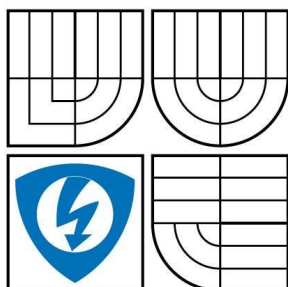


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MODULÁTOR QAM

QAM MODULATOR

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. LUDĚK DUFFEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADIM ČÍŽ

BRNO 2008

Licenční smlouva poskytovaná k výkonu práva užít školní dílo

uzavřená mezi smluvními stranami:

1. Pan

Jméno a příjmení: Bc. Luděk DUFFEK
Bytem: Kozlov 155, 588 21 Velký Beranov
Narozen (datum a místo): 12. 1. 1984 v Jihlavě
(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 602 00 Brno
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.
(dále jen „nabyvatel“)

Článek 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: Modulátor QAM
Vedoucí / školitel VŠKP: Ing. Radim ČÍŽ
Ústav: telekomunikací
Datum obhajoby VŠKP: 10. 6. 2008

VŠKP odevzdal autor nabyvateli v:

- tištěné formě - počet exemplářů 2
- 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.

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:

.....
nabyvatel

.....
autor

Abstrakt

Náplní práce je zvážit možné způsoby realizací vícecestavových modulátorů pro vytvoření laboratorního přípravku. Přípravek bude názorně ukazovat signály, které se podílí na vytváření modulací QAM. Návrh přípravku v práci obsahuje teoretický návrh schématu i samotné vyhotovení funkčního prototypu.

Začátkem práce je teoretické seznámení s vícecestavovými signály a vícecestavovými modulacemi. Podle teorie vytváření kvadraturně amplitudové modulace je stanoven jednoduchý blokový diagram přípravku. Pro každý blok jsou dále navrženy jednotlivá schémata, která jsou pak výsledně součástí celkového schématu. Ze schématu dále vychází návrh desky plošných spojů a návrh konstrukčních prvků přípravku. Další část je věnována programovému vybavení pro použitý mikroprocesor, který zajišťuje celou funkci přípravku. Posledním bodem je provedené měření, které ověřuje funkce laboratorního přípravku a správně nastavené konstelační diagramy kodéru.

Klíčová slova

QAM, PSK, Vícestavový signál, Modulace, Modulátor, Konstelační diagram, ATmega

Abstract

The thesis focuses on consideration of possible ways how to realize multi-state modulators for a laboratory instrument. This instrument will illustrate signals, which partake on generation QAM modulation. The design of the laboratory instrument includes theoretic schemata and the execution of a prototype.

There are theoretical facts of multi-state signals and multi-state modulations in this thesis. A simple block diagram of the laboratory instrument is made by the theory of creating quadrature amplitude modulation. For each block there are made several schemata, which are compiled to the global scheme. A printed circuit board and the structure items are made by the aid of the global scheme. The next part is devoted to a software facility for a used microprocessor, which ensures the whole function of the laboratory instrument. The final section deals with measuring, which checks the function of the modulator and the right setting coder's constellation diagrams.

Key Words

QAM, PSK, Multi-state signal, Modulator, Modulation, Constellation diagram, ATmega

DUFFEK, L. *Modulátor QAM*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 72 s. Vedoucí diplomové práce Ing. Radim Číž, .

Prohlášení

Prohlašuji, že diplomovou práci na téma „QAM Modulátor“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....
(podpis autora)

Seznam použitých zkratek

ADSL – technologie využívaná pro přístup k datové síti

AVR – řada mikroprocesorů firmy Atmel

CPU – centrální procesorová jednotka (MPU – mikroprocesorová jednotka)

D/A – číslicově analogový převodník

DMT – diskrétní multitónová modulace použitá v technologii ADSL

DPS – deska plošných spojů

EEPROM – elektronicky smazatelná a programovatelná paměť určená pro čtení

ISP – rozhraní pro programování procesorů přímo na DPS

LCD – displej s použitím technologie tekutých krystalů

LSB – označení nejméně významného bitu

MSB – označení nejvýznamnějšího bit

QAM – kvadrurní-amplitudová modulace (mQAM – m-stavů) / výst. signál z přípravku

PSK – modulace fázovým klíčováním (BPSK – binární, mPSK – m stavů)

RISC – redukovaná instrukční sada (oproti CISC – kompletní instrukční sada)

SRAM – paměť pro libovolný zápis a libovolné čtení, uchování dat vyžaduje napájení

SYNC – výstup synchronizačních impulzů

Obsah

1. ÚVOD	13
2. PRINCIP A VÝZNAM VÍCESTAVOVÝCH MODULACÍ	14
2.1. VÍCESTAVOVÉ SIGNÁLY A MODULACE	14
2.2. VYUŽITÍ MODULACE QAM.....	15
2.3. OBECNÉ VLASTNOSTI QAM	15
2.4. MODULACE QAM	18
2.5. DEMODULACE QAM.....	19
3. MĚŘENÍ NA INTEGROVANÉM KVADRATURNÍM (DE)MODULÁTORU	21
4. BLOKOVÝ NÁVRH LABORATORNÍHO PŘÍPRAVKU	23
4.1. MIKROPROCESOR ATMEGA16.....	23
4.2. LCD ZOBRAZOVAČ (MC1604B-SYL)	25
4.3. DA PŘEVODNÍKY (DAC0832).....	26
4.4. KVADRATURNÍ OSCILÁTOR	27
4.5. VYTVOŘENÍ QAM POMOCÍ NÁSOBIČEK (AD633).....	28
4.6. OVLÁDACÍ PRVKY	28
4.7. OBVOD OBSLUHY PŘERUŠENÍ	29
4.8. VÝSTUPNÍ OBVOD.....	30
4.9. NAPÁJECÍ ZDROJ.....	30
4.10. TLAČÍTKA.....	31
5. REALIZACE NA KONTAKTNÍM POLI	32
5.1. NÁVRH JEDNODUCHÉHO PROGRAMU	32
5.2. ZJIŠTĚNÉ NEDOSTATKY A JEJICH ŘEŠENÍ	33
6. NÁVRH CELKOVÉHO SCHÉMATU	35
7. NÁVRH DESKY PLOŠNÉHO SPOJE (DPS)	36
8. NÁVRH HLAVNÍCH FUNKČNÍHO DIAGRAMŮ – POPIS SW	39
8.1. HLAVNÍ ALGORITMUS.....	39
8.2. ALGORITMUS PŘERUŠENÍ	40
8.3. ALGORITMUS PŘERUŠENÍ OD ČASOVAČE	42
8.4. PŘEPÍNÁNÍ ALGORITMŮ	42
9. IMPLEMENTACE ALGORITMŮ V ASSEMBLERU PRO ATMEGA16	44
9.1. LCD ZOBRAZOVAČ	44
9.2. DA PŘEVODNÍKY	45
9.3. OVLÁDACÍ PRVKY	47
9.4. NASTAVENÍ PŘERUŠENÍ	47
9.5. KONVERZE DAT SYMBOLU NA DATA I A Q.....	49
9.7. PSEUDONÁHODNÝ GENERÁTOR DAT.....	51
9.7. CELKOVÝ PROGRAMOVÝ NÁVRH.....	53
10. KONSTRUKCE	54
11. OŽIVENÍ PŘÍPRAVKU	56
11.1. POČÁTEČNÍ NASTAVENÍ.....	56
11.2. NAHRÁNÍ PROGRAMU DO FLASH PAMĚTI PROCESORU.....	56
11.3. NASTAVENÍ A OVĚŘENÍ HARDWARE	57
12. UŽIVATELSKÝ POPIS PŘÍPRAVKU	59
12.1. ZAPOJENÍ PŘÍPRAVKU	59
12.2. NASTAVOVÁNÍ PARAMETRŮ	60
13. MĚŘENÍ	61

13.1. MĚŘENÍ NA KVADRATURNÍM OSCILÁTORU	61
13.2. MĚŘENÍ NA KODÉRU	62
13.3. MĚŘENÍ NA MODULÁTORU	62
13.4. MĚŘENÍ NA GENERÁTORU A NA VÝSTUPU Z PŘÍPRAVKU	63
14. ZÁVĚR.....	64
15. POUŽITÁ LITERATURA	65

Seznam obrázků

OBRÁZEK 2.1: VÍCESTAVOVÉ SIGNÁLY V ZÁKL. A PŘELOŽENÉM PÁSMU A JEJICH SPEKTRA ..	14
OBRÁZEK 2.2: VYTVOŘENÍ SYMBOLU MQAM.....	15
OBRÁZEK 2.3: ORIENTACE V KONSTELAČNÍM DIAGRAMU.....	16
OBRÁZEK 2.4 A,B,C: KONSTELAČNÍ DIAGRAMY ČASTO UŽÍVANÝCH MODULACÍ	16
OBRÁZEK 2.4 D,E,F: KONSTELAČNÍ DIAGRAMY ČASTO UŽÍVANÝCH MODULACÍ.....	17
OBRÁZEK 2.5: BLOKOVÉ SCHÉMA KVADRATURNÍHO MODULÁTORU.....	18
OBRÁZEK 2.6: ČASOVÉ PRŮBĚHY MODULÁTORU (PŘÍKLAD QPSK/4QAM)	18
OBRÁZEK 2.7: BLOKOVÉ SCHÉMA KVADRATURNÍHO DEMODULÁTORU	19
OBRÁZEK 3.1: BLOKOVÉ SCHÉMA RF2713	21
OBRÁZEK 3.2: DOPORUČENÉ ZAPOJENÍ RF2713 VÝROBCEM JAKO MODULÁTOR	22
OBRÁZEK 4.1: BLOKOVÉ SCHÉMA LABORATORNÍHO PŘÍPRAVKU „QAM MODULÁTOR“	23
OBRÁZEK 4.2: SCHÉMA ZAPOJENÍ ŘÍDÍCÍ PROCESOROVÉ JEDNOTKY PŘÍPRAVKU	24
OBRÁZEK 4.3: SCHÉMA KONEKTORU PRO „IN SYSTEM PROGRAMMING“	25
OBRÁZEK 4.4: SCHÉMA KONEKTORU PRO LCD	25
OBRÁZEK 4.5: SCHÉMA ZAPOJENÍ OBVODŮ PRO VYTVOŘENÍ SIGNÁLU I A Q.....	26
OBRÁZEK 4.6: ZAPOJENÍ DA PŘEVODNÍKU PRO UNIPOLÁRNÍ VÝSTUP	26
OBRÁZEK 4.7: SCHÉMA KVADRATUNÍHO OSCILÁTORU	27
OBRÁZEK 4.8: SCHÉMA KVADRATUNÍHO MODULÁTORU	28
OBRÁZEK 4.9: SCHÉMA PRO PŘIPOJENÍ TLAČÍTEK	29
OBRÁZEK 4.10: SCHÉMA OBVODU PRO OBSLUHU PŘERUŠENÍ.....	29
OBRÁZEK 4.11: SCHÉMA VÝSTUPNÍHO OBVODU	30
OBRÁZEK 4.12: SCHÉMA NAPÁJECÍHO ZDROJE.....	31
OBRÁZEK 4.13: SCHÉMA ZAPOJENÍ TLAČÍTEK.....	31
OBRÁZEK 5.1A: VÝSTUP Z DA PŘEVODNÍKU	33
OBRÁZEK 5.1B: VÝSTUP Z DA PŘEVODNÍKU S KONDENZÁTOREM VE ZPĚTNÉ VAZBĚ.....	33
OBRÁZEK 5.2A: REAKCE DA PŘEVODNÍKU PO PŘÍCHODU PŘERUŠENÍ	34
OBRÁZEK 5.2B: PŘÍCHOD PŘERUŠENÍ S PŘEDSTIHEM.....	34
OBRÁZEK 7.1: HLAVNÍ DESKA – SPODNÍ POHLED	36
OBRÁZEK 7.2: HLAVNÍ DESKA – VRCHNÍ POHLED	37
OBRÁZEK 7.3: HLAVNÍ DESKA – ROZMÍSTĚNÍ SOUČÁSTEK	37
OBRÁZEK 7.2: DESKA S TLAČÍTKY – SPODNÍ POHLED	38
OBRÁZEK 7.3: DESKA S TLAČÍTKY – ROZMÍSTĚNÍ SOUČÁSTEK.....	38
OBRÁZEK 8.1: HLAVNÍ ALGORITMUS	39
OBRÁZEK 8.2: ALGORITMUS VNĚJŠÍHO PŘERUŠENÍ.....	40
OBRÁZEK 8.3: ALGORITMUS PŘERUŠENÍ OD ČASOVAČE.....	42
OBRÁZEK 8.4: ČASOVÝ PRŮBĚH PŘEPÍNÁNÍ ALGORITMŮ	43
OBRÁZEK 9.1: ČASOVÝ DIAGRAM PŘENOSU DAT NA DA PŘEVODNÍKY.....	45

OBRÁZEK 9.1: GENERÁTOR PSEUDONÁHODNÝCH DAT	51
OBRÁZEK 10.1: FOTOGRAFIE ROZMÍSTĚNÍ PRVKŮ V PROTOTYPU	55
OBRÁZEK 10.2: FOTOGRAFIE KOMPLETNÍHO LABORATORNÍHO PŘÍPRAVKU	55
OBRÁZEK 11.1: NASTAVOVACÍ PRVKY NA PŘÍPRAVKU	56
OBRÁZEK 11.2A: OSCILOGRAM <u>ŠPATNĚ</u> NASTAVENÉHO KOMPARÁTORU	58
OBRÁZEK 11.2B: OSCILOGRAM <u>SPRÁVNĚ</u> NASTAVENÉHO KOMPARÁTORU	58
OBRÁZEK 12.1: POPIS PŘÍPRAVKU	59
OBRÁZEK 13.1: MĚŘENÍ KVADRATURNÍHO OSCILÁTORU (NOSNÝ SIGNÁL)	61
OBRÁZEK 13.2A: ZMĚŘENÝ KONSTELAČNÍ DIAGRAM 8PSK (REŽIM XY).....	62
OBRÁZEK 13.2B: ZMĚŘENÝ KONSTELAČNÍ DIAGRAM 64QAM (REŽIM XY).....	62
OBRÁZEK 13.3A: MĚŘENÍ NA MODULÁTORU PRO 1 PERIODU NA SYMBOL	62
OBRÁZEK 13.3B: MĚŘENÍ NA MODULÁTORU PRO 5 PERIOD NA SYMBOL	62
OBRÁZEK 13.4A: MĚŘENÍ NA VÝSTUPU A GENERÁTORU DAT PRO 1 PERIODU NA SYMBOL	63
OBRÁZEK 13.4B: MĚŘENÍ NA VÝSTUPU A GENERÁTORU DAT PRO 5 PERIOD NA SYMBOL	63

Seznam tabulek a programových kódů

TABULKA 2.1: MODULACE A JEJICH ZÁKLADNÍ PARAMETRY	18
TABULKA 4.1: VÝZNAM VSTUPNĚ-VÝSTUPNÍCH VÝVODU PROCESORU	25
TABULKA 6.1: SEZNAM POUŽITÝCH SOUČÁSTEK	35
TABULKA 9.1: OPERACE V INTERVALECH PŘI PŘENOSU NA DAC	46
TABULKA 9.2: NASTAVENÍ VNĚJŠÍHO PŘERUŠENÍ – REGISTR MCUCR	47
TABULKA 9.3: NASTAVENÍ VNĚJŠÍHO PŘERUŠENÍ – POPIS REGISTRU MCUCR.....	47
TABULKA 9.4: NASTAVENÍ VNĚJŠÍHO PŘERUŠENÍ – REGISTR GICR	48
TABULKA 9.5: NASTAVENÍ ČASOVAČE – REGISTR TCCR0.....	48
TABULKA 9.6: NASTAVENÍ ČASOVAČE – VOLBA MÓDU V REGISTRU TCCR0	48
TABULKA 9.7: NASTAVENÍ ČASOVAČE – VOLBA ZDROJE V REGISTRU TCCR0	48
TABULKA 9.8: NASTAVENÍ ČASOVAČE – REGISTR TIMSK.....	49
TABULKA 9.9: HODNOTY PRO VYTVOŘENÍ TABULKY SYMBOLŮ MODULACE 16QAM.....	50
TABULKA 9.10: HODNOTY I A Q SYMBOLŮ MODULACE 16QAM.....	50
TABULKA 10.1: BAREVNÉ ROZLIŠENÍ PŘÍSTROJOVÝCH ZDÍŘEK.....	54
TABULKA 11.1: KONFIGURAČNÍ A BEZPEČNOSTNÍ BITY	57
TABULKA 11.2: PROPOJKY JP ₁ , JP ₂ A JP ₃	58
KÓD 5.1: TESTOVÁNÍ LCD.....	32
KÓD 5.2: TESTOVÁNÍ REAKCE NA VNĚJŠÍ PŘERUŠENÍ	32
KÓD 5.3: TESTOVÁNÍ REAKCE TLAČÍTKA	32
KÓD 5.4: TESTOVÁNÍ FUNKČNOSTI DA PŘEVODNÍKŮ	32
KÓD 9.1: ZÁKLADNÍ FUNKCE PRO LCD	44
KÓD 9.2: ZÁKLADNÍ FUNKCE PRO DA PŘEVODNÍKY.....	46
KÓD 9.3: ZÁKLADNÍ FUNKCE PRO OBSLUHU TLAČÍTEK.....	47
KÓD 9.4: NASTAVENÍ A POVOLENÍ VNĚJŠÍHO PŘERUŠENÍ	48
KÓD 9.5: NASTAVENÍ ČASOVAČE A POVOLENÍ PŘERUŠENÍ.....	49
KÓD 9.6: TABULKA PRO PŘEVOD DAT MODULACE 16QAM.....	51
KÓD 9.7: POUŽÍVÁNÍ TABULEK PRO PŘEVOD DAT	51
KÓD 9.8: PSEUDONÁHODNÝ GENERÁTOR.....	52

1. Úvod

Úkolem této práce je prostudovat vícestavové modulace a navrhnout schéma a desku plošných spojů pro laboratorní přípravek modulátoru QAM (kvadrurní amplitudová modulace). V kapitolách je popsán jak princip vytváření modulace QAM, tak i funkčnost samotného přípravku, který popsáný princip používá. Protože bude přípravek použit pro laboratorní účely, musí být jeho koncepce vytvořena tak, aby byl dostatečně odolný vůči hrubému zacházení a jeho obsluha nebyla příliš složitá. K obsluze budou sloužit ovládací tlačítka a LCD.

Schémat jsou teoreticky navržena, samostatně odzkoušena a po té odladěna. Hodnoty konkrétních součástek jsou popsány u jednotlivých vyobrazení. Některé nedostatky byly odhaleny již při sestavení na kontaktním poli, ale zdaleka ne všechny vzniklé chyby v návrhu lze odhalit ještě před samotným sestavením prototypu. Deska plošných spojů je navržena v prostředí EAGLE a osazena 110 součástkami.

Obslužný program je navržen pomocí vývojových diagramů a implementován pomocí assemblerovských instrukcí. Návrh obsahuje rozhraní ISP a proto je velice jednoduché program ladit.

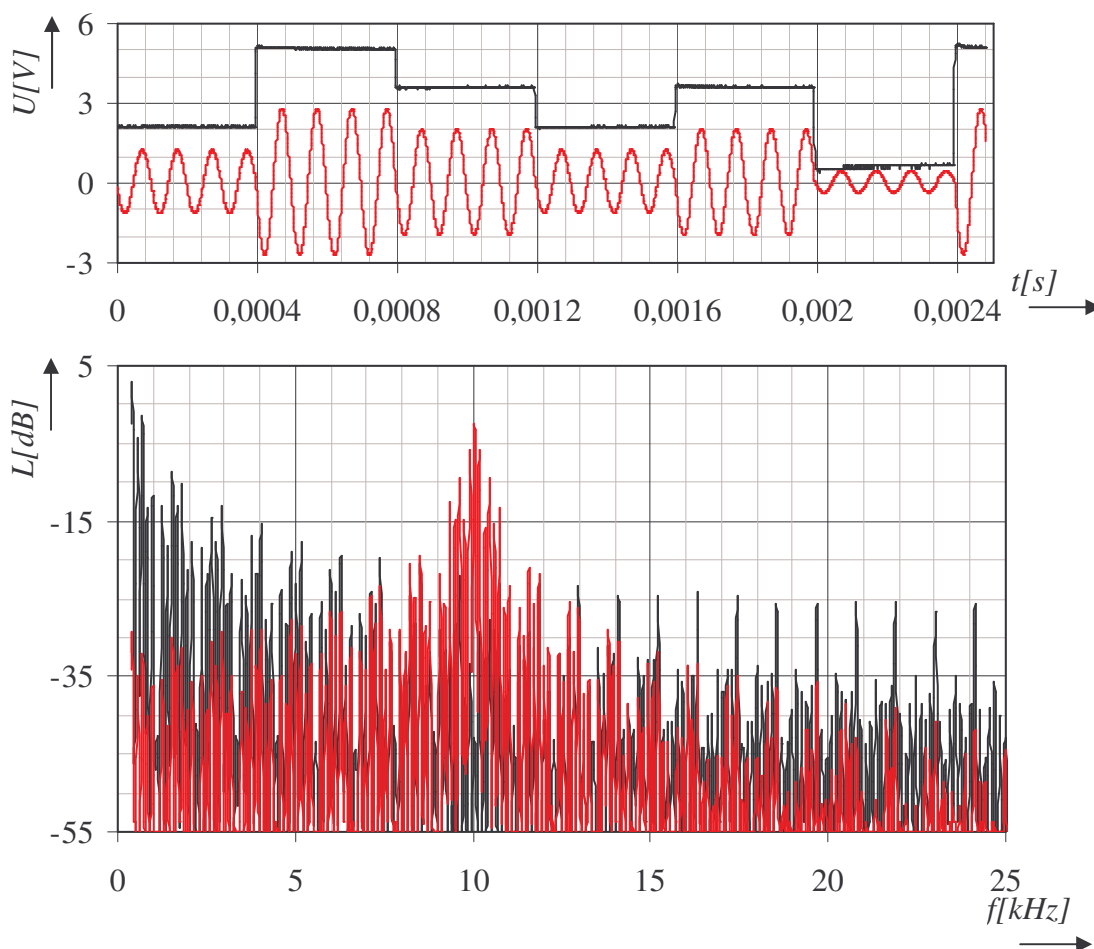
V práci je také uvedeno závěrečné měření, které ukazuje průběhy na výstupech z přípravku.

2. Princip a význam vícecestavových modulací

2.1. Vícestavové signály a modulace

Vícestavovými signály lze několikanásobně zvýšit přenosovou rychlost bez změny modulační rychlosti. Toho lze využít v systémech s omezeným přenosovým pásmem, kde již nelze zvýšit přenosovou rychlost zvýšením vysílacího kmitočtu. Vícestavový signál je vždy vytvořen z několika signálů s méně stavy, nejčastěji dvoustavových, které se běžně užívají pro přenos dat. V jednom časovém intervalu pak lze přenést více informačních bitů, spektrum tohoto signálu bývá široké a často obsahuje i stejnosměrnou složku. Tyto signály též nazýváme „signály v základním pásmu“.

Naproti tomu vícecestavové modulace, které se stále více dostávají do popředí v komunikačních systémech, využívají vícecestavových signálů v přeloženém pásmu. To znamená, že rozložení přenosového signálu (rozložení energie toho signálu) se ve spektru pohybuje kolem nosného kmitočtu jak ukazuje příklad z měření na obrázku 2.1.



Obrázek 2.1: Vícestavové signály v základním a přeloženém pásmu a jejich spektra

První obrázek vyjadřuje časové průběhy vícecestavových signálů v základním a přeloženém pásmu a druhý obrázek vyjadřuje jejich reprezentaci v kmitočtové oblasti. Černě je znázorněn průběh a spektrum vícecestavového signálu v základním pásmu a červeně pak průběh a spektrum vícecestavového signálu v přeloženém pásmu. V tomto příkladu se jedná o amplitudové klíčování s nosným kmitočtem 10 kHz.

Vícecestavové modulace lze především rozdělovat podle způsobů jakým ovlivňují nosný signál. Při přenosu informační jednotky můžeme změnit amplitudu, kmitočet nebo fázi a tyto změny můžeme provádět i současně. Na změnu amplitudy a fáze je právě zaměřena modulace QAM.

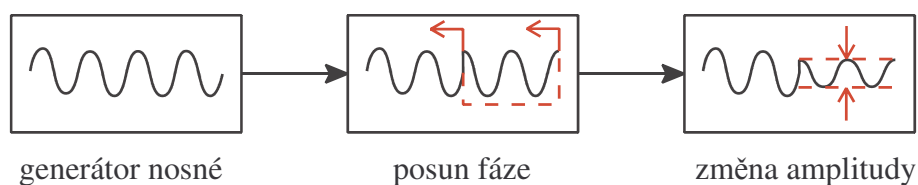
2.2. Využití modulace QAM

Princip kvadraturní amplitudové modulace je využíván především pro přenos dat s úzkým přenosovým spektrem. Princip ADSL například využívá několikanásobně tyto úzkopásmové modulace a vytváří tak multitónovou modulaci (DMT) skládající se z 255 modulací QAM.

Modulace QAM má své využití v přenosech po metalickém vedení a také pro bezdrátový přenos.

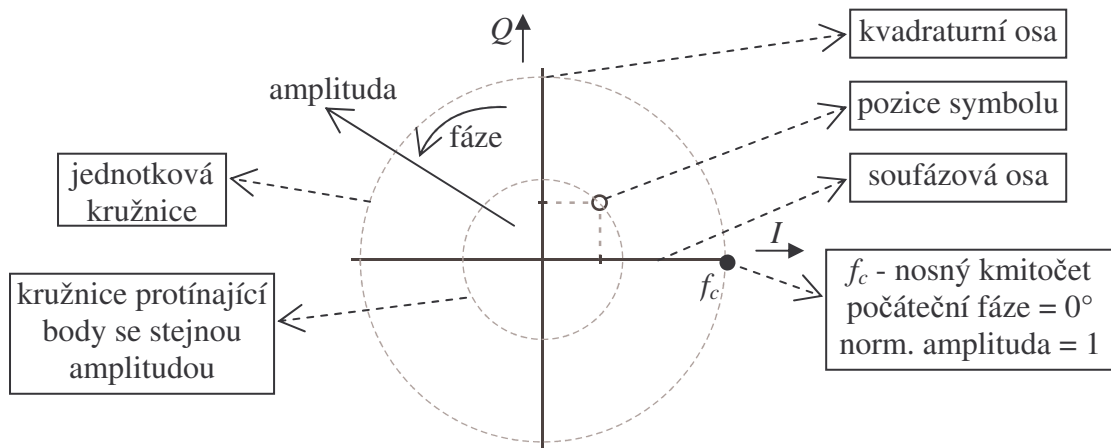
2.3. Obecné vlastnosti QAM

Jednou z výhod této modulace je přidělování tzv. symbolů k určité skupině bitů a při přenosu se použijí právě tyto symboly. Každý symbol se skládá z několika period nosného harmonického signálu a je určen dvěma parametry: amplitudou a fází. Teoreticky je princip QAM kombinací fázového (PSK) a amplitudového (ASK) klíčování. Na obrázku 2.2. je zjednodušeně znázorněna kombinace obou metod klíčování.



Obrázek 2.2: Vytvoření symbolu mQAM

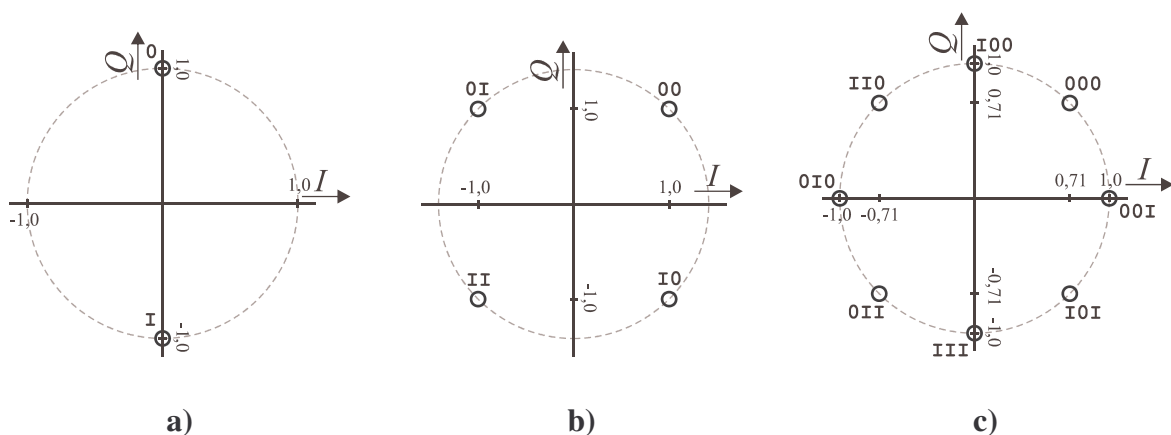
Každý druh této modulace je určen počtem používaných symbolů při modulaci. Variant těchto modulací existuje mnoho a nejčastěji se znázorňují pomocí konstelačních diagramů. Na obrázku 2.3. je zobrazena orientace v těchto diagramech.



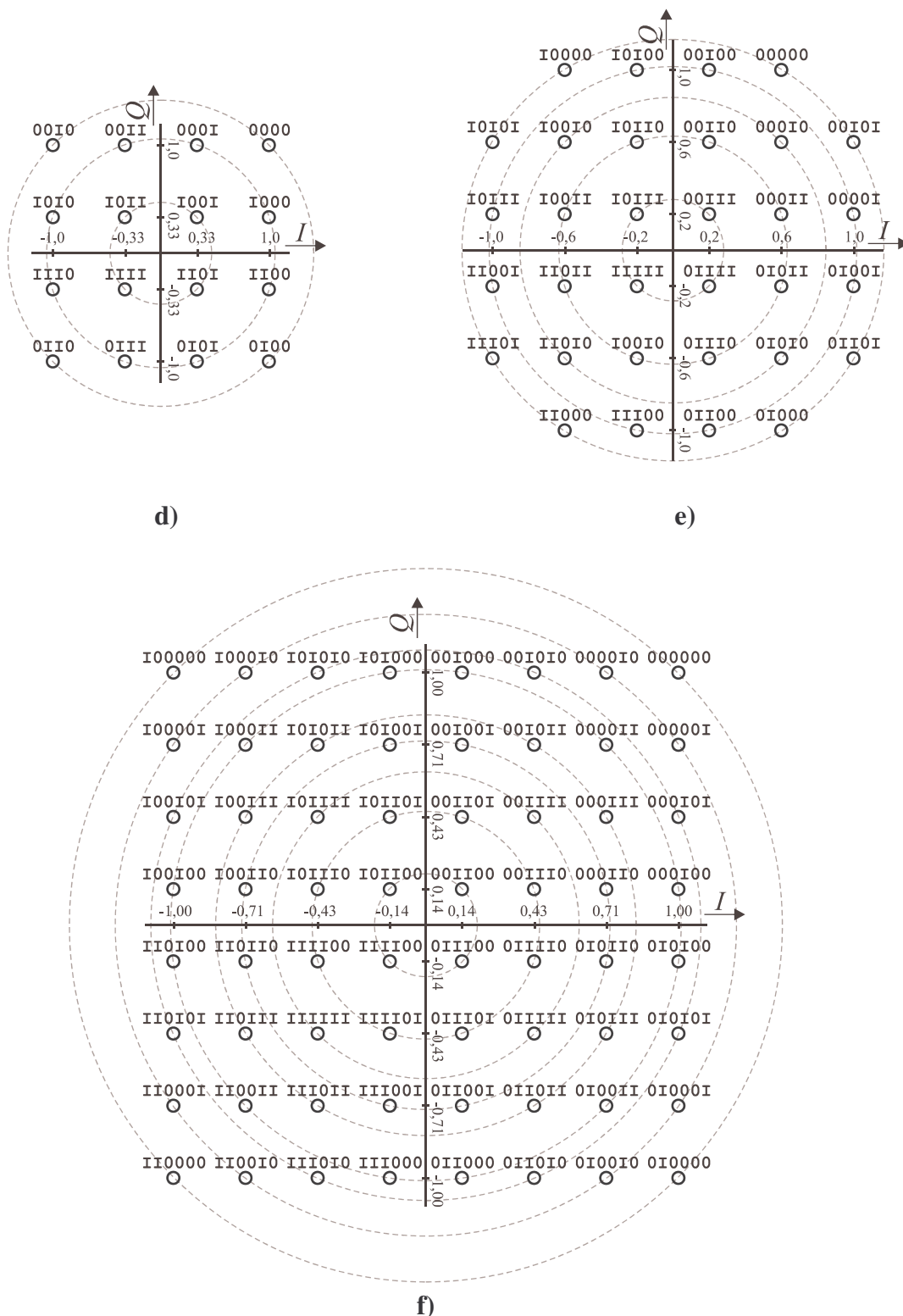
Obrázek 2.3: Orientace v konstelačním diagramu

Konstelačními diagramy se dá přehledně zobrazit souvislost bitové kombinace a příslušného symbolu k této kombinaci. Jakým způsobem se rozloží bitové kombinace v diagramu nezáleží. Pro co největší eliminaci chyb je výhodné umisťovat sousední bitové kombinace tak, aby byly rozdílné pouze v jednom bitu. Při chybném příjmu (špatně identifikovaném symbolu) dojde k záměně nejpravděpodobněji se sousedním symbolem. Chyba pak nastává minimálně a detekčním (například parita) nebo samoopravným protichybovým kódováním ji lze odhalit, případně opravit.

Na obrázku 2.4. jsou vyobrazeny diagramy vícestavových modulací. Jednotlivé body v diagramu představují pozici symbolu s příslušnou bitovou kombinací (orientaci úhlu od základní nosné a amplitudu lze určit podle obrázku 2.3). Body, které protíná jedna kružnice, jsou symboly se stejnou amplitudou.



Obrázek 2.4 a,b,c: Konstelační diagramy často užívaných modulací
 a) BPSK, b) QPSK (4QAM), c) 8PSK



Obrázek 2.4 d,e,f: Konstelační diagramy často užívaných modulací
d) 16QAM, e) 32QAM, f) 64QAM

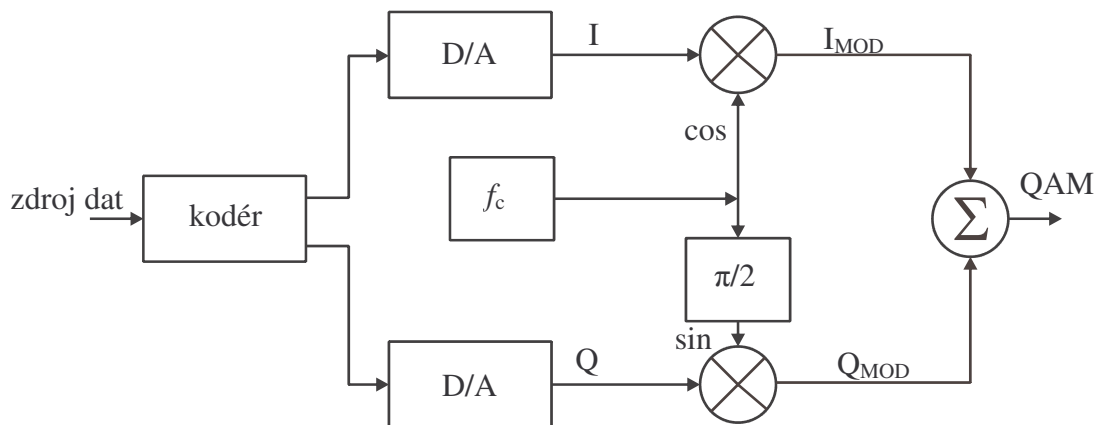
Jednotlivé varianty těchto modulací jsou určeny počtem užívaných symbolů (tj. zároveň i počtem bitů jednoho symbolu) jak ukazuje tabulka 2.1.

Tabulka 2.1: Modulace a jejich základní parametry

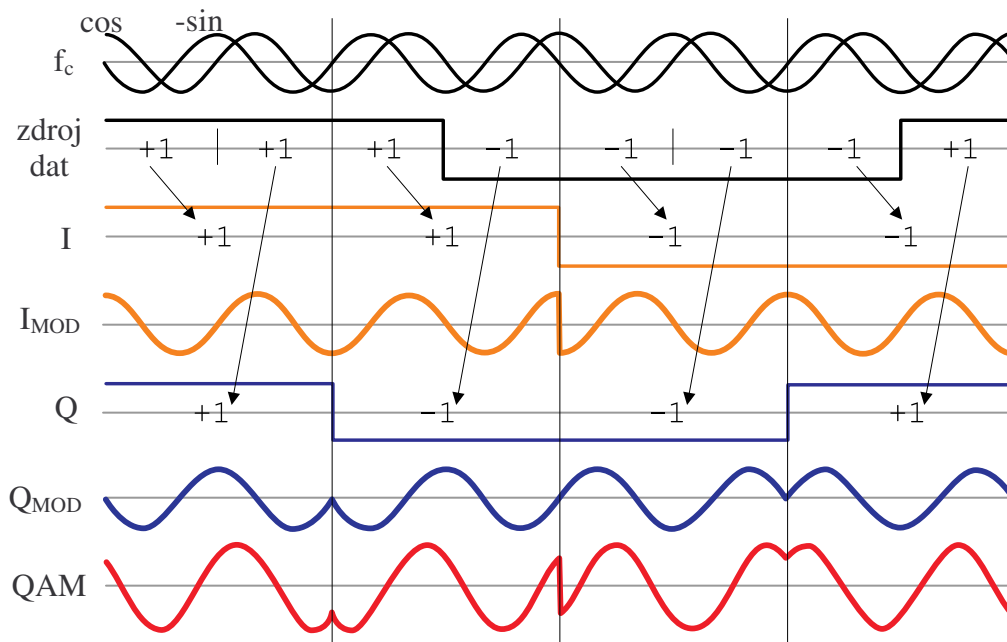
Modulace	Počet symbolů	Počet bitů na symbol	Modulace	Počet symbolů	Počet bitů na symbol
BPSK	2	1	16QAM	16	4
QPSK (4QAM)	4	2	32QAM	32	5
8PSK	8	3	64QAM	64	6

2.4. Modulace QAM

Modulace QAM používá propracovanou modulační techniku. Přicházející data jsou nejprve rozdělena do dvou toků s poloviční přenosovou rychlostí a potom modulována na dvojici nosných, vzájemně fázově posunutých o 90° . Kvadrurní modulátor využívající tohoto principu je znázorněn na obrázku 2.5. Obrázek 2.6 ukazuje časové průběhy na částech modulátoru.



Obrázek 2.5: Blokové schéma kvadrurního modulátoru



Obrázek 2.6: Časové průběhy modulátoru (příklad QPSK/4QAM)

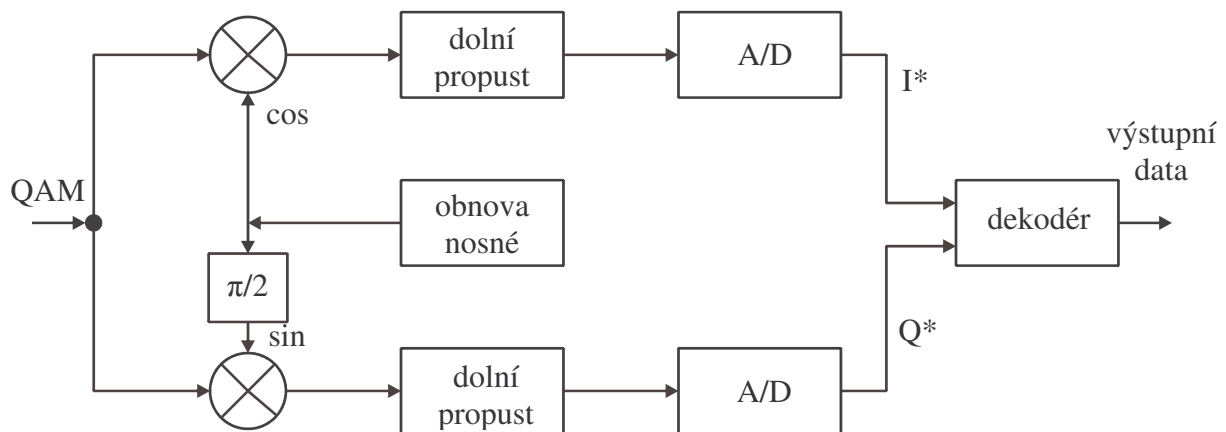
Na časových průbězích můžeme pozorovat signál QAM, který je součtem signálů I_{MOD} a Q_{MOD} . Tyto signály jsou zase tvořeny vynásobením signálu $\cos \times I$ a $\sin \times Q$. Celkový výstupní signál lze symbolicky vyjádřit zjednodušeným vztahem (2.1)

$$QAM(t) = (\cos(\omega t) \cdot I(t)) + (\sin(\omega t) \cdot Q(t)) \quad [-] \quad (2.1)$$

kde $\cos(\omega t)$ odpovídá okamžité hodnotě nosného kmitočtu a $\sin(\omega t)$ odpovídá okamžité hodnotě nosného kmitočtu posunutého o 90° . $I(t)$ a $Q(t)$ odpovídají násobku nosné a násobku nosné posunutě o 90° , které obsahuje výstupní signál $QAM(t)$. Všechny signály jsou funkcí času, proto vztah platí pouze pro výpočet okamžité hodnoty.

2.5. Demodulace QAM

Demodulování QAM signálu se provádí opačným principem než modulování. Demodulátor obsahuje kromě inverzního schématu blok, který musí identifikovat jednotlivé symboly (synchronizace) a blok pro obnovu nosného kmitočtu. Obnovená nosná musí být při procesu demodulace použita taková, která by měla stejnou frekvenci a stejnou fázi jako při modulaci. Tento blok je možná nejsložitějším obvodem v celém demodulátoru.



Obrázek 2.7: Blokové schéma kvadrurního demodulátoru

Kvadrurní demodulátor na obrázku 2.7 má podobnou skladbu jako kvadrurní modulátor (obrázek 2.5). Příchozí modulovaný signál je veden do násobiček a v nich je násoben se soufázovou nosnou a s nosnou posunutou o 90° . Po vynásobení signálu QAM a signálu se stejným kmitočtem dojde ke zdvojnásobení kmitočtu. Dolní propusti musí být navrženy tak, aby tento kmitočet zadržely, a zároveň propustily všechny ostatní dolní složky.

Matematické vztahy trigonometrických funkcí (2.2) dokazují, že při násobení harmonických funkcí o stejném kmitočtu dochází ke zdvojnásobení kmitočtu, a může docházet k posunu ve stejnosměrném směru. Tento stejnosměrný posun udává jakou mírou je složka sinus či cosinus zastoupena v přijímaném signálu. Po vyfiltrování složky o dvojnásobku nosného kmitočtu dostaneme signály I a Q (při reálné aplikaci demodulace dostaneme I a Q zasažené rušením).

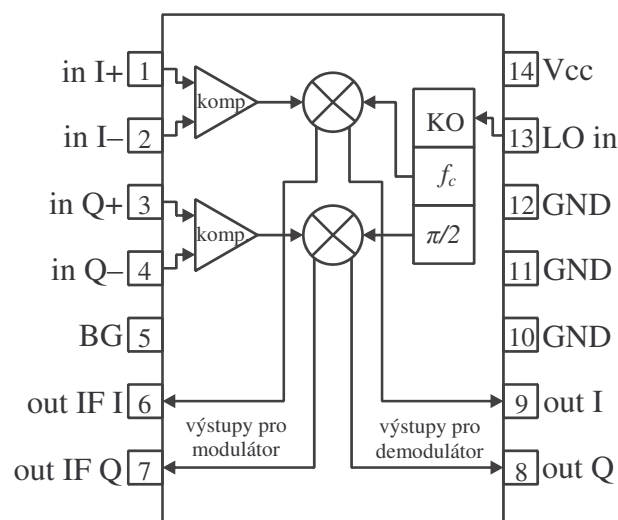
$$\begin{aligned}\sin(x) \cdot \cos(x) &= \frac{1}{2} \sin(2x) \\ \sin(x) \cdot \sin(x) &= \frac{1}{2} - \frac{\cos(2x)}{2} \quad [-] \\ \cos(x) \cdot \cos(x) &= \frac{1}{2} + \frac{\sin(2x)}{2}\end{aligned}\tag{2.2}$$

Po A/D převodu musí existovat rozhodovací funkce, která určí jaký symbol byl zachycen, aby mohl být dekodován na kombinaci bitů.

S demodulací se v přípravku nepočítá.

3. Měření na integrovaném kvadrurním (de)modulátoru

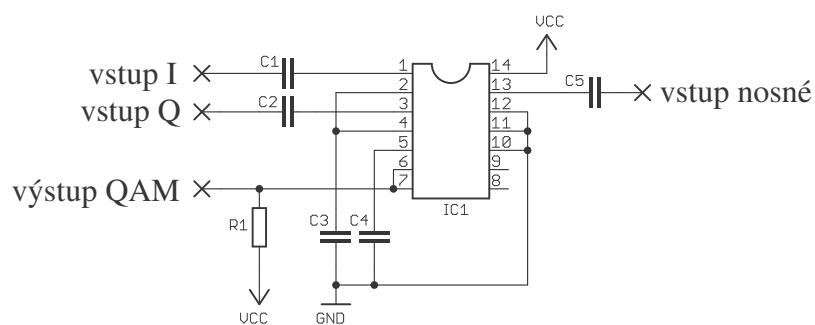
Sériově vyráběné komunikační přístroje (jako jsou různé modemy) používající modulaci QAM využívají integrované QAM modulátory. Existuje jich celá řada a jednou z možností při sestavování přípravku je právě využití některého integrovaného obvodu. Navrhovaný přípravek musí obsahovat výstupy z těch částí, které se podílejí na vytváření modulace, aby je bylo možné zobrazit na osciloskopu, a bylo možné měřený přípravek analyzovat a vyhodnotit v protokolu. Integrovaný obvod RF2713 obsahuje modulátor i demodulátor a jeho blokový diagram z obrázku 3.1 [5] připomíná modulátor z obrázku 2.5. Na první pohled můžeme vidět, že ne všechny signály budeme moci měřit a analyzovat.



Obrázek 3.1: Blokové schéma RF2713

Kmitočet vstupního nosného signálu je nejprve vydělen dvěma (výstup \cos) klopným obvodem typu D^1 a po té zpožděn o 90° (výstup \sin). Tyto signály nejsou přivedeny na žádný výstup, proto jej ani nemůžeme zobrazovat na osciloskopu. Další zjištěnou negativní vlastností je, že dělička (KO-D)¹ na vstupu je digitální. To znamená že nosný kmitočet není harmonický ale obdélníkový. V praxi tato skutečnost nevádí, protože výstup z obvodu je veden přes různé filtry. Posledním problémem byl samotný vstup signálů I a Q. V doporučeném zapojení výrobce [5] jsou vedeny přes vazební kondenzátory a to znemožňuje použití jakéhokoli vstupního signálu. Musí být bipolární a s kompenzovanou stejnosměrnou složkou. Pro laboratorní přípravek se nám toto omezení nehodí. Výrobce popisuje [5], že při jejich odstranění musí být použito předpětí o velikosti 2,7 V. Doporučené zapojení s vazebními kondenzátory je nakresleno na obrázku 3.2.

¹ Klopný obvod typu D zapojený jako dělič kmitočtu

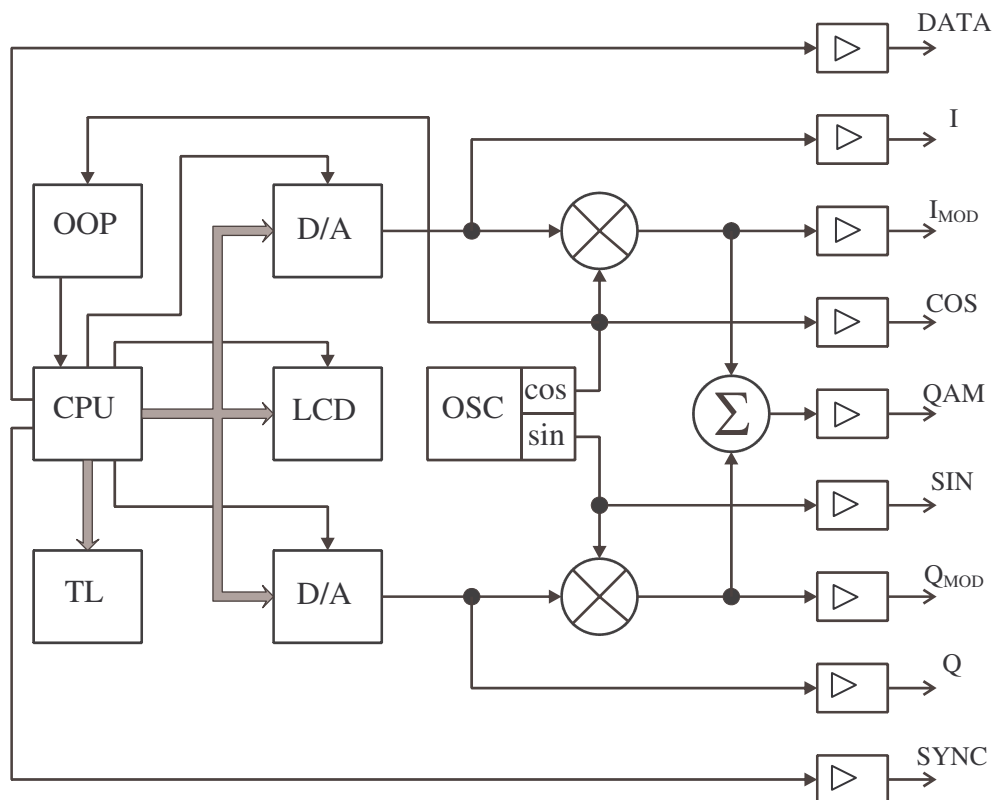


Obrázek 3.2: Doporučené zapojení RF2713 výrobcem jako modulátor

Obvod RF2713 se pro laboratorní přípravek nehodí, protože nosná a i jednotlivé symboly jsou obdélníkového charakteru. Vstup je navíc limitován bipolárním vstupem a to znemožňuje modulovat jakýkoliv signál. Na osciloskopu by nebylo možné dostatečně rozlišit jednotlivé symboly (především fázi).

4. Blokový návrh laboratorního přípravku

Každý sestavovaný přístroj je nejprve vhodné navrhnout blokově a po té řešit každý blok samostatně. Celkový problém návrhu funkčního přístroje se rozpadá do mnoha menších problémů návrhu funkčních bloků. Tyto bloky se většinou skládají z jedné hlavní součástky (integrovaného obvodu) a okolních menších součástek. Většinou jde o doporučené zapojení přímo výrobcem. Blokové schéma je znázorněno na obrázku 4.1. Jednotlivé bloky jsou popsány v dalších kapitolách.



OOP – obvod obsluhy přerušení

CPU – mikroprocesor

TL – ovládací prvky (tlačítka)

SIN, COS, I, I_{MOD}, Q, Q_{MOD}, QAM, SYNC, DATA – výstupy vedené přes zesilovače

LCD – zobrazovač

D/A – číslicově analogové převodníky

OSC – kvadrurní oscilátor

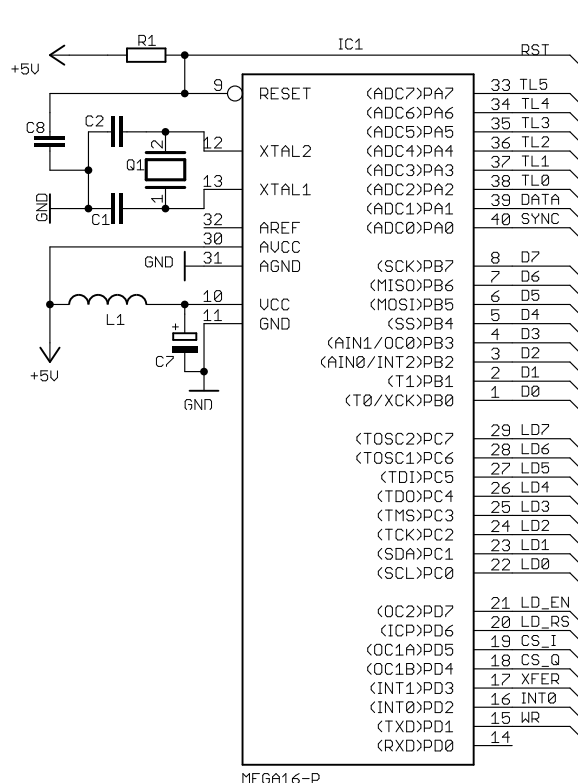
Obrázek 4.1: Blokové schéma laboratorního přípravku „QAM modulátor“

4.1. Mikroprocesor ATmega16

Jedná se o 8bitový mikroprocesor architektury RISC od firmy ATMEL patřící do řady AVR. Procesor má harvardskou architekturu, s pamětí programu 16 kilobajtů a s datovou pamětí rozdělenou na část SRAM (1 kilobajt) a EEPROM (512 bajtů). Instrukční soubor

obsahuje kolem 100 instrukcí a programátorský model je téměř shodný pro celou řadu AVR. Rozdíly jsou pouze v perifériích připojených k vnitřní datové sběrnici.

Pro návrh schématu s mikroprocesorem ATmega16 není prozatím programátorský model důležitý. Důležité jsou informace o použitém pouzdru a rozložení všech pinů (vstupně-výstupní porty, napájení, krystal). Na obrázku 4.2 je vyobrazeno schéma zapojení procesoru. Použité součástky jsou z doporučeného zapojení výrobcem [2] a jejich hodnoty nejsou pro funkčnost příliš rozhodující. Rezistor R_1 společně s kondenzátorem C_8 udávají časovou konstantu pro resetování po připojení napájecího napětí ($R_1 = 1 \text{ k}\Omega$; $C_8 = 100 \text{ nF}$). Tlačítko pro resetování je možné umístit paralelně s C_8 . V přípravku ale není nutné.



Obrázek 4.2: Schéma zapojení řídicí procesorové jednotky přípravku

Kondenzátory C_1 a C_2 jsou dle doporučení výrobce 22 pF, pro krystal 16 MHz. Tlumicí cívka L_1 společně s kondenzátorem C_7 zamezují pronikání taktovacího kmitočtu procesoru do analogové části. Protože není zatím známo jakou měrou bude pronikat signál z vnitřního oscilátoru do napájení, budou tyto součástky definovány až při měření na kontaktním poli.

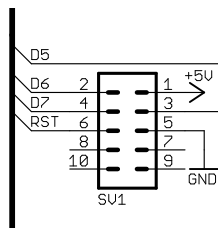
Pro přehlednost schématu jsou všechny použité vstupně-výstupní vývody připojeny na sběrnici (na obrázcích tlustou čarou). Význam jednotlivých vývodů je popsán v tabulce 4.1.

Schéma na obrázku 4.3. zobrazuje zapojení konektoru pro programování procesoru.

Tabulka 4.1: Význam vstupně-výstupních vývodu procesoru

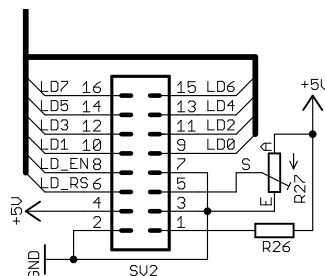
Vývod	v/v ¹⁾	Stručný popis	Vývod	v/v ¹⁾	Stručný popis
RST	I	reset (sestupná hrana)	DATA	O	sériová data
TL0-TL5	I	6 vstupů pro tlačítka	SYNC	O	synchronizace
D0-D7	O	data pro DAC	LD0-LD7	O	data pro LCD
LD_EN	O	LCD - povolení přenosu	LD_RS	O	LCD – výběr registru
CS_I	O	data jsou pro DAC1	CS_Q	O	data jsou pro DAC2
XFER	O	poslání dat na výstup DAC	INT0	I	vstup vnějšího přerušení
WR	O	DACx – povolení přenosu			

¹⁾ vstup/výstup I – vstup; O – výstup

**Obrázek 4.3:** Schéma konektoru pro „In System Programming“

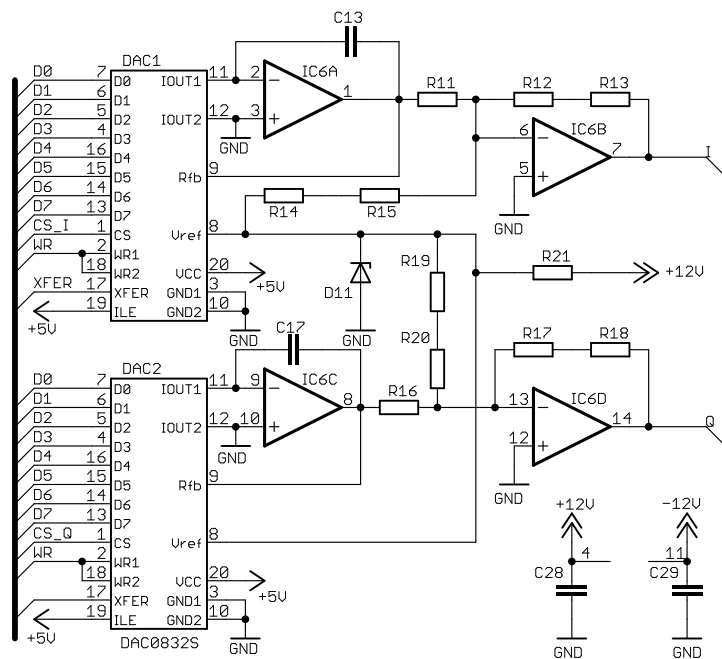
4.2. LCD zobrazovač (MC1604B-SYL)

Jako zobrazovač byl zvolen čtyřřádkový LCD, typ MC1604B-SYL s řadičem HD44780. Pro jednoduché připojení je použit 16pinový konektor, aby bylo možné LCD připevnit například na pouzdro. Zapojení je typickou aplikací od výrobce a umožňuje připojit téměř kterýkoli display s tímto řadičem. Podle doporučení je potenciometr pro řízení kontrastu roven $R_{27} = 10 \text{ k}\Omega$. Rezistor R_{26} je zde pouze pro případ, že podsvícení LCD neobsahuje předřadný odpor. Je-li předřadný rezistor přímo na desce s displejem, osadíme rezistorem 0Ω (pouze propojka). Obrázek 4.4. znázorňuje konektor pro LCD připojený na sběrnici. Význam vývodů je popsán v tabulce 4.1.

**Obrázek 4.4:** Schéma konektoru pro LCD

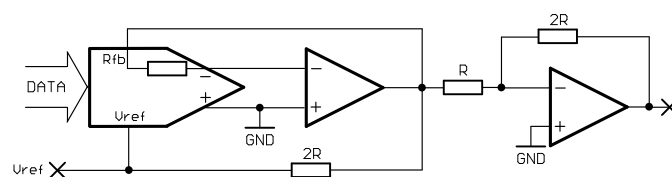
4.3. DA převodníky (DAC0832)

Číslicově-analogové převodníky jsou v tomto obvodu použity pro vytváření signálů I a Q. Každý z použité dvojice 8bitových DA převodníků obsahuje záchytný registr a registr pro udržování výstupních hodnot. To umožňuje poslat převodníkům data zvlášť do záchytných registrů a signálem XFER je zároveň zapsat do registrů výstupních. Signály I a Q se tak mění v jeden moment. Operační zesilovače IC6A a IC6C ve schématu zapojení na obrázku 4.5. patří k doporučenému zapojení výrobcem [4] a výstupem je signál $0 \text{ V} \div U_{\text{REF}}$. Zesilovače IC6B a IC6D pracují jako (invertující) sčítačky, posouvají výstupní signál o polovinu referenčního napětí zpět a násobí ho 2x. Tím dostaneme signál $-U_{\text{REF}}$ až $+U_{\text{REF}}$.



Obrázek 4.5: Schéma zapojení obvodů pro vytvoření signálu I a Q

Toto zapojení je také jedno z typických zapojení a na obrázku 4.6. je zapojení uvedené v katalogovém listu [4]. Na tomto schématu je znázorněn DA převodník s invertujícím součtovým zesilovačem. Dvojice sériově zapojených rezistorů ve schématu na obrázku 4.5. (například R_{12} a R_{13}) jsou zapojeny na obrázku 4.6. jako $2R$.



Obrázek 4.6: Zapojení DA převodníku pro unipolární výstup

Rezistory jsou jednoduše zdvojené na místo dvojnásobné hodnoty. Tak lze dosáhnout lepší přesnosti R a $2 \times R$. Osadíme-li obvod rezistory, které budou vyrobeny téměř hned za sebou, budou mít pravděpodobně i stejnou odchylku. Rezistory R_{11} ; R_{12} ; R_{13} ; R_{14} ; R_{15} ; R_{16} ; R_{17} ; R_{18} ; R_{19} ; R_{20} ; volíme $100 \text{ k}\Omega$. Zenerovu diodu volíme podle velikosti požadovaného referenčního napětí tedy $5,1 \text{ V}$ s výkonem $> 30 \text{ mW}$. Rezistor R_{21} vypočítáme podle vztahu (4.1) z rozdílu napětí ($U_{12V} - U_{ZD}$) a z minimálního proudu Zenerovou diodou (z katalogu 5 mA).

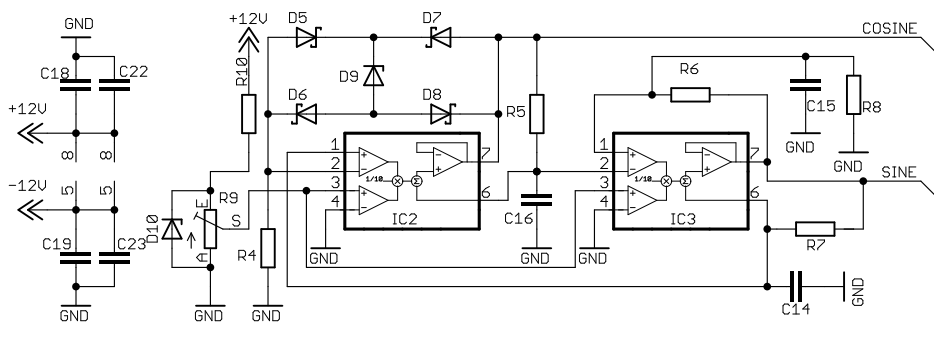
$$R_{21} = \frac{U_{12V} - U_{ZD}}{I_{ZD \min}} = \frac{12 - 5.1}{0,005} = 1150\Omega \doteq 1,2\text{k}\Omega \quad (4.1)$$

Kondenzátory C_{13} a C_{17} slouží k odstranění zákmitů na výstupu operačních zesilovačů IC6A a IC6C. Jejich použití je řešeno v kapitole 5.2 při řešení nedostatků na kontaktním poli.

Blokovací kondenzátory C_{28} a C_{29} jsou standardně 100 nF .

4.4. Kvadraturní oscilátor

Tento oscilátor vytváří dva harmonické navzájem posunuté signály o úhel 90° právě jako harmonické signály sinus a cosinus. Oba tyto signály jsou potřebné pro kvadraturní modulátor. Na obrázku 4.7 je schéma napěťově řízeného kvadraturního oscilátoru, který používá analogové násobičky AD633, a je opět jedním ze zapojení od výrobce [1]. Kmitočet oscilátoru je stanoven dvěma integračními RC články (R_5 ; C_{16} a R_7 ; C_{14}) a je přeladitelný trimrem R_9 . Dioda D_{10} zabezpečuje stabilizaci kmitočtu a diody D_5 až D_9 jsou ve funkci napěťové a teplotní stabilizace. Kondenzátory C_{18} , C_{19} , C_{22} a C_{23} jsou blokové o typické hodnotě 100 nF .



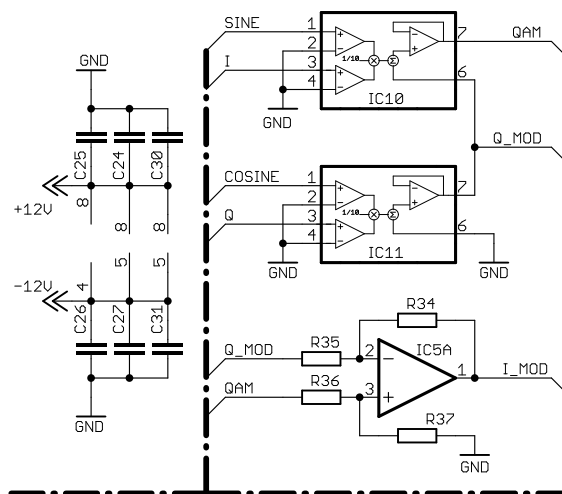
Obrázek 4.7: Schéma kvadraturního oscilátoru

4.5. Vytvoření QAM pomocí násobiček (AD633)

Signál QAM je vytvářen přesně podle blokového schématu na obrázku 2.5. Signály I a Q jsou násobeny harmonickým signálem a sečteny přímo v jedné z použitých násobiček. Jako analogové násobičky jsou použity integrované obvody AD633, které pracují přesně podle vztahu 4.2. Vstupní harmonický signál nosné je přiveden na vstup $U_{x_{in}}$ a signál I (nebo Q) na vstup $U_{y_{in}}$. Výsledný vynásobený signál je veden do sčítačky (též uvnitř integrované násobičky) a z ní veden na výstup.

$$U_{out} = \frac{U_{x_{in}} \cdot U_{y_{in}}}{10} + U_{z_{in}} \text{ [V]} \quad (4.2)$$

Kromě blokovacích kondenzátorů nejsou u integrovaného obvodu zapotřebí žádné jiné součástky. Integrovaný obvod IC5A je použit jako rozdílový zesilovač pro vytvoření signálu I_MOD, který je v jedné z násobiček přímo sčítán se signálem Q_MOD, a není vyveden z pouzdra násobičky. Obrázek 4.8. znázorňuje popsané zapojení. Hodnoty rezistorů R₃₄ až R₃₇ musí být stejné a nesmí zbytečně zatěžovat operační zesilovač. Volíme 100 kΩ, což odpovídá 50 μA při napájení 5 V.



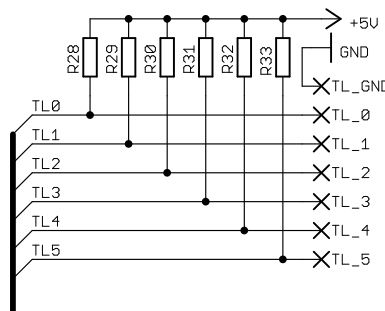
Obrázek 4.8: Schéma kvadraturního modulátoru

4.6. Ovládací prvky

Ovládání přípravku je možné za pomoci až šesti tlačítek, které budou umístěny na pouzdru. Podle obrázku 4.9. je provedeno jejich připojení k datové sběrnici.

Tlačítka jsou připojené na vývody TL_0 až TL_5 a spínají se proti společnému vývodu TL_GND. Rezistory R₂₈ až R₃₃ napomáhají pro nastavení logické „1“ při nestisknutém tlačítku. Zároveň také zamezují rušení, které může vzniknout na nezakončených vývodech.

Jako u předchozího schématu nesmí rezistory zbytečně zatěžovat napájecí zdroj. Volíme tedy 10 k Ω , což odpovídá proudu 500 μ A při stisknutí tlačítka.

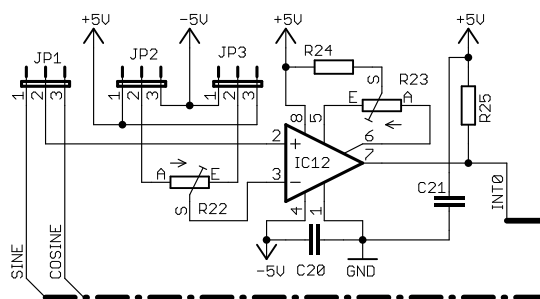


Obrázek 4.9: Schéma pro připojení tlačítek

4.7. Obvod obsluhy přerušení

Rychlost generování symbolů musí být svázána s oscilátorem, aby bylo možné symboly zřetelně zobrazit na osciloskopu. Tok dat tedy bude řízen přímo oscilátorem nosné. V praxi není příliš časté, aby jeden symbol modulace QAM obsahoval pouze jednu periodu či přesný násobek oscilátoru. Pro výukové účely a názorné zobrazení na osciloskopu se nám velice hodí, aby symbolová rychlost byla právě přesným násobkem nosného kmitočtu. Tímto způsobem pak nelze zvolit jakoukoliv přenosovou rychlost, ale každý symbol bude mít na osciloskopu jasně zřetelný začátek a konec.

Tento problém je řešen tak, že signál z oscilátoru (sinus i cosinus) je veden do komparátoru, který detekuje průchod nulou, a vytváří pro procesor impulzy pro přerušení. Programový kód přerušení pak řídí tok dat (symbolů). Přepínači JP1, JP2 a JP3 ve schématu na obrázku 4.10. je možné změnit zda bude přerušení řízeno signálem sinus nebo cosinus. Prakticky to znamená, že lze zvolit jestli soufázová větev I bude modulována signálem sinus nebo cosinus. Kvadrurní větev Q potom signálem cosinus nebo sinus. Obě dvojice jsou vzájemně fázově posunuty o 90°.

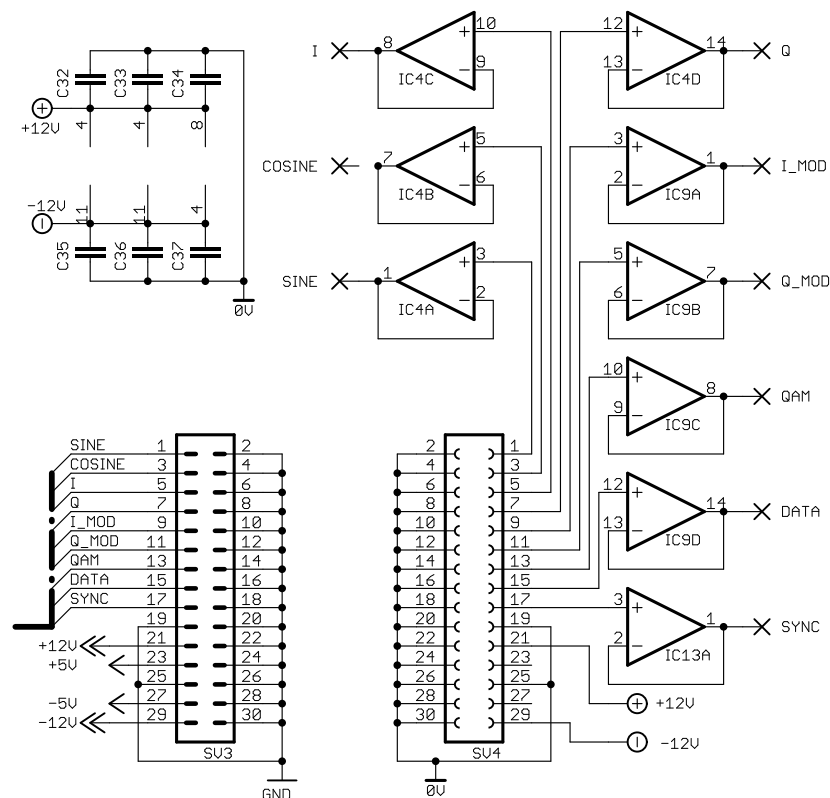


Obrázek 4.10: Schéma obvodu pro obsluhu přerušení

Trimrem R_{23} nastavujeme hysterezi (zabraňuje rozkmitání) a trimrem R_{22} nastavujeme napětí, s kterým se vstupní harmonický signál porovnává (popsáno v kapitole 5.2.).

4.8. Výstupní obvod

Pro výstup z přípravku je použit 30pinový konektor, do kterého je zapojen modul s operačními zesilovači, aby nebyly vnitřní obvody příliš zatěžovány připojenými prvky na výstupu. V případě poruchy na výstupu bude možné vyměnit pouze operační zesilovače, popřípadě celý výstupní modul. V budoucnu je také možné výstupní modul upravit (vyměnit) tak, aby každý z výstupů měl požadovanou úroveň napětí.

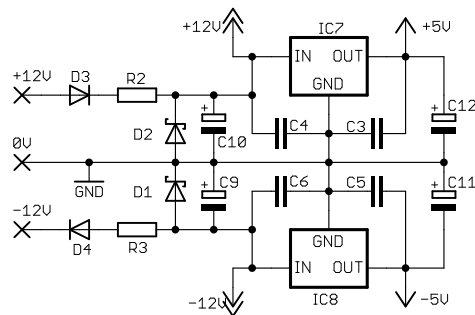


Obrázek 4.11: Schéma výstupního obvodu

4.9. Napájecí zdroj

Integrované obvody celého přípravku jsou napájeny jak napětím 5 V, tak i souměrným napájením +12 V a -12 V. Jak bylo již zmíněno v úvodu, požadavky na přípravek jsou takové, že musíme předpokládat hrubé zacházení při používání. Proto musí být napájecí zdroj (obrázek 4.12.) opatřen diodami proti prepólování (D_3 a D_4) a transily proti přepětí (D_1 a D_2). Zdroj dále obsahuje integrované stabilizátory napětí +5 V a -5 V, filtrační elektrolytické

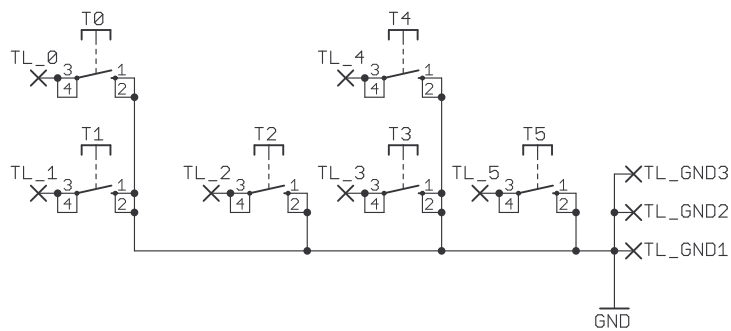
kondenzátory a blokovací kondenzátory. Usměrnovací diody D_3 a D_4 použijeme s parametry $U_f > 50 \text{ V}$ a $I_r > 500 \text{ mA}$ (z katalogu například 1N4007). D_1 a D_2 jsou transily s parametry $U_z > 12 \text{ V}$ (z katalogu například BZW06-13). Rezistory R_2 a R_3 použijeme $10 \Omega / 2 \text{ W}$.



Obrázek 4.12: Schéma napájecího zdroje

4.10. Tlačítka

Ovládací prvky (tedy tlačítka) nejsou umístěna na hlavní desce se všemi součástkami, ale na desce, která bude připevněna na horním víku krabice. Schéma samostatné desky je na obrázku 4.13.



Obrázek 4.13: Schéma zapojení tlačítek

5. Realizace na kontaktním poli

Pro otestování navržených částí je výhodné nejdříve sestavit celý nebo částečný prototyp na kontaktním poli. Spolu s vytvořením jednoduchého obslužného programu lze otestovat spolupráci jednotlivých bloků a to pomůže odhalit některé nedostatky.

5.1. Návrh jednoduchého programu

Následující jednoduché programy (kódy 5.1 až 5.4) testují samostatné části připojených periférií. Nemají téměř žádnou funkčnost, ale pro otestování stačí.

Kód 5.1: Testování LCD

```

LCDcmd:                ;procedura „poslat příkaz na LCD“
    CBI    LCD_CTRL, LCD_RS ;úroveň 0 na RS značí příkaz
    RJMP   LCDbyte
LCDdata:               ;procedura „poslat data na LCD“
    SBI    LCD_CTRL, LCD_RS ;úroveň 1 na RS značí data
LCDbyte:              ;procedura „poslat byte v registru ACC na LCD“
    OUT    LCD_BUS, ACC     ;poslat obsah ACC na sběrnici
    CBI    LCD_CTRL, LCD_ENA ;sestupná hrana ENA povolí přenos
    LDI    BUF, 0          ;nastavení časové smyčky (255 cyklů)
LCDbcyk:
    DEC    BUF             ;dekrementace ve smyčce
    BRNE   LCDbcyk        ;dokud není BUF nulový provádět smyčku
    SBI    LCD_CTRL, LCD_ENA ;konec přenosu (sběrnice volná)
    RET

```

Kód 5.2: Testování reakce na vnější přerušení

```

LDI    ACC, 3            ;přerušení od vzestupné hrany ISC00=1 a ISC01=1
OUT    MCUCR, ACC       ;zapsání do registru MCUCR
LDI    ACC, 64          ;povolení přerušení INT0=1
OUT    GICR, ACC       ;zapsání do registru GICR
SEI                    ;povolení globálního přerušení

```

Kód 5.3: Testování reakce tlačítka

```

IN     ACC, BUT         ;načtení stavu tlačítek do registru ACC
ORI    ACC, $03        ;na nejnižších bitech tlačítka nejsou
COM    ACC             ;negace - sepnuté tlačítko = 1 na určitém bitu
CPI    ACC, 64        ;test 7. bitu (TL_4)
BREQ   STISKNUTO      ;jestliže stisknuto, skoč

```

Kód 5.4: Testování funkčnosti DA převodníků

```

OUT    DAC_BUS, ACC    ;pošle data z ACC na sběrnici
CBI    DAC_CTRL, DAC_CS1 ;povolení přenosu pro první DA převodník
CBI    DAC_CTRL, DAC_WR ;povolení globálního přenosu pro oba převodníky
RCALL  WAIT           ;čekací smyčka
SBI    DAC_CTRL, DAC_WR ;přenos dokončen

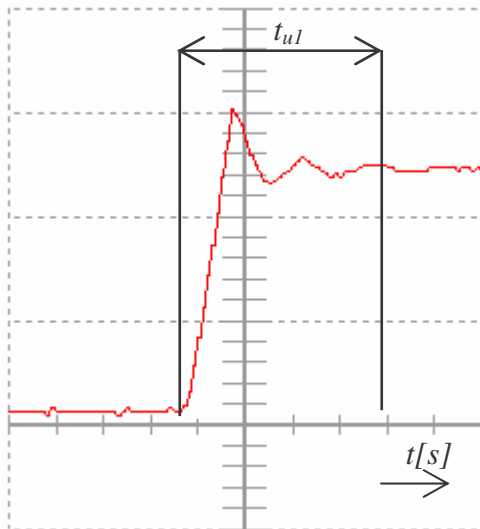
```


5.2. Zjištěné nedostatky a jejich řešení

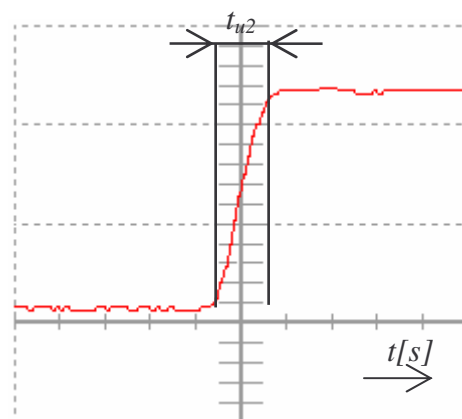
Při realizaci na kontaktním poli bylo zjištěno několik nedostatků.

1. zákmity na výstupu z DA převodníku

Při přechodu z jedné úrovně na jinou jsou hrany sice ostré, ale několikrát zakmitají (podle obrázku 5.1a). Řešení se nacházelo v katalogovém listu od výrobce [4], kde je popsán způsob zamezení kmitání při vzestupných a sestupných hranách. Přidáním kondenzátoru do zpětné vazby operačního zesilovače na výstupu DA převodníku tento problém odstraníme. Jsou přidány i do schématu na obrázku 4.5. jako C_{13} a C_{17} . Na obrázku 5.1b. je vyřešený problém pomocí kondenzátoru.



Obrázek 5.1a: Výstup z DA převodníku

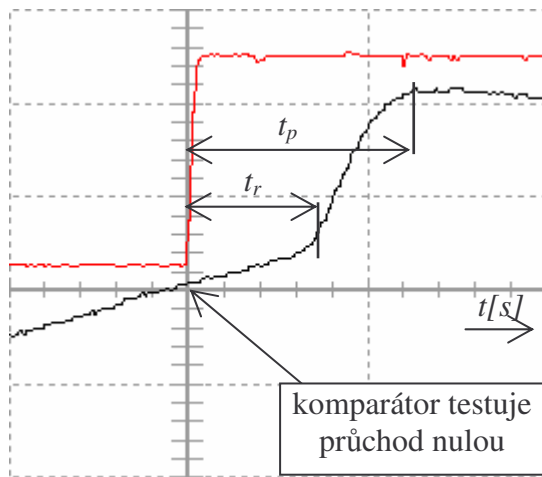


Obrázek 5.1b: Výstup z DA převodníku s kondenzátorem ve zpětné vazbě

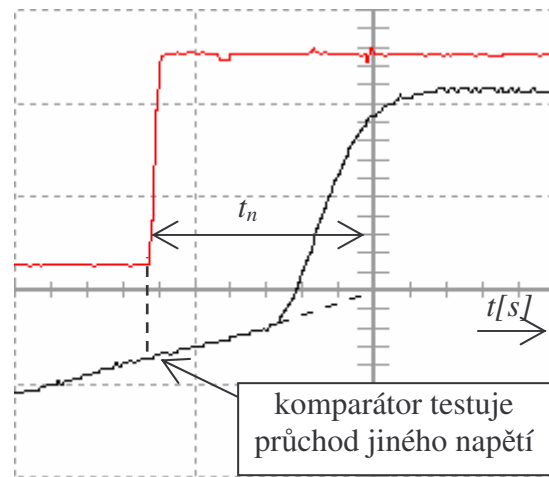
S použitým kondenzátorem o hodnotě 44 pF došlo ke značnému zlepšení ustálení. Doba se zkrátila z $t_{u1} = 7 \mu\text{s}$ na $t_{u2} = 3,5 \mu\text{s}$ a navíc neobsahuje parazitní zákmit.

2. Zpoždění symbolů z důvodu reakce na přerušení

Při příchodu přerušení na vstup INTO je v procesoru nejprve dokončena poslední prováděná instrukce a po té je možný skok na kód, který přerušení obslouží. Obsluha opět obsahuje další instrukce, které počítají který symbol se na výstup pošle.



Obrázek 5.2a: Reakce DA převodníku po příchodu přerušení



Obrázek 5.2b: Příchod přerušení s předstihem

Měřením bylo zjištěno, že od příchodu impulsu na vstup INT0 do změny symbolu, dojde ke zpoždění $t_r = 3,8 \mu\text{s}$ (obrázek 5.2a). Celková doba zpoždění i s dobou přechodu trvá $t_p = 7,5 \mu\text{s}$. Příchozí impulsy jsou řízeny průchodem nulou harmonického signálu nosné. Jestliže napětí komparátoru posuneme do záporu, nebude se harmonický signál porovnávat s nulovou hodnotou a přerušení bude přicházet dříve (obrázek 5.2b). Ve schématu na obrázku 4.10 je k tomuto účelu připraven trimr R_{22} , kterým lze porovnávací napětí regulovat. O kolik bude nutné posunout napětí, závisí na použitém taktovacím kmitočtu procesoru. Na obrázku 5.2b je předstih přerušení posunut tak, aby přechod ze symbolu sinus na symbol kosinus probíhal právě v okamžiku, kdy sinus prochází nulou. Na tomto obrázku je časová konstanta $t_n = 6,25 \mu\text{s}$.

6. Návrh celkového schématu

Celkový návrh je složen ze všech dílčích popsaných zapojení (schémata na obrázku 4.2 - 4.13). Všechny součástky jsou buď popsány a definovány, nebo jsou použity doporučené hodnoty z katalogových listů (viz použitá literatura).

Z uvedených důvodů a předpokladů jsou všechny navrhované součástky stanoveny v tabulce 6.1 a celkové schéma i se seznamem součástek je uvedeno v příloze.

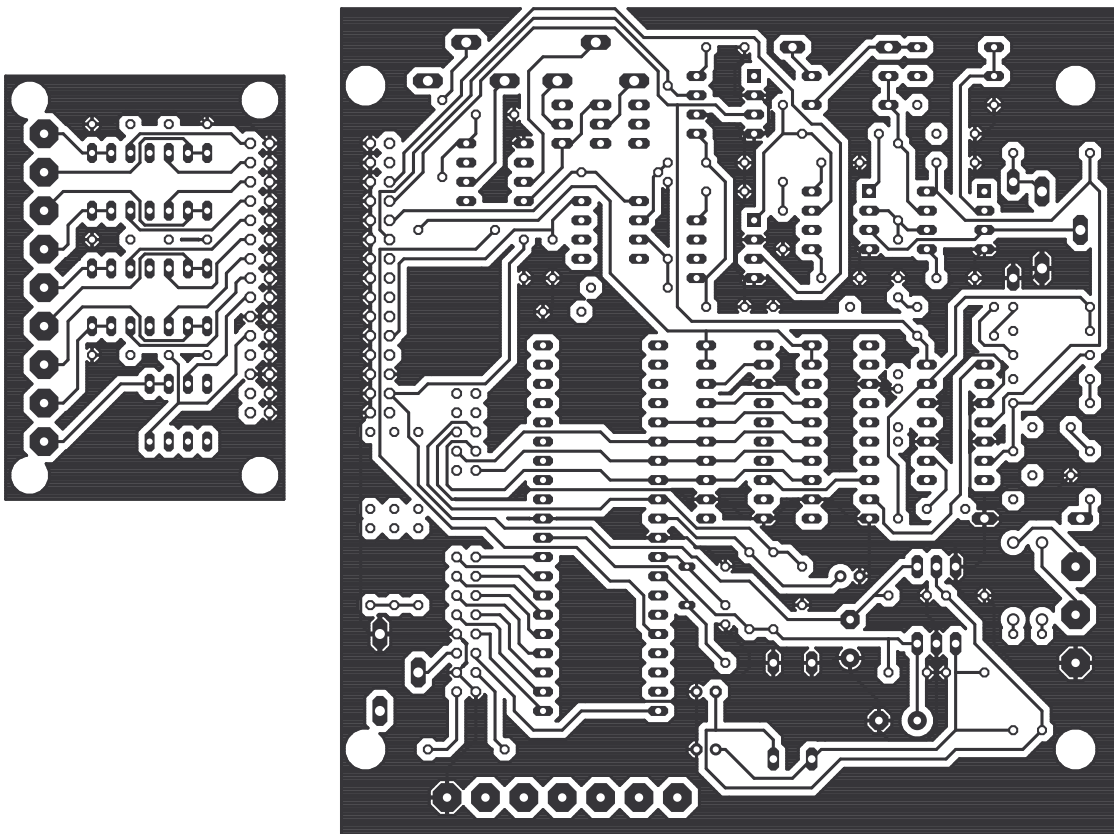
Tabulka 6.1: Seznam použitých součástek

Součástka	Typ/hodnota	Poznámka	Součástka	Typ/hodnota	poznámka
R ₁ , R ₄ , R ₂₁ , R ₂₅	1kΩ	<0,6 W	L ₁	TL.33μH	500mA/0,92Ω
R ₂ , R ₃	10Ω	<2 W	D ₁ , D ₂	BZW06-13	unip. transil
R ₅ , R ₇ , R ₃₈ , R ₃₉	1kΩ	<0,6 W	D ₃ , D ₄	1N4007	1000V/1A
R ₆	330kΩ	<0,6 W	D _{5 ÷ 8}	SD101B	schottky
R ₈	16kΩ	<0,6 W	D ₉ , D ₁₁	BZX83V005.1	ZD 5.1V
R ₉	PT15HK100	trimr 100 kΩ	D ₁₀	BZX83V008.2	ZD 8.2V
R ₁₀	1,6kΩ	<0,6 W	IC ₁	ATmega16-16PI	MPU
R _{11 ÷ 20} , R _{34 ÷ 37}	100kΩ	<0,6 W	IC ₂ , IC ₃ , IC ₁₀ , IC ₁₁	ADD633	násobička Analog Devices
R ₂₂ , R ₂₃	PT15HK005	trimr 5 kΩ	IC ₄ , IC ₆ , IC ₉	TL064	4x OZ
R ₂₄	3kΩ	<0,6 W	IC ₅ , IC ₁₃	TL062	2x OZ
R ₂₆	0Ω	propojka	IC ₇	78S05	stab. 5 V
R ₂₇	PT15HK010	trimr 10 kΩ	IC ₈	79S05	stab. -5 V
R _{28 ÷ 33}	10kΩ	<0,6 W	IC ₁₂	LM311	komparátor
C ₁ , C ₂	33pF	keramický	DAC _{1,2}	DAC0832	DA převodník
C _{3 ÷ 6} , C ₈ , C _{18 ÷ 37}	100nF	keramický	Q ₁	QM16.0MHz	krystal
C ₇ , C ₉ , C ₁₀	47μF/25V	elektrolytický	SV ₁	MLW10G	kon. ISP
C ₁₁ , C ₁₂	470μF/6,3V	elektrolytický	SV ₂	MLW16G	kon. LCD
C ₁₃ , C ₁₇	82pF	keramický	SV ₃	S2G34W	
C ₁₄ , C ₁₆	10nF	keramický	SV ₄	BLW230G	
C ₁₅	150pF	keramický			

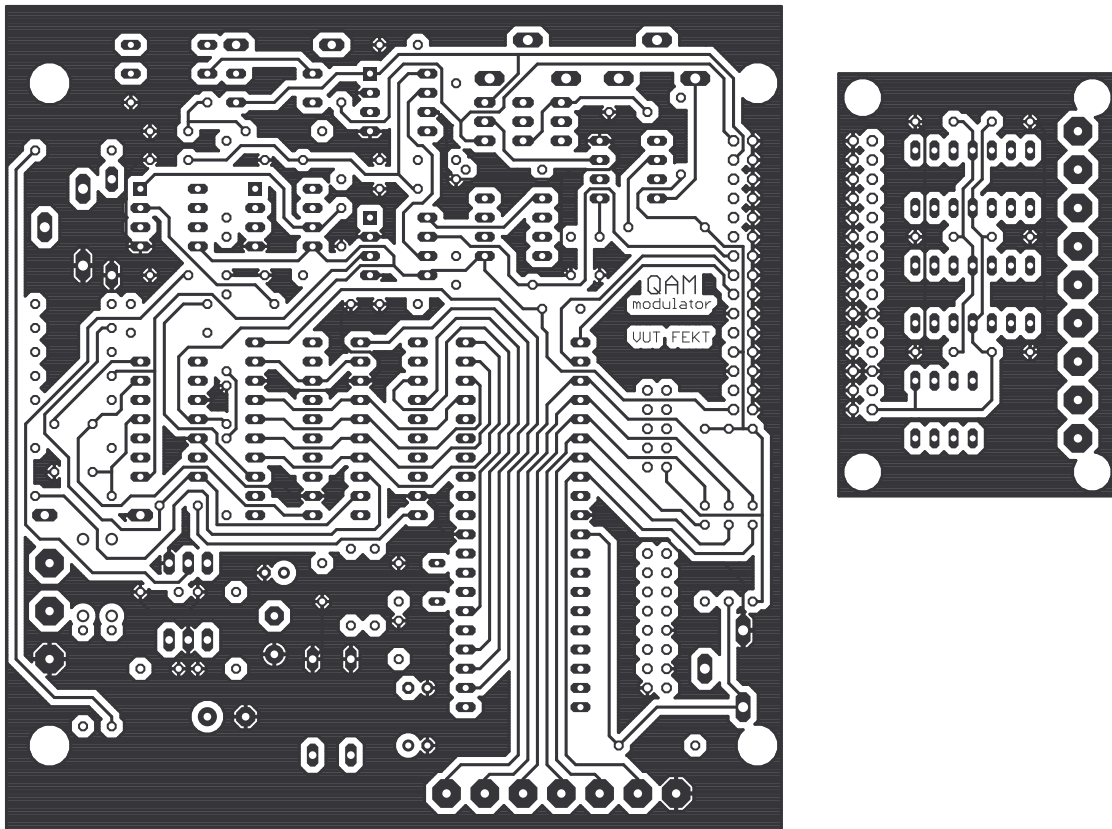
7. Návrh desky plošného spoje (DPS)

Další částí před samotným sestavením prototypu je vytvoření desky plošných spojů. Návrh je proveden za pomoci software pro kresbu a návrh plošných spojů EAGLE. Ze schématu jsou vytvořeny dvě dvouvrstvé desky. K hlavní desce bude připojena deska s výstupními obvody. Navržené schéma a oba plošné spoje obsahují celkem 110 součástek a jsou propojeny 30pinovým konektorem. Deska je navržena tak, aby bylo možné její umístění a připevnění do plastové krabice 45x190x138mm. Na obrázku 7.1 a 7.2 jsou znázorněny pohledy na desky plošných spojů v poměru 1:1. Obrázek 7.3 vyobrazuje rozmístění součástek na obou deskách.

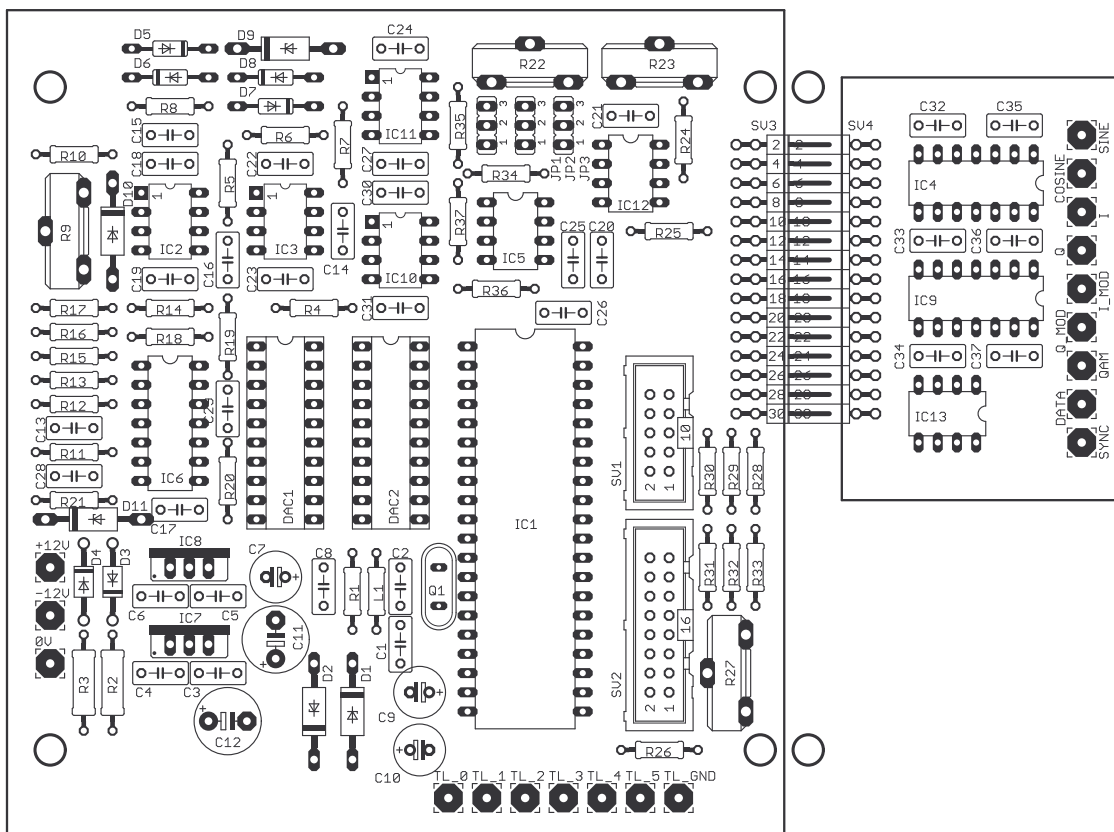
Obrázek 7.4 ukazuje desku s tlačítky, která bude umístěna na horním víku krabice. Rozmístění tlačítek ukazuje obrázek 7.5.



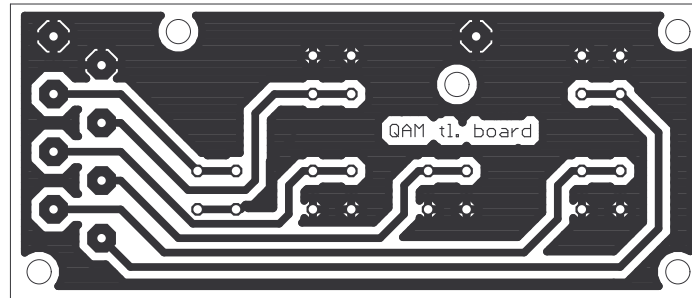
Obrázek 7.1: Hlavní deska – spodní pohled



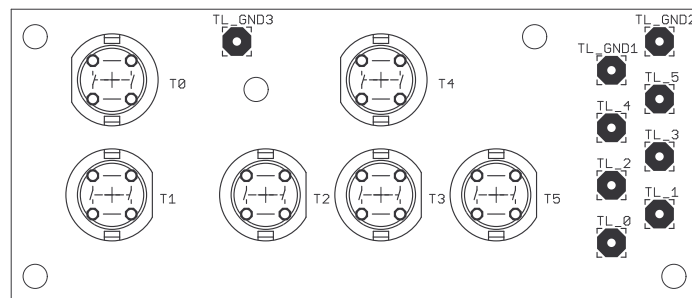
Obrázek 7.2: Hlavní deska – vrchní pohled



Obrázek 7.3: Hlavní deska – rozmístění součástek



Obrázek 7.2: Deska s tlačítky – spodní pohled



Obrázek 7.3: Deska s tlačítky – rozmístění součástek

8. Návrh hlavních funkčního diagramů – popis SW

Doposud se návrh zabýval hardwarovou částí, ve které je to nejdůležitější, co programovým vybavením provést nelze. Z blokového schématu přípravku na obrázku 4.1., lze odvodit jaké funkce musí obsahovat program uvnitř procesoru a které periférie bude obsluhovat. Mezi funkce zařadíme generování datového toku, zajištění synchronizačních impulzů, výběr použité modulace (například jedna z modulací na obrázku 2.4.), nastavení všech parametrů modulace a převod datového toku na symboly vybrané modulace. Zobrazování na LCD, komunikace s DA převodníky a reakce na tlačítka můžeme označit jako obsluha periférií.

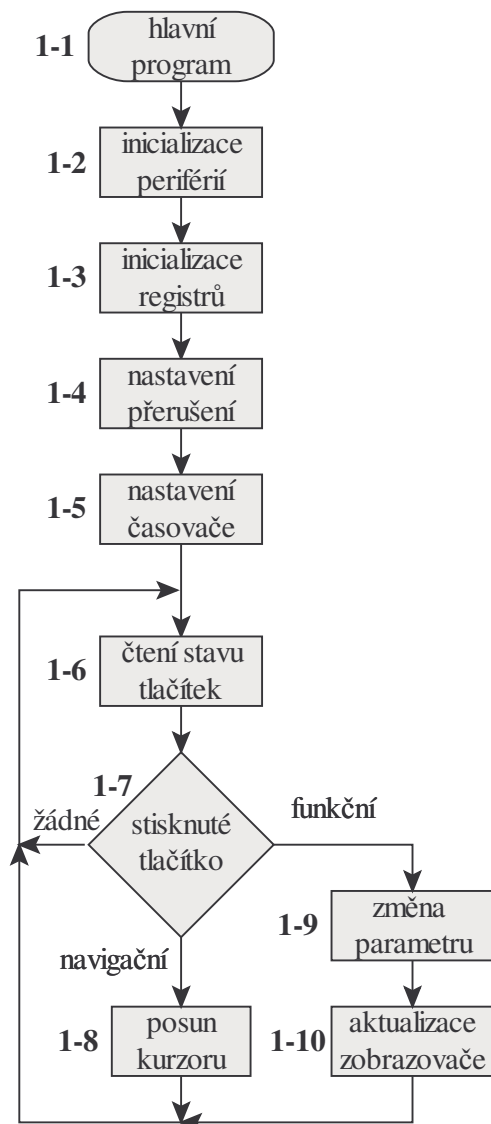
Celistvý program lze rozdělit do jednotlivých bloků. Tyto bloky se často prolínají, obzvláště při realizaci pomocí assembleru.

Myšlenka přípravku bude taková, že bude obsahovat nastavení počtu opakujících se symbolů, data jednotlivých symbolů, počtu period nosného signálu v jednom symbolu, volba modulace a volbu přednastavených ukázkových modulací.

8.1. Hlavní algoritmus

Hlavní algoritmus na obrázku 8.1. je spuštěn po připojení napájení a jeho počáteční adresa je určena vnitřní architekturou procesoru. Ihned po spuštění je nutné inicializovat (1-2) LCD a nastavit všechny vstupně-výstupní porty. Procesor a celý jeho program je závislý na stavu některých registrů a ty je nutné naplnit (1-3) potřebnými daty ještě před povolením přerušení. Jde například o informace, kde se po startu bude nacházet kurzor na zobrazovači.

Po těchto inicializacích lze povolit vnější přerušení (1-4) a nastavit jeho spouštění vzestupnou hranou na vstupním pinu INT0. Pro zajištění vysílání sériových dat bude použit časovač, který vyšle vždy



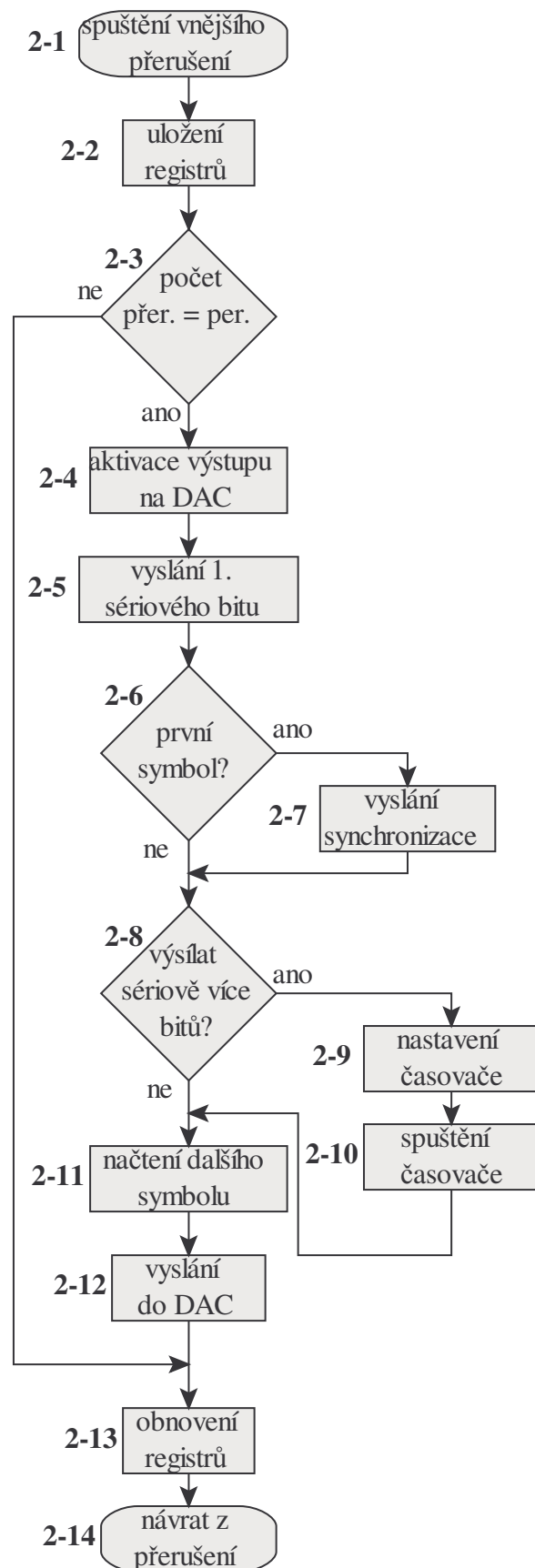
Obrázek 8.1: Hlavní algoritmus

jeden bit po přetečení časovače (1-5). Parametry časovače se musí měnit vždy společně se změnou parametrů jednotlivých modulací. Jestliže jsou nastaveny všechny prvotní potřebné parametry, modulátor bude pracovat pouze za pomoci přerušeni a časovačů. Hlavní program bude dále pracovat ve smyčce (1-6) až (1-10), která se stará o změnu parametrů za běhu modulátoru. Po přečtení obsahu tlačítek (1-6) je vyhodnocena jejich funkce (1-7). Při detekci stisknutí navigačního tlačítka program změní pozici kurzoru (1-8) na zobrazovači. Při detekci funkčního tlačítka dojde k úpravě parametru na pozici kurzoru (1-9). Následně je pak nutné zobrazit změnu na zobrazovači (1-10).

Program nelze ze smyčky (1-6) až (1-10) standardně vyvést. Smyčku lze přerušit pouze jedním signálem nemaskovatelného vnějšího přerušeni, kterým je reset a způsobí spuštění celého algoritmu od bodu (1-1).

8.2. Algoritmus přerušeni

Algoritmus vnějšího přerušeni na obrázku 8.2 je spuštěn v momentě příchodu vnějšího signálu vzestupné hrany, která je přivedena z komparátoru nosného signálu. Spuštění se provede při každé periodě a dojde k volání procedury, která leží v paměti na přesně určeném místě (vektoru přerušeni vnějšího zdroje). Obvykle jsou tyto vektory přerušeni umístovány postupně za sebou (často podle



Obrázek 8.2: Algoritmus vnějšího přerušeni

priorit) od začátku programové paměti. Dobu reakce spuštění (**2-1**) na příchod přerušení nelze dopředu předpovídat, protože než je obsluha spuštěna, musí být vykonána rozpracovaná instrukce právě zpracovávaného programu. Důležité je tedy i uložení dat (**2-2**) z registrů, které hodláme používat a pravděpodobně se mohli používat i před spuštěním obsluhy. Důležité je také uschovat stavový registr (konkrétně u ATmega registr SR) ve kterém jsou uloženy příznaky probíhajících operací. Hlavní program (nebo jiné přerušení) do kterého přerušení zasáhne nesmí být obsluhou dotčen a data které zpracovával hlavní program musí být nezměněna. Když jsou data uložena (například do zásobníku), obsluha může provést potřebný úkol.

Jestliže je nastaveno více period v jednom symbolu, je nutné provést kontrolu (**2-3**). Jednotlivé periody nosného signálu jsou počítány a je-li počet shodný s nastavenou hodnotu, program pokračuje dále (**2-4**). V opačném případě dojde k návratu dat do registrů, které byly v kroku (**2-2**) uloženy. Použité DA převodníky obsahují dvojitou paměť. První funguje jako záchytná paměť a druhá je ve funkci výstupní paměti. Aktivací výstupních dat (**2-4**) dojde k přenosu ze záchytného registru do výstupní paměti. Tyto přenosy jsou provedeny zároveň na obou převodnicích a v záchytných registrech musí existovat potřebná data (**2-12**) uložená v předchozím přerušení. Zároveň s aktivací dat (**2-4**) se vyšle první bit sériového přenosu (**2-5**). V případě, že DA převodníky vysílají první symbol (**2-6**), vyšle se synchronizační impuls (**2-7**). Použitou modulací zároveň určujeme počet přenášených bitů v jednom symbolu (viz obrázek 2.4 a tabulka 2.1). Bude-li nutné vyslat více než jeden bit (**2-8**) (všechny modulace kromě BPSK), musíme použít časovač (**2-9**) a ostatní bity vyslat opakovaným spouštěním časovače (**2-10**).

Na konci obsluhy přerušení musíme připravit další symbol (**2-11**) pro přenos do záchytného registru obou převodníků (**2-12**). Data toho symbolu budou použita v dalším přerušení. Přípravou dat v předstihu (vždy v předchozím přerušení) zmenšíme dobu reakce DA převodníků.

Na konci přerušení je nutné obnovit všechny uložené registry (**2-13**), aby se hlavnímu programu během některé operace nezměnily operandy. Návrat z přerušení (obvykle instrukcí RETI) zajistíme pokračování hlavního programu v přerušeném místě.

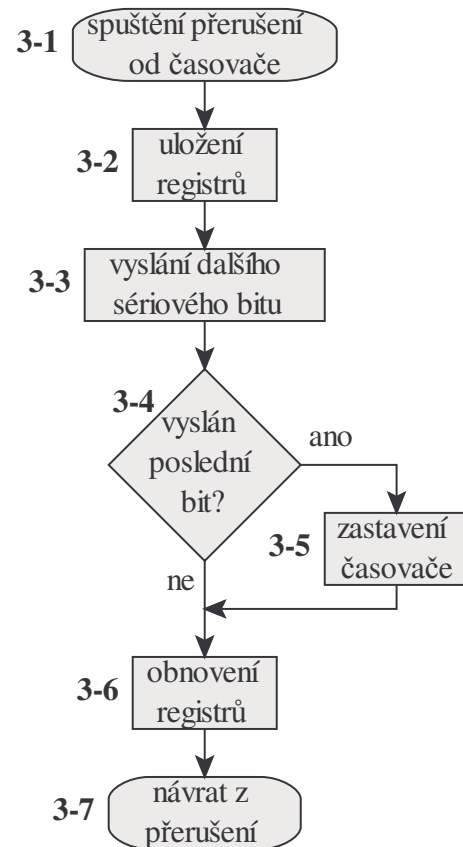
8.3. Algoritmus přerušeni od časovače

V kapitole 8.1. již bylo zmíněno, že vnitřní přerušeni od časovače je využito na vysílání sériových dat. Nastavení časovače je závislé na použité modulaci, protože různé modulace používají různý počet bitů pro vyjádření jednoho symbolu. Jestliže je oscilátor nosného kmitočtu nastaven na 10 kHz bude spuštění časovače vždy každých:

$$t = \frac{1}{b \cdot f_{osc}} \text{ [s]} \quad (8.1)$$

kde b je počet bitů, f_{osc} je kmitočet oscilátoru a t je čas spuštění časovače. Výpočet hodnot do registrů časovače bude uveden u implementace programového kódu.

Algoritmus časovače na obrázku 8.3. je spuštěn vždy po přetečení čítače, který se spustí po prvním vyslaném bitu vyslaném současně s daty I a Q. Obsluha časovače je (podobně jako obsluha vnějšího přerušeni) spuštěna z určitého místa (3-1) (vektoru) přerušeni a opět je nutné uložit obsah všech používaných registrů (3-2). Následuje vyslání dalšího bitu (3-3) a kontrola, zda je vyslaný bit poslední (3-4). V případě, že jsou vyslány všechny bity sériových dat, je časovač zastaven (3-5). Před ukončením obsluhy přerušeni (3-7) musí být obnoveny uložené hodnoty registrů (3-6).

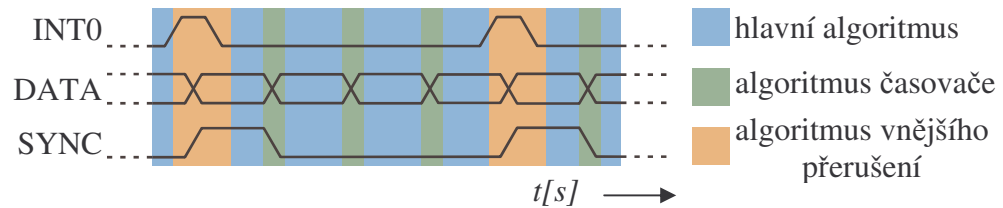


Obrázek 8.3: Algoritmus přerušeni od časovače

8.4. Přepínání algoritmů

Programový kód, který je popsán algoritmy na obrázcích 8.1 až 8.3, je spuštěn po příchodu signálu RESET (vnějšího nemaskovatelného přerušeni). Spustí se hlavní algoritmus a jsou postupně nastaveny podmínky, kdy budou spuštěny ostatní dva algoritmy. Obrázek 8.4 ukazuje, že algoritmus obsluhy vnějšího přerušeni je volán po příchodu signálu INTO. Po inicializaci připravených dat a načtení dat pro příští přerušeni je vyslán první bit sériových dat a impuls synchronizace. Zároveň je spuštěn časovač, který po vypršení spustí algoritmus obsluhy přerušeni od časovače. Vždy od skončení jednoho přerušeni do vyvolání dalšího,

zpracovává procesor instrukce hlavního algoritmu. Každý systém, který využívá přerušování musí řešit jednotlivé priority. Je nutné nastavit které přerušování lze ještě přerušit a které nelze. V našem případě teoreticky není možné, aby spuštění časovače zasáhlo do obsluhy přerušování a naopak. Obsluha vnějšího přerušování spouští časovač a ten je vždy do dalšího příchodu vnějšího přerušování neaktivní. Nastavování priorit tedy nebude nutné. Obrázek 8.4 symbolicky ukazuje časový průběh přepínání jednotlivých algoritmů.



Obrázek 8.4: Časový průběh přepínání algoritmů

9. Implementace algoritmů v assembleru pro ATmega16

Popsané algoritmy v kapitole 8 jsou striktně odděleny a jsou i samostatně realizovány. Dále popisované části programů jsou vkládány (někdy pouze volání těchto částí) do různých částí algoritmů

9.1. LCD zobrazovač

Jako zobrazovač je použit jeden z řady LCD displejů se standardním řadičem znaků HD44780. Při sestavování a testování se osvědčil 4řádkový display MC1604B-SYL, který je podsvětlený. V následujícím výpisu programu (kód 9.1), jsou vypsány používané procedury pro práci s LCD a program je platný pro jakýkoliv display kompatibilní s řadičem HD44780. Nejvíce používané procedury jsou popsány zde a ostatní jsou v elektronické příloze. Obsah funkcí a procedur se s malými změnami opakuje a proto nejsou všechny dopodrobna popsány.

Kód 9.1: Základní funkce pro LCD

```

;LCDcmd (ACC)           - pošle příkaz na LCD
;LCDdata (ACC)         - pošle data na LCD
;LCDbyte (ACC)        - pošle byte na LCD
;LCDinit              - inicializace LCD
;LCDcurON             - zapnutí blikajícího kurzoru
;LCDcurOFF           - vypnutí kurzoru
;LCDpos (ACC)        - nastaví pozici na LCD kde se budou data vypisovat
;LCDwrite            - vypíše řetězec na LCD

LCDcmd:                ;---- pošle příkaz na LCD ---- v ACC je očekáván příkaz
    CBI    LCD_CTRL,LCD_RS ;LCD_RS=0 - vyslaný byte bude příkaz
    RMP    LCDbyte        ;dále poslat byte
LCDdata:              ;---- pošle data na LCD ---- v ACC jsou očekávána data
    SBI    LCD_CTRL,LCD_RS ;LCD_RS=1 - vyslaný byte budou data
LCDbyte:              ;---- pošle byte na LCD ---- v ACC je očekáván byte
    OUT    LCD_BUS,ACC    ;zápis dat na směrnicí
    CBI    LCD_CTRL,LCD_ENA ;LCD_ENA=0 - povolení přenosu na LCD
    LDI    BUF,0
LCDbcyk:              ;spuštění čekací smyčky 255x strojový cyklus
    DEC    BUF
    BRNE   LCDbcyk
    SBI    LCD_CTRL,LCD_ENA ;LCD_ENA=1 - konec přenosu na LCD
    RET

LCDinit:              ;---- inicializace LCD ----
    CBI    LCD_CTRL,LCD_RS ;LCD_RS=0 - vyslaný byte bude příkaz
    SBI    LCD_CTRL,LCD_ENA ;LCD_ENA=1 - zakáže veškerý přenosu na LCD
    LDI    ACC,$38        ;komunikace 8 bity a použití 2 nebo 4 řádky
    RCALL  LCDcmd         ;pošle příkaz v ACC
    RCALL  LCDiwait       ;volání čekací smyčky
    LDI    ACC,$0C        ;D=1, C=0, B=1 - LCD zapnout, kurzor vypnout, blikat
    RCALL  LCDcmd         ;pošle příkaz v ACC
    RCALL  LCDiwait       ;volání čekací smyčky
    LDI    ACC,$06        ;inkrementace adr. po zápisu a kurzorem psát doprava
    RCALL  LCDcmd         ;pošle příkaz v ACC
    RCALL  LCDclrHW      ;HW smazání LCD
    RCALL  LCDiwait       ;volání čekací smyčky
    RET

LCDcurON:             ;---- zapnutí blikajícího kurzoru ----
    LDI    ACC,$0F        ;D=1, C=1, B=1 - LCD zapnout, kurzor zapnout, blikat
    RCALL  LCDcmd         ;pošle příkaz v ACC
    RET

```

```

LCDcurOFF:                ;---- vypnutí kurzoru ----
    LDI    ACC,$0C        ;D=1, C=0, B=1 - LCD zapnout, kurzor vypnout, blikat
    RCALL  LCDcmd        ;pošle příkaz
    RET

LCDpos:                   ;---- nastaví pozici na LCD ----
    ORI    ACC,$80        ;nejvyšší byt znamená adresu dat
    RCALL  LCDcmd        ;pošle příkaz v ACC
    RET

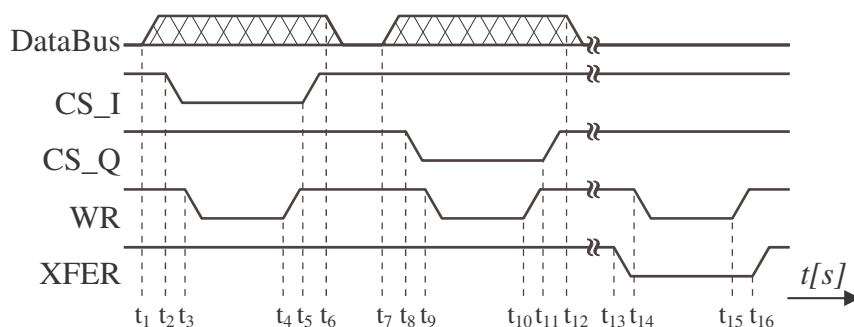
LCDwrite:                 ;---- vypíše řetězec na LCD ----
    POP    ZH            ;výběr návratové adresy - horní byte
    POP    ZL            ;výběr návratové adresy - dolní byte
    CLC
    ROL    ZL            ;vynásobení adresy 2x - datová sběrnice je 8mi bitová
    ROL    ZH            ;      oproti programové 16ti bitové sběrnici

write0:
    LPM    ACC,Z+        ;čtení dat z paměti programu
    TST    ACC           ;test obsahuje-li paměť nulu
    BRNE  writel        ;nulu neobsahuje - skok, jinak pokračovat
    LPM    ACC,Z+        ;posun Z na následující byte
    CLC
    ROR    ZH            ;vydělení adresy 2 (horní byte)
    ROR    ZL            ; ---- || ---- (dolní byte)
    IJMP
writel:
    RCALL  LCDdata        ;pošle data v ACC
    RJMP  write0         ;zpět na čtení z paměti programu

```

9.2. DA převodníky

Tyto převodníky zajišťují vysílání vícestavových signálů I a Q a v konstrukci je použita dvojice převodníků DAC0832. Jeden tento převodník obsahuje dva datové registry. Do prvního jsou data vložena při přenosu z procesoru, ale na analogovém výstupu není znatelná žádná změna, a teprve přepsáním dat z prvního do druhého registru dojde k aktivaci dat na výstupu. V katalogovém listu výrobce [4] je popsán mechanismus, jakým lze oba registry na obou převodnicích obsluhovat. Časový diagram na obrázku 9.1. ukazuje, jakým způsobem se data do převodníků přenášejí. Signálem DataBus je označena datová sběrnice pro přenos k obou převodníkům. Nastavením CS_I do úrovně L zajistíme komunikaci s převodníkem, který zajišťuje analogový signál I. Podobně lze vybrat druhý převodník CS_Q pro signál Q. Signálem WR zapisujeme data do převodníku, na kterém je CS nastaveno do úrovně L. Jestliže jsou data v prvním záchytném registru, můžeme je kombinací signálů WR a XFER poslat na výstup.



Obrázek 9.1: Časový diagram přenosu dat na DA převodníky

Výhodou tohoto zdánlivě složitého postupu je současný výstup na obou převodnicích. Obrázek 9.1. přehledně popisuje tento mechanismus a tabulka 9.1 popisuje kroky v jednotlivých intervalech.

Tabulka 9.1: Operace v intervalech při přenosu na DAC

Interval	Operace
$t_{1\div 6}$	platná data I
$t_{2\div 5}$	výběr převodníku I
$t_{3\div 4}$	povolení přenosu
$t_{7\div 12}$	platná data Q
$t_{8\div 11}$	výběr převodníku Q
$t_{9\div 10}$	povolení přenosu
$t_{13\div 16}$	signál pro přepis do druhé paměti (obou převodníků)
$t_{14\div 15}$	povolení přenosu

Následujícím výpisem (kód 9.2) je princip přepsán do assembleru a stručně popsán.

Kód 9.2: Základní funkce pro DA převodníky

```

;DACinit          - inicializace obou DA převodníků
;DAC_Ibuf(ACC)   - pošle data I do prvního záchytného registru DAC-I
;DAC_Qbuf(ACC)   - pošle data Q do prvního záchytného registru DAC-Q
;DACout          - DAC přepíše data ze záchytných registrů do výstupních

DACinit:         ;--- inicializace obou DA převodníků ---
    SBI    DAC_CTRL,DAC_CS_I ;výběr převodníku I reaguje na sestupnou hranu
    SBI    DAC_CTRL,DAC_CS_Q ;výběr převodníku Q reaguje na sestupnou hranu
    SBI    DAC_CTRL,DAC_WR   ;zápis dat reaguje na sestupnou hranu
    SBI    DAC_CTRL,DAC_XFER ;signál pro aktivaci výstupu reaguje na sest. hranu
    RET

DAC_Ibuf:        ;--- pošle data I do prvního registru DAC-I ----
    OUT    DAC_BUS,ACC      ;v ACC jsou očekávána data - pošlou se na sběrnici
    CBI    DAC_CTRL,DAC_CS_I ;výběr DAC-I - reaguje ne sestupnou hranu
    CBI    DAC_CTRL,DAC_WR   ;povolení přenosu - reaguje ne sestupnou hranu
    RCALL  WAIT             ;volání čekací smyčky
    SBI    DAC_CTRL,DAC_WR   ;dokončení přenosu (úroveň H)
    SBI    DAC_CTRL,DAC_CS_I ;zrušit výběr DAC-I (úroveň H)
    RET

DAC_Qbuf:        ;--- pošle data Q do prvního registru DAC-Q ----
    OUT    DAC_BUS,ACC      ;v ACC jsou očekávána data - pošlou se na sběrnici
    CBI    DAC_CTRL,DAC_CS_Q ;výběr DAC-Q - reaguje ne sestupnou hranu
    CBI    DAC_CTRL,DAC_WR   ;povolení přenosu - reaguje ne sestupnou hranu
    RCALL  WAIT             ;volání čekací smyčky
    SBI    DAC_CTRL,DAC_WR   ;dokončení přenosu (úroveň H)
    SBI    DAC_CTRL,DAC_CS_Q ;zrušit výběr DAC-Q (úroveň H)
    RET

DACout:         ;--- nařídí převodníkům přepsat data . . . ----
    CBI    DAC_CTRL,DAC_XFER ;signál pro aktivaci výstupu reaguje na sest. hranu
    CBI    DAC_CTRL,DAC_WR   ;povolení přenosu - reaguje ne sestupnou hranu
    RCALL  WAIT             ;volání čekací smyčky
    SBI    DAC_CTRL,DAC_WR   ;dokončení přenosu (úroveň H)
    SBI    DAC_CTRL,DAC_XFER ;vrátí zpět nastavení signálu XFER (úroveň H)
    RET

```

9.3. Ovládací prvky

Načítání stavu tlačítek je opakovaně spouštěno v hlavním programu podle vývojového diagramu na obrázku 8.1. Tato funkce načte stav tlačítek a uloží číslo stisknutého tlačítka do registru. Jakým způsobem bude provedena reakce na tlačítko, závisí na programu, který tuto funkci spustil. Následující kód 9.3 popisuje funkci `GetBut`, která uloží do registru číslo stisknutého tlačítka. V případě, že tlačítko stisknuté není, vloží do registru nulu. Funkce `WaitBut` čeká na uvolnění tlačítka a je zpravidla volána po reakci na určité tlačítko.

Kód 9.3: Základní funkce pro obsluhu tlačítek

```

GetBut:                ;--- uloží stav tlačítek ----
RCALL WAIT            ;volání čekací smyčky
IN    ACC,BUT         ;načtení stavu tlačítek z
COM    ACC             ;stisk tlačítek - sestupná hrana, proto negace
RCALL WAIT            ;volání čekací smyčky (eliminace zákmitů)
RET

WaitBut:              ;--- čeká na uvolnění ----
RCALL WAIT            ;volání čekací smyčky
RCALL Getbut          ;volání čtení stavu tlačítek
CPI    ACC,0          ;zjištění jestli je stále stisknuté tlačítko
BRNE  WaitBut        ;pokud je stisknuté opakuj
RCALL WAIT            ;pokud není, volej čekací smyčku (eliminace zákmitů)
RET

```

9.4. Nastavení přerušení

Pro složitější zařízení je systém přerušení velice výhodný. Během zpracovávání hlavního programu lze vykonávat i kratší programy při příchodu vnějšího přerušení nebo při vypršení vnitřního časovače. Správné nastavení těchto přerušení je dobrým předpokladem pro správnou funkci. Protože návrh obsahuje procesor ATmega16 (standardní AVR), nastavení vnějšího přerušení je provedeno za pomoci registru MCUCR (tabulce 9.2) a registru GICR (tabulce 9.4). Přerušení musí přijít vždy při detekci průchodu nulou funkce sinus (tedy vzestupná hrana z komparátoru). V tabulkách jsou popsány jen ty bity, které využijeme. Ostatní popis můžeme nalézt v popisu výrobce [2].

Tabulka 9.2: Nastavení vnějšího přerušení – registr MCUCR

MSB	7	6	5	4	3	2	1	0	LSB
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	

Tabulka 9.3: Nastavení vnějšího přerušení – popis registru MCUCR

ISC01	ICS00	Popis
0	0	úroveň L na INT0 spouští přerušení
0	1	změna úrovně na INT0 spustí přerušení
1	0	sestupná hrana spustí přerušení
1	1	vzestupná hrana spustí přerušení

Z tabulky 9.2 a 9.3 vidíme, že musíme nastavit $ISC01 = 1$ a $ISC00 = 0$. Registr MCUCR naplníme hodnotou \$02. V registr GICR dále povolíme přerušení pouze od vnějšího zdroje podle tabulky 9.4. Bit $INT0 = 1$ a registr GICR naplníme hodnotou \$40

Tabulka 9.4: Nastavení vnějšího přerušení – registr GICR

MSB	7	6	5	4	3	2	1	0	LSB
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	

Následující programový kód 9.4 nastavuje popsané registry pro vnější přerušení:

Kód 9.4: Nastavení a povolení vnějšího přerušení

```
LDI ACC, $03 ;2 - sestupná; 3 - vzestupná
OUT MCUCR, ACC ;nastaví vnější přerušení na sestupnou hranu
LDI ACC, $40
OUT GICR, ACC ;povolení přerušení od vnějšího přerušení 0; INT0=1

SEI ;povolení globálního přerušení
```

Časovač je použit pro vysílání sériových dat a je nastaven za pomoci registru TCCR0 (tabulka 9.5), OCR0 (vztah 9.1) a TIMSK (tabulka 9.8). Registrem TCCR0 nastavíme mód a zdroj impulsů pro časovač podle tabulky 9.6 a 9.7.

Tabulka 9.5: Nastavení časovače – registr TCCR0

MSB	7	6	5	4	3	2	1	0	LSB
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	

Tabulka 9.6: Nastavení časovače – volba módu v registru TCCR0

Mód	WGM01	WGM00	Popis
0	0	0	normal – čítá od \$00 ÷ \$FF
1	0	1	PWM
2	1	0	CTC – čítá od \$00 po obsah registru OCR0
3	1	1	rychlá PWM

Tabulka 9.7: Nastavení časovače – volba zdroje v registru TCCR0

CS02	CS01	CS00	Popis
0	0	0	bez zdroje (časovač vypnut)
0	0	1	clk (oscilátor)
0	1	0	clk/8 (výstup z předděličky)
0	1	1	clk/64 (výstup z předděličky)
1	0	0	clk/256 (výstup z předděličky)
1	0	1	clk/1024 (výstup z předděličky)
1	1	0	externí zdroj na vstupu T0 – sest. hrana
1	1	1	externí zdroj na vstupu T0 – vzest. hrana

Časovač musí během jednoho symbolu spustit tolikrát, kolik je nutné přenést bitů. V případě přípravku s kmitočtem nosné 10 kHz a nastavenou modulací například 16QAM musíme během jednoho symbolu přenést čtyři bity. Přerušení od časovače musí spouštět čtyřikrát rychleji než vnější přerušení. V našem případě pro 16QAM musí být kmitočet 40 kHz. Výpočet registru OCR0 pro časovač módu 2 o kmitočtu 40 kHz je následující:

$$OCR0 = \frac{f_{osc}}{presc} = \frac{16 \cdot 10^6}{40 \cdot 10^3} = 50 \text{ [-, Hz]} \quad (9.1)$$

Kde f_{osc} je kmitočet oscilátoru, $presc$ předdělička a $f_{counter}$ požadovaný kmitočet pro časovač. Kmitočet 40 kHz tedy nastavíme (podle tabulek 9.5–9.7) TCCR0 = \$0A a OCR0 = 50.

Tabulka 9.8: Nastavení časovače – registr TIMSK

MSB	7	6	5	4	3	2	1	0	LSB
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	

Nastavením bitu OCIE0 zapneme přerušení, které se spustí při dosažení registru TCNT0 do hodnoty uložené v OCR0, tedy při rovnosti $OCR0 = TCNT0$. Registr TIMSK naplníme hodnotou \$02.

Následující kód 9.5 nastavuje popsané potřebné registry pro časovač a jeho přerušení.

Kód 9.5: Nastavení časovače a povolení přerušení

LDI	ACC, \$0A	
OUT	TCCR0, ACC	; nastaví (oscilátor/8) jako zdroj signálu časovače
LDI	ACC, 50	
OUT	OCR0, ACC	; porovnávací hodnota přerušení (záleží na modulaci)
LDI	ACC, \$02	
OUT	TIMSK, ACC	; povolí přerušení od časovače; OCIE0=1
SEI		; povolení globálního přerušení

9.5. Konverze dat symbolu na data I a Q

Datovou konverzi pro větve I a větve Q je výhodné provést za pomoci tabulek. Algoritmus pro různé modulace zůstává stejný a tabulky lze kdykoliv měnit a přidávat. Paměťová náročnost stoupá s použitým počtem modulací, ale při výběru z tabulky dochází k velmi rychlému převodu.

Vytváření tabulek pro jednotlivé modulace vychází z obrázků 2.4. Maximální úrovně složek I a Q jsou normovány a tedy označeny úrovní 1. Výstupní kodér (oba DA převodníky) je osmibitový a úroveň -1 v obrázku 2.4 odpovídá nejnižší hodnotě DA převodníku. Úroveň 1 v obrázku bude odpovídat maximální hodnotě. Další úrovně jsou podle vztahu 9.2, kde je normovaná hodnota označena jako NH . Příklad pro modulaci 16QAM je v tabulce 9.9.

$$I, Q = \frac{1 + NH}{2} \cdot 255 \quad [-] \quad (9.2)$$

Tabulka 9.9: Hodnoty pro vytvoření tabulky symbolů modulace 16QAM

NH	I, Q ₍₁₀₎	I, Q ₍₁₆₎
-1,0	0	0x00h
-0,33	85	0x55h
-0,33	170	0xAAh
1,0	255	0xFFh

Za pomoci obrázku 2.4d a tabulky 9.9 seřadíme jednotlivé symboly a dostaneme tabulku symbolů 9.10 pro modulaci 16QAM

Tabulka 9.10: Hodnoty I a Q symbolů modulace 16QAM

kvadrant	symbol ₍₁₀₎	symbol ₍₂₎	I ₍₁₆₎	Q ₍₁₆₎
1	0	0000	0xFFh	0xFFh
	1	0001	0xAAh	0xFFh
2	2	0010	0x00h	0xFFh
	3	0011	0x55h	0xFFh
4	4	0100	0xFFh	0x00h
	5	0101	0xAAh	0x00h
3	6	0110	0x00h	0x00h
	7	0111	0x55h	0x00h
1	8	1000	0xFFh	0xAAh
	9	1001	0xAAh	0xAAh
2	10	1010	0x00h	0xAAh
	11	1011	0x55h	0xAAh
4	12	1100	0xFFh	0x55h
	13	1101	0xAAh	0x55h
3	14	1110	0x00h	0x55h
	15	1111	0x55h	0x55h

Tabulku dat přepíšeme do zdrojového kódu tak, aby v paměti na nižším byte jednoho slova byla informace I a na vyšším byte informace Q (kód 9.6). Ze znalosti uspořádání uložených dat v tabulce můžeme vytvořit program (kód 9.7) pro používání těchto dat.

Kód 9.6: Tabulka pro převod dat modulace 16QAM

QAM16Tab:		; tabulka symbolů 16QAM	
.DB	0xFF, 0xFF	; symbol	0000 -- 0
.DB	0xAA, 0xFF	; symbol	0001 -- 1
.DB	0x00, 0xFF	; symbol	0010 -- 2
.DB	0x55, 0xFF	; symbol	0011 -- 3
.DB	0xFF, 0x00	; symbol	0100 -- 4
.DB	0xAA, 0x00	; symbol	0101 -- 5
.DB	0x00, 0x00	; symbol	0110 -- 6
.DB	0x55, 0x00	; symbol	0111 -- 7
.DB	0xFF, 0xAA	; symbol	1000 -- 8
.DB	0xAA, 0xAA	; symbol	1001 -- 9
.DB	0x00, 0xAA	; symbol	1010 -- 10
.DB	0x55, 0xAA	; symbol	1011 -- 11
.DB	0xFF, 0x55	; symbol	1100 -- 12
.DB	0xAA, 0x55	; symbol	1101 -- 13
.DB	0x00, 0x55	; symbol	1110 -- 14
.DB	0x55, 0x55	; symbol	1111 -- 15

Podobně jako jsme vytvořili tabulku pro modulaci 16QAM můžeme vytvořit ostatní tabulky použitých modulací pomocí obrázků 2.4 a vztahu 9.1. Tabulky pro převod ostatních modulací jsou uvedeny v programu v elektronické příloze.

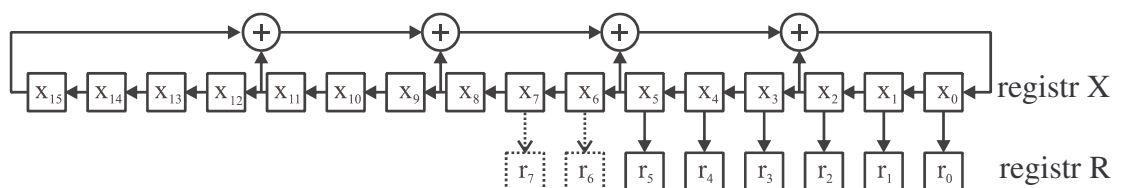
Kód 9.7: Používání tabulek pro převod dat

QAM16code:		
LDI	ZH, HIGH(QAM16Tab*2);	načtení adresy tabulky do registru Z (horní byte)
LDI	ZL, LOW(QAM16Tab*2);	načtení adresy tabulky do registru Z (dolní byte)
ANDI	ACC, \$0F	; maskování - spodní čtyři bity využijeme
CLC		
ROL	ACC	; číslo symbolu vynásobíme, hledáme dvojnásobnou adr.
ADD	ZL, ACC	; přičtení k registru Z; dolní byte
CLR	ACC	
ADC	ZH, ACC	; možné přetečení ZL přes 0xFF přičteme do horního byte
LPM	IBUF, Z+	; registr Z ukazuje (jeho adresa) na I; vyzvedneme a Z=Z+1
LPM	QBUF, Z+	; registr Z ukazuje na Q; vyzvedneme a Z=Z+1
RET		

9.7. Pseudonáhodný generátor dat

Generátor pseudonáhodné posloupnosti je nakreslen na obrázku 9.1. Jde o 16bitový posuvný registr, který je doplněn čtyřmi odbočkami pro exklusivní sčítání (XOR). Při rotaci celého registru X je bit x_0 naplněn hodnotu, která je výsledkem operace ze vztahu 9.3.

$$x_0 = x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{15} [-] \quad (9.3)$$

**Obrázek 9.1:** Generátor pseudonáhodných dat

Rotace v posuvném registru je spuštěna tolikrát, kolik bitů obsahuje jeden symbol. To zajistí naplnění všech potřebných bitů v registru R náhodnou posloupností. Registr se po vymaskování² pouze potřebných bitů využije jako zdroj dat pro symbol. Pro modulaci 16QAM je rotace provedena o 4 bity a dále podle tabulky 2.1. Posun o více než 6 bitů není nutný, protože používáme modulaci nejvýše 64QAM s 6 bity. Následující okomentovaný výpis programu (kód 9.8) řeší pseudonáhodný generátor ve zdrojovém kódu.

Kód 9.8: Pseudonáhodný generátor

```

GenerateSymb:                ;funkce generování náhodného byte (pouze n bitů)
    LDS    REG,Bits_per_symb.;načtení počtu bitů na jeden symbol
GenerateSmb:
    RCALL  Gener              ;volání funkce pro rotaci registru
    DEC   REG                 ;počítání provedených cyklů
    CPI   REG,0               ;není-li u konce
    BRNE  GenerateSmb        ;opakovat
    RET

Gener:                        ;funkce pro jednu rotaci registrem generátoru
    LDS   REG1,GenDat1        ;vzvednutí nižšího byte registru pro generátor ze SRAM
    LDS   REG2,GenDat2        ;vzvednutí vyššího byte registru pro generátor ze SRAM
    CLC
    ROL   REG1                ;rotace doleva nižšího byte
    ROL   REG2                ;rotace doleva vyššího byte
    BRCS  SetReg1_HI          ;jestliže přetekl byte přes vyšší byte skok
    CLT
    RUMP  BitStore            ;uložit uživatelský příznak
SetReg1_HI:
    SET
    ;nastavení uživatelského příznaku
BitStore:
    BLD   REG1,0              ;uložit uživatelský příznak na nejnižší bit
    STS   GenDat1,REG1        ;uloží 1x zrotovaný spodní byte zpět do SRAM
    STS   GenDat2,REG2        ;uloží 1x zrotovaný horní byte zpět do SRAM

    LDS   REG1,GenDat2        ;přečtení horního byte ze SRAM
    BST   REG1,3              ;přečtení třetího bitu - odpovídá x11
    CLR   REG1                ;vynulování registru
    BLD   REG1,0              ;zapsání uloženého bitu x11 na pozici 0 nového registru
    EOR   REG2,REG1           ;exclusive-or s přečteným bitem

    LDS   REG1,GenDat2        ;přečtení horního byte ze SRAM
    BST   REG1,0              ;přečtení prvního bitu - odpovídá x8
    CLR   REG1                ;vynulování registru
    BLD   REG1,0              ;zapsání uloženého bitu x11 na pozici 0 nového registru
    EOR   REG2,REG1           ;exclusive-or s přečteným bitem

    . . . . .                ;dále podobně pro x5 a x2 v SRAM GenDat1

    LDS   REG1,GenDat1        ;načtení vyššího byte z paměti
    EOR   REG1,REG2           ;exclusive-or s existujícím nultým bitem a registrem REG2
    STS   GenDat1,REG1        ;návrát dat zpět do SRAM, v REG1 jsou výstupní data
    RET

```

Výstupní posloupnost tohoto generátoru je jednoznačně určena strukturou (obrázek 9.1) a počátečním stavem registrů GenDat1 a GenDat2. Parametry generátoru nelze jednoduše uživatelsky změnit, protože je nutné upravit inicializační data, nebo část programu. Tento generátor ale naprosto postačuje, protože změna posloupnosti není nutná. Dobré vlastnosti naznačují i výsledky měření dále v této práci.

² Vymaskování se provádí operací AND. Například pro modulaci 64QAM se 6ti bity na symbol (viz tab. 2.1) se použije maska 00111111

9.7. Celkový programový návrh

Nejdůležitější popisované části kódů v kapitole 9 jsou ve výsledném programu použity s drobnými změnami a doplněny dalším kódem pro práci s tlačítky a LCD. Celkový program se skládá z několika souborů, které jsou součástí elektronické přílohy k této práci.

Soubory s kódem a jejich obsah:

QAM_Main.asm	– hlavní program
QAM_MemoryMap.inc.asm	– uspořádání registrů a paměti SRAM
QAM_InitStart.inc.asm	– inicializační část programu
QAM_CoderFunctions.inc.asm	– funkce generátoru a kodéru
QAM_BUTFunctions.inc.asm	– funkce pro tlačítka
QAM_DACFunctions.inc.asm	– funkce pro DAC převodník
QAM_LCDFunctions.inc.asm	– funkce pro LCD zobrazovač
QAM_Preset.inc.asm	– hodnoty tabulek presetů
QAM_ModTable.inc.asm	– hodnoty tabulek modulací

10. Konstrukce

Při návrhu desky plošných spojů již bylo počítáno s umístěním do plastové krabice o rozměrech 45x190x138mm.

K vytvoření otvorů do plastové krabice slouží šablona, která je v měřítku 1:1 součástí přílohy. Přiložením této šablony na horní víko lze označit středy pro vyvrtání otvorů.

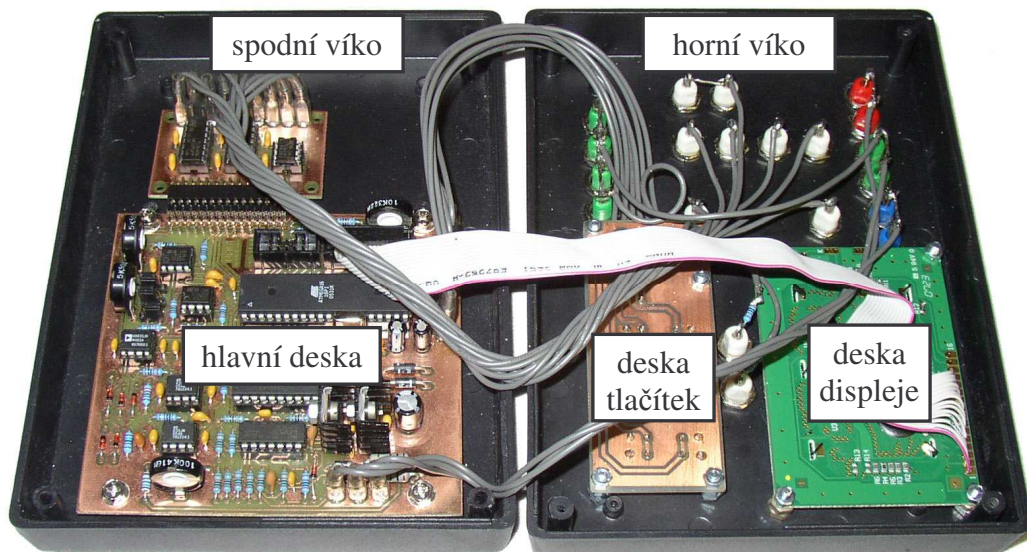
Chemicky ošetřená a osazená hlavní deska plošných spojů je pomocí čtyř šroubů umístěna na spodní část krabice (vlevo na obrázku 10.1). Deska displeje a deska s tlačítky je připevněna na horním víku (vpravo na obrázku 10.1).

Potisk přípravku je sestaven z bloků uvedených na obrázku 2.5. Před nalepením potisku je nutné namontovat LCD displej a tlačítka. Černobílý potisk (negativní i pozitivní) je součástí přílohy.

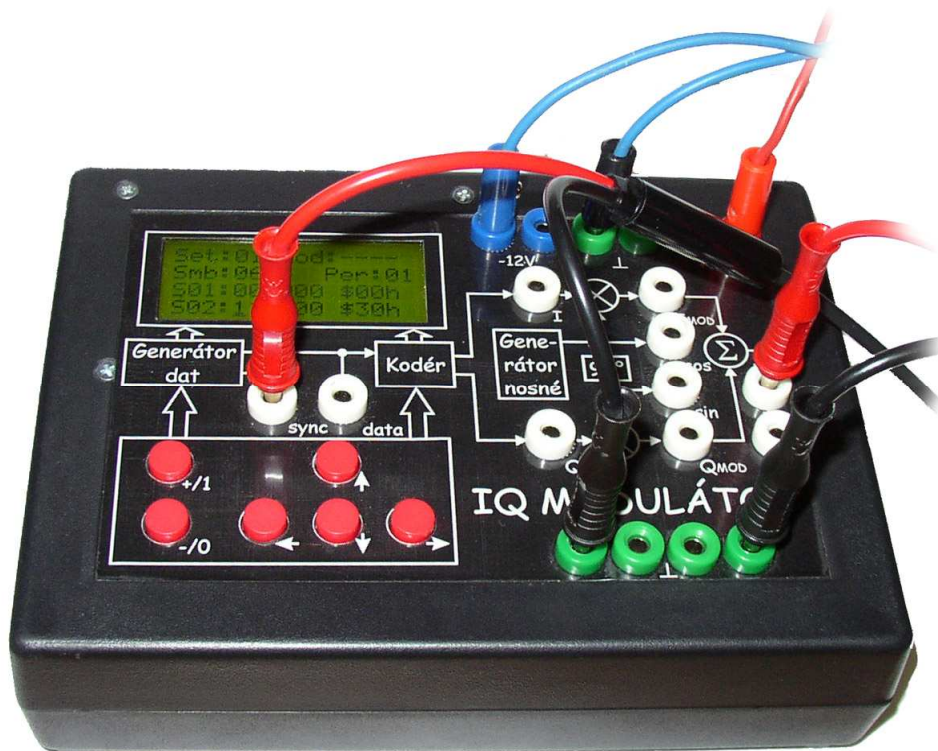
Barevné rozvržení přístrojových zdírek je podle tabulky 10.1 a obrázku 12.1. Rozmístění vidíme i na fotografii na obrázku 10.2

Tabulka 10.1: Barevné rozlišení přístrojových zdírek

Barva	Použití	Počet
červená	vstupní svorka napájení +12 V	2
modrá	vstupní svorka napájení -12 V	2
zelená	vstupní i výstupní svorka 0 V	6
bílá	výstupní signálová svorka	10



Obrázek 10.1: Fotografie rozmístění prvků v prototypu

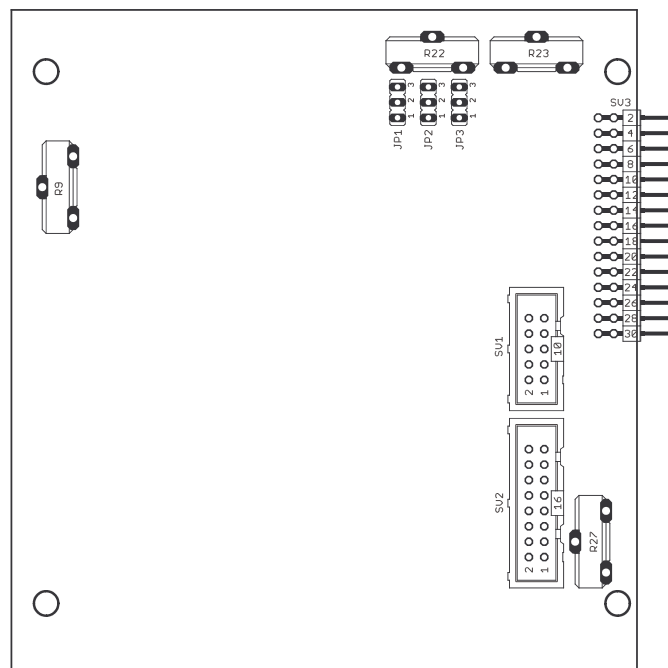


Obrázek 10.2: Fotografie kompletního laboratorního přípravku

11. Oživení přípravku

11.1. Počáteční nastavení

Před připojením přípravku ke zdroji napětí je nutné provést základní nastavení. Všechny trimry (R_9 , R_{22} , R_{23} a R_{27}) nastavíme do střední polohy. Všechny tři propojky JP_1 , JP_2 a JP_3 nastavíme na pozici 2-3. Do konektoru SV_2 připojíme LCD displej a do konektoru SV_1 programátor s rozhraním ISP. Poloha těchto trimrů a propojek je na obrázku 11.1.



Obrázek 11.1: Nastavovací prvky na přípravku

11.2. Nahrání programu do flash paměti procesoru

Program do procesoru ATmega16 nahrajeme pomocí rozhraní ISP (SV_1 na obrázku 11.1). Při programování prototypu byl použit programátor STK200 s vývojovým prostředím AVR Studio 4. Konfigurační a zabezpečovací bity nastavíme podle tabulky 11.1, které jsou podrobně popsány ve [2]. Nastavením těchto bitů zajistíme spuštění vnitřního oscilátoru za pomoci vnějšího krystalu a vyvolání resetu, když napájecí napětí dosáhne hodnoty 4 V. Nastavením bitů Lock na úroveň 1 dovolíme libovolné čtení programu z flash paměti za pomoci programátoru.

Tabulka 11.1: Konfigurační a bezpečnostní bity

-	-	BootLock12	BootLock11	BootLock02	BootLock02	Lock2	Lock1
x	x	1	1	1	1	1	1

OCDEN	JTAGEN	-	CKOPT	EESAVE	BOOTSZ1	BOOTSZ0	BOOTRST
1	1	x	0	1	0	0	1

BODLEVEL	BODEN	SUT1	SUT0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
1	1	1	0	1	1	1	0

11.3. Nastavení a ověření hardware

1. Nastavení kontrastu

Po naprogramování můžeme přípravek spustit (spuštění indikuje zapnuté podsvícení displeje) a vyzkoušet reakce tlačítek a zobrazování údajů na displeji. Jestliže nejsou údaje na displeji čitelné, vyladíme kontrast trimrem R₂₇ (pozice vyobrazena na obrázku 11.1).

2. Nastavení kmitočtu

Nastavení kmitočtu nosného signálu provedeme pomocí čítače nebo digitálního osciloskopu, který připojíme do výstupu z generátoru (zdířka s označením „cos“). Nastavení přesného kmitočtu 10 kHz zajistíme pomocí trimru R₉ (pozice vyobrazena na obrázku 11.1)

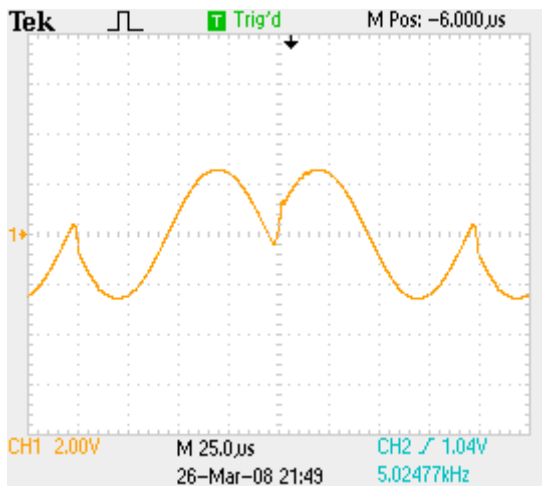
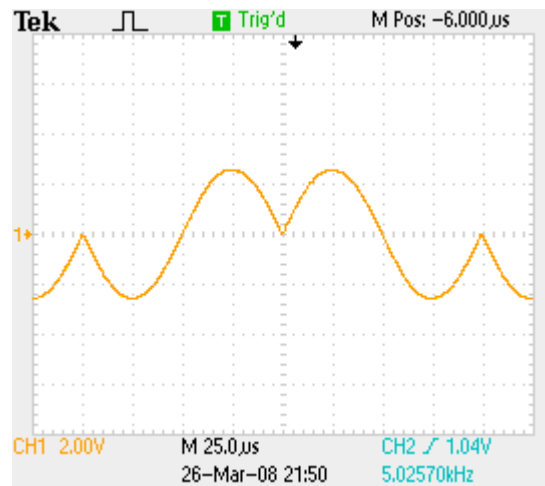
3. Vyladění komparátoru

Přerušení je spouštěno vzestupnou hranou, proto nastavíme záporné předpětí (ještě než projde sinusový signál nulou, projde signál nastaveným záporným předpětím) propojkami JP₂ a JP₃ na pozici 2-3. Tyto propojky mají souvislost s použitým programem v procesoru. Propojkou JP₁ nastavíme modulování soufázové větve kosinusoidou a kvadrurní větve sinusoidou (pozice 2-3). Tabulka 11.2 popisuje nastavení ostatních možností JP₁, JP₂ a JP₃.

Na komparátoru je nutné nastavit předstih tak, aby se symboly generovaly ve správný okamžik. K vyladění potřebujeme osciloskop připojený na výstup (zdířka „out“ na přípravku). Na přípravku nastavíme modulaci BPSK a střídání symbolů 0 a 1. Vyladění provedeme pomocí trimru R₂₂ tak, abychom dostali stejný výsledek jako na obrázku 11.2b. Před vyladěním komparátoru musí být kmitočet nosného signálu již nastaven.

Tabulka 11.2: Propojky JP₁, JP₂ a JP₃

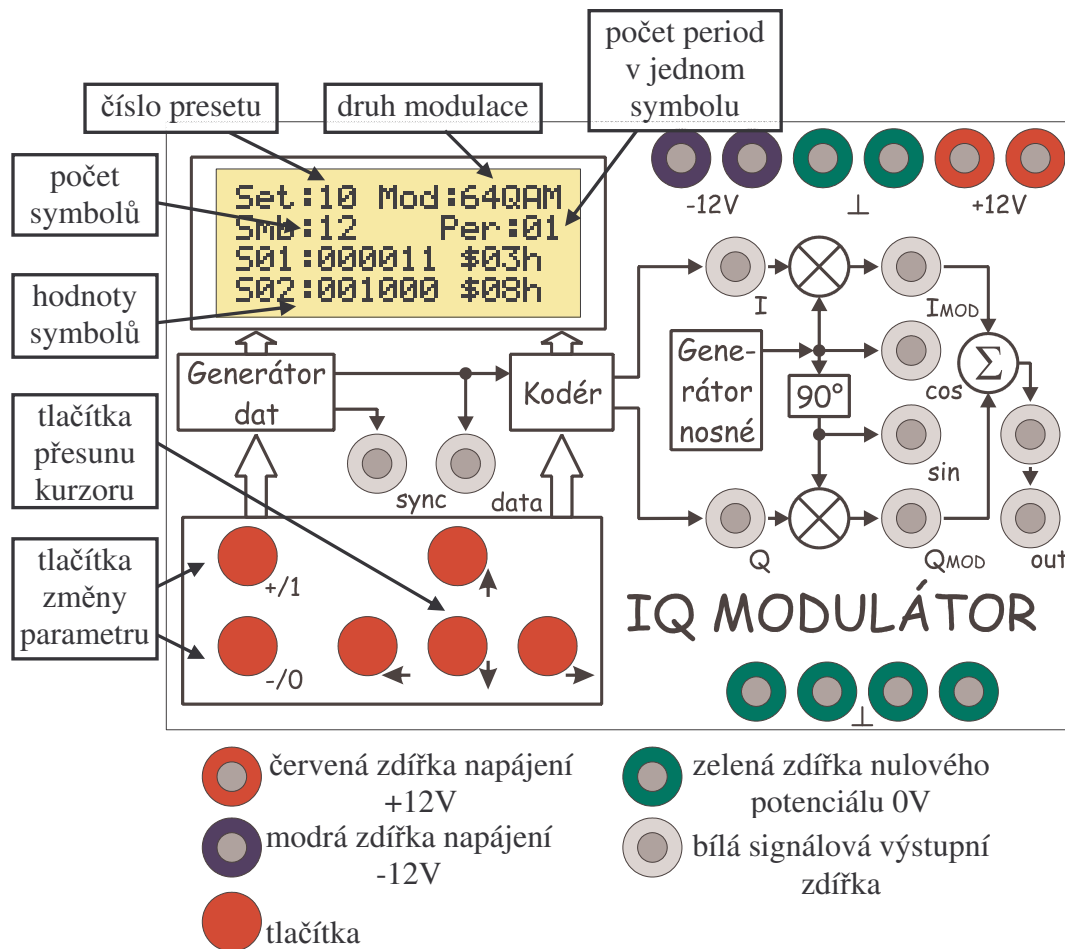
přepínače		Funkce
JP ₁	1-2	Soufázová větev je modulována sinusoidou, kvadrurní větev je modulována kosinusoidou. Vztah popisující výstup: $out = I \cdot \sin + Q \cdot \cos$
	2-3	Soufázová větev je modulována kosinusoidou, kvadrurní větev je modulována sinusoidou. Vztah popisující výstup: $out = I \cdot \cos + Q \cdot \sin$
JP ₂ , JP ₃	1-2	Kladné předpětí komparátoru - sestupná hrana spouští přerušení (závisí na programu)
	2-3	Záporné předpětí komparátoru - vzestupná hrana spouští přerušení (závisí na programu)

**Obrázek 11.2a:** Oscilogram špatně nastaveného komparátoru**Obrázek 11.2b:** Oscilogram správně nastaveného komparátoru

4. Hystereze komparátoru

Hysterezní křivka komparátoru zamezuje rozkmitání při průchodu nulou. Ke změně hystereze můžeme použít trimr R₂₃ (pozice vyobrazena na obrázku 11.1), který je popsán v zapojení od výrobce [6]. Pozici trimru ze střední polohy není nutné měnit.

12. Uživatelský popis přípravku



Obrázek 12.1: Popis přípravku

12.1. Zapojení přípravku

K zapojení přípravku využijeme symetrický napájecí zdroj $\pm 12\text{V}$. Napájecí zdířky jsou zdvojené a barevně odlišeny. Do modré zdířky zapojíme napájecí napětí -12V , do červené $+12\text{V}$ a do zelené nulový potenciál (GND). Na přípravku není žádná zdířka ve funkci vstupu a všechny ostatní (bílé) zdířky jsou výstupní. Výstupní zdířka „sync“ slouží pro synchronizaci osciloskopu a impuls na tomto výstupu vždy označuje první bit prvního symbolu. Na výstupu označeném jako „data“ můžeme měřit sériový přenos dat od generátoru ke kodéru. Přenáší se vždy jako první nejvýznamnější bit (MSB). Na ostatních zdířkách můžeme měřit signály, které naznačuje potisk na přípravku i obrázek 12.1.

12.2. Nastavování parametrů

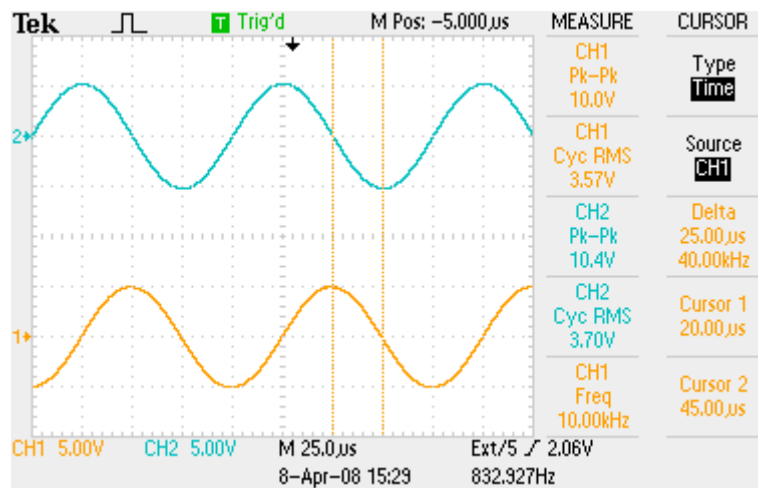
Přípravek lze obsluhovat pomocí šesti tlačítek. Čtyři slouží pro pohyb kurzoru a dvě pro změny parametrů. Rozmístění vidíme na obrázku 12.1. Přípravek obsahuje několik přednastavených presetů, které můžeme vybírat tlačítky +/1 a -/0 jakmile kurzor nastavíme na parametr `Set`. Nastavením presetu číslo nula budou symboly generovány pomocí pseudonáhodného generátoru. Změnou parametru `Mod` můžeme volit typ modulace. Parametr `Per` slouží pro určení počtu period nosného signálu v jednom symbolu. Kromě přednastavených presetů můžeme nastavit až 40 uživatelských symbolů. Počet symbolů určuje parametr `Smb` a tlačítka +/1 a -/0 můžeme počet měnit. Po nastavení počtu můžeme symboly naplňovat uživatelskými daty. Symboly jednotlivých modulací nastavíme podle bitové kombinace, které určují konstelační diagramy na obrázku 2.4. Nastavení opět provádíme za pomoci tlačítek přesunu kurzoru a tlačítek +/1 a -/0. Přípravek opakovaně posílá všechny uživatelsky nastavené symboly a po vyslání posledního symbolu začne vysílání opět od prvního symbolu.

13. Měření

Měření na přípravku je uvedeno s krátkým popisem tak, aby bylo možné měření zopakovat.

13.1. Měření na kvadrurním oscilátoru

Při měření nosného signálu na kvadrurním oscilátoru není nutné přípravek nějak nastavovat. Na nosném signálu můžeme hodnotit přesně nastavený kmitočet a ověřit posun fáze obou výstupních signálů kvadrurního oscilátoru. Chyba v těchto signálech se dále zanáší do výsledného modulovaného signálu. Na obrázku 13.1 můžeme vidět naměřený oscilogram s vyznačeným časovým posunem.



Obrázek 13.1: Měření kvadrurního oscilátoru (nosný signál)

Fázový posun vypočteme pomocí jednoduchého vztahu 13.1

$$\alpha = t_p \cdot f \cdot 360 \text{ [}^\circ, \text{s, Hz]} \quad (13.1)$$

Kde α je úhel posunu, t_p je časový posun dvou signálů a f je kmitočet nosného signálu.

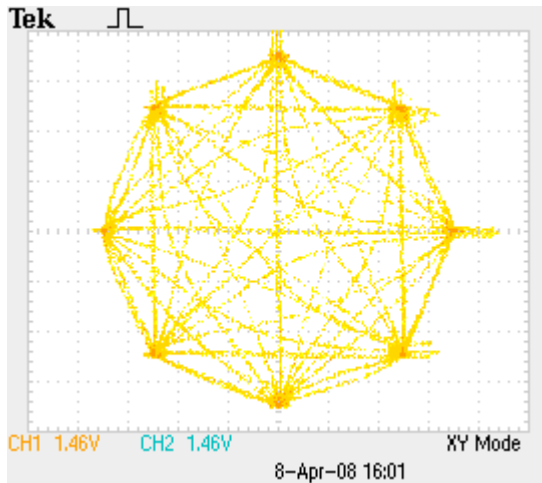
Naměřené hodnoty:
 $f = 10,00\text{kHz}$; $t_p = 25 \mu\text{s}$

Výpočet fázového posunu:
 $\alpha = t_p \cdot f \cdot 360 = 25 \cdot 10^{-6} \cdot 10 \cdot 10^3 \cdot 360^\circ = 90^\circ$

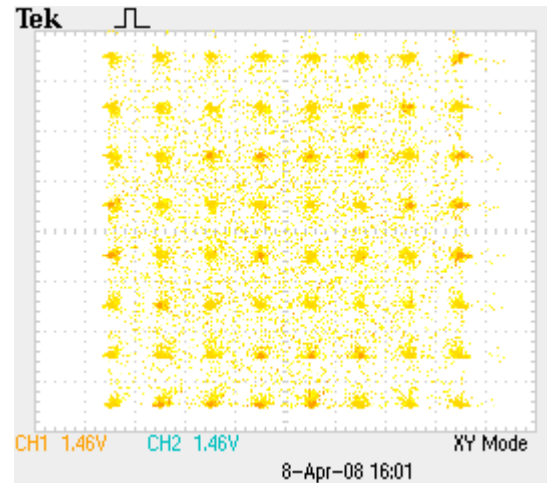
Signál označený jako „sin“ je o 90° otočen za signálem „cos“.

13.2. Měření na kodéru

V přípravku má kodér funkci převodu sériových dat na dvojici vícecestavových signálů. Na těchto signálech můžeme ověřit správně nastavené konstelační diagramy. Na obrázcích 13.2a a 13.2b jsou výstupy z osciloskopu v režimu XY se zapnutým pseudonáhodným generátorem.



Obrázek 13.2a: Změřený konstelační diagram 8PSK (režim XY)

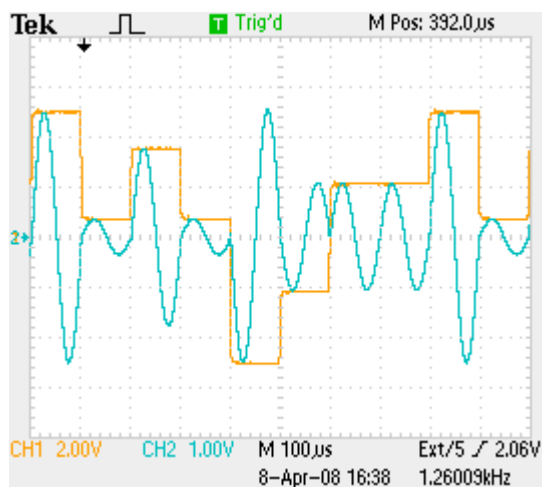


Obrázek 13.2b: Změřený konstelační diagram 64QAM (režim XY)

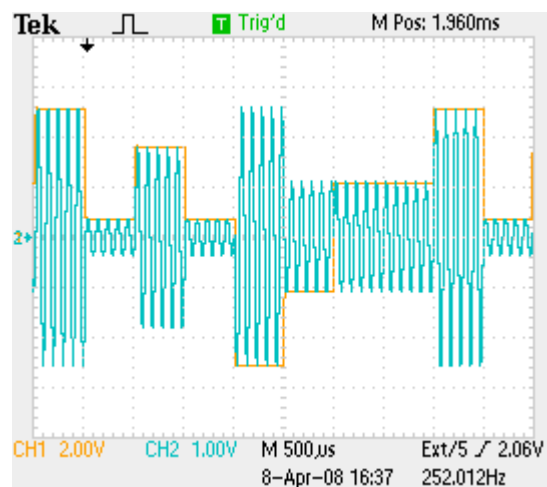
Naměřené diagramy odpovídají obrázkům 2.4c a 2.4f

13.3. Měření na modulátoru

Přípravek obsahuje jeden modulátor pro každou větev I a Q. Měření bylo provedeno pro dva různé parametry p_{er} na větvi Q a s osmi symboly. Kvadrurní větev Q je modulována harmonickým sinusovým signálem, jak je vidět na obrázcích 13.3a a 13.3b



Obrázek 13.3a: Měření na modulátoru pro 1 periodu na symbol



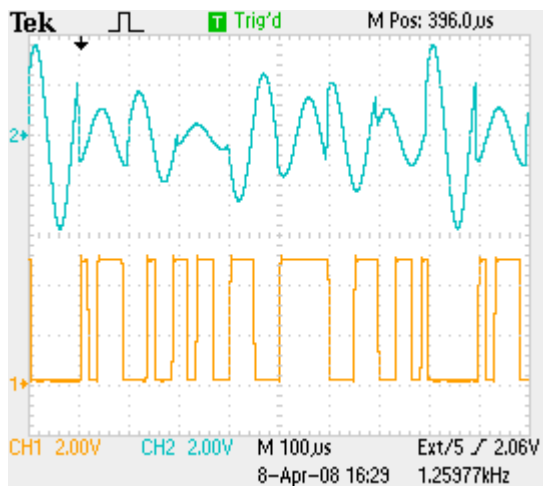
Obrázek 13.3b: Měření na modulátoru pro 5 periodu na symbol

13.4. Měření na generátoru a na výstupu z přípravku

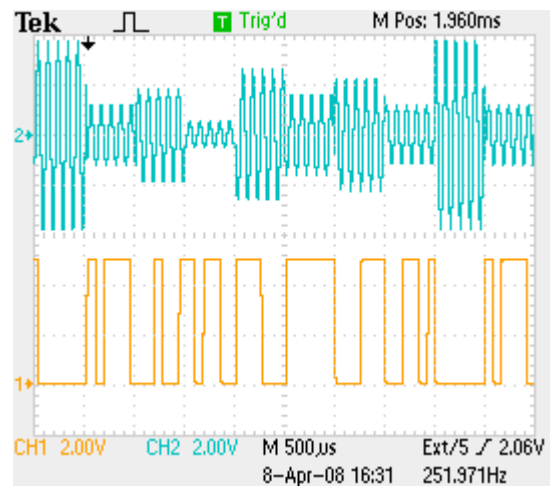
Přenosová rychlost dat, které poskytuje generátor, se mění v závislosti s použitou modulací, protože modulační rychlost zůstává stejná (při konstantní hodnotě parametru p_{er} v přípravku). Se zvyšováním počtu bitů na jeden symbol (změna modulace) roste přenosová rychlost. Výsledky měření lze předpokládat podle vztahů 13.2

$$m = \frac{f}{p} [\text{Bd}] \qquad r = \frac{f \cdot b}{p} = m \cdot b [\text{b/s}] \qquad (13.2)$$

Kde m je modulační rychlost, r je přenosová rychlost sériových dat, f je kmitočet nosného signálu, b je počet bitů jednoho symbolu a p je počet period nosného signálu v jednom symbolu. Na obrázcích 13.4 je naměřen výstupní signál společně s výstupem z generátoru.



Obrázek 13.4a: Měření na výstupu a generátoru dat pro 1 periodu na symbol



Obrázek 13.4b: Měření na výstupu a generátoru dat pro 5 period na symbol

Na obrázku 13.4a můžeme vidět změnu fáze a amplitudy pro jednotlivé symboly. Každý symbol je složen pouze z jedné periody harmonického signálu. Obrázek 13.4b ukazuje totéž, ale s pěti periodami nosného signálu v jednom symbolu. Přenosovou rychlost z měření na obrázku 13.4a vypočteme podle vztahu 13.2.

$$f = 10 \cdot 10^3; b = 6; p = 1; r = \frac{f \cdot b}{p} = \frac{10 \cdot 10^3 \cdot 6}{1} = 60 \cdot 10^3 = 60 \text{ kb/s}$$

Přenosová rychlost z měření na obrázku 13.4b:

$$f = 10 \cdot 10^3; b = 6; p = 5; r = \frac{f \cdot b}{p} = \frac{10 \cdot 10^3 \cdot 6}{5} = 12 \cdot 10^3 = 12 \text{ kb/s}$$

V zamýšlených laboratorních úlohách budou zobrazovány právě uvedené signály. Všechny změřené průběhy plně vyhovují zadaným požadavkům.

14. Závěr

Úkolem této práce bylo navrhnout celkové schéma s deskou plošných spojů a celkovou konstrukci pro laboratorní přípravek modulátoru QAM. V jednotlivých kapitolách je popsán princip vytváření modulace QAM a popis dílčích funkčních bloků samotného přípravku. K pohodlnému ovládání je přípravek vybaven tlačítky a čtyřřádkovým LCD.

Všechna schémata byla samostatně i společně odzkoušena na kontaktním poli a společně s krátkým programem jsou odladěna. Některé nedostatky byly odhaleny hned na kontaktním poli, a některé byly odstraněny při testování prototypu.

Schéma obsahuje 110 součástek a vytvořená dvouvrstvá deska plošných spojů je navržena pro umístění do plastové krabice 45x190x138mm.

Pro sestavený prototyp byl vytvořen a odladěn obslužný program, který je součástí elektronické přílohy.

Měření a testování parametrů modulátoru, které je uvedeno v závěru této práce, může posloužit jako inspirace pro případné budoucí laboratorní úlohy s tímto přípravkem. Naměřené oscilogramy (viz kap. 13) ukazují, že přípravek vyhovuje požadavkům pro názorné zobrazení signálů na laboratorním přípravku.

15. Použitá literatura

- [1] ANALOG DEVICES: *Low Cost Analog Multiplier – AD633* [katalogový list]. 2002 REV E
- [2] ATMEL: *8-Bit AVR Microcontroller with 16K Bytes In-System Programmable Flash – ATmega16* [katalogový list]. 2005
- [3] ITU-T Recommendation G.992.1: *Asymmetric digital subscriber line (ADSL) transceivers*. 1999-06
- [4] NATIONAL SEMICONDUCTOR: *8-Bit μ P Compatible, Double-Buffered D to A Converters – DAC0832* [katalogový list]. 1999
- [5] RFMD: *Quadrature Modulator/Demodulator – RF2713* [katalogový list]. 2005 Rev A4
- [6] SGS-THOMSON MICROELECTRONICS: *Voltage Comparators – LM311* [katalogový list]. 1996
- [7] SKALISKÝ, P.: *Přístrojové aplikace mikropočítačů* [Skriptum ČVUT]. Praha: 2004 ISBN 80-01-03111-X
- [8] ŠEBESTA, V.: *Teorie sdělování* [Skriptum VUT v Brně], Brno: 2001. ISBN 80-214-1843-5
- [9] ŠEBESTA, V., SMÉKAL Z.: *Signály a soustavy* [Skriptum FEKT VUT], Brno 2003 ISBN 80-214-2434-6
- [10] TEXAS INSTRUMENTS: *Low-power JFET-input operational amplifiers – TL062* [katalogový list]. 1976 Rev SEPTEMBER 2004
- [11] WIKIPEDIA: *Quadrature amplitude modulation* [online] [cit. 2008-04-15].
URL: <http://en.wikipedia.org/wiki/Quadrature_amplitude_modulation>

Seznam příloh

PŘÍLOHA 1: CELKOVÉ SCHÉMA	67
PŘÍLOHA 2: DESKY PLOŠNÝCH SPOJŮ	68
PŘÍLOHA 3: SEZNAM POUŽITÝCH SOUČÁSTEK	69
PŘÍLOHA 4: ŠABLONA PRO VYTVOŘENÍ OTVORŮ	71
PŘÍLOHA 5: POTISK PŘÍPRAVKU	72

Schéma hlavní desky

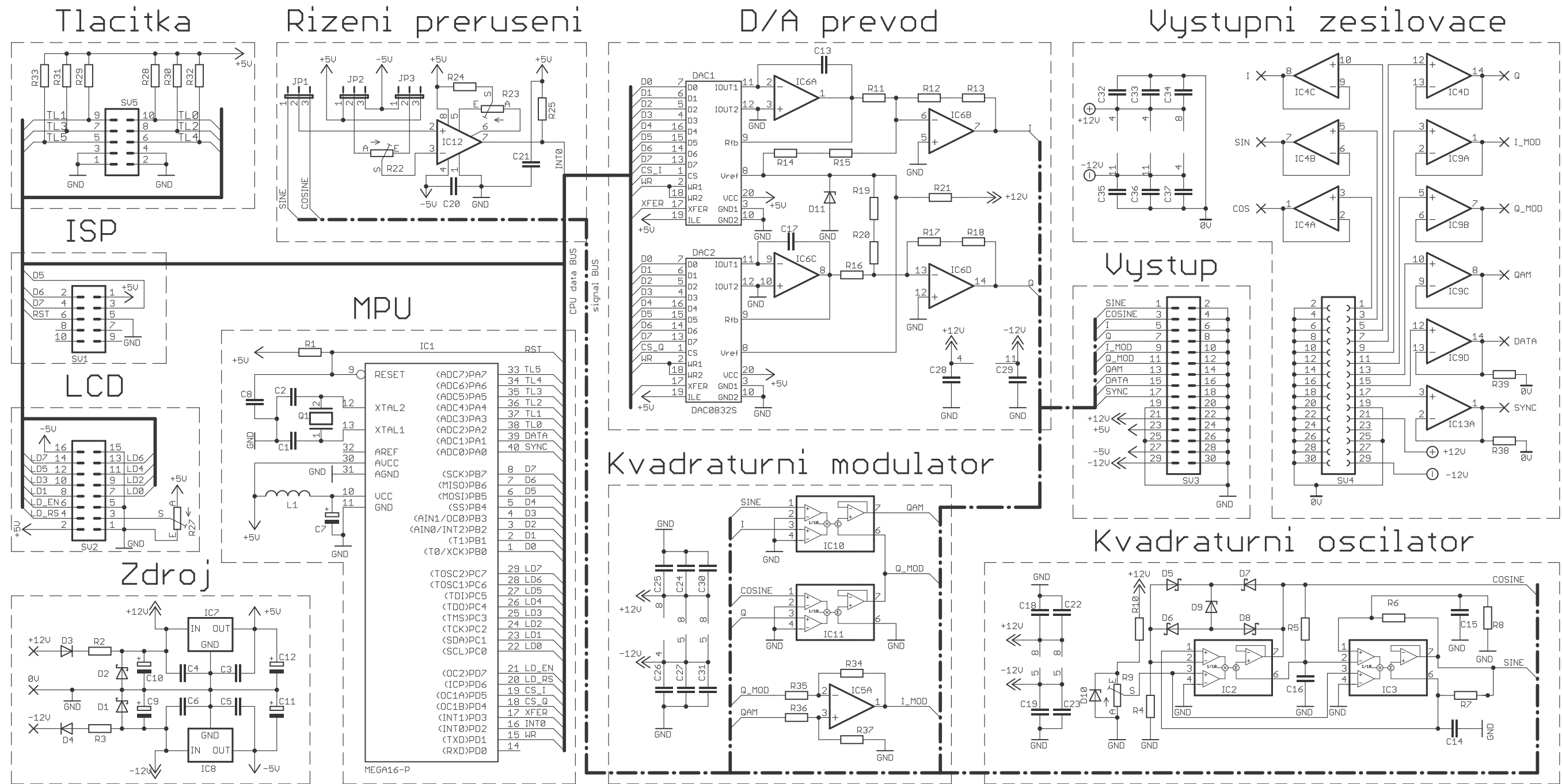
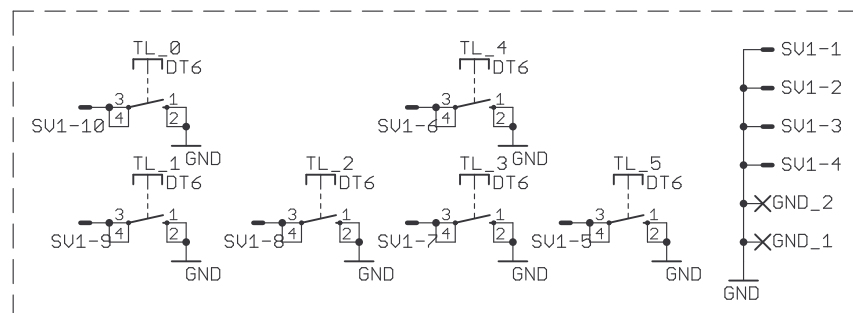
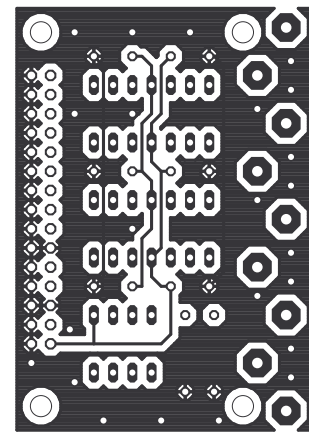
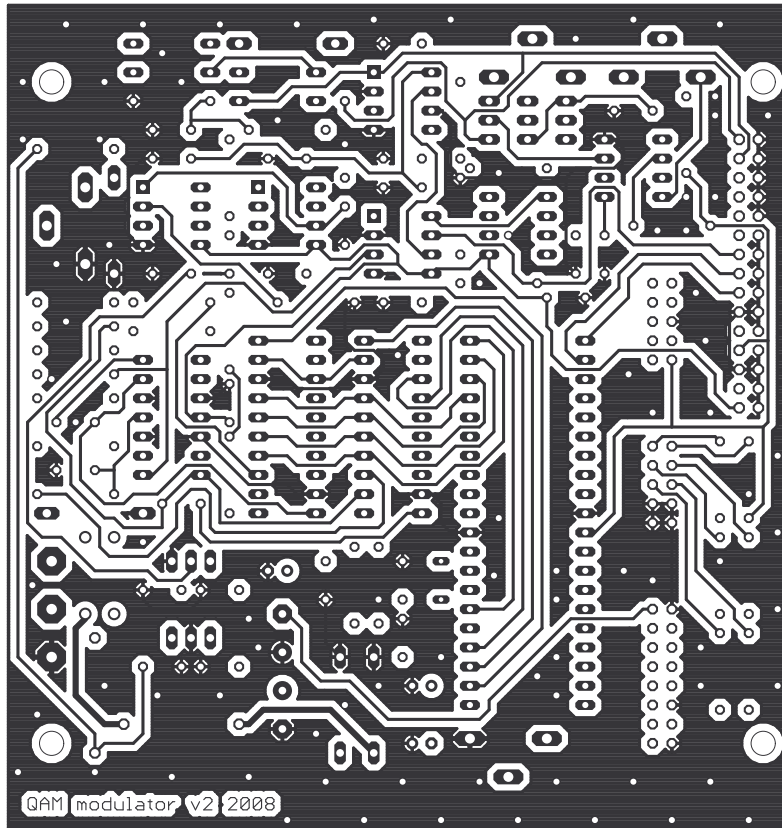
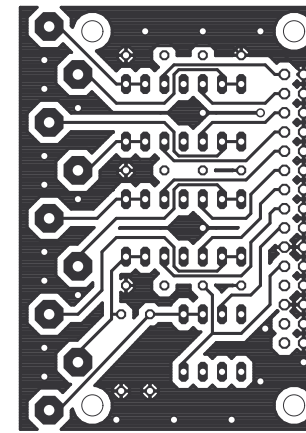


Schéma desky s tlačítky

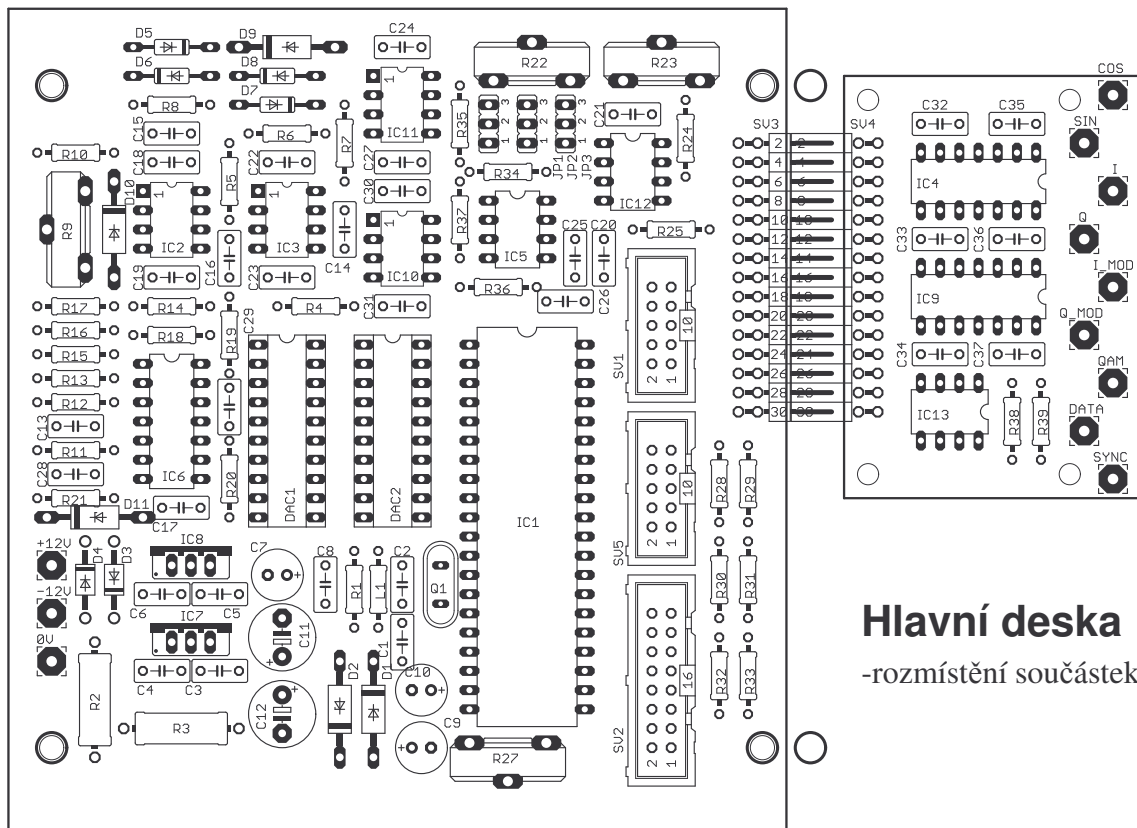
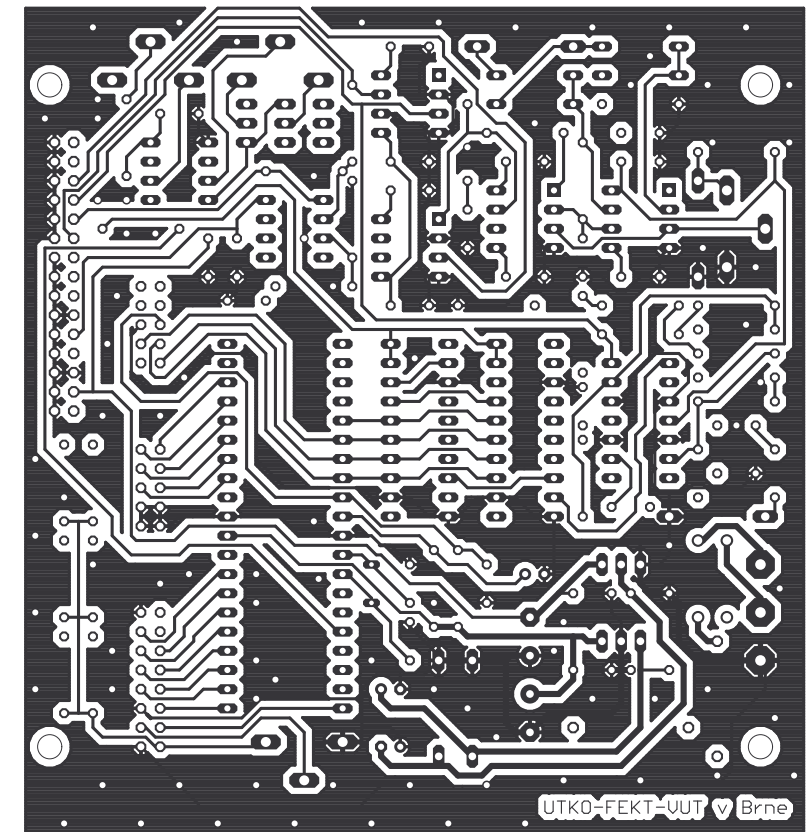




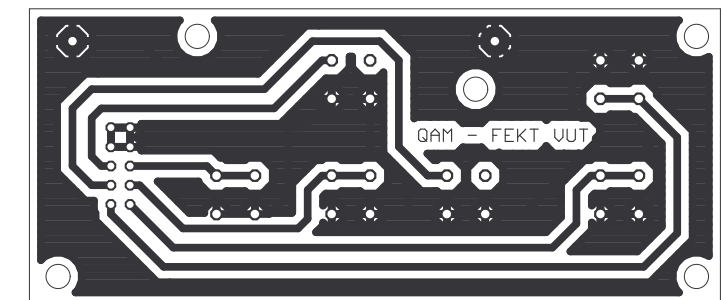
Hlavní deska
-vrchní vrstva



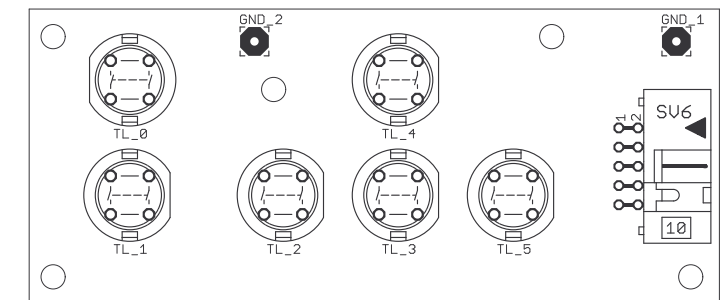
Hlavní deska
-spodní vrstva
(pohled zespodu)



Hlavní deska
-rozmístění součástek



Deska tlačítek
-spodní vrstva
(pohled zespodu)



Deska tlačítek
-rozmístění součástek

Počet	Název	Číslo GME	Popis	Označení ve schématu
8x	RR1K	110-073	rezistor 0,6W; 1k Ω	R1; R4; R5; R7; R21; R25; R38; R39
1x	RR330K	110-133	rezistor 0,6W; 330k Ω	R6
1x	RR16K	110-102	rezistor 0,6W; 16k Ω	R8
1x	RR1K6	110-078	rezistor 0,6W; 1,6k Ω	R10
14x	RR100K	110-121	rezistor 0,6W; 100k Ω	R11 - R20; R34 - R37
1x	RR3K	110-084	rezistor 0,6W; 3k Ω	R24
6x	RR10K	110-097	rezistor 0,6W; 10k Ω	R28-R33
2x	RR W2 E010	114-040	rezistor 2W; 10 Ω	R2; R3
1x	PT15HK100	112-066	trimr PIHER 15mm; 100k Ω	R9
2x	PT15HK005	112-064	trimr PIHER 15mm; 5k Ω	R22; R23
1x	PT15HK010	112-065	trimr PIHER 15mm; 10k Ω	R27
2x	CKS 33P/50V	120-133	keramický kondenzátor 33pF; 50V	C1; C2
25x	CK 100N/63V	120-060	keramický kondenzátor 100nF; 63V	C3 - C6; C8; C18 - C37
2x	CKS 82P/50V	120-151	keramický kondenzátor 82pF; 50V	C13 - C17
2x	CKS 10N/50V	120-108	keramický kondenzátor 10nF; 50V	C14; C16
1x	CKS 150pF/50V	120-112	keramický kondenzátor 150pF; 50V	C15
3x	E47M/25VT	123-096	elektrol. kondenzátor 47 μ F; 25V	C7; C9; C10
2x	E470M/6.3VT	123-151	elektrol. kondenzátor 470 μ F; 6.3V	C11; C12
1x	TL. 33 μ H	611-007	tlumivka 500mA; 0,92 Ω	L1
1x	QM 16.000MHZ	131-075	krystal 16MHz, miniaturní	Q1
2x	BZW06-13	222-207	transil 12,8V	D1; D2
2x	1N4007	220-002	usměrňovací dioda 1000V; 1A	D3; D4
4x	SD101B	223-025	schottky dioda; 50V; 15mA	D5 - D8
1x	BZX83V008.2	222-023	zenerova dioda 8,2V; 0,5W	D10
2x	BZX83V005.1	222-018	zenerova dioda 5,1V; 0,5W	D9; D11
1x	78S05	330-018	stabilizátor 5V; 2A	IC7
1x	7905	330-023	stabilizátor -5V; 1,5A	IC8
1x	ATmega16-16PI	432-175	MCU AVR 512 byte EEPROM	IC1
4x	ADD633		GME v nabídce neposkytuje	IC2; IC3; IC10; IC11
3x	TL064	311-011	4x OZ J-FET	IC4; IC6; IC9
2x	TL062	311-010	2x OZ J-FET	IC5; IC13
1x	LM311	314-004	napěťový komparátor	IC12
2x	DAC0832	321-014	8bit. DAC uP kopat. DIP20 0,2%	DAC1; DAC2
1x	MC1604B-SYL	513-073	LCD displej 16 x 4; STN; LED	-
1x	S1G20	832-017	lámací lišta jednořadá 20 pinů	JP1; JP2; JP3
1x	S2G34W	832-028	lámací lišta dvouřadá 34 pinů 90°	SV3
1x	BLW230G	832-077	dutinková lišta 2 x 15 pinů 90°	SV4
1x	MLW10A	800-005	vidlice 2 x 5 pinů, 90°	SV6
2x	MLW10G	800-035	vidlice 2 x 5 pinů, přímá	SV1; SV5
1x	MLW16G	800-006	vidlice 2 x 8 pinů, přímá	SV2
2x	PFL10	800-007	samořezná zásuvka 2 x 5 pinů	-
1x	PFL16	800-009	samořezná zásuvka 2 x 8 pinů	-
3x	JUMP-SW BL.	832-013	zkratovací propojka černá	součást JP1; JP2; JP3
7x	SOKL 8	823-011	obyčejná patice 8 pinů	součást IC
3x	SOKL 14	823-001	obyčejná patice 14 pinů	součást IC
2x	SOKL 20	823-004	obyčejná patice 20 pinů	součást IC
1x	SOKL 40	823-009	obyčejná patice 40 pinů	součást IC
6x	P-DT6RT	630-042	spínač tlač. Červený	TL_0 - TL_5
2x	DO2A	620-020	ALU eloxovaný 13x19x9, TO220	součást IC7 a IC8
2x	SIS-TO220	621-015	silikonová podložka TO220	součást IC7 a IC8

Počet	Název	Číslo GME	Popis	Označení ve schématu
2x	IB2	621-012	izolační průchodka 3,5mm TO220	součást IC7 a IC8
12x	FS7018-BS	834-029	2,8 x 0,8 faston, zásuvka	-
12x	LST1315	834-005	2,8 x 0,5 faston vidlice PLS	vývody z DPS
12x	FS25030PVC	834-023	PVC izolace	-
10x	SBZ WHITE	811-009	zdířka pro banánek bílá	-
2x	SBZ RED	811-008	zdířka pro banánek červená	-
6x	SBZ GREEN	811-007	zdířka pro banánek zelená	-
2x	SBZ BLUE	811-006	zdířka pro banánek modrá	-
0,5m	AWG28-16H	650-045	plochý kabel šedý, 16 žil	-

