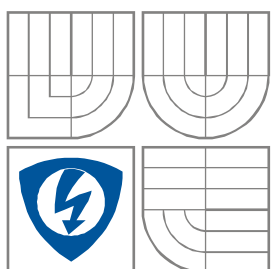


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

LADITELNÁ SMYČKOVÁ ANTÉNA PRO RADIOAMATÉRSKÉ POUŽITÍ

TUNABLE LOOP ANTENNA FOR HAM RADIO

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAROSLAV ŠMERDA

VEDOUcí PRÁCE
SUPERVISOR

Ing. ZBYNEK LUKEŠ, Ph.D.

BRNO, 2010

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Jaroslav Šmerda
Bytem: Lelekovice 317, 664 31
Narozen/a (datum a místo): 16. srpna 1987 v Brně

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
- diplomová práce
- bakalářská práce
- jiná práce, jejíž druh je specifikován jako
(dále jen VŠKP nebo dílo)

Název VŠKP: Laditelná smyčková anténa pro radioamatérské použití

Vedoucí/ školitel VŠKP: Ing. Zbyněk Lukeš, Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy
(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 28. května 2010

.....
Nabyvatel

.....
Autor

ANOTACE

Tato bakalářská práce je zaměřena na návrh a konstrukci smyčkové antény, kterou můžeme ladit pomocí kondenzátorů krokovým motorem. Anténa je konstruována pro krátké vlny využívané radioamatéry. Mým úkolem je sestavit ladění antény za pomoci čtyřpolohového krokového motoru ovládaného mikroprocesorem, který bude nastavovat krokový motor do poloh, které změni kapacitu kondenzátorů na potřebnou pro naladění antény pro dané kmitočtové pásmo.

ANNOTATION

This bachelor thesis is focused on the design and construction of loop antenna that can tune the stepper motor using capacitors. The antenna is designed for short waves used by radio amateurs. My task is to establish an antenna tuning using 4-position stepper motor controlled by a microprocessor, which will set the stepper motor to the positions that will change the capacity of capacitors needed to tune the antenna for the frequency band.

Klíčová slova:

smyčková anténa, laditelný kondenzátor, mikroprocesor

Keywords:

loop antenna, tunable capacitor, microprocessor

Bibliografická citace práce

ŠMERDA, J. *Laditelná smyčková anténa pro radioamatérské použití*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 50 s. Vedoucí bakalářské práce Ing. Zbyněk Lukeš, Ph.D.

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma Laditelná smyčková anténa pro radioamatérské použití 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 práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 4. ledna 2010

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Zbyňkovi Lukešovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 4. ledna 2010

.....
podpis autora

Obsah

1 Úvod.....	7
2 Antény.....	8
2.1 Princip činnosti	8
2.2 Dělení antén podle:	9
2.3 Typy antén.....	9
2.4 Magnetické smyčkové antény (MLA).....	11
3 Návrh obvodu.....	13
3.1 Výpočty parametrů.....	14
3.2 Zapojení a výpočet kapacitorů	16
3.3 Návrh řídicí části	18
3.4 Mikroprocesor.....	22
3.5 Programátor BiProg	27
3.6 Řízení krokového motoru a obvod ULN2803A	27
4 Návrh programu	30
4.1 Nastavení registrů	30
4.2 Funkce automatického měření	33
4.3 Funkce manuálního řízení	35
4.4 Vlastní funkce použité pro zjednodušení programu	36
5 Realizace práce	38
6 Závěr	40
7 Seznam použité literatury.....	41
8 Seznam použitých zkratk a symbolů	42
9 Přílohy.....	43
Příloha 1: DPS - Řídicí deska.....	43
Příloha 2: DPS – Deska s laditelným kondenzátorem.....	44
Příloha 3: Program pro mikroprocesor.....	46
Příloha 4: fotografie celého zapojení	50

1 Úvod

Antény jsou zařízení, která transformují elektrickou energii na vedení na elektromagnetickou energii v podobě elektromagnetické vlny šířící se prostorem, kde ji zachytává přijímací anténa a transformuje zpět na energii elektrickou. Tato práce se věnuje magnetickým anténám smyčkovým, což jsou antény využívající jen magnetickou složku elektromagnetické vlny. Protože těchto antén hojně využívají radioamatéři, budeme se zabývat anténou využitelnou pro radioamatérské účely v pásmech krátkých vln.

Samotná smyčková anténa se skládá z hlavní smyčky, vazební smyčky a ladícího kondenzátoru. Laditelný kondenzátor je laděný pomocí krokového motoru ovládaného mikroprocesorem. Ten má za úkol naladit smyčkovou anténu na rezonanční kmitočet.

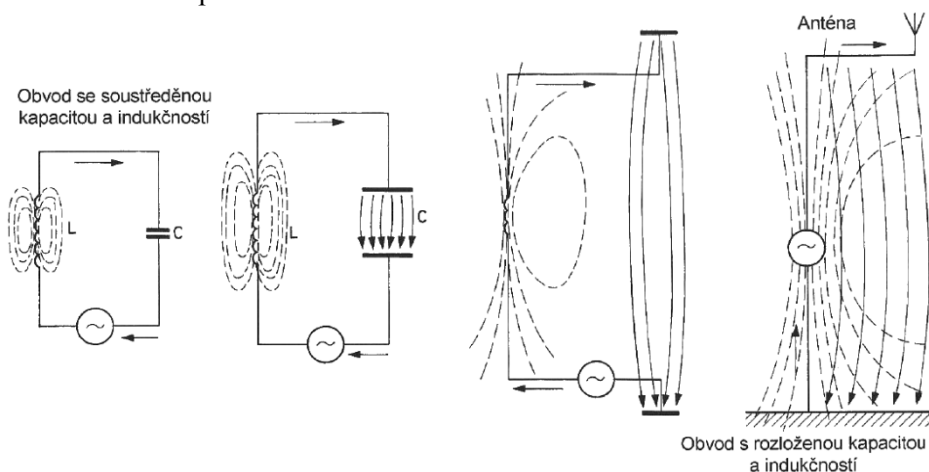
2 Antény

2.1 Princip činnosti

Anténa je zařízení, sloužící k vysílání a přijímání radiového signálu. Je nejdůležitější částí radiokomunikačního zařízení protože ji nelze nahradit žádným jiným zařízením nebo elektrickým obvodem (viz [2]). Antény přetváří elektromagnetické vlnění, šířící se podél vedení na elektromagnetické vlnění šířící se v prostoru a naopak. Pomocí této transformace vysokofrekvenční energie můžeme uskutečnit bezdrátové spojení i na velice dlouhé vzdálenosti.

Indukování elektrické energie z prostoru na vodič se velice často považuje za nechtěné a řeší se tyto případy různým stíněním, antény však této možnosti využívají a proto může být anténou i obyčejný vodič s průřezem a délkou, která určí, které elektromagnetické vlny z okolí se budou podílet a jak na výsledném proudu tekoucím anténou.

Rezonanční obvody sestavené z cívky a kondenzátoru, napájené potřebným napětím, se rozkmitají na frekvenci danou jejich parametry a vytváří tak kolem sebe slabé elektromagnetické pole. Elektrické pole vzniká mezi deskami kondenzátoru a magnetické pole se soustřeďuje okolo cívky. Toto elektromagnetické pole je však málo účinné a působí na malou vzdálenost. Vzdálenost zvýšíme oddálením desek kondenzátoru, což způsobí zmenšení kapacity, kterou musíme řešit zvětšením plochy desek kondenzátoru. Představíme-li si desky tohoto kondenzátoru jako dva vodiče, z nichž jeden je umístěný v určité výšce a druhý je umístěný na zemi nebo s ní spojený, vznikne otevřený rezonanční obvod. Vodiče tohoto otevřeného rezonančního obvodu nemají jen kapacitu ale i svou vlastní indukčnost, která je dána tím že každý vodič, kterým protéká proud, má svoji indukčnost. Pokud je daný rozměr vodičů otevřeného rezonančního obvodu srovnatelný s vlnovou délkou frekvence, na kterou je naladěn stává se účinnou anténou. Přibližuje-li se délka vodiče polovině vlnové délky, zvyšuje se účinnost zářiče v přímém směru.



Obr. 1: Uzavřený a otevřený rezonanční obvod (převzato z [2])

Hlavní parametry antén

- rezonanční kmitočet f_{rez}
- šířka pásma B
- zisk antény G
- vstupní impedance Z_{vst}
- vyzařovací odpor R
- poměr stojatých vln PSV
- předozadní poměr ČZZ/ČZP

2.2 Dělení antén podle:

1. Způsobu použití:
 - stacionární antény (trvale instalované)
 - mobilní antény (pro pohyblivé radiostanice na vozidlech, lodích letadlech)
 - přenosné antény (pro přenosné radiostanice)
2. Konstrukce:
 - drátové (zavěšené)
 - trubkové (samonosné)
 - plošné nebo prostorové (např. parabolické antény)
3. Vyzařovací charakteristiky:
 - všesměrové
 - směrové

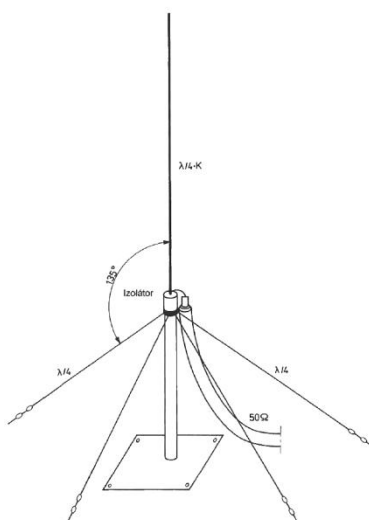
2.3 Typy antén

VŠESMĚROVÉ ANTÉNY

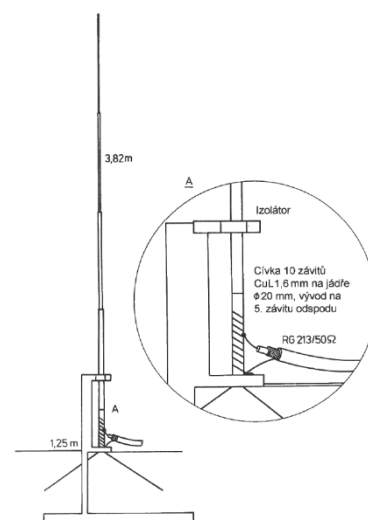
Nejjednodušší všesměrovou anténou je anténa grand plane, je to anténa, která se skládá ze zářiče o délce $\lambda/4$ a několika protiváhových prvků ze spodní strany zářiče, které s ním svírají úhel zpravidla 90° až 135° (někdy až 180°). Impedance antény je na tomto úhlu značně závislá, při 90° je impedance přibližně 20Ω , při 180° je impedance přibližně 80Ω . Zisk antény je asi -1 dB vzhledem k zisku půlvlnného dipólu.

Půlvlnný dipól je také velice populární anténou. Jak již bylo zmíněno, půlvlnný dipól je referenční anténou při měření zisku antén. Skládá se ze dvou stejně dlouhých vodičů, které mají dohromady velikost $\lambda/2$. Abychom mohli tuto anténu považovat za všesměrovou, musíme dodat, že mluvíme o svislém (vertikálním) dipólu, vodorovný (horizontální) dipól má totiž diagram záření ve tvaru osmičky. Impedance dipólu je závislá na jeho výšce nad zemí, pohybuje se kolem $50 - 70 \Omega$.

Obľíbenou všesměrovou anténou je také stacionární vertikální anténa o délce $5/8\lambda$ s protiváhou (unipól). Protiváha je tu realizována cívkou v patě zářiče. Zisk je přibližně $3,5 \text{ dB}_i$ (vzhledem k izotropickému zářiči). Izotropická anténa je teoretická, bezztrátová anténa vyzařující rovnoměrně do všech směrů v prostoru. Zvýšení zisku je možné dosáhnout tzv. kolineární anténou což je moderní řešení principiálně založené na sousém uspořádání několika dipólů.



Obr. 2: Anténa Grand plane
SMĚROVÉ ANTÉNY (převzato z [2])

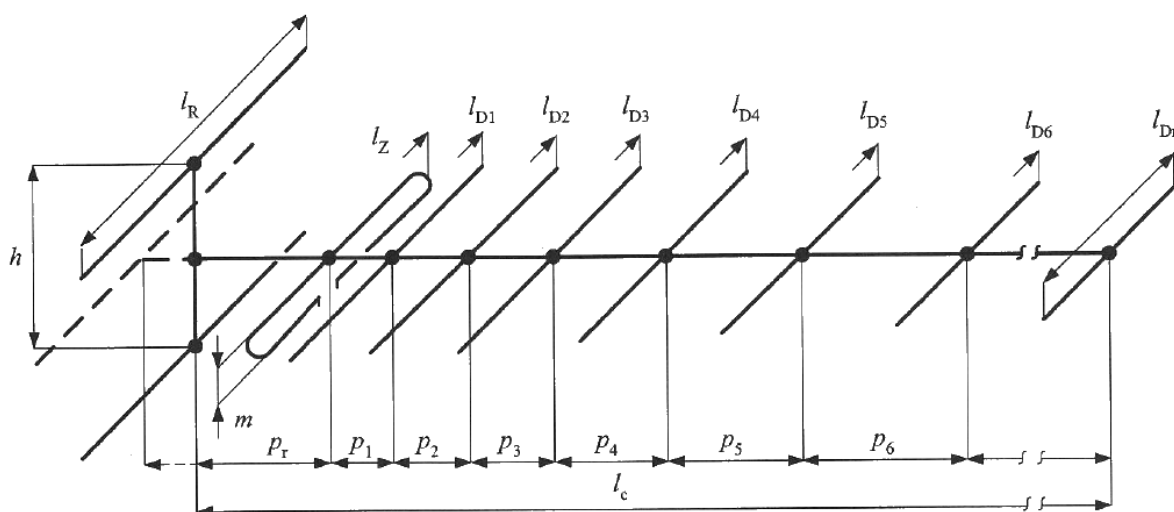


Obr. 3: Anténa $5/8\lambda$ (převzato z [2])

Nejoblíbenější a nejčastější směrovou anténou je anténa typu Yagi. Je to anténa složená z jednoho zářiče (aktivní prvek) a reflektorů a direktorů (pasivní prvky). Zářič je tvořen dipólem a je to jediný napájený prvek. Zářič je umístěn mezi reflektorem a nejbližším direktorem. Reflektor je pasivní prvek způsobující soustředění vyzařované energie do opačného směru tím, že je umístěn ve správné vzdálenosti od zářiče tak aby se v něm indukovali proudy v protifázi oproti zářiči a on tak zpět odrážel elektromagnetickou energii. Zvyšuje se tak činitel zpětného záření. Direktor je pasivní prvek umístěný před zářič ve směru záření. Způsobuje zvýšení směrovosti a zisku a ovlivňuje impedanci antény. Většinou se před zářič umísťuje větší množství direktorů s různou roztečí. Orientační zisk antény v závislosti na počtu direktorů udává tabulka:

3prvková anténa Yagi	4dB _d
6prvková anténa Yagi	9dB _d
9prvková anténa Yagi	12dB _d
15prvková anténa Yagi	15dB _d
28prvková anténa Yagi	19dB _d

Tab. 1: Tabulka zisku antén Yagi (převzato z [2])

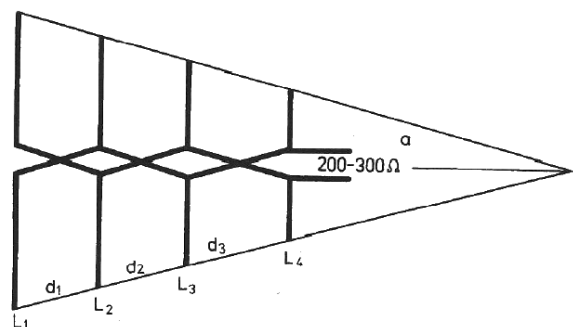


Obr. 4: Anténa typu Yagi (převzato z [2])

Není však tolik výhodné dělat příliš dlouhé antény Yagi protože jejich zisk nestoupá s počtem prvků lineárně a při dalším zvětšování by bylo zvýšení zisku zanedbatelné.

Jsou však i jiné antény založené na principu Yagiho antény, například anténa Cubical Quad, která má čtvercové (Quagi) nebo kruhové (Loop) prvky. Tyto antény jsou rozměrově menší a mají o 1 až 2 dB větší zisk než stejně dlouhé antény Yagi, jsou však konstrukčně náročnější.

Známé jsou také logaritmicko-periodické antény (LPDA). Jsou sice podobné na antény Yagi, ale všechny jejich prvky jsou napájené a fungují každý jako dipól. Každý z prvků je napájen s fázovým posuvem 180° , proto mají tyto antény dobrou směrovost. Mají také poměrně velký zisk, dobrý činitel zpětného příjmu a značnou šířku pásma. Je možné kombinovat s anténami Yagi tím, že přidáme k anténě nějaké direktory nebo reflektory.



Obr. 5: Anténa LPDA (převzato z [2])

Pro pásma KV je tu magnetická smyčková anténa, tvořená jedním nebo více závitů vodiče stočeného do smyčky tvaru kruhu nebo čtverce. Na koncích vodiče je přivedeno napájecí napětí. Vyzařovací diagram je podobný jako u dipólu orientovaného rovnoběžně s osou smyčky s minimem v ose smyčky ale jsou zaměněny vektory pole E a H. Právě na takovou anténu bude zaměřena tato práce.

2.4 Magnetické smyčkové antény (MLA)

V počátcích rozvoje rozhlasu se smyčkové antény navíjely na dřevěné rámy a používaly se jako přijímací antény, v současnosti se jako přijímací antény stále používají, ale bývají navinuty na feritové tyče. Mají dobré směrové vlastnosti. Odpor záření daný vzorcem (viz [1])

$$R_Z \approx 31,2(n \cdot A/\lambda^2)^2 \cdot 10^3 [\Omega] \quad (1)$$

kde A je plocha smyčky [m²], n je počet závitů smyčky, λ je vlnová délka [m]. To znamená, že odpor záření je závislý na těchto veličinách tak, že čím je víc závitů smyčky, tím větší je vyzařovací odpor, čím větší je plocha antény, tím větší je vyzařovací odpor. Velikost vlnové délky není volitelná, ale je závislá na přijímaném kmitočtu, takže abychom dosáhli maximálního odporu záření, musí být poměr mezi plochou A a vlnovou délkou co největší. Toho dosáhneme tím, že za tvar smyčky zvolíme kruh. Maximální napětí na svorkách udává vzorec

$$|U| = l_{ef} \cdot |E| \cdot n [V] \quad (2)$$

kde $l_{ef} = 2\pi A/\lambda$, takže je to v případě kruhu polovina z druhé mocniny obvodu smyčky, E je intenzita elektrického pole [V/m] a n je počet závitů smyčky.

Další parametry popisující magnetickou smyčkovou anténu jsou (viz [3]):

Plocha kruhové smyčky: $A = \frac{\pi}{4} \cdot D^2 [m^2] \quad (3)$

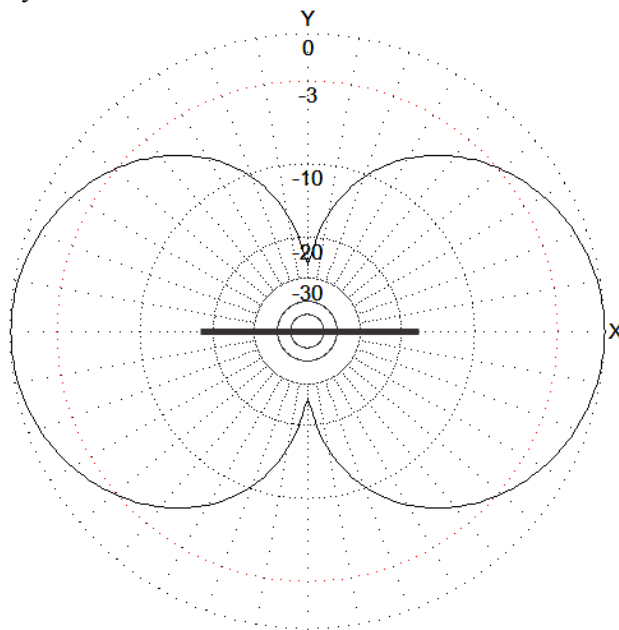
Ztrátový odpor: $R_L = 9,96 \cdot 10^{-4} \cdot \sqrt{f} \frac{U}{D} [\Omega] \quad (4)$

Účinnost: $\eta = \frac{R_Z}{R_Z + R_L} \quad (5)$

Činitel kvality: $Q = \frac{X_L}{2(R_Z + R_L)} = \frac{f}{B} \quad (6)$

Směrová funkce: $F = \sin^2 \theta \quad (7)$

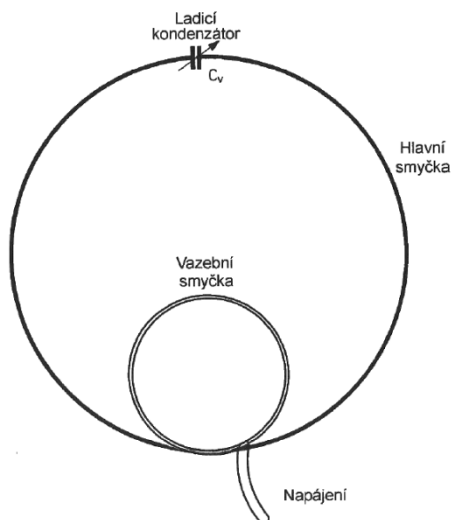
kde D je průměr hlavní smyčky [m], f je pracovní kmitočet [Hz], B je šířka pásma antény [Hz] a θ je úhel vyzařování v rovině xy .



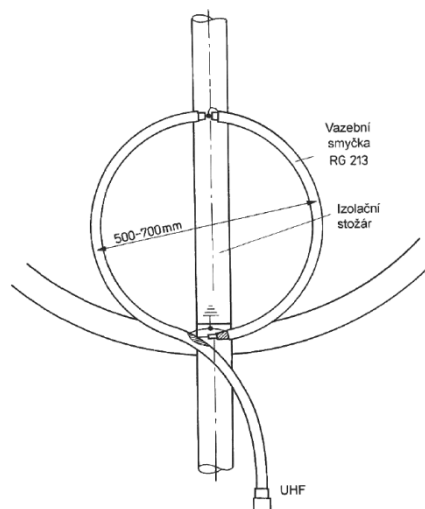
Obr. 6: Vyzařovací diagram magnetické smyčkové antény (převzato z [3])

3 Návrh obvodu

Smyčková anténa se skládá ze tří částí: hlavní smyčky, vazební smyčky a ladícího kondenzátoru.



Obr. 7: Magnetická smyčková anténa (převzato z [3])



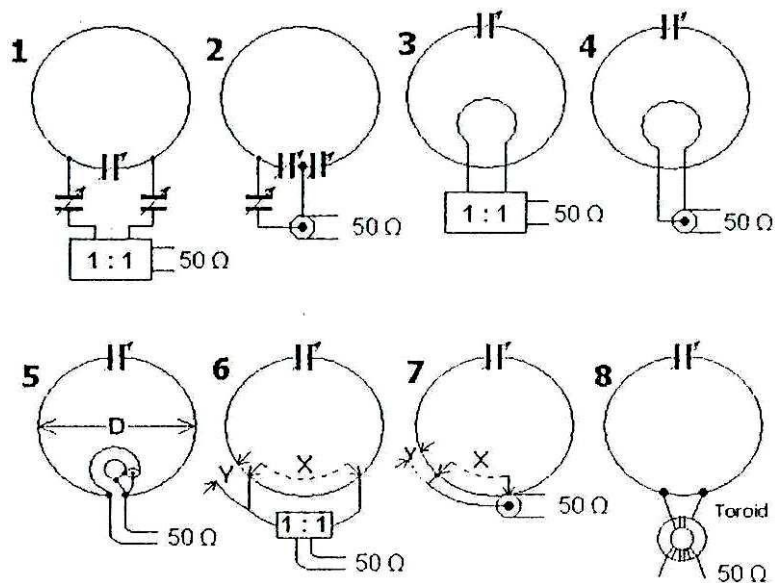
Obr. 8: Vazební smyčka (převzato z [3])

Hlavní smyčka je zářičem antény, má obvykle jeden závit ale pro delší vlny se používá závitů víc. Délka vodiče smyčky je odvozena z vlnové délky přijímaného signálu, nejčastěji 0,1 až 0,2 λ . Indukčnost smyčky je dána vzorcem (viz [3]):

$$L = 0,2 \cdot U \cdot N^2 \cdot \left(\ln \frac{U}{d} - 1,07 \right) \text{ [}\mu\text{H]} \quad (8)$$

kde L je indukčnost smyčky, U je obvod hlavní smyčky, N je počet závitů smyčky a d je průměr vodiče hlavní smyčky. Z tohoto vztahu tedy vyplývá, že čím víc má hlavní smyčka závitů, tím víc se zmenší její průměr a můžeme tak používat anténu o menších rozměrech i na nízkých kmitočtech (velkých vlnových délkách).

Vazební smyčka (někdy také indukční smyčka) je malá smyčka umístěná do hlavní smyčky. Tato smyčka má průměr většinou 5krát menší než je průměr hlavní smyčky. Umístění vazební smyčky bývá naproti ladícímu kondenzátoru ve vzdálenosti 0 až 6 cm od hlavní smyčky. Tato vzdálenost je třeba naladit spolu s ladícím kondenzátorem, protože ovlivňuje pracovní kmitočet antény a hodnotu poměru stojatých vln. Dva nejpoužívanější typy vazební smyčky jsou č. 4 a č. 7 na obrázku



Obr. 9: Typy vazební smyčky (převzato z [3])

Ladící kondenzátor je kondenzátor s proměnnou kapacitou. Má dvě části: stator a rotor. Na obou těchto částech jsou desky zasouvající se do sebe a tím měnící kapacitu kondenzátoru. Tento kondenzátor slouží k přeladění kmitočtu ve velkém rozsahu. Kapacita ladícího kondenzátoru a indukčnost hlavní smyčky tvoří rezonanční obvod antény na kmitočtu daném známým Thomsonovým vztahem. Při rezonanci dochází k velmi vysokému napětí na koncích smyčky antény a tedy i deskách použitého kondenzátoru, musí být proto brán zřetel na průrazné napětí dielektrika. Proto používáme vícesegmentové kondenzátory nebo sériové a paralelní řazení kondenzátorů.

3.1 Výpočty parametrů

délka vodiče antény $l = 7,5\text{m}$

průměr vodiče $d = 8\text{mm}$

počet závitů $N = 3$

$$\text{obvod hlavní smyčky } U = \frac{l}{N} = \frac{7,5}{3} = 2,5\text{m} \quad (9)$$

$$\text{průměr hlavní smyčky } D = \frac{U}{\pi} = \frac{2,5}{\pi} = 0,796\text{m} \quad (10)$$

$$\text{obvod vazební smyčky } U_v = \frac{U}{5} = \frac{2,5}{5} = 0,5\text{m} \quad (11)$$

$$\text{průměr vazební smyčky } D_v = \frac{U_v}{\pi} = \frac{0,5}{\pi} = 0,159\text{m} \quad (12)$$

indukčnost hlavní smyčky

$$L = 0,2 \cdot U \cdot N^2 \cdot \left(\ln \frac{U}{d} - 1,07 \right) = 0,2 \cdot 2,5 \cdot 9 \cdot \left(\ln \frac{2,5}{0,008} - 1,07 \right) = 21,04\mu\text{H} \quad (13)$$

$$\text{vlnová délka } \lambda_1 = l \cdot 10 = 75\text{m} \quad (14)$$

$$\lambda_2 = l \cdot 5 = 37,5\text{m} \quad (15)$$

$$\text{kmitočet} \quad f_1 = \frac{c}{\lambda_1} = \frac{3 \cdot 10^8}{75} = 4 \text{ MHz} \quad (16)$$

$$f_2 = \frac{c}{\lambda_2} = \frac{3 \cdot 10^8}{37,5} = 8 \text{ MHz} \quad (17)$$

abychom bez problémů dosáhli těchto hodnot, zvolíme:

$$f_1 = 3 \text{ MHz}$$

$$f_2 = 10 \text{ MHz}$$

$$\text{šířka pásma} \quad B = f_1 - f_2 = 10 - 3 = 7 \text{ MHz} \quad (18)$$

vlnové délky ze zvolených kmitočtů

$$\lambda_1 = \frac{c}{f_1} = \frac{3 \cdot 10^8}{3 \cdot 10^6} = 100 \text{ m} \quad (19)$$

$$\lambda_2 = \frac{c}{f_2} = \frac{3 \cdot 10^8}{10^7} = 30 \text{ m} \quad (20)$$

$$\text{plocha smyčky antény} \quad A = \frac{\pi}{4} \cdot D^2 = \frac{\pi}{4} \cdot 0,796^2 = 0,5 \text{ m}^2 \quad (21)$$

vyzařovací odpor

$$R_{Z1} \approx 31,2(n \cdot A/\lambda_1^2)^2 \cdot 10^3 = 31,2(3 \cdot 0,5/10^4)^2 \cdot 10^3 = 0,702 \text{ m}\Omega \quad (22)$$

$$R_{Z2} \approx 31,2(n \cdot A/\lambda_2^2)^2 \cdot 10^3 = 31,2(3 \cdot 0,5/900)^2 \cdot 10^3 = 86,7 \text{ m}\Omega \quad (23)$$

ztrátový odpor

$$R_{L1} = 9,96 \cdot 10^{-4} \cdot \sqrt{f_1} \frac{U}{D} = 9,96 \cdot 10^{-4} \cdot \sqrt{3 \cdot 10^6} \frac{2,5}{0,796} = 16,38 \text{ m}\Omega \quad (24)$$

$$R_{L2} = 9,96 \cdot 10^{-4} \cdot \sqrt{f_2} \frac{U}{D} = 9,96 \cdot 10^{-4} \cdot \sqrt{10^7} \frac{2,5}{0,796} = 29,68 \text{ m}\Omega \quad (25)$$

účinnost

$$\eta_1 = \frac{R_{Z1}}{R_{Z1} + R_{L1}} = \frac{0,702}{0,702 + 16,38} = 4,1\% \quad (26)$$

$$\eta_2 = \frac{R_{Z2}}{R_{Z2} + R_{L2}} = \frac{86,7}{86,7 + 29,68} = 74,5\% \quad (27)$$

reaktance hlavní smyčky

$$X_{L1} = 2\pi f_1 L = 2\pi \cdot 3 \cdot 10^6 \cdot 21,04 \cdot 10^{-6} = 396,6 \Omega \quad (28)$$

$$X_{L2} = 2\pi f_2 L = 2\pi \cdot 10^7 \cdot 21,04 \cdot 10^{-6} = 1322 \Omega \quad (29)$$

činitel kvality

$$Q_1 = \frac{X_{L1}}{2(R_{Z1} + R_{L1})} = \frac{396,6}{2(0,000702 + 0,01638)} = 11\,608,7 \quad (30)$$

$$Q_2 = \frac{X_{L2}}{2(R_{Z2} + R_{L2})} = \frac{1322}{2(0,0867 + 0,02968)} = 5679,7 \quad (31)$$

3.2 Zapojení a výpočet kapacitorů

Při návrhu ladění smyčkové antény ladícím kondenzátorem je třeba nejdříve znát rozsah kapacit potřebných k ladění rezonančního obvodu antény, který tvoří právě laditelný kondenzátor a indukčnost hlavní smyčky antény. Potřebné kapacity kondenzátoru spočítáme z krajních pracovních kmitočtů antény podle vzorců:

$$C_1 = \frac{1}{(2\pi f_1)^2 \cdot L} = \frac{1}{(2\pi \cdot 3 \cdot 10^6)^2 \cdot 21,04 \cdot 10^{-6}} = 133,8 \text{ pF} \quad (32)$$

$$C_2 = \frac{1}{(2\pi f_2)^2 \cdot L} = \frac{1}{(2\pi \cdot 10^7)^2 \cdot 21,04 \cdot 10^{-6}} = 12 \text{ pF} \quad (33)$$

Tyto vzorce jsou odvozeny ze známého Thomsonova vztahu (viz [4]):

$$f = \frac{1}{2\pi \cdot \sqrt{LC}} \rightarrow C = \frac{1}{(2\pi f)^2 \cdot L} \quad (34)$$

Laditelný kondenzátor, který máme k dispozici, je čtyřsegmentový. Dva segmenty mají větší rozsah a větší kapacitu, druhé dva mají menší rozsah a menší kapacitu. Maximální a minimální kapacity každého segmentu najdeme v tabulce 2. Z této tabulky můžeme vidět, že první dva segmenty mají tak malou kapacitu, že je pro naše zapojení nemůžeme použít. Použijeme proto segmenty 3 a 4.

Protože na kondenzátor bude přivedeno vysoké napětí, musíme jej chránit před průrazem. Abychom dosáhli nižšího napětí na ladícím kondenzátoru, zapojíme do série s ladícím kondenzátorem ještě jiný kondenzátor s velkým průrazným napětím. Čím větší kondenzátor zapojíme do série s ladícím kondenzátorem, tím větší bude napětí na sériovém kondenzátoru a menší napětí na ladícím kondenzátoru. Sériový kondenzátor však nemůžeme dělat příliš velký, protože tím zmenšujeme celkovou kapacitu zapojení a rozsah přeladění celkové kapacity. Kompromisem velikosti kapacity sériového kondenzátoru je zvolit jeho velikost stejně velkou jako je maximální velikost kapacity laditelného kondenzátoru. Tím se maximální napětí na laditelném kondenzátoru sníží na polovinu a rozsah ladění zůstane na dostačující úrovni.



Obr. 10: Laditelný kondenzátor

číslo segmentu	1	2	3	4
max. kapacita [pF]	23,5	23,5	337	397
min. kapacita [pF]	9,5	9,5	13	13

Tab. 2: Tabulka kapacit segmentů laditelného kondenzátoru

Z tabulky 2 vidíme, že kapacita vybraných segmentů 3 a 4 je dostatečně velká a proto zapojíme vybrané segmenty do série, kdy zmenšíme jejich ladící rozsah a jejich celkovou kapacitu. Zvýšíme tím však jemnost ladění a jejich odolnost vůči průrazu. Kapacity vypočítáme pomocí vzorce pro sériové řazení kondenzátorů.

$$C = \frac{C_1 \cdot C_2}{C_1 + C_2} \quad (35)$$

Maximální a minimální kapacitu spočítáme dosazením maximálních a minimálních hodnot kapacit segmentů.

$$C_{min} = \frac{C_{3min} \cdot C_{4min}}{C_{3min} + C_{4min}} = \frac{13 \cdot 13}{13 + 13} = 6,5 \text{ pF} \quad (36)$$

$$C_{max} = \frac{C_{3max} \cdot C_{4max}}{C_{3max} + C_{4max}} = \frac{337 \cdot 397}{337 + 397} = 182,3 \text{ pF} \quad (37)$$

Abychom dosáhli hodnoty kapacity sériového kondenzátoru přibližně stejné jako je maximální hodnota kapacity laditelného kondenzátoru, zvolíme sériovou kombinaci dvou kondenzátorů o velikosti 330 pF, dosáhneme tím i toho že jednotlivé kondenzátory nebudou tolik napětově namáhány.

$$C_S = \frac{C_{S1} \cdot C_{S2}}{C_{S1} + C_{S2}} = \frac{330 \cdot 330}{330 + 330} = 165 \text{ pF} \quad (38)$$

Vypočteme-li výsledné hodnoty maximální a minimální naladitelné kapacity a rozsah ladění:

$$C'_{min} = \frac{C_{min} \cdot C_S}{C_{min} + C_S} = \frac{6,5 \cdot 165}{6,5 + 165} = 6,25 \text{ pF} \quad (39)$$

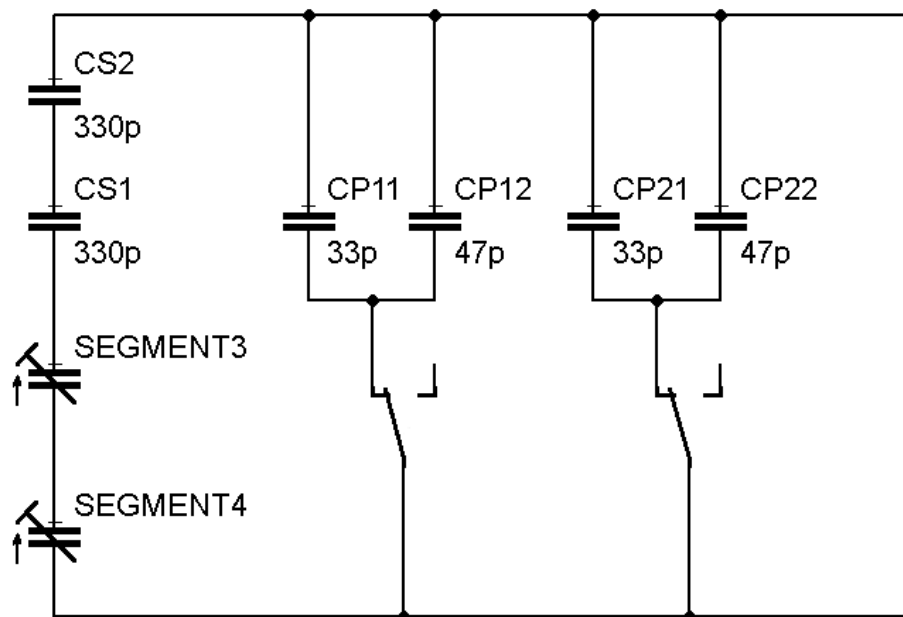
$$C'_{max} = \frac{C_{max} \cdot C_S}{C_{max} + C_S} = \frac{182,3 \cdot 165}{182,3 + 165} = 86,6 \text{ pF} \quad (40)$$

$$\Delta C = C'_{max} - C'_{min} = 86,6 - 6,25 = 80,35 \text{ pF} \quad (41)$$

Zjistíme, že rozsah ladění je příliš malý. Rozsah ladění nyní zvětšit už nemůžeme, můžeme však tento rozsah ladění posouvat na jiné hodnoty kapacit. Toho docílíme zapojením paralelních kondenzátorů k celému kondenzátorovému zapojení. Protože výsledná hodnota kapacity paralelního zapojení kondenzátorů je součet kapacit jednotlivých kondenzátorů, musí být hodnota kapacity paralelně připojeného kondenzátoru stejně velká nebo menší než rozsah ladění sériové části zapojení, nebo by mohlo dojít k tomu, že některé kmitočty v daném rozsahu by nebyli naladitelné. Abychom se mohli vracet na kapacity, které potřebujeme, budeme k původnímu sériovému zapojení zapojovat podle potřeby kondenzátory o velikosti 80 pF (jako rozsah ladění) a tak posuneme rozsah o tuhle hodnotu výš. Vzniknou nám takhle určitá pásma, která se rozloží přes celý rozsah hodnot kapacit, které potřebujeme. Potřebnou kapacitu o hodnotě 80 pF vytvoříme paralelní kombinací kapacit 33 pF a 47 pF. Rozsah kapacit pásem udává tabulka 3.

pásma	1	2	3
max. kapacita [pF]	6,25	86,25	166,25
min. kapacita [pF]	86,6	166,6	246,6

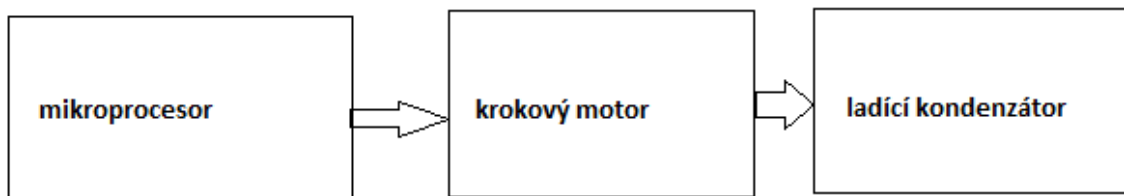
Tab. 3: Tabulka kapacit pásem



Obr. 11: Schéma zapojení laditelného kondenzátoru

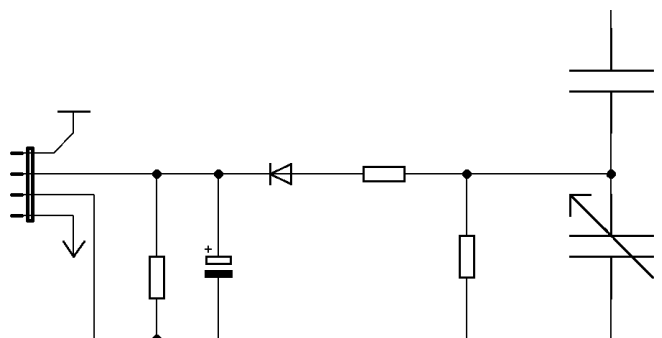
3.3 Návrh řídicí části

Ladící kondenzátor, kterým ladíme hlavní smyčku antény, je třeba nějakým způsobem řídit. Protože ladící kondenzátor ladíme jeho hřídelí a potřebujeme tuto hřídel nastavovat do různých poloh, zvolíme na nastavování hřídele krokový motor. Krokový motor bude nastavovat polohy ladícího kondenzátoru a bude řízen mikroprocesorem. Tyto části tvoří základní část ladění antény.



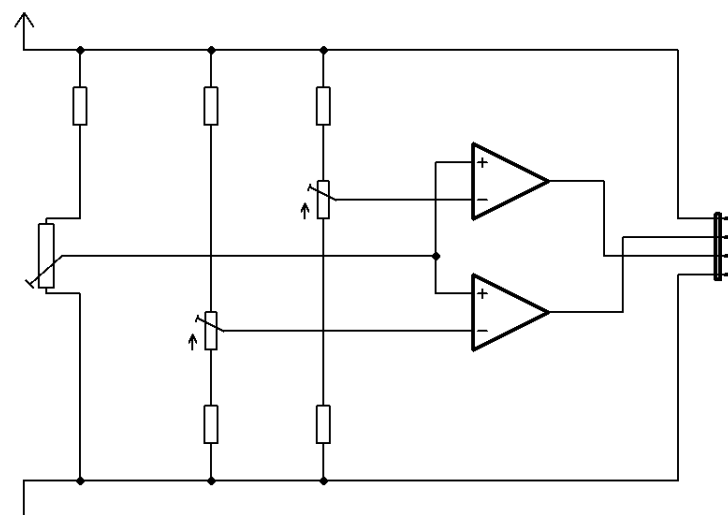
Obr. 12: Blokové schéma základních prvků ladění

Aby mikroprocesor mohl snímat úroveň přijímaného signálu, je z ladícího kondenzátoru vyveden detektor, který snímá úroveň a nechává ji zpracovat mikroprocesorem. Detektor je tvořen odporovým děličem, diodou a filtrem. Takto detekovaný signál je přiveden na konektor a přes něj na A/D převodník mikroprocesoru.



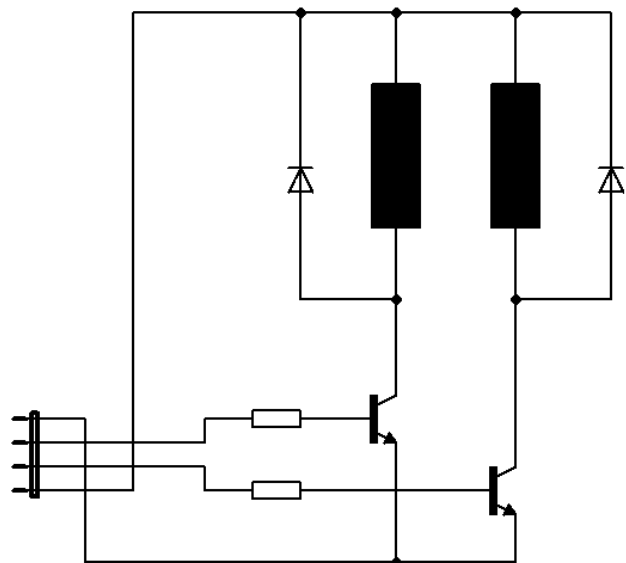
Obr. 13: Schéma detektoru úrovně signálu antény

Úroveň výstupního signálu antény však není vše, co mikroprocesor potřebuje pro ovládání ladícího kondenzátoru. Abychom věděli v jaké poloze je ladící kondenzátor, budeme jeho polohu snímat potenciometrem. Potenciometr je použit jako nastavitelný dělič napětí. Pomocí konstantního stejnosměrného napětí přivedeného na potenciometr můžeme snímat výstupní napětí potenciometru a porovnávat ho komparátory s napětím nastavitelným pomocí odporových trimrů. Na jednom trimru nastavíme napětí pro jednu krajní polohu ladícího kondenzátoru a na druhém druhou krajní polohu a výstupy z komparátorů budou přes konektory přivedeny na některé vstup-výstupní piny mikroprocesoru.

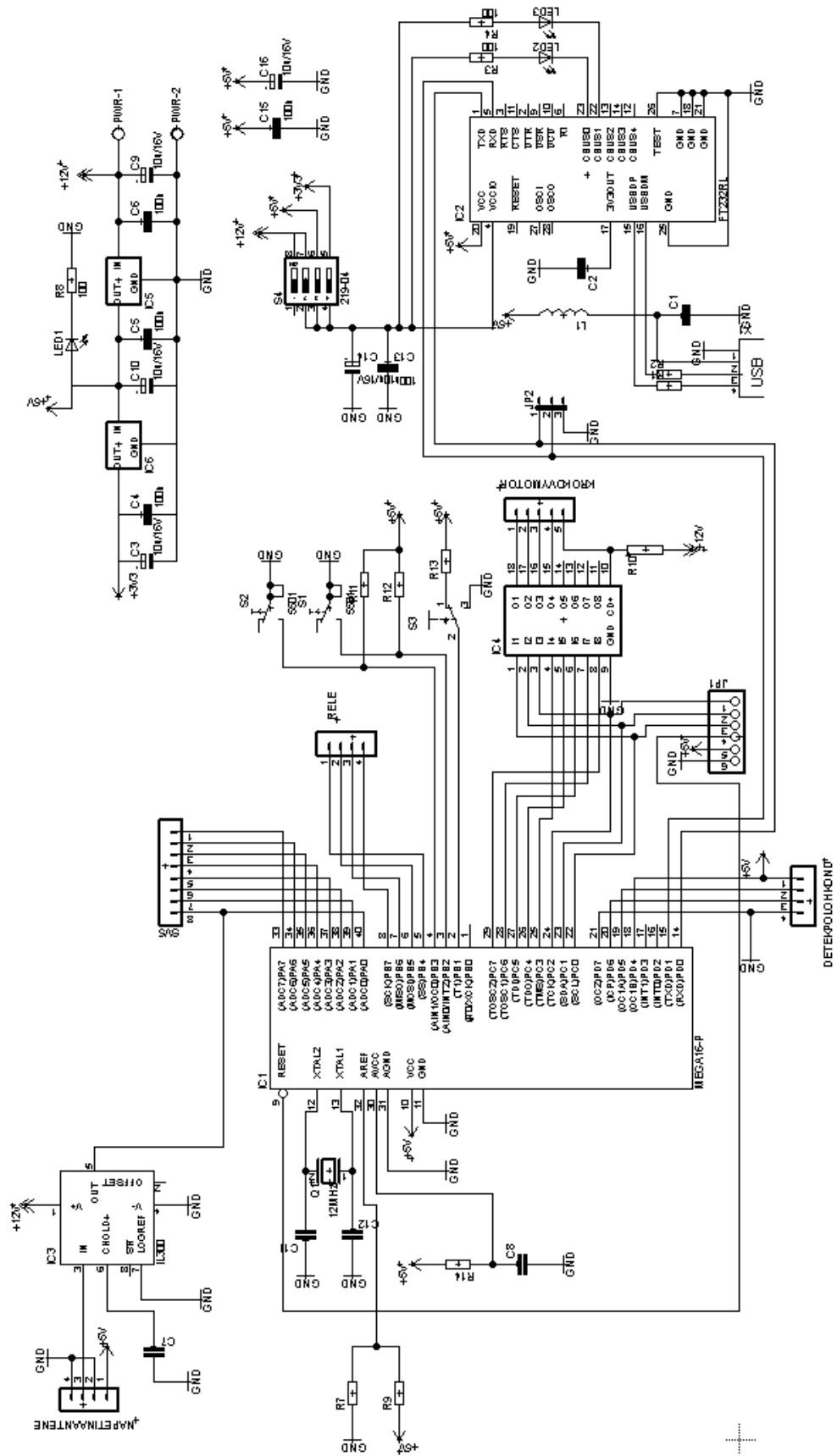


Obr. 14: Schéma snímání polohy laditelného kondenzátoru

Paralelní kondenzátory připojitelné k ladícímu kondenzátoru musí být připojované elektronicky pomocí mikroprocesoru. K tomu slouží spínací relé. Spínací proud relé musí být však větší než je výstupní proud signálu z mikroprocesoru, proto musíme řídicí proud zesílit pomocí tranzistorů. Pro použití našich tří pásem potřebujeme dva spínače (relé). Cívky relé musí být doplněny o ochranné diody.



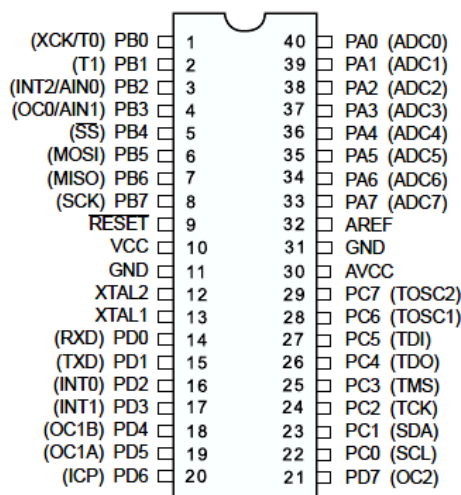
Obr. 15: Schéma zapojení cívek relé



Obr. 16: Schéma zapojení řídicí části obvodu

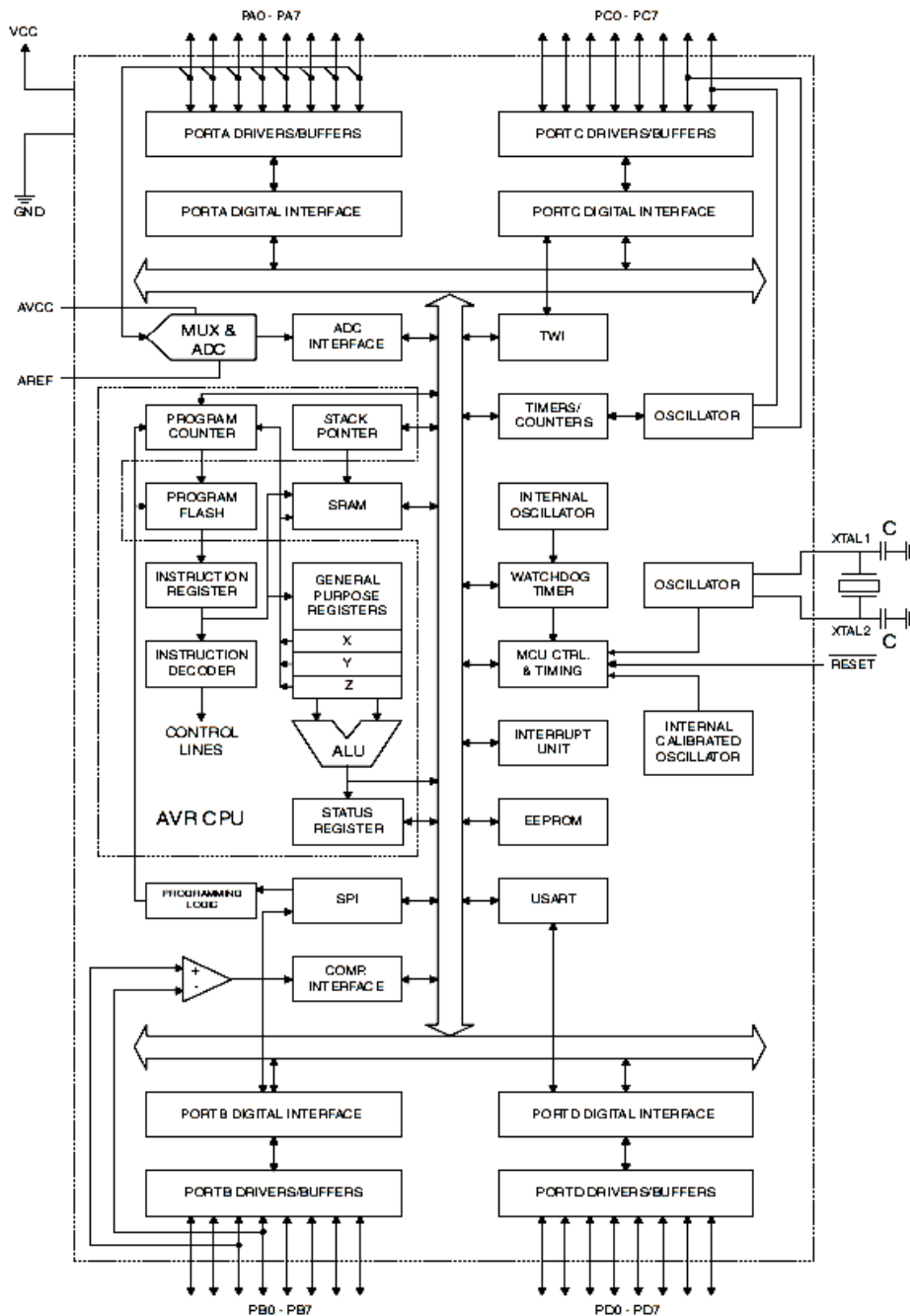
3.4 Mikroprocesor

Nejdůležitější částí celého obvodu je mikroprocesor ATmega16 (viz [5]), který řídí všechny části obvodu.



Obr. 17: Popis jednotlivých pinů procesoru ATmega16 (převzato z [5])

ATmega16 je osmibitový mikrokontroler se zvětšenou architekturou RISC, 131 výkonných instrukcí, 32 x 8 univerzálních pracovních registrů. Programové a datové paměti 16KB In-System programovatelné Flash, 512 bytů EEPROM a vnitřní SRAM velikosti 1KB. Periferní funkce: dva osmibitové čítače/časovače, jeden 16-ti bitový časovač, čítač s vlastním oscilátorem, osmikanálový desetibitový A/D převodník, sériová sběrnice I2C, USART, programovatelný časovač Watchdog a analogový komparátor. Je možnost připojit externí krystalový oscilátor. ATmega16 má 4 vstup-výstupní porty.



Obr. 18: Vnitřní blokové schéma procesoru ATmega16 (převzato z [5])

I/O Porty (viz [5])

Port A (PA7 – PA0): 8-bitový vstup/výstupní port
další funkce: Všech 8 pinů slouží jako analogové vstupy pro A/D převodník.

Port B (PB7 – PB0): 8-bitový vstup/výstupní port
další funkce: PB7 – SCK (rozhraní SPI – Bus Serial Clock)
PB6 – MISO (rozhraní SPI – Bus Master Input/Slave Output)
PB5 – MOSI (rozhraní SPI – Bus Master Output/Slave Input)
PB4 – SS (rozhraní SPI – Slave Select Input)
PB3 – AIN1 (záporný vstup analogového komparátoru)
- OC0 (porovnávací výstup z čítače/časovače 0)
PB2 – AIN0 (kladný vstup analogového komparátoru)

- INT2 (vstup externího přerušení 2)
- PB1 – T1 (externí vstup čítače/časovače 1)
- PB0 – T0 (externí vstup čítače/časovače 0)
- XCK (externí synchronizační signál rozhraní USART)

Port C (PC7 – PC0): 8-bitový vstup/výstupní port
 další funkce: PC7 – TOSC2 (pin 2 pro oscilátor k časovači)
 PC6 – TOSC1 (pin 1 pro oscilátor k časovači)
 PC5 – TDI (rozhraní JTAG – Test Data In)
 PC4 – TDO (rozhraní JTAG – Test Data Out)
 PC3 – TMS (rozhraní JTAG – Test Mode Select)
 PC2 – TCK (rozhraní JTAG – Test Clock)
 PC1 – SDA (sběrnice I2C – Data Input/Output Line)
 PC0 – SCL (sběrnice I2C – Clock Line)

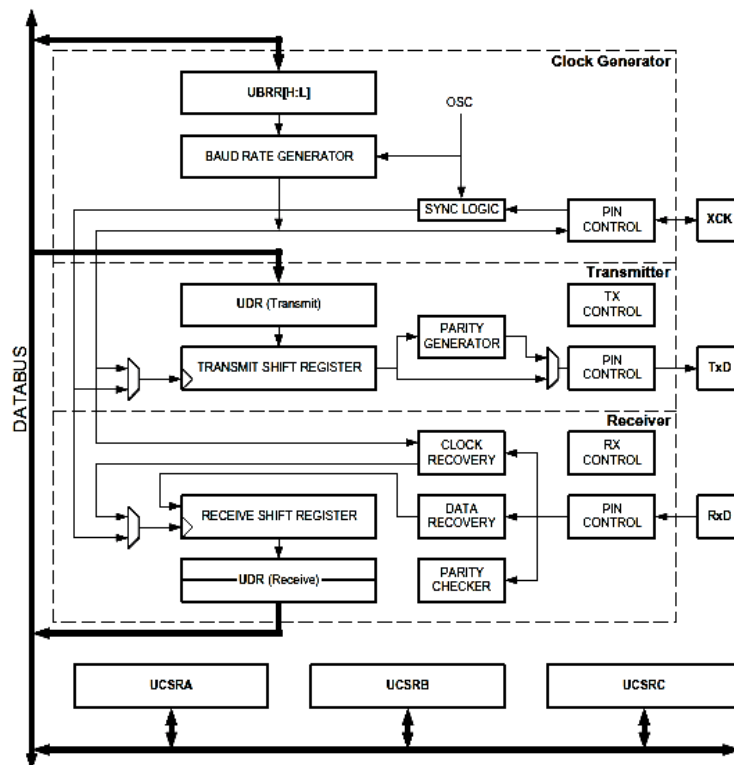
Port D (PD7 – PD0): 8-bitový vstup/výstupní port
 další funkce: PD7 – OC2 (porovnávací výstup z čítače/časovače 2)
 PD6 – ICP (vstupní pin čítače/časovače 1)
 PD5 – OC1A (porovnávací výstup A z čítače/časovače 1)
 PD4 – OC1B (porovnávací výstup B z čítače/časovače 1)
 PD3 – INT1 (vstup externího přerušení 1)
 PD2 – INT0 (vstup externího přerušení 0)
 PD1 – TXD (vysílací pin rozhraní USART)
 PD0 – RXD (přijímací pin rozhraní USART)

USART (viz [5])

- univerzální synchronní a asynchronní sériový přijímač a vysílač

Vysoce flexibilní sériové komunikační zařízení. Hlavními rysy jsou:

- plně duplexní provoz
- asynchronní nebo synchronní provoz
- synchronní provoz Master nebo Slave
- generátor pro velkou přenosovou rychlost
- podporuje sériové rámce s 5, 6, 7, 8 nebo 9 bity dat + 1 nebo 2 stop bity
- generování liché nebo sudé parity
- podpora hardwarové kontroly parity
- detekce překročení množství dat
- rámcová detekce chyb
- filtrace šumu
- tři samostatná přerušení: TX kompletní, TX datový, RX kompletní
- asynchronní komunikaci s dvojnásobnou rychlostí



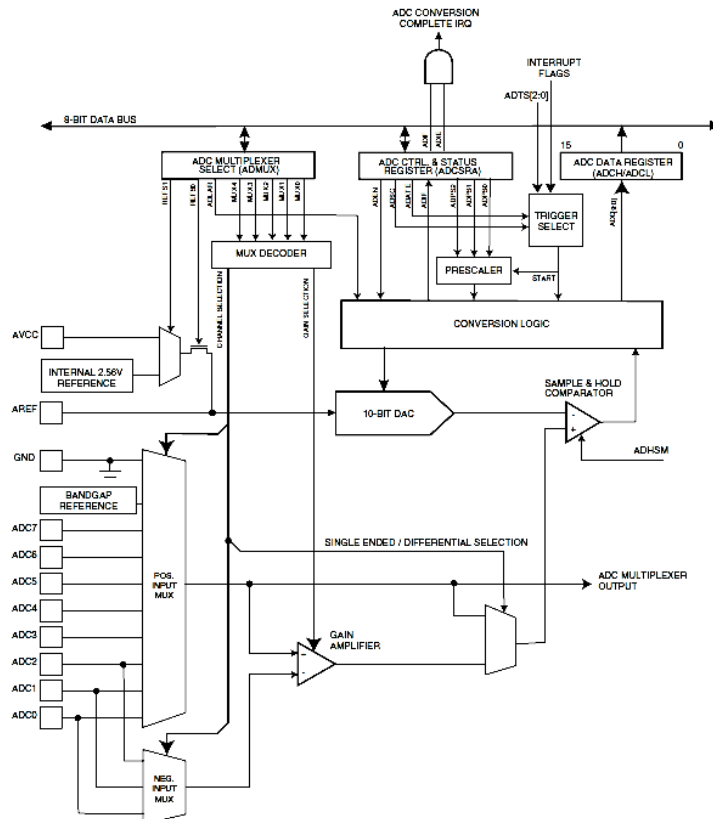
Obr. 19: Blokové schéma rozhraní USART (převzato z [5])

A/D převodník (viz [5])

-zařízení převáděcí analogový signál na digitální signál

Vlastnosti:

- rozlišení 10 bitů
- integrální nelinearita 0,5 LSB
- absolutní přesnost ± 2 LSB
- doba převodu 65 až 260ms
- rychlost až 15kS/s
- 8 samostatných vstupních kanálů
- 7 diferencíálních vstupních kanálů
- 2 diferencíální vstupní kanály s volitelným ziskem
- rozsah vstupního napětí 0 až VCC
- volitelné referenční napětí 2,56V
- spouštění převodu automatické nebo pomocí přerušení
- vyvolání přerušení po dokončení převodu



Obr. 20: Blokové schéma A/D převodníku (převzato z [5])

Analogově - číslicový převodník je v našem případě zařízení, které převádí analogovou velikost napětí snímaného na anténě na číselnou velikost, která naměřené napětí srovnává s referenční úrovní nastavenou na A/D převodníku. ATmega16 má převodník, na kterém můžeme nastavit referenční napětí na 2,56V nebo můžeme přivést referenční napětí externě na pin AREF. Tento převodník je 10-ti bitový, to znamená, že má 1024 úrovní, kterých může výsledný převod dosáhnout. Pro n-bitový A/D převodník platí rovnice (viz [13]):

$$2^n = \text{počet úrovní} \quad 2^{10} = 1024 \quad (42)$$

Z tohoto plyne rovnice pro vyjádření závislosti mezi vstupním analogovým napětím a výstupní digitální hodnotou napětí (viz [13]):

$$ADCH: L = \frac{V_{in}}{V_{ref}} \cdot (2^n - 1) \quad (43)$$

kde ADCH a ADCL jsou dva registry, do kterých se ukládá výsledná hodnota A/D převodu, V_{in} je vstupní analogové napětí a V_{ref} je nastavená referenční úroveň napětí. Protože 10-ti bitový A/D převodník má 10-ti bitovou výslednou hodnotu převodu, nemůže být tento výsledek zapsán do jediného osmibitového registru, proto jsou použity registry dva, jeden co obsahuje vrchní část desetibitového čísla a druhý spodní část desetibitového čísla. Pokud bit ADLAR registru ADMUX je nastaven na nulu (počáteční nastavení), obsahuje registr ADCH pouze vrchní 2 bity výsledného 10-ti bitového čísla a registr ADCL obsahuje spodních 8 bitů. Pokud ale nastavíme bit ADLAR na 1, bude registr ADCH obsahovat vrchních 8 bitů a registr ADCL spodní 2 bity. Doba převodu se pohybuje mezi 65 a 260ms, převodník je schopen vykonat rychlost až 15kS/s, což znamená udělat 15000 vzorků za sekundu. Samotný převod může být vyvolán automaticky programem po určitých časových intervalech nebo může být vyvolán externím přerušením, v našem případě stiskem tlačítka. Zahájením A/D převodu se vypnou povolení přerušení a povolí se opět až se A/D převod dokončí a vykoná přerušení signalizující konec převodu.

3.5 Programátor BiProg

Programátor BiProg (viz [8]) je univerzální programátor procesorů firmy Atmel řady AVR. Tento programátor slouží k ISP programování mikroprocesorů. Programuje se většinou pomocí vývojového prostředí AVR Studio.

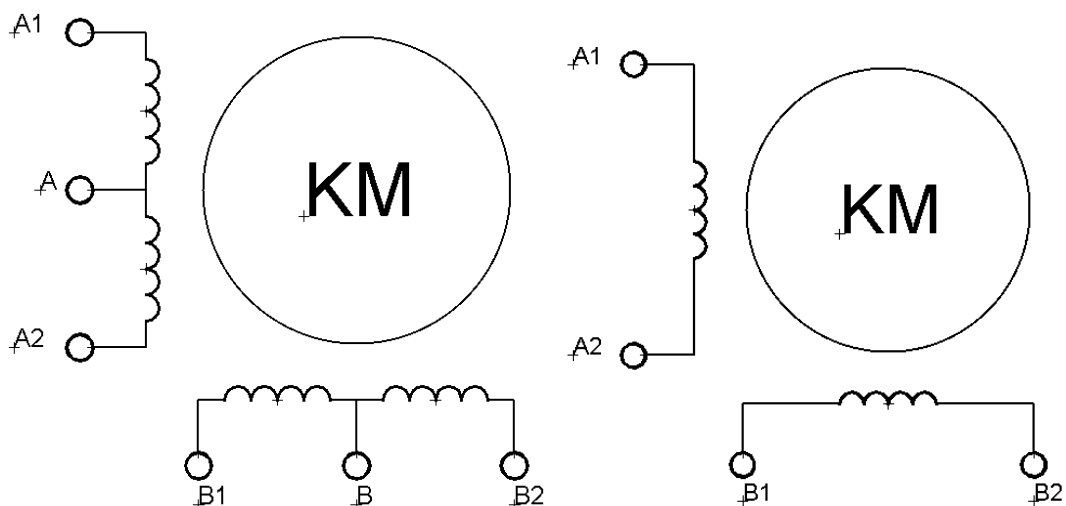
Programátor BiProg je kombinací programátorů, které jsou vytvořeny na základě aplikační poznámky AVR910, využívající především program AVRprog, a programátorů využívajících STK500 plugin AVR Studia na základě aplikační poznámky AVR068. Programátor dokáže automaticky měnit režim podle ovládacího programu, ukládá polaritu signálu RESET a poslední použitou frekvenci SCK. Je možné použít tzv. bootloader pro update. Bootloader umožňuje měnit v aplikaci obsah programové i EEPROM paměti pomocí AVRprogu. Zhotovení přípravku je jednoduché, na jednostranné desce plošných spojů.



Obr. 21: Fotografie programátoru BiProg (převzato z [8])

3.6 Řízení krokového motoru a obvod ULN2803A

Krokový motor (viz [6]) je speciální druh synchronního motoru, u něhož není tolik důležité rychlost otáčení ale nastavení výsledné polohy rotoru. Tento motor funguje na principu proudu procházejícímu cívkou, která kolem sebe vytvoří magnetické pole, které přitáhne magnet rotoru a nastaví tak rotor do správné polohy. V této poloze je schopen vydržet než se proud přepne do jiné cívky nebo změní svůj směr ve stejné cívce. Podle toho kolik má krokový motor cívek, je možné ho nastavovat po určitých krocích. Podle [6] můžeme rozdělit krokové motory podle počtu vývodů. Motory se šesti vývody – Unipolární a motory se čtyřmi vývody – Bipolární.



Obr. 22: Zapojení cívek v unipolárním a bipolárním krokovém motoru (převzato z [6])

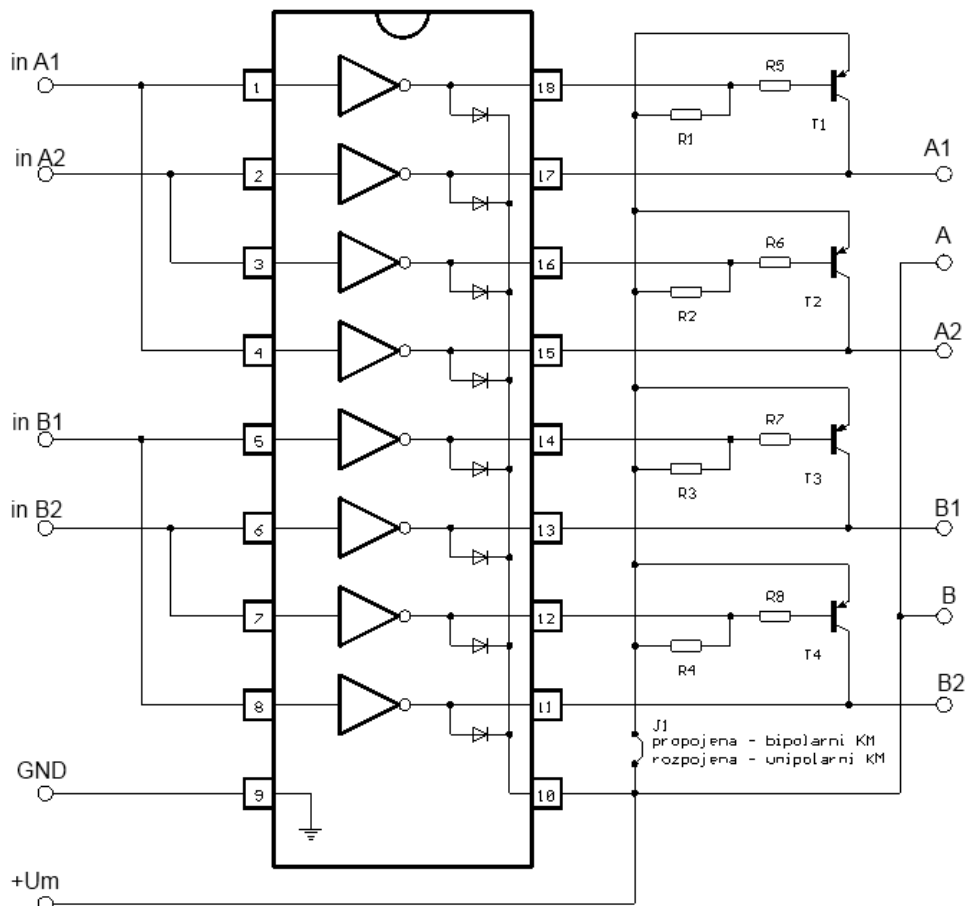
Oba typy krokových motorů se liší především buzením. Buzení unipolárního krokového motoru je realizováno jedinou polaritou napětí, kdy se napětí přivádí na jednu ze čtyř cívek statoru a podle toho se motor natočí. Bipolární krokový motor má jen dvě cívky ve statoru a proto se na jednu z cívek přivede napětí jedné nebo druhé polarizace. Existují také krokové motory s pěti vývody, jsou to ale jen unipolární krokové motory, které mají vyvedený společný střed všech cívek místo toho aby měly vyvedeny středy cívek pouze po dvou.

Pokud vezmeme unipolární motor, spojíme jeho středové vývody a připojíme na společný středový vývod 12V, můžeme pomocí minusové svorky řídit otáčení rotoru. Protože ze začátku nevíme na které poloze je rotor natočený, nemůžeme předpovídat, na kterou stranu se motor otočí. Postupným zkoušením můžeme zjistit, jak jdou kontakty po sobě a můžeme tak zjistit jak krokový motor řídit na kterou stranu. Při zkoušení tak mohou nastat tři situace: motor se otočí na jednu stranu, motor se otočí na druhou stranu nebo motor zůstane na místě. Pokud motor zůstane na místě, znamená to, že poloha, kterou jsme právě nastavili, je poslední poloha, která byla na motoru nastavena před naším nynějším připojením. Právě proto, že nemůžeme předem určit polohu krokového motoru, musíme v našem programu použít proměnnou *pol*, která tuhle polohu zaznamenává a přenáší do všech režimů a funkcí v celém programu. Pokud necháme napájení motoru na jedné pozici a pokusme se mechanicky pootočit rotorem, zjistíme, že motor drží svou pozici docela velkou silou.



Obr. 23: Fotografie použitého unipolárního krokového motoru

Řízení unipolárního krokového motoru je možné například pomocí obvodu ULN2803A (viz [6]). Tento obvod obsahuje 8 darlingtonových zapojení tranzistorů s otevřeným kolektorem, které dokáže spínat proud až 500mA a napětí do 50V. Obvod je říditelný mikroprocesorem a ovládá napájení krokového motoru. Každý výstup je opatřen ochranou diodou z důvodu spínání indukční zátěže. Schematicky je znázorněn na obrázku 24 jako osm invertorů.



Obr. 24: Zapojení obvodu ULN2803A (převzato z [6])

Tento obrázek ukazuje obvod ULN2803A v zapojení s tranzistory, které představují ve skutečnosti takzvané H – můstky, které se používají pro řízení bipolárních krokových motorů. Protože pro řízení unipolárního krokového motoru tyto H – můstky nepotřebujeme, je přidána do schématu propojka J1, kterou nastavíme, zda chceme řídit unipolární nebo bipolární krokový motor. V případě že je propojka rozpojena, je zvolený unipolární krokový motor a napětí se na emitory tranzistorů nedostanou. Pro nás je však důležité zapojení vstupů do tohoto obvodu. Připojením napětí na piny 1 a 4 nastavíme polohu A1, napětím na pinech 2 a 3 nastavíme polohu A2, napětím na pinech 5 a 8 nastavíme polohu B1, napětím na pinech 6 a 7 nastavíme polohu B2.

4 Návrh programu

Program budeme vyvíjet ve vývojovém prostředí AVR Studio (viz [11]). Je to program pro vytváření, odzkoušení a odladění programu pro mikroprocesory AVR. Můžeme s ním prohlížet stavy všech registrů, vstupů a výstupů mikroprocesoru. AVR Studio poskytuje pohodlné editační prostředí a kvalitní simulátor. Program má také možnost napojit se také na ISP programátor. Programovat tyto mikroprocesory je možné pomocí dvou programovacích jazyků: jazyka symbolických adres nebo jazyka C. Jazyk symbolických adres (JSA) neboli také assembler je nízkourovňový programovací jazyk, který je tvořen reprezentací jednotlivých strojových instrukcí, které tvoří strojový kód programu řídící všechny procesy, které se dějí v mikroprocesoru. Tento strojový kód je kód sestavený z posloupností číselných kódů. Každý kód znamená pro procesor jednu instrukci, kterou se řídí. Protože JSA jsou tvořeny zpravidla výrobci procesorů, jsou určeny jen pro konkrétní mikroprocesory a nejsou přenositelné na jiné mikroprocesory jiných typů nebo výrobců.

Jazyk C je vyšší programovací jazyk (viz [12]), který se používá pro vývoj aplikací častěji. Je oproti jazyku symbolických adres přehlednější, snadno přenositelný mezi různými architekturami, jednodušší a programování pomocí takového jazyka je rychlejší. Nevýhodou je ovšem delší výsledný strojový kód a pomalejší výkon. To je dáno tím, že převádění takového jazyka do strojového kódu probíhá nejdříve převodem do JSA a následně převodem do strojového kódu. Překladač, který překládá jednotlivé kódy mezi sebou, není schopen přeložit daný kód tak aby zároveň eliminoval části kódu, které vznikly univerzálním přeložením, aby výsledný kód fungoval pro každou aplikaci instrukce ve vyšším programovacím jazyku. Ve strojovém kódu tak vznikají nadbytečné instrukce, které zpomalují aplikaci. Přesto se vyšší programovací jazyky používají více a já jsem si jej zvolil také.

4.1 Nastavení registrů

Před samotným programováním je třeba si uvědomit, se kterými zařízeními budeme v rámci mikroprocesoru pracovat a jakými registry se tyto zařízení řídí a nastavují (viz [10]). V naší aplikaci se tedy zaměříme hlavně na registry spojené se vstup/výstupními porty, přerušováními a A/D převodem.

Vstup/výstupní porty (viz [10])

DDRn - registr pro každý port (n = A, B, C nebo D), který určuje, které bity daného portu jsou vstupní a které výstupní. Pokud je bit nastaven do 0 tak je vstupní a pokud do 1 tak je výstupní.

PORTn - registr do kterého zapisujeme v případě, že jeho bity jsou nastaveny jako výstupní. Hodnoty bitů zapsaných do tohoto registru se potom objeví na pinech mikroprocesoru. Každý port má opět svůj registr (n = A, B, C nebo D).

PINn - registr ze kterého čteme vstupní hodnoty přivedené na piny mikroprocesoru, pokud jsou dané piny nastaveny jako vstupní. Každý port má opět svůj registr (n = A, B, C nebo D).

Řízení přerušování (viz [10])

GICR – Global Interrupt Control Registr

INT1	INT0	INT2	-	-	-	IVSEL	IVCE
------	------	------	---	---	---	-------	------

INT1 – Povolení externího přerušování od INT1

INT0 – Povolení externího přerušování od INT0

INT2 – Povolení externího přerušování od INT2

IVSEL – start vektoru přerušení v bootovací sekci nebo programové paměti flash
 IVCE – nastavení umožní změnit bit IVSEL do 4 cyklů, po tuto dobu jsou přerušení neaktivní

GIFR – Global Interrupt Flags Register

INTF1	INTF0	INTF2	-	-	-	-	-
-------	-------	-------	---	---	---	---	---

INTF1 – příznak externího přerušení INT1
 INTF0 – příznak externího přerušení INT0
 INTF2 – příznak externího přerušení INT2

MCUCR – MCU Control Register

SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00
-----	----	-----	-----	-------	-------	-------	-------

Pomocí bitů ISC11 a ISC10 ovlivňujeme citlivost externího přerušení INT1.
 Pomocí bitů ISC01 a ISC00 ovlivňujeme citlivost externího přerušení INT0.
 Význam jednotlivých dvojic bitů ISC

ISCx1	ISCx0	popis
0	0	vstup INTx se aktivuje logickou 0 (úrovňově citlivý vstup)
0	1	jakákoli změna stavu INTx generuje přerušení
1	0	vstup INTx se aktivuje sestupnou hranou (hranově citlivý vstup)
1	1	vstup INTx se aktivuje náběžnou hranou (hranově citlivý vstup)

Podle toho, o které externí přerušení jde, nabývá hodnota x 0 nebo 1.

Tab. 4: Tabulka nastavení bitů ISCx1 a ISCx0 (převzato z [10])

TIMSK – masky přerušení čítačů/časovačů

OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
-------	-------	--------	--------	--------	-------	-------	-------

OCIE2 – povolení přerušení od čítače/časovače 2 při shodě
 TOIE2 – povolení přerušení od čítače/časovače 2 při přetečení
 TICIE1 – povolení přerušení při záchytu obsahu TCNT1 v registru ICR1
 OCIE1A – povolení přerušení při shodě TCNT1 s OCR1A
 OCIE1B – povolení přerušení při shodě TCNT1 s OCR1B
 TOIE1 – povolení přerušení od čítače/časovače 1 při přetečení
 OCIE0 – povolení přerušení od čítače/časovače 0 při shodě
 TOIE0 – povolení přerušení od čítače/časovače 0 při přetečení

TIFR – příznaky přerušení čítačů/časovačů

OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
------	------	------	-------	-------	------	------	------

OCF2 – nastaví se při shodě čítače/časovače 2
 TOV2 – nastaví se při přetečení čítače/časovače 2
 ICF1 – nastaví se při záchytu hodnoty TCNT1 do registru ICR1
 OCF1A – nastaví se při shodě TCNT1 s OCR1A
 OCF1B – nastaví se při shodě TCNT1 s OCR1B
 TOV1 – nastaví se při přetečení čítače/časovače 1
 OCF0 – nastaví se při shodě čítače/časovače 0
 TOV0 – nastaví se při přetečení čítače/časovače 0

TCCR_x – řídicí registr čítače/časovače x (x = 0, 1A, 1B nebo 2)

FOCx	WGMx0	COMx1	COMx0	WGMx1	CSx2	CSx1	CSx0
------	-------	-------	-------	-------	------	------	------

FOCx – v případě, že je nastaven, je generátor průběhu pro OC0 řízen přímo shodou TCNTx s OCRx

WGMx0 a WGMx1 – volba režimu generování průběhu

WGMx0	WGMx1	režim
0	0	normální
0	1	fázově korigovaný PWM
1	0	CTC
1	1	rychlý PWM

Tab. 5: Tabulka nastavení bitů WGMx0 a WGMx1 (převzato z [10])

COMx1 a COMx0 – řídí chování vývodů OCx, je-li některý z těchto bitů nastaven, je překryta normální funkce vstup/výstupních vývodů OCx. V závislosti na bitech WGMx0 a WGMx1 nastavují režimy vývodů OCx například na negace stavu vývodů při shodě, vynulování vývodů při shodě, nastavení invertujícího režimu, neinvertujícího režimu nebo odpojení vývodů od časovače.

CSx2, CSx1 a CSx0 – vybrání hodinového signálu pomocí předděličky

CSx2	CSx1	CSx0	popis
0	0	0	stop, čítač/časovač x je nastaven
0	0	1	f_{clk} – hodiny mikrokontroleru
0	1	0	$f_{clk}/8$
0	1	1	$f_{clk}/64$
1	0	0	$f_{clk}/256$
1	0	1	$f_{clk}/1024$
1	1	0	sestupná hrana Tx
1	1	1	náběžná hrana Tx

Tab. 6: Tabulka nastavení bitů CSx2, CSx1 a CSx0 (převzato z [10])

TCNTx – datový registr čítačů/časovačů x ($x = 0, 1$ nebo 2). V tomto registru je tedy aktuální stav čítače/časovače, který je srovnáván s hodnotou v registru OCRx a vyvolává událost. Pokud budeme v programu měnit obsah tohoto registru, nemusí vůbec dojít ke shodě registrů.

Řízení A/D převodu (viz [10])

ADMUX – výběr analogového kanálu

REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
-------	-------	-------	------	------	------	------	------

REFS1 a REFS0 – nastavení referenčního napětí

REFS1	REFS0	zvolený zdroj referenčního napětí
0	0	AREF
0	1	AVCC s vnějším kondenzátorem na vývodu AREF
1	0	vyhrazeno
1	1	vnitřní reference 2,56V, na vývod AREF musí být připojen kondenzátor

Tab. 7: Tabulka nastavení bitů REFS1 a REFS0 (převzato z [10])

ADLAR – zarovnáva výsledek převodu v registrovém páru ADCH a ADCL

MUX4 až MUX0 – bity sloužící pro výběr kanálu.

- Pro kombinace bitů 00000 až 00111 jde o zvolení jen jednoho kanálu pro A/D převod
- Kombinace bitů 01000 až 11101 nastavují vstupy pro diferenciální použití
- Kombinace 11110 přivede na vstup A/D převodníku napětí 1,22V
- a kombinace 11111 přivede na vstup převodníku napětí z vývodu označeného AGND.

ADCSRA – řídicí a stavový registr A/D převodníku

ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
------	------	-------	------	------	-------	-------	-------

ADEN – zapnutí A/D převodníku, vypnutí A/D převodníku předčasně ukončí převod

ADSC – start A/D převodu, nastavením spustíme jednotlivé A/D převody nebo v případě nastavení automatického převodu, spustíme postupné převádění

ADATE – povolení autospouštění A/D převodníku

ADIF – příznak přerušení A/D převodníku, nastaví se, když je dokončen převod a v registrech ADCH a ADCL je uložen výsledek převodu

ADIE – povolení přerušení A/D převodníku, je-li nastaveno, na konci převodu je vyvoláno přerušení

ADPS2 až ADPS0 – předdělička hodinového signálu pro A/D převodník

ADPS2	ADPS1	ADPS0	dělicí faktor
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Tab. 8: Tabulka nastavení bitů ADPS2, ADPS1 a ADPS0 (převzato z [10])

ADCH + ADCL = ADCW – dvojregistr do kterého se ukládá výsledek A/D převodu. Výsledek má 10 bitů, proto je potřeba ukládat jej do dvou registrů. Pokud je bit ADLAR registru ADMUX nastaven na nulu, desetibitová hodnota výsledku převodu zůstává zarovnána do nižšího registru ADCL a zbývající dva bity jsou uloženy v ADCH. Pokud je bit ADLAR nastaven na 1 zarovnáva se desetibitová hodnota výsledku převodu do vyššího registru ADCH a nejnižší dva bity jsou uloženy v registru ADCL.

Programování začneme tím, že zjistíme, jestli přepínač je v poloze AUTOMATICKÉ MĚŘENÍ nebo v poloze MANUÁLNÍ ŘÍZENÍ. Na obě tyto možnosti vytvoříme vlastní funkce.

4.2 Funkce automatického měření

Tato funkce má za úkol nastavit laditelný kondenzátor do počáteční polohy, krok po kroku nastavovat jednotlivé polohy, snímat a zaznamenávat jednotlivé hodnoty napětí na anténě, ze všech naměřených hodnot určit tu největší a nastavit kondenzátor do této polohy.

Nastavení kondenzátoru do počáteční polohy není příliš velký problém, protože máme detektory krajních poloh kondenzátoru. Nastavíme tedy kondenzátor do první krajní polohy, vypneme obě relé a dostaneme se tak do počáteční polohy.

```

while (bit_is_set(PIND,4))      //posunuti motoru na zacatek
{
    pol = posunmotor1 (pol);
}
PORTB = 0b10000000;          //vypni obe rele

```

Cyklus *while* se opakuje tak dlouho, dokud je splněna podmínka. Podmínka: *bit_is_set* nám říká, že dokud je daný bit nastavený, opakuj cyklus. Vynulování vstupního bitu zajistí detektor krajních poloh kondenzátoru. Proměnná *pol* je vždy nastavena na aktuální polohu motoru, to je důležité, aby program věděl, kam bude motor následně otočen. Funkce *posunmotor1* je funkce která na základě proměnné *pol* otáčí motor proti směru hodinových ručiček. Nastavení registru PORTB na danou hodnotu vypínáme obě relé.

Pokud máme nastavený kondenzátor do počáteční polohy, můžeme začít nastavovat polohy kondenzátoru a odečítat napětí při všech polohách kondenzátoru.

```

for (m = 0; m<13; m++)
{
    if (!bit_is_set(PIND,5))
    {
        p[m][0] = sejminapeti (m,0);    //zaznam vseh hodnot
        pol = posunmotor2 (pol);
    }
}

```

Tato část zdrojového kódu obsahuje cyklus *for*. Tento velmi používaný cyklus se opakuje tak dlouho, dokud je splněna podmínka *m<13*, která nám říká, že se daný cyklus bude opakovat vždy právě 13x. Toto číslo má svou důležitost v počtu sejmutých napětí při stejném stavu obou relé. Hnaná hřídel laditelného kondenzátoru se totiž ze své jedné krajní polohy do druhé krajní polohy otočí o 1080° (3x360°). Otáčíme-li hřídel po 90°, otočíme ji tak právě 12x. Přičteme-li sejmutí hodnoty napětí ještě na výchozí pozici, dostaneme číslo 13.

Následuje podmínka *if*, která říká, že pokud je podmínka splněna, vykoná danou posloupnost instrukcí. V podmínce najdeme proměnou *p*. Proměnná *p* je typu pole, vidíme, že jedna souřadnice je daná proměnnou *m*, která udává polohu laditelného kondenzátoru. Druhá souřadnice je rovna nule. Tahle souřadnice totiž udává počet sepnutých relé. V následující části kódu jsou postupně zapnuta obě relé a s každým zapnutým relé je zvýšena tahle souřadnice. Samotná proměnná *p* slouží k ukládání naměřených hodnot napětí na měřené anténě. Do téhle proměnné zapisuje funkce *sejminapeti*, která má za vstupní parametry také souřadnice dané polohy kondenzátoru a relé. Funkce *posunmotor2* je obdobou již zmíněné funkce *posunmotor1*, funkce *posunmotor2* ale posouvá polohu motoru po směru hodinových ručiček, tedy opačným směrem než funkce *posunmotor1*. Podmínka *if* kontroluje nastavení krajní polohy kondenzátoru pomocí detektoru krajní polohy, je-li podmínka porušena, došli jsme na druhou krajní polohu a můžeme pokračovat dál v dalších instrukcích.

```

int v,i,j;
for (n = 0; n<3; n++)
{
    for (m = 0; m<13; m++)
    {
        if ( p[m][n]>v)
        {
            v = p[m][n];    //nejvetsi hodnota napeti
            i = m;         //jeji souradnice m
            j = n;         //jeji souradnice n
        }
    }
}

```

Pokud jsme prošli všechny polohy laditelného kondenzátoru a odečetli jsme na nich všechna napětí, je čas na zpracování výsledků a určení největší naměřené hodnoty, k tomu slouží tahle část zdrojového kódu. Porovnává velikosti naměřených hodnot a tu největší запиše a srovnává s ní další hodnoty. Byli tu vytvořené nové proměnné typu *int*, je to proměnná *v*, do které je uložena největší hodnota napětí a hodnoty *i* a *j* představující souřadnice této největší hodnoty. Souřadnice *m* a *n* jsou pracovní, proto je nemůžeme použít pro záznam souřadnic této hodnoty.

```

switch (j)                                //nastaveni souradnice n
{
    case 0:
        PORTB = 0b10000000;                //vypni obe rele
        break;
    case 1:
        PORTB = 0b11000000;                //zapni jedno rele
        break;
    case 2:
        PORTB = 0b11100000;                //zapni obe rele
        break;
    default:
        break;
}
for (m = 0; m<=i; m++)                     //nastaveni souradnice m
{
    pol = posunmotor2 (pol);                //posun motor na max napeti
}

```

Protože už máme nejdůležitější výsledek – souřadnice polohy *i* a *j* s největším napětím, zbývá nám jen poslední věc a tou je nastavení daných souřadnic. To provedeme tak, že podobně jako na začátku našeho programu, posuneme krokový motor do výchozí polohy a napíšeme výše uvedenou část. Na téhle části zdrojového kódu se nachází přepínač *switch* a známý cyklus *for*. Přepínač *switch* slouží k tomu, aby udělal potřebnou posloupnost instrukcí podle toho, jakou hodnotu má proměnná v podmínce přepínače. V našem případě bere souřadnici *j* největší hodnoty napětí, která představuje počet zapnutých relé. Cyklus *for* nastavuje potom druhou souřadnici největší hodnoty napětí.

```

while(1)                                    //nekonecna smycka
{
    if (!bit_is_set(PINB,6))
    {
        pol = manual (pol);                 //prepnuti na manualni rizeni
    }
}

```

Když je všechno nastaveno, je tahle část programu zakončena nekonečnou smyčkou, která kontroluje, jestli není nastavený přepínač na manuální ovládání, aby se program přesunul do jiné části zdrojového kódu. Až se program přepne do manuálního řízení, bude potřebovat znát aktuální polohu krokového motoru, aby ji nemusel zjišťovat přesunováním se zpět do výchozí polohy, proto je důležité vložit do funkce *manual* vstupní proměnnou *pol*.

4.3 Funkce manuálního řízení

Protože součástí přípravku není žádná zobrazovací jednotka, slouží funkce manuálního řízení pouze k otáčení krokového motoru na jednu nebo druhou stranu.

```

int manual (int pol)           //funkce manualního řízení
{
    while (1)
    {
        if (bit_is_set(PINB,4)) //je-li stisknuto tlačítko
        {
            pol = posunmotor1 (pol); //posun motor 1
        }
        if (bit_is_set(PINB,5)) //je-li tisknuto tlačítko
        {
            pol = posunmotor2 (pol); //posun motor 2
        }
        if (bit_is_set(PINB,6)) //je-li prepnut prepínač
        {
            pol = automatic (pol); // spust funkce automatického nastavení
        }
    }
    return pol;
}

```

Funkce manuálního řízení tak kontroluje, jestli je zmáčknuto některé tlačítko nebo jestli je přepnut přepínač. Sepnutí jednoho tlačítka znamená otočení motoru v jednom směru pomocí funkce *posunmotor1* a sepnutí druhého tlačítka znamená otočení motoru v druhém směru pomocí funkce *posunmotor2*. Přepnutí přepínače způsobí návrat na začátek funkce automatického měření.

4.4 Vlastní funkce použité pro zjednodušení programu

Celým popisem tohoto programu nás provází několik funkcí, které jakoby automaticky vyřešili problém daný určitou částí programu. Místo toho abychom museli zjišťovat, jak je nastaven krokový motor, abychom jej mohli posunout ve směru, který chceme, používáme funkce *posunmotor1* a *posunmotor2*, které nám bez problémů posunou krokový motor směrem, který potřebujeme. Dále funkce *sejminapeti*, která zajistí odečtení napětí z pinu mikroprocesoru a zapíše je do potřebné proměnné.

Funkce Posunmotor1

Jak už víme, tahle funkce slouží k posouvání krokového motoru proti směru hodinových ručiček o jeden krok, v našem případě o 90°.

```

int posunmotor1 (int pol)     //funkce posunutí směrem 1
{
    switch (pol)
    {
        case 0b10010000:
            pol = 0b00001001; //z a1 do b1
            break;
        case 0b00001001:
            pol = 0b01100000; //z b1 do a2
            break;
        case 0b01100000:
            pol = 0b00000110; //z a2 do b2
            break;
        case 0b00000110:
            pol = 0b10010000; //z b2 do a1
            break;
        default: break;
    }
    PORTC = pol;
    return pol;
}

```

Jediná proměnná, která nás v téhle funkci zajímá je proměnná *pol*, která udává aktuální polohu motoru, proto je tato proměnná vstupní i výstupní veličinou této funkce. Nejdůležitější část této funkce je přepínač *switch*, který podle aktuální polohy krokového motoru v podobě dané sekvence jedniček a nul rozhodne o následující poloze v podobě jiné dané sekvence. Následující poloha vybraná přepínačem *switch* je uložena zpět do proměnné *pol* a následně také do výstupního registru ovládacího krokový motor.

Funkce Posunmotor2

Tato funkce je podobná funkci *posunmotor1*. Funkce *posunmotor1* posouvala rotor motoru proti směru hodinových ručiček, funkce *posunmotor2* posouvá rotor motoru po směru hodinových ručiček.

```
int posunmotor2 (int pol)          //funkce posunutí směrem 2
{
    switch (pol)
    {
        case 0b10010000:
            pol = 0b00000110;    //z a1 do b2
            break;
        case 0b00001001:
            pol = 0b10010000;    //z b1 do a1
            break;
        case 0b01100000:
            pol = 0b00001001;    //z a2 do b1
            break;
        case 0b00000110:
            pol = 0b01100000;    //z b2 do a2
            break;
        default: break;
    }
    PORTC = pol;
    return pol;
}
```

Jediná proměnná v této funkci je opět proměnná *pol*. Rozdíl mezi těmito funkcemi je jen vzájemné přepínání mezi sekvencemi jedniček a nul, znamenajících nastavenou polohu motoru. Všimněme si například, že funkce *posunmotor2* přepíná z polohy A1 do polohy B2, zatímco funkce *posunmotor1* přepíná do polohy B2 z polohy A2 a z polohy B2 přepíná na polohu A1, což je právě naopak. To znamená, že přepínání poloh v obou směrech funguje.

Funkce sejminapeti

Funkce *sejminapeti* slouží k odečtení napětí z pinu mikroprocesoru a zapsání jeho hodnoty do potřebné proměnné.

```
int sejminapeti (int m, int n, int *up)    //funkce zaznamu hodnoty napeti
{
    ADCSRA = (1<<ADSC);
    while (!bit_is_set(ADCSRA,4)){
        *up = ADCW;
        *(up+1);
        return *up;
    }
}
```

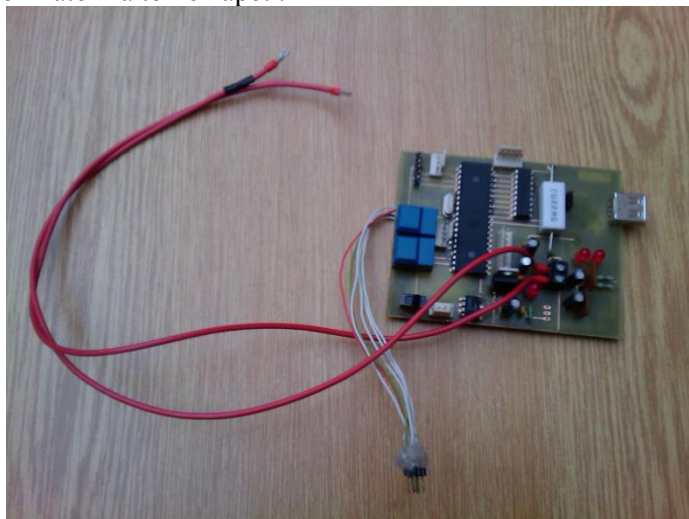
Vstupní proměnné této funkce jsou dvě souřadnice *m* a *n*, na kterých se bude napětí odečítat a jeden ukazatel na proměnnou do které se bude hodnota napětí ukládat. Ukládání pomocí ukazatelů bylo nutné zavést, protože není možné dát jako vstupní proměnnou pole, do kterého se údaje zapisují. Na prvním řádku funkce je instrukce přiřazení logické 1 do bitu ADSC (analog to digital start conversion) registru ADCSRA. Tento registr slouží k řízení vlastního A/D převodu. Následující cyklus *while* slouží k tomu, aby čekal, než bude nastavený bit číslo 4 v registru ADCSRA. Tento bit se nazývá ADIF a slouží k indikaci ukončení A/D převodu. Než A/D převodník dokončí převod, cyklus *while* opakuje svoje prázdné tělo cyklu. Když je A/D převod hotový, hodnota napětí uložená v registrech ADCH a ADCL je uložena na adresu v ukazateli. Výsledek převodu je v programu uložen v registru ADCW, tento registr je vlastně dvojregistr obsahující registry ADCH a ADCL. Na následujícím řádku je adresa uložena v ukazateli zvětšená o 1, to zajistí uložení následující hodnoty do pole *p* do další volné pozice.

5 Realizace práce

Konstrukce je sestavena na dvou deskách plošných spojů. Jedna deska má název *Řídící deska* a druhá je *Deska s kondenzátorem*. Tyto desky jsou spojeny propojovacími kabely.

Řídící deska je oboustranná deska plošných spojů. Její hlavní částí je mikroprocesor ATmega16, která ovládá krokový motor. Mikroprocesor má ještě další dva druhy vstupů, jsou to signály z detekce polohy kondenzátoru a dvě tlačítka s přepínačem. Má také ještě jeden výstup a to pro řízení spínání relé.

Při návrhu této desky byly použity sítě spojů napětových úrovní 12V, 5V a zem. Dvanácti voltová síť musela být zavedena z důvodu napájení krokového motoru a napájení obvodu LF398A, což je obvod Samle and Hold, tzn., že vzorkuje příchozí signál a vzorek, který zachytí, podrží po nějakou dobu, aby mohl být dále zpracován dalšími částmi obvodu, v našem případě mikroprocesorem. Pět voltová napájecí síť slouží k napájení mikroprocesoru a jako spínané napětí na tlačítkách a přepínači a je také vedena propojovacími kabely do druhé desky. K mikroprocesoru je připojen převodník z USART na USB, který se nazývá RS232. Tento obvod může být napájen 12V, 5V nebo 3,3V, proto byla na desce sestavena i síť pro 3,3V, tato síť ale není nezbytná a proto nebyl na desku připojen ani stabilizátor na tohle napětí.



Obr. 25: Fotografie *Řídící desky* s mikroprocesorem

Deska s kondenzátorem je oboustranná deska plošných spojů stejně jako deska řídicí. Hlavní částí je velký laditelný kondenzátor, který je řízený krokovým motorem. Laditelný kondenzátor má přední a zadní hřídel, na přední hřídel je připojen krokový motor a na zadní hřídel je připojen potenciometr. Pomocí snímaného napětí na tomto potenciometru můžeme určit polohu laditelného kondenzátoru a detekovat tak jeho krajní polohy pomocí nastavitelných odporových trimrů. Na této desce jsou tři elektrické obvody zvlášť napájené z řídicí desky napětím 5V. Všechny tři obvody jsou vzájemně propojeny pouze mechanicky. Obvod pro detekci krajních poloh laditelného kondenzátoru je s obvodem kondenzátorů spojen jen hřídelí potenciometru a obvod spínání relé je s obvodem kondenzátorů spojen právě pomocí relé, které spíná paralelní kondenzátory k laditelnému kondenzátoru.



Obr. 26: Fotografie desky s laditelným kondenzátorem

Fotografii celého zapojení obvodů obou desek a krokového motoru můžeme najít v příloze.

6 Závěr

Cílem bakalářské práce byl návrh desky plošných spojů pro ladění magnetické smyčkové antény. Výhody této antény spočívají v malých rozměrech proti vlnové délce. Tyto malé rozměry dosahujeme tím, že můžeme tuto anténu „smotat“ do menších rozměrů a vytvořit jí závity, přesto takto zmenšenou anténu je možné naladit na původní kmitočet. Zmenšení antény má však i své nevýhody, dojde tím totiž ke zmenšení plochy, která má být maximalizována anténou tvaru kruhu. Čím větší má magnetická smyčková anténa plochu tím má větší zisk. Zmenšení tak sice umožní laditelnost potřebné frekvence ale za cenu menšího zisku. Kompromisem byla zvolena smyčková anténa se třemi závity.

Kapitola 3.2 je věnována návrhu sérioparalelního zapojení kondenzátorů pro ladění smyčky antény. Nejdůležitější z kondenzátorů je laditelný kondenzátor, kterým měníme kapacitu a tak ladíme frekvenci přijímanou nebo vysílanou anténou. Je-li připojen k tomuto kondenzátoru sériový kondenzátor, zabezpečíme tím vyšší odolnost proti průrazu dielektrika, zvětšíme tím jemnost ladění, ovšem zmenšíme tím frekvenční šířku pásma, ve které se tak můžeme pohybovat. Je to řešeno paralelními kondenzátory, které způsobují posunutí laditelného pásma. Tím je dosaženo zpět potřebné laditelnosti.

Aby mohlo být všechno ovládáno, je zapojen do obvodu mikroprocesor, všechno tak bude řízeno automaticky a elektronicky. K mikroprocesoru je připojen detektor z kondenzátorů na snímání napětí na anténě, z osy laditelného kondenzátoru je snímána potenciometrem jeho poloha a za pomoci relé přepínána laditelná pásma.

V dalších kapitolách je uvedeno více informací o mikroprocesoru ATmega16, krokových motorech a zkompletování těchto částí do jednoho zařízení. Program navržený pro mikroprocesor je podrobně rozebrán a je na něm detailně vysvětlená celá funkčnost zařízení.

Při praktické realizaci došlo omylem k propojení 12V sítě napětí a 5V sítě napětí, což pravděpodobně způsobilo proražení mikroprocesoru. Napětí 12V se mohlo dostat na 5V síť také přes stabilizátor, který byl zapojený vstupem na výstup a proto došlo také k průrazu. Protože se nepovedlo přesně určit chybné místo, nebylo na 5V síti napětí nikdy dosaženo správného napětí.

7 Seznam použité literatury

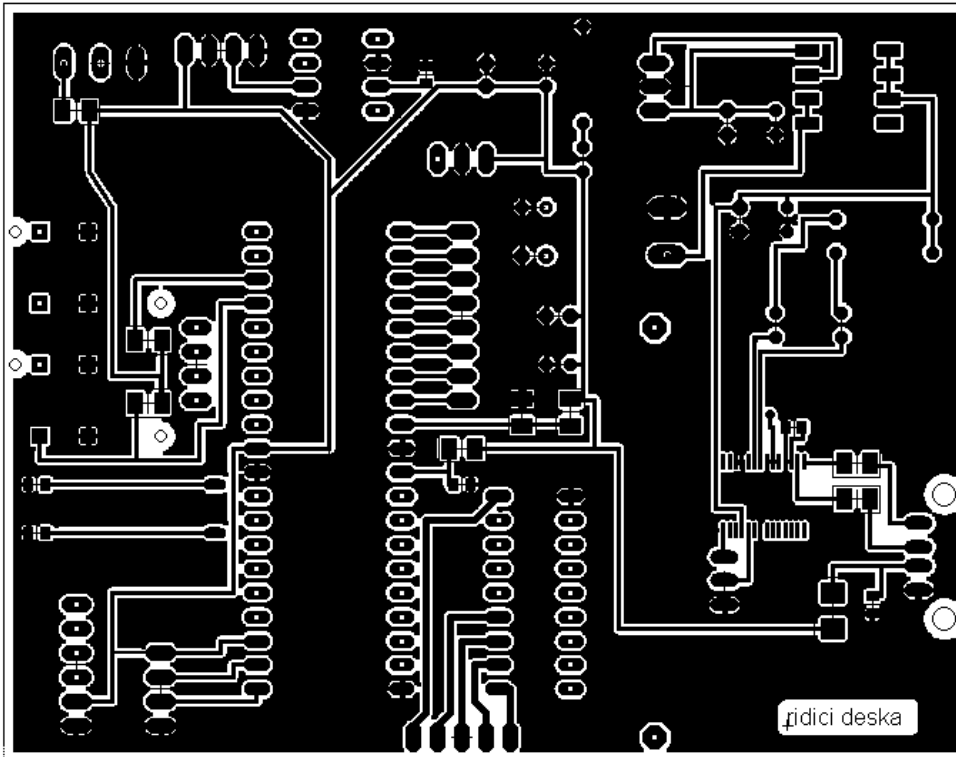
- [1] PROCHÁZKA, M. *Antény – encyklopedická příručka*. BEN – technická literatura. Praha 2005. 384 stran + CD. 3. rozšířené vydání. ISBN 80-7300-166-7
- [2] MATUSZCZYK, J. *Antény prakticky*. BEN – technická literatura. Praha 2005. 240 s. 3. české vydání. ISBN 80-7300-178-0
- [3] DVORSKÝ, M. Magnetická anténa. *Elektrorevue*. Březen 2008. Dostupné na WWW: <<http://www.elektrorevue.cz/cz/clanky/komunikacni-technologie/0/magnetickaantena/>>. ISSN 1213-1539
- [4] MILAR, P. *Elektromagnetické kmitání a vlnění*. Dostupné na WWW: <http://fyzika.gbn.cz/vyuk_materialy/elmag_vlneni/index.htm>
- [5] ATMEL CORPORATION. *ATmega16 datasheet [online]*. květen 2009. Dostupné na WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf>
- [6] ZAJAČIK, M. *Krokové motory*. Dostupné na WWW: <<http://www.zajacikm.wz.cz/KMotory/kmotory.html>>
- [7] WIKIPEDIE, *Jazyk symbolických adres*. 18. 5. 2010. Dostupné na WWW: <<http://cs.wikipedia.org/wiki/Asembler>>
- [8] RŮCKL, L. *BiProg – nejen ISP programátor*. Dostupné na WWW: <<http://web.quick.cz/ruckl/biprogram/biprogram.html>>
- [9] POVALAČ, A. *BiProg_urel.pdf*. Dostupné na WWW: <http://www.urel.feec.vutbr.cz/web_documents/dilna/BiProg/biProg_urel.pdf>
- [10] MATOUŠEK, D. *Práce s mikrokontroléry Atmel ATmega16*. BEN – technická literatura. Praha 2006. 320 stran + CD. 4. vydání. ISBN 80-7300-174-8
- [11] FRÝZA, T. *bmpt_pc_01.pdf*. Mikroprocesorová technika. BRNO: FEKT VUT v Brně.
- [12] FRÝZA, T. *bmpt_pc_05.pdf*. Mikroprocesorová technika. BRNO: FEKT VUT v Brně.
- [13] FRÝZA, T. *bmpt_pc_07.pdf*. Mikroprocesorová technika. BRNO: FEKT VUT v Brně.

8 Seznam použitých zkratek a symbolů

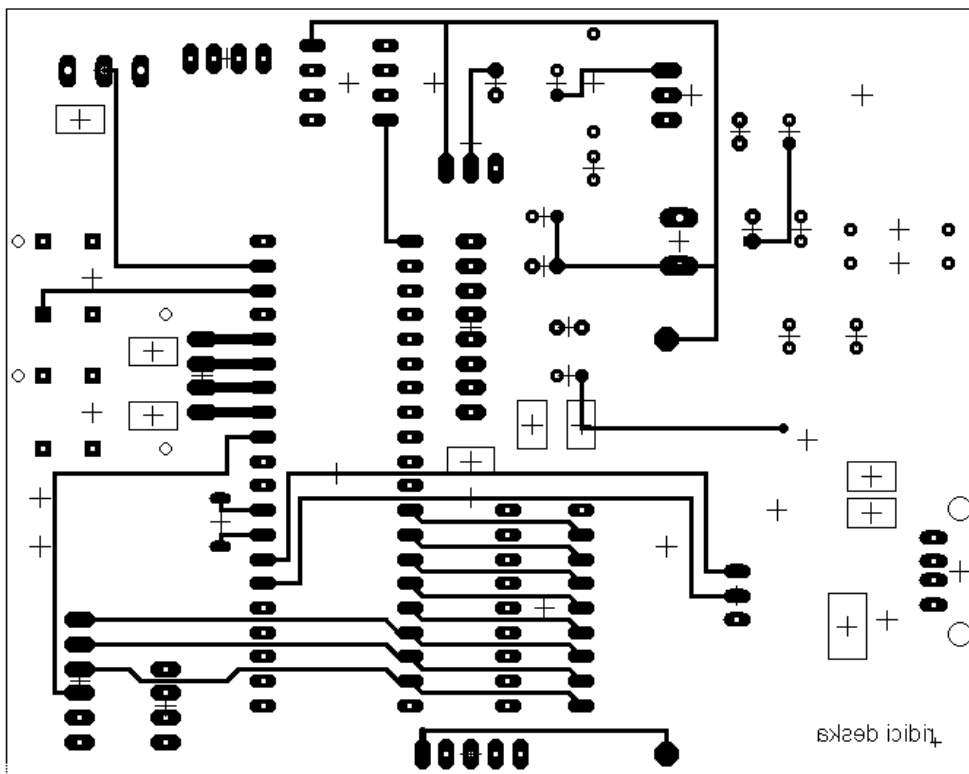
A	plocha hlavní smyčky
B	šířka pásma
C	kapacita
C_P	kapacita paralelního kondenzátoru
C_S	kapacita sériového kondenzátoru
d	průměr vodiče
D	průměr hlavní smyčky
D_v	průměr vazební smyčky
DPS	deska plošných spojů
f	kmitočet
F	směrová funkce
KV	krátké vlny
l	délka vodiče antény
L	indukčnost
LPDA	logaritmicko-periodická anténa
MLA	magnetická smyčková anténa
N	počet závitů hlavní smyčky
Q	činitel kvality
RISC	Reduced Instruction Set Computer redukovaná instrukční sada
R_L	ztrátový odpor
R_Z	odpor záření
U	obvod hlavní smyčky
U_v	obvod vazební smyčky
USART	Universal Synchronous and Asynchronous serial Receiver and Transmitter univerzální synchronní a asynchronní sériový přijímač a vysílač
USB	Universal Serial Bus univerzální sériová sběrnice
X_L	reaktance hlavní smyčky
λ	vlnová délka
η	účinnost

9 Přílohy

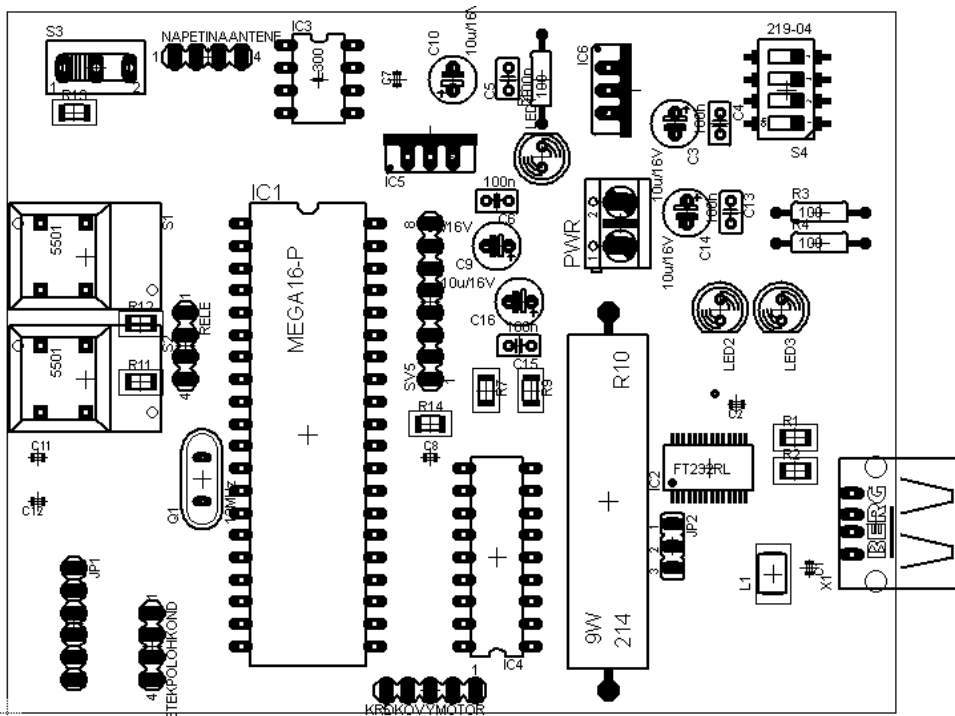
Příloha 1: DPS - Řídicí deska



Horní strana řídicí desky

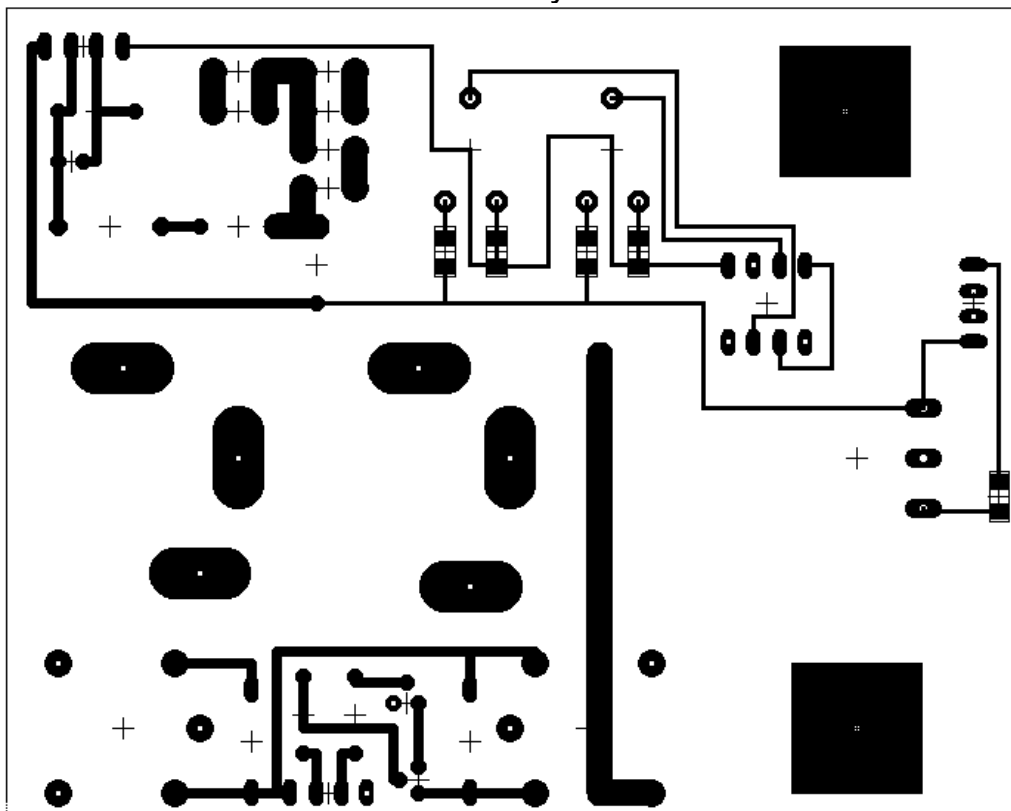


Spodní strana řídicí desky

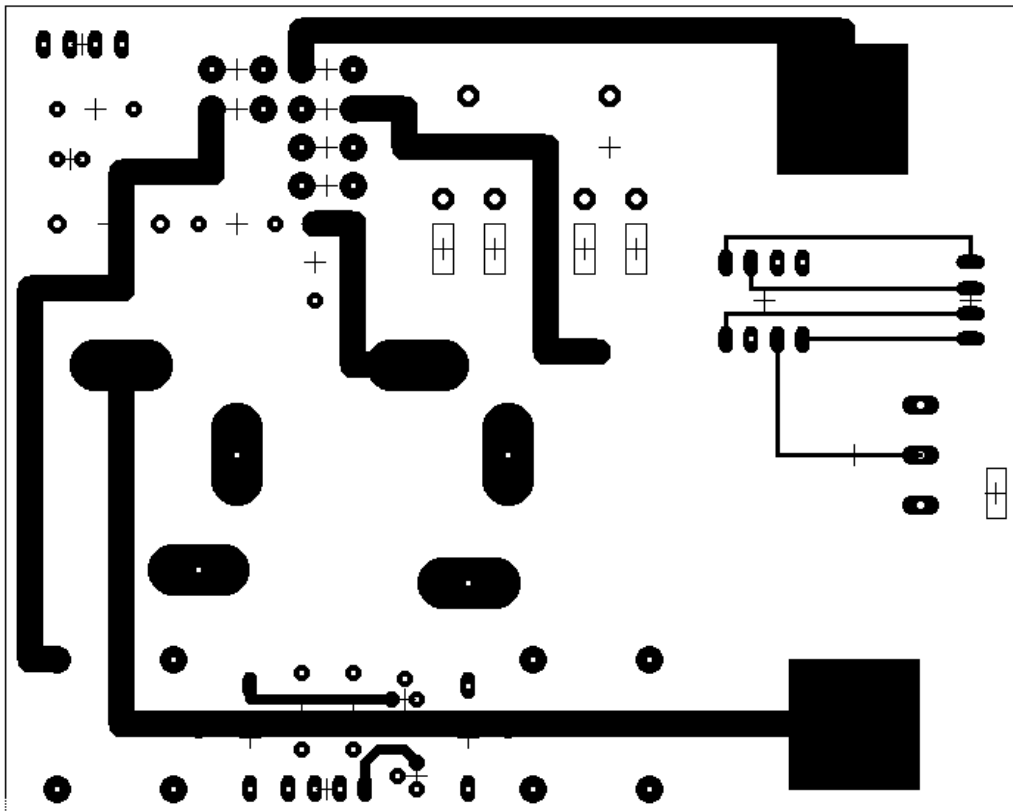


Součástky na řídicí desce

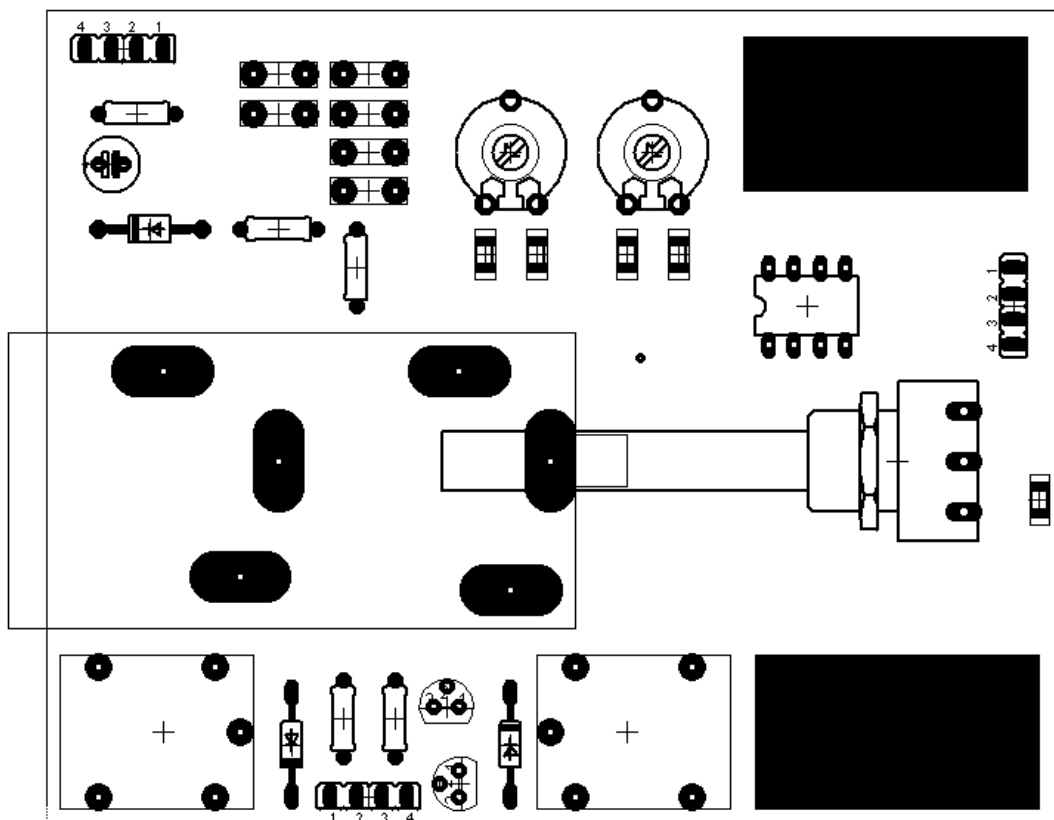
Příloha 2: DPS – Deska s laditelným kondenzátorem



Horní strana desky s laditelným kondenzátorem



Spodní strana desky s laditelným kondenzátorem



Součástky na desce s laditelným kondenzátorem

Příloha 3: Program pro mikroprocesor

```
#include <avr/io.h> // hlavickovy soubor popisujcmikrokontroler
#include <avr/interrupt.h> // knihovna potrebna pro prerusen
#include <stdio.h>
```

```
int posunmotor1 (int pol) //funkce posunuti smerem 1
{
    switch (pol)
    {
        case 0b10010000:
            pol = 0b00001001; //z a1 do b1
            break;
        case 0b00001001:
            pol = 0b01100000; //z b1 do a2
            break;
        case 0b01100000:
            pol = 0b00000110; //z a2 do b2
            break;
        case 0b00000110:
            pol = 0b10010000; //z b2 do a1
            break;
        default: break;
    }
    PORTC = pol;
    return pol;
}
```

```
int posunmotor2 (int pol) //funkce posunuti smerem 2
{
    switch (pol)
    {
        case 0b10010000:
            pol = 0b00000110; //z a1 do b2
            break;
        case 0b00001001:
            pol = 0b10010000; //z b1 do a1
            break;
        case 0b01100000:
            pol = 0b00001001; //z a2 do b1
            break;
        case 0b00000110:
            pol = 0b01100000; //z b2 do a2
            break;
        default: break;
    }
    PORTC = pol;
    return pol;
}
```

```
int automatic (int pol) //funkce automatickeho nastaveni
{
    int m, n;
    int p[13][3];
```

```

int *up;

//sejmuti vsech hodnot
while (bit_is_set(PIND,4)) //posunuti motoru na zacatek
{
    pol = posunmotor1 (pol);
}
PORTB = 0b10000000; //vypni obe rele
for (m = 0; m<13; m++)
{
    if (!bit_is_set(PIND,5))
    {
        p[m][0] = sejminapeti (m,0); //zaznam vsech hodnot pro vypnuta rele
        pol = posunmotor2 (pol);
    }
}
while (bit_is_set(PIND,4)) //posunuti motoru na zacatek
{
    pol = posunmotor1 (pol);
}
PORTB = 0b11000000; //zapni jedno rele
for (m = 0; m<13; m++)
{
    if (!bit_is_set(PIND,5))
    {
        p[m][1] = sejminapeti (m,1); //zaznam vsech hodnot pro jedno rele
        pol = posunmotor2 (pol);
    }
}
while (bit_is_set(PIND,4)) //posunuti motoru na zacatek
{
    pol = posunmotor1 (pol);
}
PORTB = 0b11100000; //zapni obe rele
for (m = 0; m<13; m++)
{
    if (!bit_is_set(PIND,5))
    {
        p[m][2] = sejminapeti (m,2); //zaznam vsech hodnot pro obe rele
        pol = posunmotor2 (pol);
    } //ejmipoli(p) sejmipole(int *p)
}
up = &p[0][0]; //ulozeni adresy p[0][0] do ukazatele up
//zpracovani vysledku
int v,i,j;
for (n = 0; n<3; n++)
{
    for (m = 0; m<13; m++)
    {
        if ( p[m][n]>v)
        {
            v = p[m][n]; //nejvetsi hodnota napeti
            i = m; //jeji souradnice m
            j = n; //jeji souradnice n
        }
    }
}

```



```

    }
}
while (bit_is_set(PIND,4)) //posunuti motoru na zacatek
{
    pol = posunmotor1 (pol);
}
switch (j) //nastaveni souradnice n
{
    case 0:
        PORTB = 0b10000000; //vypni obe rele
        break;
    case 1:
        PORTB = 0b11000000; //zapni jedno rele
        break;
    case 2:
        PORTB = 0b11100000; //zapni obe rele
        break;
    default:
        break;
}
for (m = 0; m<=i; m++) //nastaveni souradnice m
{
    pol = posunmotor2 (pol); //posun motor na max napeti
}
return pol;
while(1) //nekonecna smycka
{
    if (!bit_is_set(PINB,6))
    {
        pol = manual (pol); //prepnuti na manualni rizeni
    }
}

int sejminapeti (int m, int n, int *up) //funkce zaznamu hodnoty napeti
{
    ADCSRA = (1<<ADSC);
    while (!bit_is_set(ADCSRA,4)){ }
    *up = ADCW;
    *(up+1);
    return *up;
}

int manual (int pol) //funkce manualniho rizeni
{
    while (1)
    {
        if (bit_is_set(PINB,4)) //je-li stisknute tlacitko
        {
            pol = posunmotor1 (pol); //posun motor 1
        }
        if (bit_is_set(PINB,5)) //je-li tisknute tlacitko
        {
            pol = posunmotor2 (pol); //posun motor 2
        }
    }
}

```

```

        if (bit_is_set(PINB,6))          //je-li prepnut prepinač
        {
            pol = automatic (pol); // spust funkci automatickeho nastaveni
        }
    }
    return pol;
}

```

```

int main (void)
{
    DDRA = 0b11111110; //vstup z anteny
    DDRB = 0b11110001; //vystup rele; vstup tlacitka, prepinač
    DDRC = 0b11111111; //vystup krokovy motor
    DDRD = 0b11110011; //vstup poloha kondenzatoru

```

```

    PORTB = 0b00000000;
    PORTC = 0b10010000;

```

```

    ADMUX = 0b00000000;
    ADCSRA = 0b10000000;

```

```

int pol;

```

```

    if (bit_is_set(PINB,6))          //je-li nastaven prepinač
    {
        pol = automatic (pol); //spust automaticke nastaveni
    }
    else                               //kdyz neni
    {
        pol = manual (pol);          //spust manualni rizeni
    }
}

```

Příloha 4: fotografie celého zapojení

