikaci se snímačem polohy a přípravy a odeslání takto získaných dat do PC. Blokový návrh ovladače lze vidět na obr. 4.1.

Při měření je nutno dodržet podmínku pro minimální vzdálenost obou antén (rovnice 1.2) a jejich vzájemné polohy s ohledem na polarizaci vlnění. Měření by mělo probíhat v bezodrazových prostorách nebo na volném prostranství, aby nedocházelo ke zkreslení či znehodnocení měření vlivem odrazů od blízkých předmětů nebo osob.

Rovnice 1.2 vzdálená oblast měřené antény je vzdálenost r která je větší než r_{min} dána vztahem

$$r > r_{min} = \frac{2d^2}{\lambda} \quad (1.2)$$

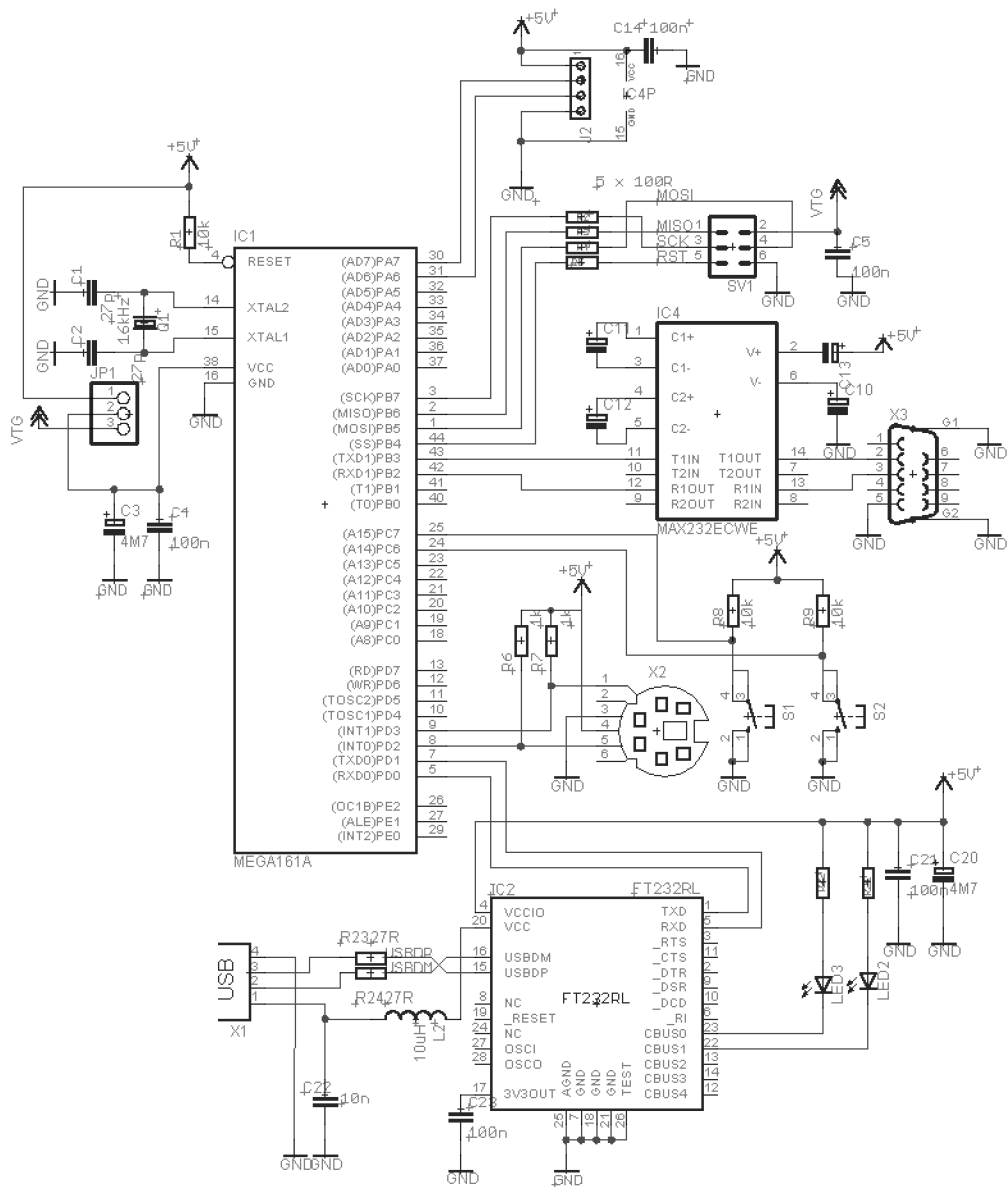
kde d je největší příčný rozměr měřené antény a λ je vlnová délka.

5.2 Původní konstrukční provedení ovladače pro optický snímač

Konstrukce ovladače je založená na mikroprocesoru AT MEGA 162, který je propojen pomocí obvodu FT232R firmy FTDI s konektorem USB typ B. Dále je mikroprocesor spojen přes sériovou linku a převodník MAX232CWE na konektor CAN9.

Tlačítka jsou řešena tak, že v nestisknutém stavu je přes zvedací rezistor přivedena na port mikroprocesoru úroveň *HIGH* a při stisku tlačítka je port přes toto tlačítko uzemněn.

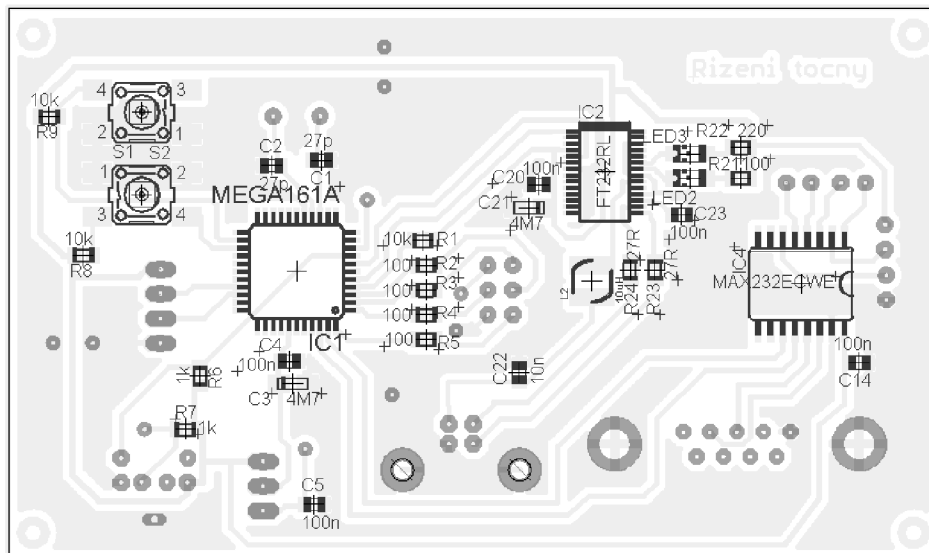
Snímač je připojena přímo k pinům MCU, externě jsou zapojeny jen zvedací odpory - konkrétně hodinový signál na **PIND.2** a datový signál na **PIND.3**. Otevřené kolektory jsou vytvořeny softwarově pomocí **DDR.D.x** bitů - **DDR.D.x=1** pro *LOW* úroveň a **DDR.D.x=0** pro *HIGH* úroveň. **PORT.x** bity jsou trvale 0.



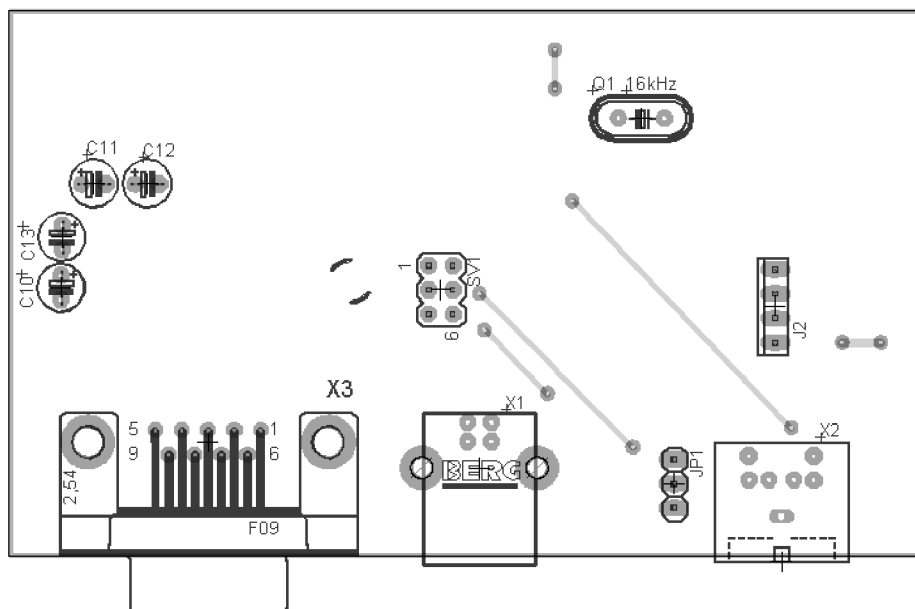
Obr. 5.2: Schéma zapojení navrhovaného ovladače.

Programování mikroprocesoru je zajištěno přes SPI komunikaci. Programátor se pak připojuje do konektoru SW1. Pro účely programování mikroprocesoru je nutno přesunout zkratovací propojku na pinech **JP1** do polohy blíže okraji DPS, aby se nastavilo napájení procesoru na napájení z programátoru.

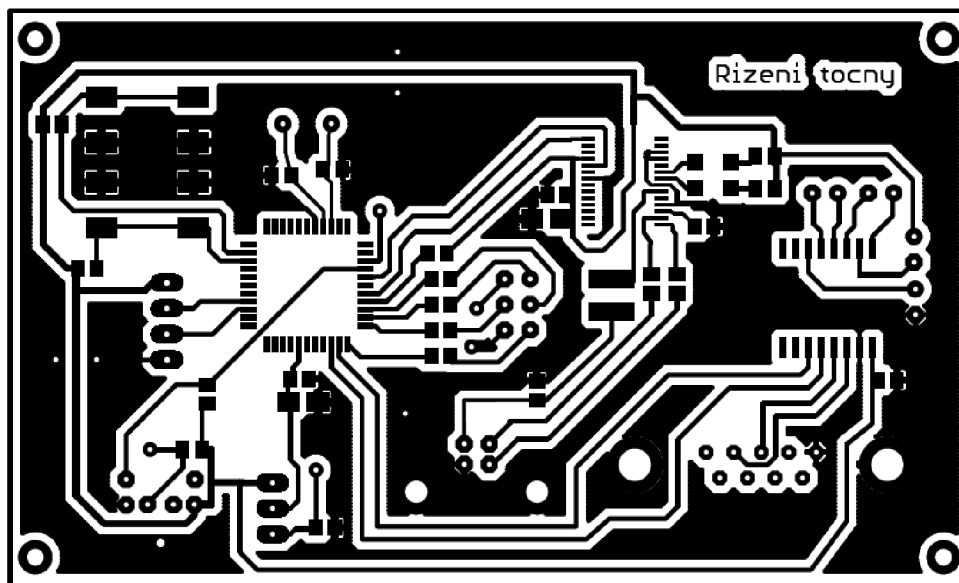
Pro jednodušší výrobu byl tento plošný spoj navrhnut, jako jednostranný viz obr. 4.5. Součástky byly preferovány SMD z toho důvodu, že procesor měl při provedení SMD možnost použít vzorkovací frekvenci 16 MHz . FT232R firmy FTDI se vyrábí pouze jako SMD.



Obr. 5.3: Zapojení ovladače točny - pohled ze strany spojů.



Obr. 5.4: Zapojení ovladače točny - pohled ze strany součástek.



Obr. 5.5: Obrázek plošného spoje – strana spojů.

5.3 Konstrukční provedení ovladače pro magnetický snímač

5.3.1 Schéma zapojení ovladače

Konstrukce zapojení ovladače pro měřicí pracoviště vychází z předchozího návrhu ovladače pro optický snímač. Základem je mikroprocesor ATmega162-16AU. Dále je použita možnost připojení čtyř ovládacích tlačítek. Z čeho jsou využity tři, a to na manuální otáčení točny doprava, doleva a kalibraci (nalezení nulové polohy točny). Tlačítka jsou řešena tak, že v nestisknutém stavu je přes zvedací rezistor přivedena na port mikroprocesoru úroveň *HIGH* a při stisku tlačítka je port přes toto tlačítko uzemněn.

Mikroprocesor komunikuje se snímačem polohy. Snímač udává svou polohu pomocí dvou o 90° od sebe posunutých signálů A a B, a jednoho indexového signálu vyslaného po přechodu snímače na další magnetický pár. Pro danou aplikaci je přesné určení polohy prioritou, proto byly zvoleny pro signál A (PD2) a Indexový (PD3) signál porty, které umožňují přerušování a zbylý signál B byl připojen k portu mikroprocesoru PD4. Na plošném spoji se ještě nacházejí pro každý signálový vodič zdvihací rezistory, jelikož je výstup snímače typu otevřený drain, a zásuvka RJ45, kde je pro každý signálový vodič využit jeden kroucený pár a čtvrtý pár je využit pro napájení snímače.

Komunikace mikroprocesoru s počítačem je řešena pomocí sériového rozhraní. Nyní již počítače, zvláště pak notebooky, jednoduchou sériovou linku RS232 nemívají, tak je v ovladači tento protokol převeden na sběrnici USB pomocí obvodu FTDI232RL a s počítačem pak propojen USB kabelem typu B.

S měřicím přístrojem se pak mikroprocesor komunikuje za pomoci druhého sériového rozhraní **USART1**. K úpravě signálů z TTL logiky na logiku sériového rozhraní RS-232C je použit integrovaný obvod firmy MAXIM v označení MAX232. Jeho vnitřní zapojení obsahuje 2 nábojové pumpy, invertor napětí, dva vstupy TTL se dvěma výstupy logiky sériového rozhraní RS-232C a dva vstupy sériového rozhraní RS-232C se dvěma výstupy na TTL logiku. V zapojení je použit konvertor TTL na RS-232C (MAX232).

Aby bylo možno procesor naprogramovat, je na plošném spoji vyveden ještě konektor pro programátor. K programování mikroprocesoru se používá SPI rozhraní. Konektor je má šest pinů a má označení MLW06G. Zapojení k jednotlivým pinům konektoru je 1 – MISO, 2 – VCC, 3 – SCK, 4 – MOSI, 5 – RST, 6 – GND.

5.3.2 Výkonová část

Výkonovou část ovladače tvoří dva výkonové rezistory a tři bipolární kondenzátory zapojené do pí článku. Vývody kondenzátoru *C6* jsou připojeny na svorkovnici na piny 1 a 2. Sepnutím relé *K1* je zkratován odpor *R15* a kondenzátor *C15*. Tím se také kondenzátor *C6* paralelně připojí k sériové kombinaci odporu *R16* a kondenzátoru *C16*. To znamená, že na pin 2 je připojena druhá svorka z napájecího transformátoru. Pin 1 je spojen taktéž s druhou svorkou transformátoru, ale přes sériově-paralelné zapojení *C6*, *C16* a *R16*. Pin číslo 3 na svorkovnici je napájen prvním koncem transformátoru 24 V~ pouze přes pojistku. V případě sepnutí relé *K2* je situace obdobná.

5.3.3 Ovládání relé

Relé *K1* a *K2* jsou ovládané pomocí tranzistorů *T1* a *T2* jehož báze jsou připojeny přes bázevový rezistor na výstupní port mikroprocesoru, konkrétně piny PA7 a PA6.

Diody *D6* a *D8* jsou zapojeny paralelně k indukčnostem *K1* a *K2* opačně pólovány mi diodami, které zajišťují ochranu proti špičkovým proudům při přepínání relé.

5.3.4 Indikace

12 V rozvod napětí na desce plošných spojů napájí především dvě relé, která ovládají směr otáčení točny, připojená přes svorkovnici *MSTBA3*, za pomoci pí článku. Dále jsou napájeny indikační diody *LED1* až *LED3* přes předřadné rezistory o velikosti 680 Ω. Indikace zapnutí přístroje je řešena podsvíceným síťovým vypínačem, zelené indikují směry točení točny vpravo nebo vlevo. Dále je červenou LED diodou indikována případná ztráta magnetického pole snímače.

5.3.5 Manuální ovládání

Na ovladači jsou ještě umístěny tlačítka pro manuální ovládání točny. Dvě tlačítka na čelním panelu slouží pro ruční otáčení točnou doprava a doleva. Červené tlačítko na zadním panelu provede kalibraci snímače.

5.3.6 Napájení Ovladače

Napájení celého ovladače je zajištěno síťovým transformátorem s výstupem 2x12 V~, přičemž je využito jedno sekundární vinutí pro logickou a ovládací část a Obě vinutí v sérii pro motor anténního rotátoru. Střídavé napětí je vedeno pojistkou se jmenovitým proudem 2 A, na gretzův můstek, za nímž je měřitelné nestabilizované stejnosměrné napětí. Gretzův můstek je řešen pomocí integrovaného obvodu *B250C1500F*, jeho výstup je přiveden na vstup integrovaného stabilizátoru napětí *7812T*, který má odfiltrován vstup i výstup kondenzátory katalogových hodnot.

5.3.7 Mechanická konstrukce ovladače točny

Celá elektronika ovladače je uložena v plastové krabičce UKO95 s vnitřními rozměry $140 \times 66 \times 100 \text{ mm}$. Napájí se síťovým kabelem přes transformátor s dvěma výstupy $12 \text{ V} \sim$, TS25/026 $2 \times 12 \text{ V}$ s jmenovitým výkonem 25 W . Zapojení je chráněno proti přetížení $2 \text{ A}/250 \text{ V}$ pojistkou osazenou na **desce plošných spojů** (dále jenom **DPS**). Napájení transformátoru síťovým napětím je vedeno ze síťového konektoru, přes síťový vypínač kabelem, jehož ochranný vodič je šroubkem spojen s kostrou transformátoru. Transformátor je připájen přímo na DPS. Celé zapojení s transformátorem, pojistkou, zdrojem 12 V a 5 V , relé, porty, je na jedné jednostranné DPS o rozměrech $139,4 \times 92,5 \text{ mm}$. Signálové spoje mají šířku spojů $0,3$ až $0,6 \text{ mm}$, napájecí a výkonové mají šířku spojů $0,6$ až $1,6 \text{ mm}$.

Ze zadní strany panelu ovladače vyčnívá konektor pro připojení svorkovnice napájení motoru anténního rotátoru, kde jednotlivé svorky zleva doprava 1 – 3 odpovídají svorkám 1 – 3 na anténním rotátoru. Následuje je konektor sériového rozhraní RS-232C (samice) pro komunikaci s PROLINK 1B. Další je konektor USB typu B pro komunikaci s PC a nejvíce vpravo je umístěna zásuvka pro konektor RJ-45 pro komunikaci se snímačem polohy.

Na zadním panelu je ještě vpravo nahoře umístěna vidlice síťového napájení typu EURO a tlačítko sloužící k nakalibrování snímače na známou (nulovou) polohu na točně. Otvory v panelu byly nejdříve vyvrtány menším vrtákem a dopilovány pilníkem. Celá DPS je umístěna do plastové krabičky vložením na spodní díl. Do čtyř rohových děr DPS průměru 8 mm musí zapadat spojovací plastové díly krabičky. V případě potřeby je nutné zvětšit díry po okrajích DPS.

Ze síťového napájecího konektoru je přes rozpojovací svorky připojen vodič vedoucí na síťový vypínač, který je umístěn na předním panelu krabičky. Krabička je rozebíratelná na čtyři díly a je spojena čtyřmi samořeznými šroubky.

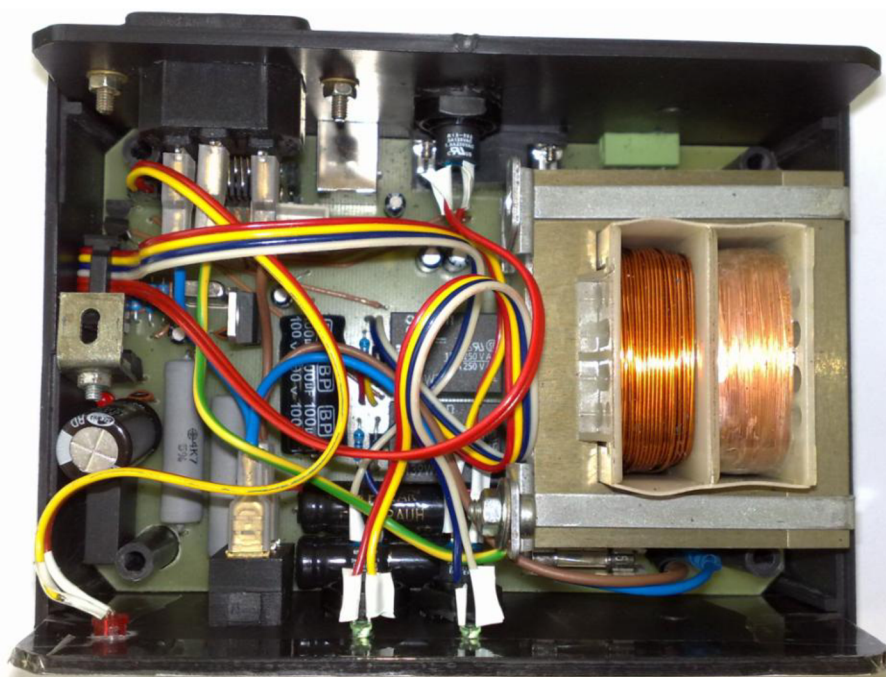
Výsledek konstrukčního provedení zobrazuje následující obrázek 5.6.



Obr. 5.6: Čelní panel krabičky ovladače.



Obr. 5.7: Zadní panel krabičky ovladače.



Obr. 5.8: Pohled do krabičky ovladače.

5.4 Softwarové řešení ovladače pro optický snímač

Při softwarové řešení programu do mikroprocesoru byl nejvíce kladen důraz na komunikaci se snímačem. Prioritou bylo, aby neklesala přesnost odečítání úhlu natočení antény a to i při několikanásobném opakování měření. Celkové řešení tedy je postaveno na tom, že přijetí bajtu od snímače je řízeno přerušením a v době vysílání či příjmu bajtu z přijímače se procesor nevěnuje ničemu jinému. Vývojový diagram tohoto programu je vidět na obr. 5.9.

5.4.1 Nastavení parametrů pro chod programu

Nejprve se nastaví vstupní a výstupní porty podle toho, který se potřebuje jako vstupní (tlačítka) a výstupní (směry otáčení). Port pro komunikaci se snímačem se bude nastavovat

jako vstupní/výstupní až v průběhu programu. Dále bude nastaven časovač/čítač **T/C0** a **T/C1** prozatím bez hodinového signálu, spouštět se budou až v běhu programu pro potřebu zpoždění. Nyní se povolí přerušení obecně nastavením SEI.

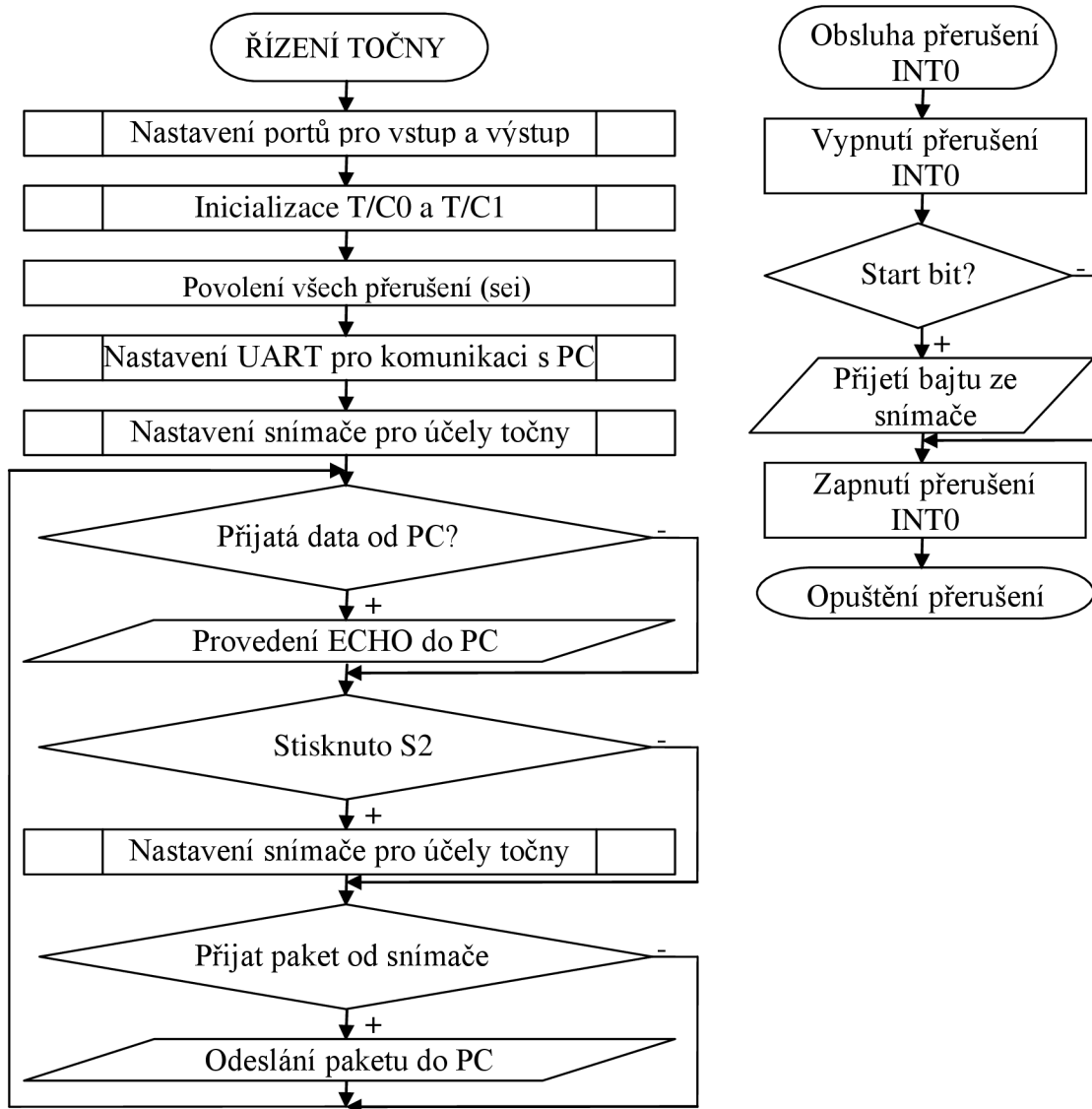
U sériové komunikace UART se nejdříve nastaví dělička na rychlost *19200 bd*, posléze se povolí příjem a vysílání a na závěr formát rámce na osm datových bitů, bez parity a dva stopbity.

Při inicializaci snímače je nejprve poslán příkaz pro jeho restart (*0xFF*) po kterém snímač vrátí hodnotu (*0xFA*) potvrzení, (*0xAA*) selftest v pořádku a (*0x00*) že se jedná o klasickou dvoutlačítkovou počítačovou myš. Poté je nastavena velikost kroku snímače (*0xE8*), opět snímač potvrdí (*0xFA*) poté velikost kroku (*0x00*) *1/mm* v případě, že to snímač podporuje, tak odpoví (*0xFA*). Ovšem ale na toto nastavení testované myši nereagovaly a neustále si držely své rozlišení, konkrétně použitá myš má rozlišení *18 pulzů/mm*

Další důležité nastavení pro tuto aplikaci je maximální rychlost vzorkování s jakou může snímač posílat data do ovladače. To se provede příkazem (*0xF3*), snímač odpoví (*0xFA*), poté se pošle hodnota maximální rychlosti vzorkování, která je ve formátu unsigned char, kdy většina myši podporuje rychlosti 10, 20, 40, 60, 80, 100 a 200 vzorků/s. V našem případě je *20 vzorků/s* dostačující, takže odešleme (*0x14*) a snímač opět potvrdí (*0xFA*).

Nyní se nastaví snímač, aby fungoval v měřítku 1:1 (kde o kolik se posune snímač, tolik pulzů do ovladače pošle). To se provede příkazem (*0xE6*), snímač odpoví (*0xFA*). Další možné (ale nepoužité) nastavení snímače je totiž měřítko 2:1 (*0xE7*), které se chová následovně: pohyb/odesláno 1/1, 2/1, 3/3, 4/6, 5/9, N/2N.

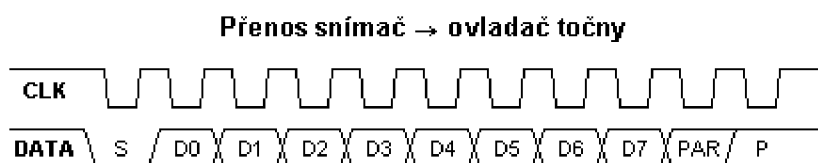
Posledním příkazem (*0xF4*) odpověď je opět (*0xFA*), se zapne v myši „stream mode“, který automaticky po změně polohy snímače, ale maximálně se zadanou vzorkovací frekvencí, odešle do ovladače paket dat, na kterém lze přečíst změna polohy snímače.



Obr. 5.9: Vývojový diagram programu mikroprocesoru - optický snímač.

5.4.2 Obsluha příjmu bajtu ze snímače polohy

Příjem bajtu ze snímače je inicializován přerušením od INTO vyvolané sestupnou hranou hodinového signálu. Když chce snímač poslat bajt, tak nejprve stáhne data do úrovně *LOW*, v zápětí na to přijde první sestupná hrana hodin, na kterou zareaguje přerušení INTO viz. obr. 4.7. Následně je v podprogramu přerušení INTO zakázáno toto přerušení, aby nerušilo příjem. Hodiny vždy generuje snímač a to s frekvencí *110 kHz* až *16 kHz* a tím pádem je jedna perioda tohoto signálu dlouhá nejméně *62 μs* a půlperioda *31 μs*.



Obr. 5.10: Přenos ze snímače polohy do ovladače točny[4].

Mikroprocesor by měl číst hodnotu na datové lince vždy přesně uprostřed *LOW* úrovně hodin. Toho se docílí tak, že po přijetí hrany signálu, v našem případě po přerušení INT0, počkáme $16 \mu s$ na prostředek *LOW* úrovně hodin a poté odečteme, zda-li je na datové lince startbit. Jestliže tam startbit není, došlo k nějaké chybě detekce první sestupné hrany hodin a bude znovu povoleno přerušení INT0 a návrat z podprogramu přerušení. Byl-li startbit detekován, při každé další sestupné hraně hodin bude počkáno $16 \mu s$ na čas čtení datového bitu a po přečtení všech 8 bitů přijde ještě následně paritní bit a stopbit s úrovní *HIGH*. Přijatý bajt uložíme do pomocného bufferu, ve kterém se skládá datový paket snímače. Následně se povolí přerušení INT0 a opustí se podprogram přerušení. Vývojový diagram tohoto přerušení lze vidět na obr. 4.6.

5.4.3 Hlavní smyčka programu

Hlavní smyčka programu obsahuje pro prozatímní testovací účely pouze pár funkcí, které budou popsány níže.

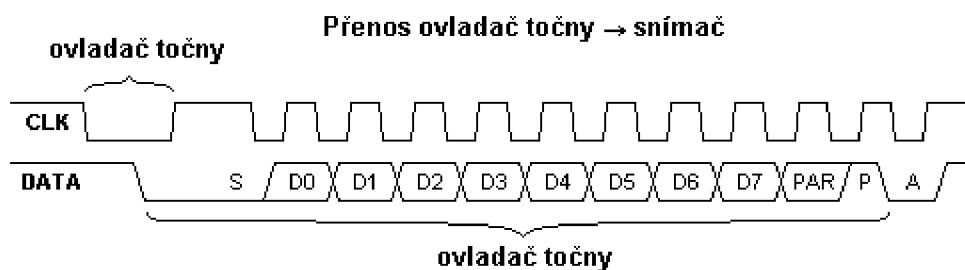
Zkontroluje, zda byla přijata data od PC, pokud ano tak pro ověření správnosti budou data poslána nazpět do PC.

Zjistí, jestli je zmáčknuuto tlačítko **S2** pokud ano tak provede restart snímače a jeho opětovné nastavení.

Jestli byly přijaty tři bajty ze snímače polohy, tak je přijat celý paket a ten je odeslán do PC a zakončen znaky **CR** (návrat vozíku) a **LF** (nový řádek).

5.4.4 Podprogram pro odeslání bajtu do snímače

Signál, který pro odeslání bajtu do snímače je nutno vytvořit je ukázán na obr. 4.8.



Obr. 5.11: Přenos z ovladače točny do snímače polohy (převzato z [4]).

Ovladač točny dá najevo, že by chtěl přenášet vysílat data tím, že stáhne hodiny do *LOW* úrovně alespoň na dobu $100 \mu s$, čímž bezpečně zablokuje případný přenos opačným směrem. Následně master stáhne do *LOW* i data (což je start bit) a počká dalších alespoň $5 \mu s$, pak uvolní hodiny, na což by měl snímač polohy zareagovat nejhůře do $10 ms$ tak, že začne generovat hodiny. Polarita hodin je při přenosu do zařízení přesně opačná – snímač polohy vzorkuje data přesně uprostřed *HIGH* úrovně hodin, master mění hodnotu dat během *LOW* úrovně hodin (nejlépe přesně uprostřed).

Toto chování umožňuje použít stejný algoritmus pro detekci sestupné hrany a čekání

16 μ s na prostředek *LOW* úrovně hodin. Poté mikroprocesor změní výstup na datové komunikaci podle následujícího vyššího bitu vysílaného bajtu. Při vysílání každé jedničky, je ještě obrácena logická úroveň paritního bitu, aby nakonec po odeslání celého bajtu dat dávala lichou paritu.

Následně je tato parita odeslána po ní ještě stop bit úrovně *HIGH* a při dalším prostředku *LOW* úrovně hodin je čten potvrzovací bit odeslaný snímačem s *LOW* úrovní.

5.5 Softwarové řešení ovladače pro magnetický n-kodér

U tohoto řešení je opět nejvyšší důraz kladen na správné načítání polohy, toho je dosaženo tak, že celý program probíhá v hlavní smyčce, pouze obsluha příchodu informace ze snímače je řešena pomocí přerušení. Vývojový diagram lze vidět na obrázku 5.12.

5.5.1 Nastavení parametrů pro chod programu

Nejprve se nastaví vstupní a výstupní porty podle toho, který se potřebuje jako vstupní (tlačítka, snímač) a výstupní (směry otáčení). Dále je nastaven časovač/čítač T/C1 prozatím bez hodinového signálu, spouštět se bude až v běhu programu pro potřebu zpoždění a T/C3 se nastaví jako časovač s předděličkou 1024. Nyní se povolí přerušení obecně nastavením SEI.

U sériové komunikace UART se nejdříve nastaví dělička na rychlost 19200 *bd*, posléze se povolí příjem i vysílání a na závěr formát rámce na osm datových bitů, bez parity a dva stopbity. Toto nastavení se provede pro oba UART kanály.

Ještě se nastaví přerušovací systém, aby přerušení od INT0 reagovalo na vzestupnou hranu a INT1 reagovalo na sestupnou hranu. Povolí se přerušení od INT0 a INT1 a vynuluje se proměnná která, obsahuje informaci o poloze.

5.5.2 Hlavní smyčka programu

Hlavní smyčka programu obsahuje funkce, které probíhají, pokud nepřijde nějaká informace s magnetického snímače polohy.

Zkontroluje, zda je dostatečná síla magnetického pole, pokud je velikost magnetického pole slabá, rozsvítí informační červenou LED na čelním panelu a zařídí, aby o tom byl informován obslužný program v PC.

Zkontroluje, zda byla přijata data od PC, pokud ano, tak spustí proceduru, která tuto událost obslouží.

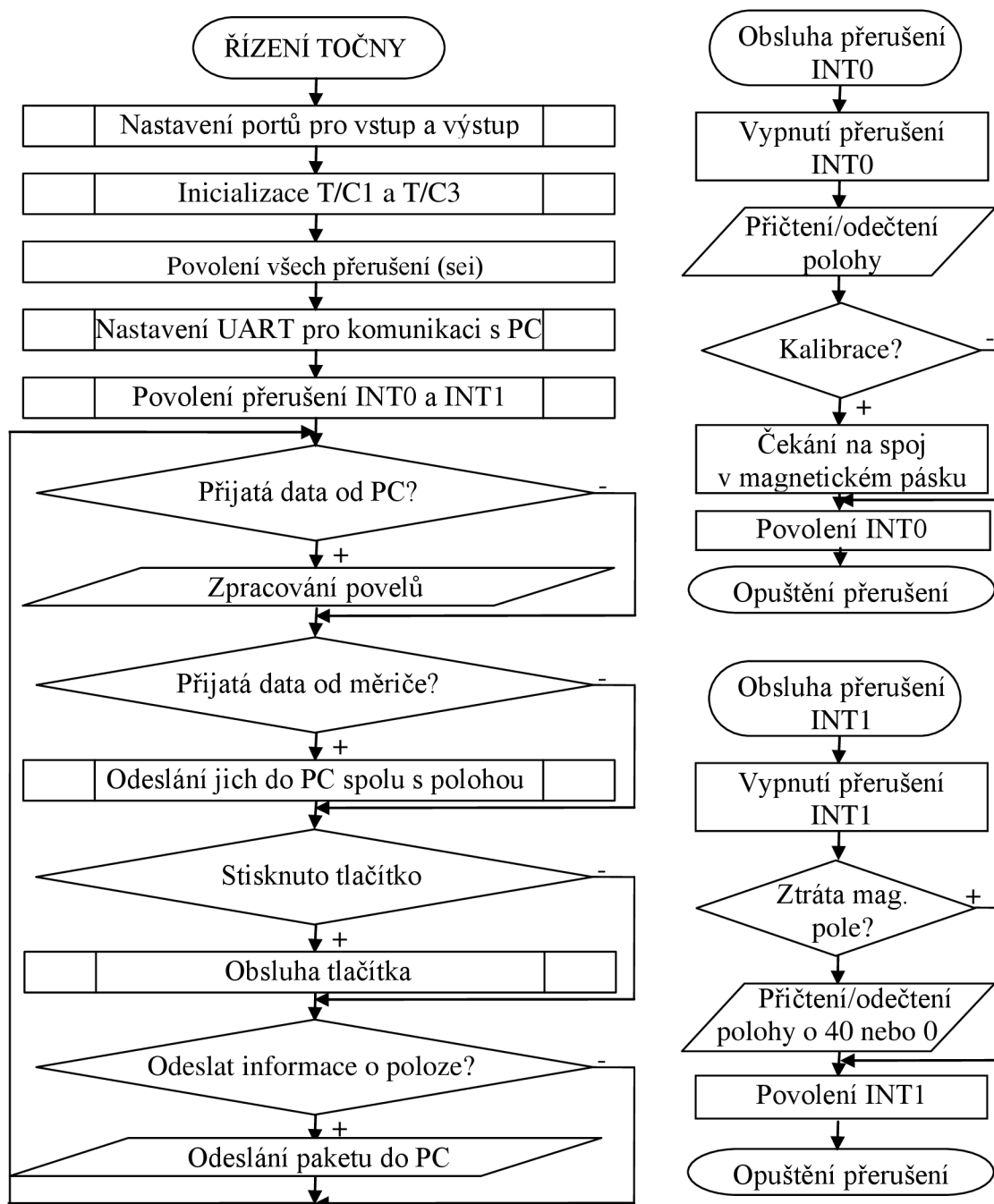
Zkontroluje, zda byla přijatá data z měřiče úrovní, pokud ano, tak spustí proceduru, která tuto událost obslouží.

Zjistí, zda je zmáčknuo tlačítka vpravo, pokud ano tak spustí otáčení točny vpravo. Při stisknutí tlačítka vlevo provede podobnou událost a roztočí točnu vlevo.

Zkontroluje stisk tlačítka **kalibrace** a v případě jeho stisku, jestliže kalibrace nebyla už jednou spuštěna, najde rotátor svou výchozí polohu a vynuluje si svou vnitřní proměnnou s informací o poloze.

Dále zkontroluje, zda některá část programu nechce vyslat do PC informaci o úhlu natočení točny, pakliže ano, odešle do PC informaci o poloze ve formátu #xy+CR+LF kde x

je vyšší bajt proměnné **poloha** a y je nižší bajt proměnné **poloha** zakončený znaky CR (návrat vozíku) a LF (nový řádek). V případě že, byla zaznamenána ztráta magnetického pole u snímače, bude ještě navíc tato informace zaslána ve tvaru „_e“.



Obr. 5.12: Vývojový diagram programu mikroprocesoru magnetický n-kodér.

5.5.3 Obsluha snímače

Snímač je k mikroprocesoru připojen k portu D a to: signál **A** k PD2 (INT0), **B** k PD4 a **Index** k PD3 (INT1). Vnější přerušení INT0 je nastaveno na náběžnou hranu signálu a vnější přerušení INT1 je z důvodu, aby se nekrylo s INT0 (obr 3.5) nastaveno na sestupnou hranu.

Obsluha hlavního přerušení od signálu **A** INT0 je provedena následovně, nejprve je

zakázáno přerušení INT0.

Podle momentální logické úrovně signálu **B** zjistí, zda odečíst nebo přičíst jeden krok v poloze.

Dále si zjistí, zda není zapnuta kalibrace, a pokud je, tak odečte napočítaný počet impulzů v časovači C/T3 menší než 250. Číslo 250 je určeno z empirického měření, kdy už se projeví magnetický defekt ve spojení magnetického pásku v časovém sledu po sobě přičítajících impulzů snímače. Bylo naměřeno, že při čase rovnajícím se napočítání 200 – 350 impulzů v časovači C/T3 (12,8 – 22,4 ms) je tento defekt dobře rozpoznatelný. Z důvodu, že rotátor se nebude točit rychleji než při měření, kdy nebyl zatížen, byla výsledná hodnota stanovena na 250 impulzů, při nichž dojde k bezpečnému rozeznání tohoto defektu. Najde-li program tento defekt, vynuluje polohu a zastaví točnu. Následně bude do PC odeslána informace o hotové kalibraci „_k“.

Pokud kalibrace právě neprobíhá, program přesto nuluje C/T3, povolí přerušení INT0 a vrátí se zpět k aktuálně probíhajícímu programu.

Obsluha přerušení pro signál index probíhá následovně - nejprve je zakázáno přerušení INT1 a následně je vyhodnoceno, jestli jde o průchod mezi magnetickými páry, tj. musí být alespoň jeden ze signálů **A** a **B** v log úrovni *HIGH*, není-li tomu, tak je povoleno přerušení INT1 a opouští se obsluha přerušení.

Pokud program usoudí, že se jedná o přechod mezi magnetickými páry, tak si zjistí o kolik se od posledního přijmutí tohoto impulzu, posunula poloha. Je-li tento posun větší než 20 impulzů, přičte k bývalé poloze 40 impulzů, pokud je mezi 20 a -20, použije původní polohu a pokud je menší než 20 impulzů, tak od staré polohy odečte 40 impulzů. Tím opraví chyby vzniklé ztrátou některého impulzu.

Povolí přerušení INT1 a opouští se obsluha přerušení.

5.5.4 Obsluha příjmu znaku z PC a z měřicího přístroje

Tato procedura porovnává přijímaný znak s vnitřními příkazy:

r – zapne otáčení doprava.

l – zapne otáčení doleva.

n – vypne otáčení.

d – vrátí aktuální polohu ve tvaru #xy+CR+LF kde *x* a *y* dají dohromady proměnnou polohu (*x* – vyšší bajt, *y* – nižší bajt), ta je typu znaménkový šestnáctibitový integer.

k – spustí kalibraci, pokud nebyla provedena, do PC vrátí _k.

* – počínaje tímto bajtem, přeposílá příkazy do měřicího přístroje.

Při příjmu s měřicího přístroje tyto znaky přeposílá do PC, pouze si kontroluje znak „LF“, při kterém oznámí programu ukončení příjmu. Toho je využito při zasílání údajů o poloze.

5.5.5 Funkce kalibrace

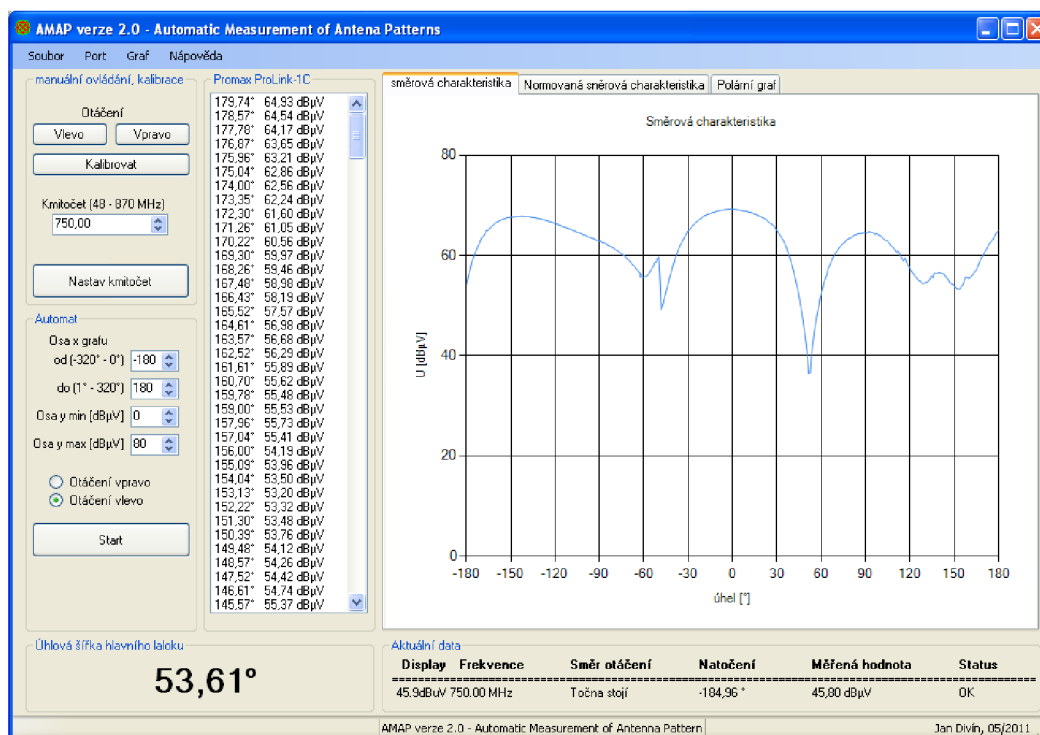
Ta je spuštěna buď stiskem tlačítka kalibrace na zadním panelu ovládací krabíčky, nebo posláním znaku „k“ z PC. Nejprve si funkce zjistí, zda už neproběhla, a pokud ještě ne, tak spustí otáčení doprava, vynuluje časovač C/T3 a počká přibližně sekundu, než se točna roztočí. Pak čeká ve vlastní smyčce, než je nalezen defekt při spojení magnetického pásku. Nakonec zastaví točnu a odešle údaj o hotové kalibraci do PC ve formě „_k“.

6 OVLÁDACÍ PROGRAM AMAP

6.1 Seznámení se s programem

Název aplikace AMAP vychází z anglické zkratky názvu *Automatic Measurement of Antenna Patterns*. Aplikace AMAP verze 2.0 navazuje na verzi 1.2. Novější verze z původní přebrala pouze vzhled, ale není již zpětně kompatibilní.

AMAP nabízí hlavně pohodlnost a přehlednost při automatizovaném měření směrových charakteristik antén. Po důkladném nastavení je možné sledovat průběh aktuálního měření v grafické i textové formě. Výstupem je hlavně grafická závislost změřených intenzit elektromagnetické indukce na úhlu otáčení antény v kartézských i v polárních souřadnicích. Export dat do textových souborů urychluje další zpracování výsledků. Vzhled programu lze vidět na obrázku 6.1.



Obr. 6.1: Vzhled programu AMAP verze 2.0.

6.2 Popis funkcí programu

Mezi hlavní funkce AMAPu patří funkce k řízení, měření a zachytávání dat. AMAP si pamatuje posledně nastavený sériový port a posledně nastavenou měřicí frekvenci pro komunikaci s ovladačem. Je možné točit anténou na obě strany a to jak manuálně, tak automaticky při měření. Manuálním točením je nastavován výchozí úhel měření, automatickým točením je prováděno měření.

AMAP kontroluje přítomnost měřiče. V průběhu chodu AMAPu je možné měřič připojit

nebo zapnout. S ohledem na otevření portu a jeho používání je při zavádění AMAPu zabráněno dalšímu spuštění stejné aplikace. Po uzavření aplikace AMAP je možné ji znovu zapnout, ale vždy pouze jako jedinou instanci.

Při měření jsou průběžně do grafu vynášena změřená data. Grafy jsou po doměření normovány. Po doměření je spočtena úhlová šířka hlavního laloku charakteristiky. AMAP vypisuje data do grafu vždy podle směru točení rotátoru.

6.3 Ovládání programu

6.3.1 Spuštění programu

Po spuštění AMAPu je uživateli zobrazen hlavní formulář zobrazen na obr. 12. Pokud se uživatel pokusí spustit AMAP více než jednou, je nahlášená chyba. Uživatel má možnost zvolení komunikačního rozhraní COM v hlavním menu pod položkou **Port**. Toto nastavení je uloženo do **nastaveni.txt** souboru na disk počítače. S tímto nastavením se bude počítat při dalším zapnutí AMAPu. Po nastavení portu je možné ovládat točnu.

6.3.2 Kalibrace a nastavování

Prvním krokem před samotným započítáním měření je kalibrace. Kvůli otáčení točny vpravo při kalibraci je doporučeno manuálně natočit točnu s dostatečnou rezervou doleva. Samotná kalibrace se spustí tlačítkem **kalibrovat** a počká se na odezvu programu, která nastane poté, co si anténní rotátor nalezne svou výchozí polohu.

Program AMAP si pamatuje poslední nainstalovanou anténu a její nulovou polohu 0° , kdy je měřená anténa natočena přímo na vysílací anténu. Při výměně antény se tato poloha najde ručním posuvem doprava doleva a stiskne se tlačítko „**Nastavit nulu**“, které tento úhel označí jako nula.

Další doporučený postup v přesném nastavení antény je nastavit měřicí úhel na točně v rozmezí -15° až 15° , provést automatické měření a po jeho skončení v nabídce **soubor** vybrat volbu **nastav nulu automaticky**. Automatickým přepočtem naměřených dat určí nulový úhel jako úhel, na kterém má měřená anténa nejsilnější příjem.

AMAP umožňuje na Promax Prolink – 1B zasílat požadavky o změně měřicí frekvence. Měřič se, po korektním přijetí příkazu, automaticky přeladí. Není-li měřič připojen k ovladači, nebo není-li zapnut, je zobrazeno chybové hlášení. Potvrzuje se po zadání číselné hodnoty tlačítkem **Nastav kmitočet**. Automatické měření je možné zahájit po kalibraci úhlu a nastavení kmitočtu měřiče Promax Prolink – 1B.

V panelu **Automat** jsou umístěny prvky na nastavení os grafu. Souřadnicí osy y lze změnit výchozí rozsah -180° až 180° na jiný. Číslice lze zadávat z klávesnice nebo posuvníky. Úhel 0° by měl směřovat na zdroj přijímaného signálu.

6.3.3 Automatizované měření

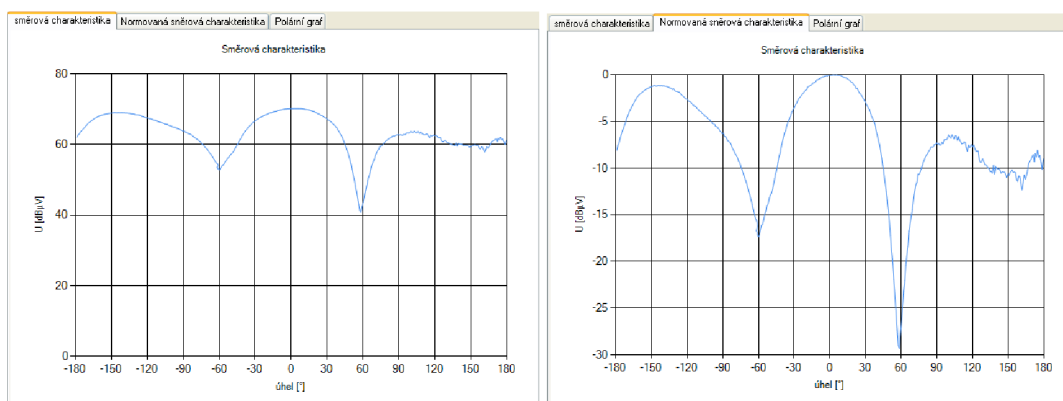
Druhá část je věnována automatickému měření v panelu **Automat**. Volbou je také směr prvotního otáčení. Kliknutím na tlačítko **Start** se zahájí automatické měření. Točna se po doměření automaticky zastaví, ale je ji možné zastavit i manuálně stisknutím tlačítka **Stop**. Na zastavení točny je nastavena ochranná doba *2000 ms*, po jejíž uplynutí je možné dále

pracovat s programem.

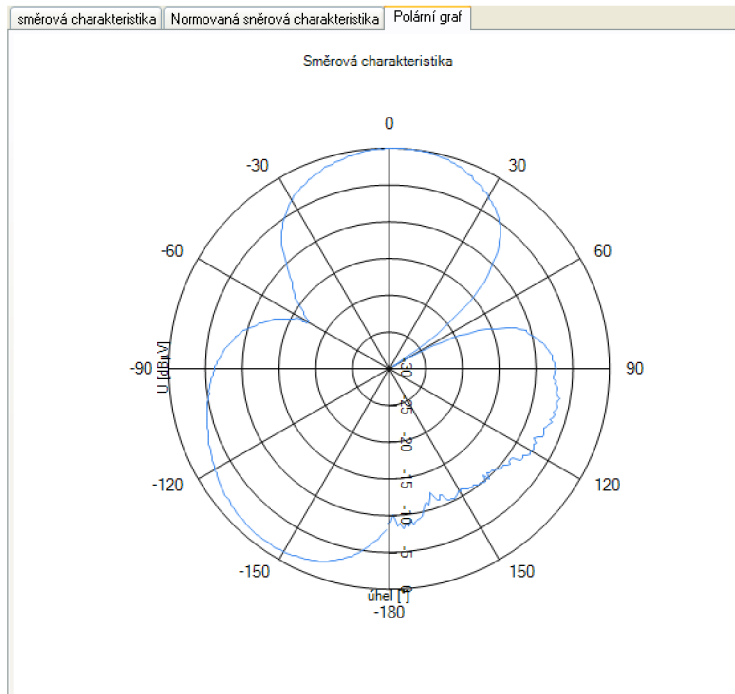
Po úspěšném samozastavení točny jsou k dispozici všechna data. Přepínač směru otáčení je automaticky přepnut na druhou stranu. Tím je zabráněno omotávání kabelu kolem točny.

Další částí je pole Promax Prolink 1B, kde jsou v průběhu automatického měření vypisovány aktuální hodnoty k danému úhlu natočení. Tato data lze po měření uložit kliknutím na **Soubor** a **Uložit** hlavním menu aplikace. Před samotným uložením se ještě program zeptá, jaký chce uživatel použít typ oddělovače mezi jednotlivými daty, kvůli pozdějšímu importu do rozličných programů.

Výsledek měření si lze prohlédnout v grafech (obr. 6.2 a 6.3).



Obr. 6.2: Graf s naměřenými daty a graf s normovanou směrovou charakteristikou.



Obr. 6.3: Normovaná směrová charakteristika v polárním grafu.

6.3.4 Výstupy

AMAP 2.0 ukládá textové soubory, v nichž na prvním řádku je uložena měřená frekvence a na dalších úhel natočení a odpovídající naměřená hodnota z Promax Prolink-1B.

Další panel je pojmenován jako *Úhlová šířka hlavního laloku*. Jeho obsah prozrazuje samotný název. Do něj je po úspěšném doměření a zastavení točny vypočítána úhlová šířka hlavního laloku ve stupních. Jedná se o rozdíl úhlů při poklesu o 3 dB vzhledem k maximum celé charakteristiky.

6.3.5 Vystředování grafů

Vystředování grafu je aktivní pouze po zachycení všech hodnot kolem točny. Ovládací panel, který je na obr. 6.4, se zobrazí po kliknutí na **Vystředění grafu** v hlavním menu nabídky, položky **Graf**.

Vystředit na úhel znamená zobrazit graf tak, aby zadaný úhel zaujímal místo v bodě 0° . Celý graf lze posouvat oběma směry. Zároveň se posouvají všechny grafy.



Obr. 6.4: Vystředování grafu.

Další možností jak vystředit graf je použití tlačítka v nabídce **Soubor**, volbu **nastav nulu automaticky**. Tato volba posune i nulu měření a při opakovaném měření bude již nejvyšší přijímaná úroveň signálu vystředěná na úhel 0° .

7 PROGRAMÁTORSKÁ PŘÍRUČKA

Programová příručka popisuje zkonstruování AMAPu ve vývojovém prostředí Microsoft Visual Studio. Jsou zde popsány vybrané základní funkce programu.

Ovládání ovladače je založeno na posílání informací do ovladače přes sériové rozhraní. K tomuto účelu slouží následující příkazy:

r – zapne otáčení doprava.

l – zapne otáčení doleva.

n – vypne otáčení.

d – vrátí aktuální polohu ve tvaru #xy+CR+LF kde x a y dají dohromady proměnnou polohu (x – vyšší bajt, y – nižší bajt), ta je typu znaménkový šestnáctibitový integer.

k – spustí kalibraci, pokud nebyla provedena, do PC vrátí _k.

* – počínaje tímto bajtem, přeposílá příkazy do měřicího přístroje.

Ovladač přeposílá celou komunikaci směrem od měřicího přístroje do PC.

7.1 Struktura NamerenyBod

Zde se definuje struktura jedné datové buňky naměřeného bodu, která se potom využívá pro zaznamenávání naměřených dat, které se dále zaznamenávají do datové struktury **fronta**, která implementuje rozhraní **IEnumerable**.

```
class NamerenyBod : IComparable<NamerenyBod>
{
    public double Hodnota { get; set; } //naměřená úroveň signálu
    public double Poloha { get; set; } //úhel natočení točny
    public int Id { get; set; } //pomocná proměnná

    public override string ToString() //přetížení metody ToString
    {
        return String.Format("{0}°\t{1} dBμV",
this.Poloha.ToString("F"), this.Hodnota.ToString("F"));
    }

    int IComparable<NamerenyBod>.CompareTo(NamerenyBod other)
    {
        if (other == null)
            return 1;
        if (this.Id > other.Id)
            return 1;
        if (this.Id < other.Id)
            return -1;

        return 0;
    } //řazení podle ID
}
```

7.2 Funkce timer_Tick

Tato funkce se spustí jako událost komponenty timer, při uplynutí jedné sekundy. Tato funkce

informuje o aktuálním stavu na Promax Prolink-1B a o aktuálním úhlu natočení točny.

```
private void timer_Tick(object sender, EventArgs e)
{
    SendSerial("*?A6\r"); //požádá o měření vstupní úrovně
    typDat = 1;           //proměnná pro vnitřní potřeby programu
    timerMeasure.Enabled = true; //kvůli odeslání stavu
    displeje měřiče Promax Prolink-1B
}
```

7.3 Funkce SendSerial

Tato funkce odešle zvolený řetězec znaků typu **string** do ovládací krabčky.

```
private void SendSerial(string sendString) //SEND SERIAL
{
    try
    {
        if (!_serialPort.IsOpen) //otevře port není-li otevřen
            _serialPort.Open();
        // char[] buffer;
        // buffer = new char[1];
        // buffer[0] = 'n';

        if (kontrola) //jestli PC nepřijal odpověď s ovladače
        {
            lbStatusy.Text = "chyba krabičky";
            statusBar.Text = "Info: Vypnutá krabička nebo
měřicí přístroj";
        }
        else
        {
            if (lbStatusy.Text == "chyba krabičky")
            {
                lbStatusy.Text = "OK";
                statusBar.Text = "";
            }
        }
        if (sendString[0] == '*') kontrola = true;

        _serialPort.Write(sendString);

        // _serialPort.Close();
    }

    catch (Exception ex)
    {
        lbStatusy.Text = "chyba portu";
        statusBar.Text = "Error otevření/zápis na serial port
:: " + ex.Message;
    }
}
```

7.4 Funkce OnReceived

Je spuštěna při přijetí bajtu přes sériovou linku.

```
private void onReceived(object sender, SerialDataReceivedEventArgs e)
{
    try
    {
        /* Cteme z prijimaciho bufferu dokud neni prazdny */
        while ((_serialPort.IsOpen) &&
(_serialPort.ReadBufferSize != 0))
        {
            /* Prichodzi byte precteme a ulozime do pomocne
promenne */
            PrijemChar((char)_serialPort.ReadByte());
        }
    }
    catch (Exception)
    {
    }
}
```

7.5 Funkce prijemChar

Jedna z nejzákladnějších funkcí, přijme znak, který určí co je zač, a buď ho předá nějaké funkci, nebo zjišťuje, kam další přijaté znaky budou patřit.

```
private void PrijemChar(char receivedChar)
{
    kontrola = false;
    if (lbStat.InvokeRequired)
    {
        lbStat.Invoke(new Action<char>(PrijemChar), receivedChar);
    }
    else
    {
        if (typPrijem == "#") //přijem úhlu natočení má přednost
        {
            prijemPolohy(receivedChar);
        }
        else
        {
            switch (receivedChar) //rozeznání prvního znaku
            {
                case '*':
                    stavPrijem = 0;
                    prijemDotazu(receivedChar);
                    typPrijem = "*";
                    break;
                case '#':
                    stavPrijem = 0;
                    typPrijem = "#";
                    break;
                case '_':
                    stavPrijem = 0;
                    typPrijem = "_";
                    break;
            }
        }
    }
}
```

```

        case '\u0011':
            stavPrijem = 0;
            break;
        default:
            break;
    }
    if (stavPrijem == 1)
    {
        prijato = "";
        switch (typPrijem) //rozeznání druhého znaku
        {
            case "*":
                switch (receivedChar)
                {
                    case 'A':
                        typPrijem = "*A";
                        break;
                    case 'F':
                        typPrijem = "*F";
                        break;
                    case '?':
                        typPrijem = "*?";
                        break;
                }
                break;
            case "_":
                switch (receivedChar)
                {
                    case 'e':
                        prijemChyby(receivedChar);
                        break;
                    case 'c':
                        prijemChyby(receivedChar);
                        break;
                    case 'k':
                        kalibrovano = 1;
                        lbSmer.Text = "Točna stojí";
                        btKalibrovat.Text = "Nastavit nulu";
                        break;
                }
                break;
        }
    }
    else if (stavPrijem > 1)
    {
        switch (typPrijem) //odeslání znaku do správného
        {
            case "*A":
                prijemOdezvy(receivedChar);
                break;
            case "*F":
                prijemDotazu(receivedChar);
                break;
            case "*?":
                prijemDotazu(receivedChar);
                break;
            default:
                lbStatusy.Text = "Chyba příjmu";
                break;
        }
    }
}

```

podprogramu

```

        }
    }
    stavPrijem++;
}
}

```

7.6 Funkce příjemPolohy

Převede poslanou informaci o natočení točny na číslo (úhel), který lze zpracovat.

```

private void příjemPolohy(char prijatyChar)           //PRIJEM POLOHY
{
    int cislo;
    cislo = (int)prijatyChar;
    if (stavPrijem == 1) //vyšší bajt polohy
    {
        poloha = 0;
        poloha = cislo;
        poloha = poloha << 24;
        poloha /= 65536;
    }
    else if (stavPrijem == 2) //nižší bajt polohy
    {
        cislo &= 0x00ff;
        poloha |= cislo;
        uhelSkutecny = ((double)poloha / 7.6666667);
        uhel = uhelSkutecny - nula;
        lbStat.Text = uhel.ToString("F") + " °";
    }
    else if (stavPrijem > 2) // při třetím bajtu a probíhajícím automatickém
měření spustí záznam naměřených hodnot
    {
        if (prijatyChar == '\n')
        {
            if (measure != 0)
            {
                Measure();
            }
            typPrijem = "";
        }
    }
}
}

```

7.7 Funkce btStart_Click

Tato funkce je vyvolána při stisku tlačítka **Start** u automatického měření. Funkce připraví proměnné a další události aby mohlo být započato automatické měření. A slouží i k zastavení tohoto měření.

```

private void btStart_Click(object sender, EventArgs e)           //START
{
    if (measure == 0)
    {
        btStart.Text = "Stop";
        spravceZalozek.SelectedIndex = 0; //nastaví obyčejný graf
        rbVlevo.Enabled = false; rbVpravo.Enabled = false;
    }
}

```

```

        ID = 1;

        while (frontaHodnot.Count != 0) //vymaže stará data
        {
            frontaHodnot.Dequeue();
        }
        lbVysledky.Items.Clear(); //vymaže staré data
        grafSmerova.Series["Series1"].Points.Clear(); //počáteční nastavení
grafu
        grafSmerova.ChartAreas["ProstorGrafu"].AxisX.Minimum =
(double)numeOd.Value;
        grafSmerova.ChartAreas["ProstorGrafu"].AxisX.Maximum =
(double)numeDo.Value;
        double interval = ((double)numeDo.Value - (double)numeOd.Value) %
12.0;
        if (interval == 0.0) interval = ((double)numeDo.Value -
(double)numeOd.Value) / 12.0;
        else interval = ((double)numeDo.Value - (double)numeOd.Value) / 10.0;
        grafSmerova.ChartAreas["ProstorGrafu"].AxisX.Interval = interval;

        typDat = 0;
        timer.Enabled = false; timerMeasure.Enabled = true;
        measure = 1;
    }
    Else //tlačítko STOP
    {
        lbSmer.Text = "Točna stojí";
        SendSerial("n");
        measure = 0;
        btStart.Text = "Start";
        rbVlevo.Enabled = true; rbVpravo.Enabled = true;
        timerMeasure.Enabled = false; timer.Enabled = true;
        measure = 0;
    }
}
}

```

7.8 Funkce Measure

Nejrozsáhlejší funkce v tomto programu, slouží k zaznamenávání naměřených hodnot a k ovládání chování automatizovaného pracoviště při automatickém měření. Tato funkce se spouští po příchodu kompletního datového balíčku od ovládací krabičky.

```

private void Measure() //MEASURE
{
    double uhelMin = (double)numeOd.Value;
    double uhelMax = (double)numeDo.Value;
    if (rbVlevo.Checked) //-----otaceni vlevo
    {
        switch (measure) //stavový automat měření
        {
            case 1:
                if (uhel > uhelMin)
                {
                    lbSmer.Text = "Otáčení vpravo";
                    SendSerial("r");
                    measure = 2;
                }
                else measure = 3;
            }
        }
    }
}

```

```

        break;
    case 2:
        if (uhel < uhelMin)
        {
            lbSmer.Text = "Točna stojí";
            SendSerial("n");
            timerMeasure.Enabled = false;
            timerDelay.Enabled = true;
        }
        break;
    case 3:
        lbSmer.Text = "Otáčení vlevo";
        SendSerial("l");
        measure++;
        break;
    case 4:
        if (uhel > uhelMin)
        {
            frontaHodnot.Enqueue(new NamerenyBod { Id = ID, Hodnota =
hodnota, Poloha = uhel });
            lbVysledky.Items.Add(uhel.ToString("F") + "°\t" +
hodnota.ToString("F") + " dB $\mu$ V");
            grafSmerova.Series["Series1"].Points.AddXY(uhel, hodnota);
            ID++;
        }
        if (uhel > uhelMax)
        {
            lbSmer.Text = "Točna stojí";
            SendSerial("n");
            timerMeasure.Enabled = false;
            timerDelay.Enabled = true;
        }
        break;
    case 5:
        timerMeasure.Enabled = false; timer.Enabled = true;
        measure = 0;
        rbVlevo.Enabled = true; rbVpravo.Enabled = true;
        rbVpravo.Checked = true; rbVlevo.Checked = false;
        btStart.Text = "Start";
        break;
    default:
        lbStatusy.Text = "Chyba automatu";
        break;
}

}

}
else //-----otaceni vpravo
{
    switch (measure) //stavový automat měření
    {
        case 1:
            if (uhel < uhelMax)
            {
                lbSmer.Text = "Otáčení vlevo";
                SendSerial("l");
                measure = 2;
            }
            else measure = 3;
            break;
        case 2:
            if (uhel > uhelMax)
            {

```



```

        lbSmer.Text = "Točna stojí";
        SendSerial("n");
        timerMeasure.Enabled = false;
        timerDelay.Enabled = true;
    }
    break;
case 3:
    lbSmer.Text = "Otáčení vpravo";
    SendSerial("r");
    measure++;
    break;
case 4:
    if (uhel < uhelMax)
    {
        frontaHodnot.Enqueue(new NamerenyBod { Id = ID, Hodnota =
hodnota, Poloha = uhel });
        lbVysledky.Items.Add(uhel.ToString("F") + "°\t" +
hodnota.ToString("F") + " dB $\mu$ V");
        grafSmerova.Series["Series1"].Points.AddXY(uhel, hodnota);
        ID++;
    }
    if (uhel < uhelMin)
    {
        lbSmer.Text = "Točna stojí";
        SendSerial("n");
        timerMeasure.Enabled = false;
        timerDelay.Enabled = true;
    }
    break;
case 5:
    timerMeasure.Enabled = false; timer.Enabled = true;
    measure = 0;
    rbVlevo.Enabled = true; rbVpravo.Enabled = true;
    rbVlevo.Checked = true; rbVpravo.Checked = false;
    btStart.Text = "Start";
    break;
default:
    lbStatusy.Text = "Chyba automatu";
    break;
}

}
if (measure == 0) //výpočet velikosti hlavního laloku
{
    double uhel1 = 1000;
    double uhel2 = 1000;
    double max = 0;
    int i = 0;
    int nalezeno = 0;
    var maxik = from h in frontaHodnot orderby h.Hodnota select h.Hodnota;
    foreach (var hodn in maxik)
    {
        max = hodn;
    }

    var uhlik = from h in frontaHodnot where (h.Poloha > 0) orderby
h.Poloha select h;
    foreach (var hodn in uhlik)
    {
        if (max - hodn.Hodnota >= 3)
        {
            if (nalezeno == 0)

```

```

        {
            uhel1 = hodn.Poloha;
            nalezeno++;
        }
    }
    i++;
}
var uhlik2 = from h in frontaHodnot where (h.Poloha < 0) orderby
h.Poloha descending select h;
nalezeno = 0;
foreach (var hodn in uhlik2)
{
    if (max - hodn.Hodnota >= 3)
    {
        if (nalezeno == 0)
        {
            uhel2 = hodn.Poloha;
            nalezeno++;
        }
    }
    i++;
}
if (uhel2 == 1000) lbUhlovaSirka.Text = "Nezměřeno";
else lbUhlovaSirka.Text = Math.Abs(uhel1 - uhel2).ToString("F") + "°";
najdiNuluAutomatickyToolStripMenuItem.Enabled = true;
}
}

```

7.9 Funkce pro nastavení frekvence na Prolink-1B

Tato funkce nastaví zadanou frekvenci, na které bude prováděno měření, v Promax Prolink-1B.

```

private void btNastavK_Click(object sender, EventArgs e)
{
    nastavFreq();
}

private void nastavFreq()
{
    double fIn = (double)numeKmitocet.Value;
    int fCislo = (int)((fIn + 33.375) * 16);
    string freq = fCislo.ToString("X4");
    SendSerial("*F"+freq+"\r");
}

```

7.10 Funkce uložitToolStripMenuItem_Click

Tato funkce slouží k exportování naměřených dat do textového souboru pro další zpracování. Umožňuje navíc vybrat typ oddělovače dat.

```

private void uložitToolStripMenuItem_Click(object sender, EventArgs e)
{
    string delimiter = "\t";
    FormDelimiter delimiterDialog = new FormDelimiter();
    // Zobrazí dialog jako Modální a pokračuje jestli DialogResult = OK.
    if (delimiterDialog.ShowDialog(this) == DialogResult.OK)

```

```

{
    // přečtení zvoleného oddělovače.
    switch ((string)delimiterDialog.cbDelimiter.SelectedItem)
    {
        case "tabulátor": delimiter = "\t"; break;
        case ";": delimiter = ";"; break;
        case ".": delimiter = "."; break;
        case "mezera": delimiter = " "; break;
        default: delimiter = "\t"; break;
    }
}
delimiterDialog.Dispose();

if (dlgUlozit.ShowDialog() == DialogResult.OK)
{
    using (StreamWriter zapisovac = new StreamWriter(dlgUlozit.FileName))
    {
        zapisovac.WriteLine(umeKmitocet.Value.ToString() + "\tMHz");
        zapisovac.WriteLine("");
        zapisovac.WriteLine("Uhel [°]\tHodnota [dBuV]");
        var dataVse = from h in frontaHodnot select h;
        foreach (var hodn in dataVse)
        {
            zapisovac.WriteLine("{0}\t{1}", hodn.Poloha, hodn.Hodnota);
        }
    }
}
}
}

```

8 ZÁVĚR

Tato diplomová práce shrnula možnosti měření a automatizace měření směrových charakteristik antén. Pro tyto účely bylo navrženo zapojení pracoviště, které by mělo zaručit pohodlné měření směrových charakteristik antén

Dále byla v této práci podrobně, především pro potřeby použití jako snímače polohy, popsána optická počítačová myš, která pro komunikaci používá PS/2 konektor. Stejným způsobem byl popsán i magnetický snímač.

Při celkovém zhodnocení těchto dvou možných snímačů byl nakonec použit magnetický n-kodér, a to především díky svojí spolehlivosti odečítání správného úhlu natočení anténní točny i po provedení několika měření.

Pro účely automatizovaného měřicího pracoviště pro měření směrových antén byla navržena ovládací krabice, která obsahuje výkonovou část pro ovládání anténní točny a napájení vnitřní logiky. Dále umožňuje komunikovat nejen se snímačem polohy, ale i s PC a měřicím přístrojem. Tento ovladač je schopen pomocí tlačítek otáčet točnou bez použití PC a kalibrovat snímač polohy.

K tomuto hardwarovému návrhu byl navržen i software, který umožňuje pohodlně měřit směrové charakteristiky antén, zobrazovat naměřené data v kartézských souřadnicích, normovanou vyzařovací charakteristiku v kartézských souřadnicích a normovanou vyzařovací charakteristiku zobrazenou v polárním grafu.

V návaznosti na tuto diplomovou práci by v budoucnu mohlo být umožněno připojení více druhů měřicích zařízení k měřicímu pracovišti. Zlepšení by mohla být provedena i v části, která se zabývá exportem dat a grafů (např. do MS Excel).

LITERATURA

- [1] MIHÁLIK E. *Automatizace měření směrových charakteristik antén*. Brno, 2007. 44 s., 8 příl. Diplomová práce. Vysoké učení technické, FEKT.
- [2] DRDLA K. *Měření směrových charakteristik antén*. Brno, 2008. 44 s.. Diplomová práce. Vysoké učení technické, FEKT.
- [3] ŘEHOŘ, Z. *INKREMENTÁLNÍ ČIDLO IRC-K* [online]. [cit. 2009-05-03]. Dostupné na www: <http://astrotechnika.wz.cz/WWW/mysoid.htm>.
- [4] MAŠLAŇ, S.. *ŘÍZENÍ PS/2 MYŠI*. [online]. [cit. 2009-05-03]. Dostupné na www: http://elektronika.kvalitne.cz/ATMEL/necoteorie/tutorial/PS2/PS2_mouse.html.
- [5] Austriamicrosystems. *Datasheet AS5304, AS5306*. [online]. Austria: Austriamicrosystems, 2011 – [cit. 19. května 2011]. Dostupné na www: <http://www.austriamicrosystems.com/eng/Products/Magnetic-Encoders/Linear-Encoders/AS5304>.
- [6] Atmel. *Datasheet ATmega162, ATmega162V*. [online]. Atmel Corporation, 2009 – [cit. 19. května 2011]. Dostupné na www: http://www.atmel.com/dyn/resources/prod_documents/doc2513.pdf.
- [7] Promax. *Prolink-1B : Instruction manual*. [online]. Promax electronica, s.a. 1999 – [cit. 19. května 2011]. Dostupné na www: <http://www.promaxprolink.com/pdf/Prolink-1b%20Instruction%20Manual.pdf>.
- [8] LOUIS, D., MEJZLÍK, P., VIRIUS, M. *Jazyky C a C++ podle normy ANSI/ISO: KOMPLETNÍ KAPESNÍ PRŮVODCE*. 1.vyd. Praha: Grada Publishing. 1999. 644 s. ISBN 80-7169-631-5.
- [9] MATOUŠEK, David. *C++ Builder – vývojové prostředí : 1,díl*. 3.vyd. Praha: BEN – technická literatura, 2002. 688 s. ISBN 80-7300-064-4.
- [10] SHARP, John. *Microsoft Visual C# 2010 : krok za krokem*. 1.vyd. Brno: Computer Press, 2010. 696 s. ISBN 978-80-251-3147-3.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

2θ	Úhlová šířka hlavního laloku
AO	Analog Output
COM	Označení sériového portu v operačních systémech Microsoft
CR	znak - návrat vozíku
D_{max}	Činitel směrovosti
DPS	deska plošných spojů
G_{max}	Zisk antény
LF	znak - nový řádek
MCU	Microcontroller - Mikroprocesor
PC	Osobní počítač
SMD	Surface mount device - součástka pro povrchovou montáž plošných spojů
TTL	transistor-transistor-logic - tranzistorově-tranzistorová logika
UART	Asynchronní sériové rozhraní
USB	Univerzální sériové rozhraní počítače

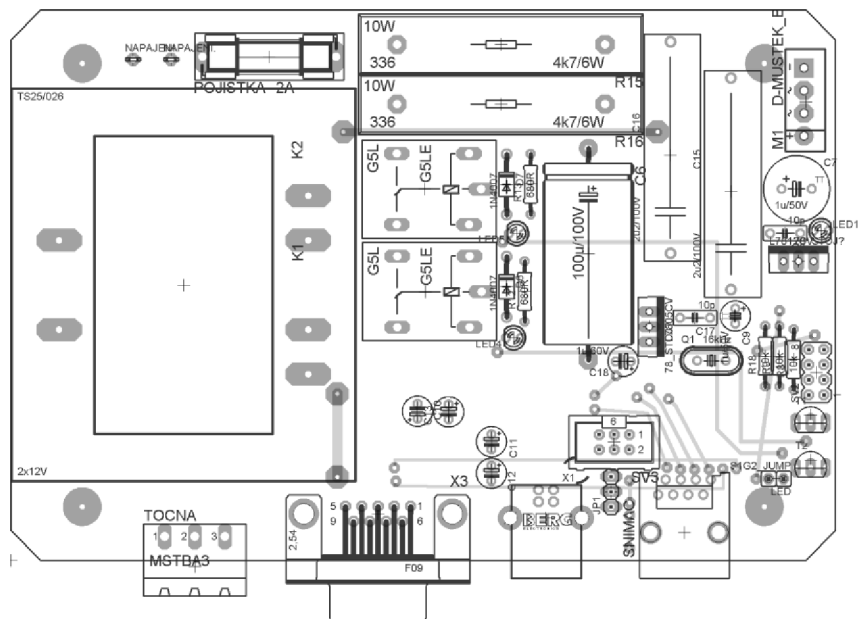
SEZNAM PŘÍLOH

A	Plošný spoj Ovladače	47
A.1	Deska plošného spoje – top (strana součástek)	47
A.2	Deska plošného spoje – zapojení bottom (strana spojů)	47
A.3	Obvodové zapojení	48
A.4	Deska plošného spoje – bottom (strana spojů)	49
A.5	Seznam součástek	49
B	Zdrojový kód pro mikroprocesor v jazyku C	51
B.1	Optický snímač	51
B.2	Obsluha magnetického n-kodéru	53

PŘÍLOHY

A PLOŠNÝ SPOJ OVLADAČE

A.1 Deska plošného spoje – top (strana součástek)



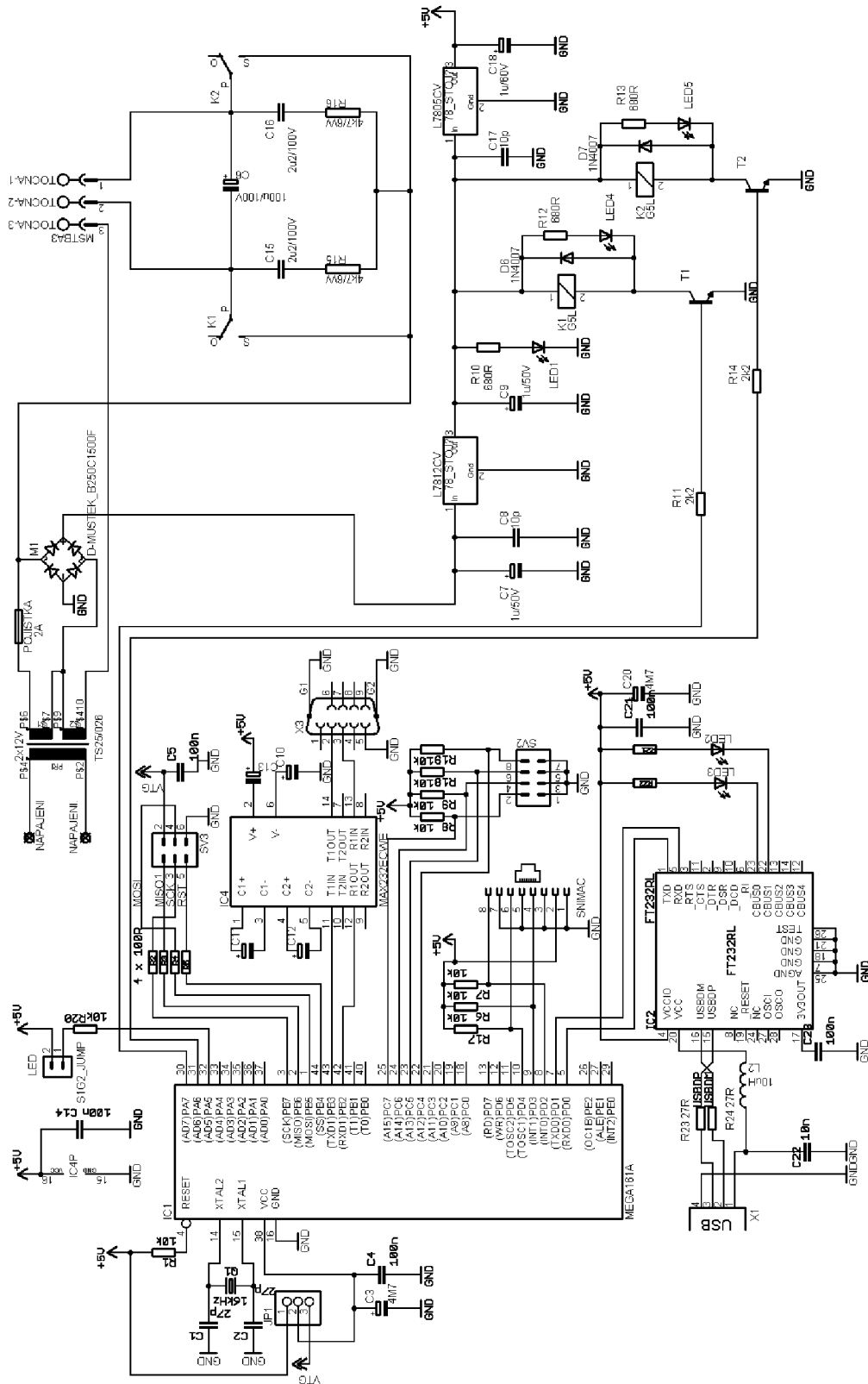
Obr. A.1: Rozměr 139,4 x 92,5 [mm].

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



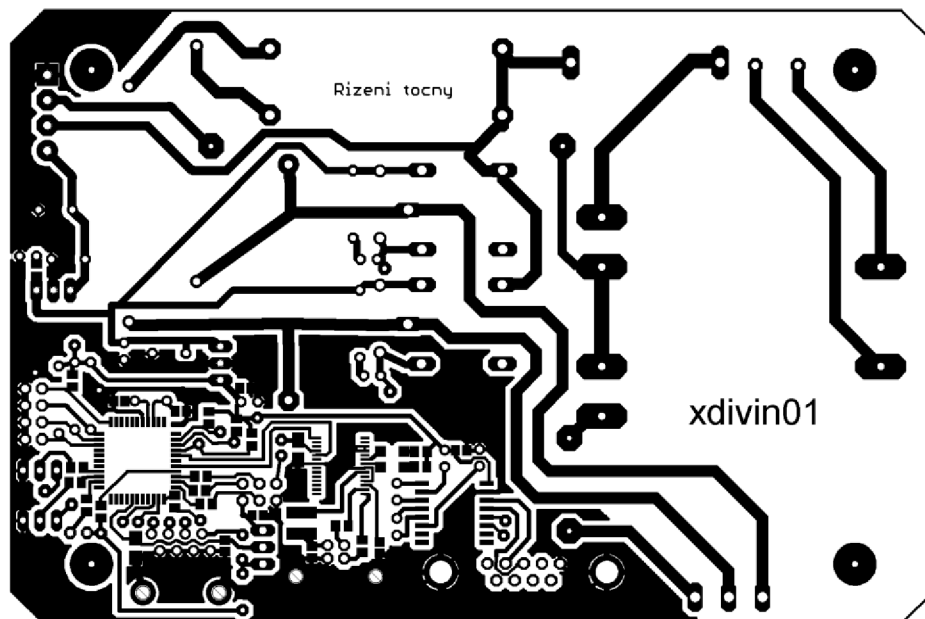
Obr. A.2: Rozměr 139,4 x 92,5 [mm].

A.3 Obvodové zapojení



Obr. A.3: Schéma zapojení ovladače

A.4 Deska plošného spoje – bottom (strana spojů)



Obr. A.4: Rozměr 139,4 x 92,5 mm.

A.5 Seznam součástek

Kondenzátory / indukčnost

C1, C2	27p
C3, C20	4M7
C4, C5, C14, C21	
C23	100n
C6	100u/100V Bipolární
C7	1u/50V
C8	10p
C9	1u/50V
C10 -C13	1u
C15, C16	2u2/100V Bipolární
C17	10p
C18	1u/60V
C22	10n
L2	10uH /MS50

Rezistory

R1, R6 -R9, R17, R19	10k /M0805
R2 -R5, R21	100R /M0805
R10	680R /M0805
R11, R16	2k2 /M0805

R20, R22	220R /M0805
R23, R24	27R /M0805
R12, R13	680R /0207/10
R14, R15	4k7/6W
R18	10k /0207/10

Diody

D6, D7	1N4007
LED	LED5MM
LED1, LED4, LED5	LED3MM
LED2, LED3	LEDCHIPLED_1206

Konektory

NAPAJENI	Konektor fast-on 2,8x0,8
NAPAJENI.	Konektor fast-on 2,8x0,8
SNIMAC	215877-4
SV2	MA04-2
SV3	ML6
TOCNA	MSTBA3
X1	PN61729
X3	F09HP
JP1	PINHD-1X3

Tranzistory

T1	BC337-NPN-TO92-EBC
T2	BC337-NPN-TO92-EBC

Integrované obvody

IC1	ATmega162-16AU
IC2	FT232RL
IC4	MAX232ECWE
L7805CV	5V stabilizátor
L7812CV	12V stabilizátor
M1	B250C1500F

Ostatní součástky

TS25/026	transformátor 2x12V
POJISTKA	2A
Q1	16kHz
K1	12 V relé RAS 1215
K2	12 V relé RAS 1215

B ZDROJOVÝ KÓD PRO MIKROPROCESOR V JAZYKU C

B.1 Optický snímač

Není zde uveden kompletní program, ale jen část co se týká komunikace se snímačem pomocí protokolu PS/2.

```
//-----SPOZDENI 120 us-----
void spozdeni_120u( void )
{
    TCNT1H = 0x08;
    TCNT1L = 0x00;
    TCCR1B |= (1 << CS10); //start CT1 pro zpozdeni 120 us
    while ( bit_is_clear(TIFR,TOV1) );
    TCCR1B &= ~(1 << CS10);
    TIFR |= (1<<TOV1); //stop CT1 pro zpozdeni
}

//-----DETEKCE HRANY A SPOZDENI-----

void spozdeni_hrany( void )
{
    while ( bit_is_clear(PIND,HODINYS) ); //detekce sestupne hrany
    while ( bit_is_set(PIND,HODINYS) );
    TCCR0 |= (1 << CS00); //Zpozdeni 16us
    while ( bit_is_clear(TIFR,TOV0) );
    TCCR0 &= ~(1 << CS00);
    TIFR |= (1<<TOV0);
}

//-----ODESLANI BAJTU DO SNIMACE-----

void posli_snimac( unsigned char znak ) //prijati pres uart
{
    GICR &= !(1<<INT0); //vypnu dalsi preruseni od hodin
    unsigned char parita;
    unsigned char i;
    // unsigned char symbol;
    spozdeni_120u(); //doporucovane zpozdeni
    PORTD &= ~( (1<<DATAS) | (1<<HODINYS) );
    DDRD &= ~( (1<<DATAS) | (1<<HODINYS) ); //nastaveni otevreného kolektoru
    DDRD |= (1<<HODINYS); //stahnu hodiny

    spozdeni_120u();
    DDRD |= (1<<DATAS); //stahne DATA do nuly - START BIT

    TCCR0 |= (1 << CS00); //prvni ochrany interval na START BIT
    while ( bit_is_clear(TIFR,TOV0) );
}
```

```

TCCR0 &= ~(1 << CS00);
TIFR |= (1<<TOV0);
DDR &= ~(1<<HODINYS); //uvolni hodiny
parita = 1;
for (i=0; i<8;i++)
{
    spozdeni_hrany();
    if (bit_is_set(znak,i) )
    {
        DDRD &= ~(1<<DATAS); //posilam1
        parita ^= 0x01;      //pocita paitu
    }
    else DDRD |= (1<<DATAS); //posila 0
}
spozdeni_hrany();
if (parita) DDRD &= ~(1<<DATAS); //odeslani parity1
else DDRD |= (1<<DATAS); //odeslani parity0
spozdeni_hrany();
DDRD &= ~(1<<DATAS); //stopbit
spozdeni_hrany();
if (bit_is_clear(PIND,DATAS) )
    PORTA ^= (1<<VLEVO); //potvrzeni od snimace
GICR |= (1<<INT0); //zapnu preruseni pro prijem
}

//-----INICIALIZACE SNIMACE-----

void snimac_Init( void )
{
    posli_snimac(0xff); //reset
    RxComplete=0;
    while (!RxComplete); //ack
    RxComplete=0;
    while (!RxComplete); //selftest OK
    RxComplete=0;
    while (!RxComplete); //id MOUSE

    //velikost kroku
    posli_snimac(0xE8);
    RxComplete=0;
    while (!RxComplete);
    posli_snimac(0x00); //velikost kroku 18/mm
    RxComplete=0;
    while (!RxComplete);

    //sample rate 20
    posli_snimac(0xF3);
    RxComplete=0;
    while (!RxComplete);
    posli_snimac(0x14);
    RxComplete=0;
    while (!RxComplete);

    //scaling 1:1
    posli_snimac(0xE6);
    RxComplete=0;
    while (!RxComplete);

    //stream mode
    posli_snimac(0xF4);
    RxComplete=0;
}

```

```

    while (!RxComplete);
    byte = 0;
}

//-----PRIJETI POMOCI PRERUSENI-----

ISR(INT0_vect) //prijeti bajtu
{
    unsigned char parita;
    unsigned char znak = 0x00;
    unsigned char i;

    GICR &= !(1<<INT0); //vypnu dalsi preruseni

    TCCR0 |= (1 << CS00); //cekani na start bit
    while ( bit_is_clear(TIFR,TOV0) );
    TCCR0 &= ~(1 << CS00);
    TIFR |= (1<<TOV0);

    if ( bit_is_clear(PIND,DATAS) ) //test start bitu
    {
        for (i=0; i<8;i++)
        {
            spozdeni_hrany();
            if (bit_is_set(PIND,DATAS) ) znak |= (1<<i);
            else znak &= ~(1<<i);
        }
        spozdeni_hrany();
        if (bit_is_set(PIND,DATAS) ) parita = 0x01; else parita = 0x00;
        spozdeni_hrany();

        prijaty[byte]=znak;
        byte++;
        RxComplete = 1;
    }

    GICR |= (1<<INT0); //zapnupreruseni int0
}

```

B.2 Obsluha magnetického n-kodéru

Zdrojový kód pro obsluhu magnetického n-kodéru.

```

//-----PRIJETI POMOCI PRERUSENI-----
-
ISR(INT0_vect) //    Prijem pulzu
{
    GICR &= ~(1<<INT0); //vypnu dalsi preruseni
        //rozeznání smyslu otáčení
        if ( bit_is_set(PIND,STAV_B) ) poloha --;
        else poloha ++;
    //    PORTA ^= (1<<VPRAVO);

    if (kalibrace == 1)//v případě kalibrace vyhledá výchozí polohu
    {
        if (TCNT3 < 250)
        {

```

```

        poloha = 0;//vynuluje polohu
        pulzu = 0;// pomocna promena
        oldPoloha = 0;//stará hodnota při průchodu mag. páru
        kalibrace = 2; //ukončení kalibrace
        PORTA &= ~(1<<VPRAVO) | (1<<VLEVO);//zastavení
točny
    }
    TCNT3 = 0;
}

GICR |= ((1<<INT0) | (1<<INT1)); //povolí INT0
}

//-----PRIJETI INDEX SIGNALU-----

ISR(INT1_vect) // příjem INDEX signalu
{
    GICR &= ~(1<<INT1); //vypnu dalsi přerušení
    //zkontroluji jestli jde opravdu o přechod přes mag. pár
    if ( (bit_is_set(PIND,0x02)) || (bit_is_set(PIND,STAV_B)) )
    {
        pulzu = poloha - oldPoloha; //kam se točna posunula
        if (pulzu > 20) poloha = oldPoloha + 40;
        else if (pulzu < -20) poloha = oldPoloha - 40;
        else poloha = oldPoloha;

        oldPoloha = poloha; n//uložení nové polohy
    }
    GICR |= ((1<<INT0) | (1<<INT1)); //povolení přerušení INT1
}

```