

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**MODUL PRO ZPRACOVÁNÍ SIGNÁLU Z
INKREMENTÁLNÍHO ROTAČNÍHO SNÍMAČE**
MODUL FOR SIGNALS PROCESS FROM INCREMENTAL ROTARY ENCODER

BAKALÁŘSKÁ PRÁCE
BACHELORS THESIS

AUTOR PRÁCE
AUTHOR

MARTIN PAVLÁČEK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. TOMÁŠ MARADA, PH.D.

BRNO 2009

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

(na místo tohoto listu vložte originál a nebo kopii zadání Vaší práce)

LICENČNÍ SMLOUVA

(na místo tohoto listu vložte vyplněný a podepsaný list formuláře licenčního ujednání)

ABSTRAKT

Bakalářská práce se zabývá návrhem a realizací modulu pro zpracováním signálu z inkrementálního rotačního snímače

Realizace modulu je rozdělena na dvě části. V první je návrh hardwaru a jeho sestavení, v druhé potom naprogramování mikrokódu pro oživení modulu.

ABSTRACT

Bachelors thesis deal with design a realization module unit for signal process from Incremental rotary encoder.

Realization of module unit has two parts. First is draft and assemble hardware components, second is program microcode for module's microcomputer.

KLÍČOVÁ SLOVA

IRC, Inkrementální rotační snímač, Kvadraturní dekodér

KEYWORDS

IRC, Incremental rotary encoder, Quadrature decoder

PODĚKOVÁNÍ

Rád bych poděkoval především vedoucímu bakalářské práce Ing. Tomáši Maradovi Ph.D. za vedení práce a podnětné rady, které mi poskytl v průběhu mé práce.

Obsah:

	Zadání závěrečné práce.....	3
	Licenční smlouva.....	5
	Abstrakt.....	7
	Poděkování	9
1	Úvod.....	13
2	Inkrementální rotační snímač.....	15
2.1	Princip inkrementálního rotačního snímače.....	15
2.2	Signál inkrementálního rotačního snímače.....	16
2.3	Dekódování signálu z inkrementálního snímače.....	17
3	Hardware modulu.....	19
3.1	Volba hardware pro modul.....	19
3.2	Blokové schéma modulu.....	19
3.3	Kvadrurní dekodér.....	20
3.3.1	Rychlost dekodování HCTL-2032.....	21
3.3.2	Zapojení HCTL-2032.....	22
3.3.3	Připojení I-kanálu.....	24
3.4	Zdroj hodinového signálu pro kvadrurní dekodér.....	24
3.5	Mikrokontrolér ATmega128.....	25
3.5.1	Zapojení řídicího bloku s ATmega128.....	26
3.6	LCD displej.....	27
3.6.1	Řadič displeje HD44780.....	28
3.6.2	Zapojení a komunikace s LCD displejem.....	29
3.7	Klávesnice.....	30
3.7.1	Zapojení klávesnice.....	30
3.8	Převodník úrovní TTL-RS232.....	31
3.8.1	Zapojení převodníku úrovní TTL-RS232.....	31
3.9	Napájení modulu.....	32
3.9.1	Zapojení zdroje modulu.....	32
4	Realizace modulu.....	33
4.1	Eagle 5.4.0.....	33
4.1.1	Editor knihoven součástek.....	34
4.1.2	Návrh schématu	35
4.1.3	Návrh desky plošných spojů.....	36
4.2	Výroba desky plošných spojů.....	37
4.2.1	Technologie výroby.....	37
4.2.2	Předloha.....	38
4.2.3	Osvit.....	38
4.2.4	Vyvolání.....	38
4.2.5	Leptání.....	38
4.2.6	Osazení součástek.....	39
4.3	Oživení a finální sestavení modulu.....	40
4.3.1	Oživení modulu.....	40
4.3.2	Sestavení modulu.....	41
4.4	Programování řídicího mikrokontroléru	42
4.4.1	AVRStudio.....	42
4.4.2	WinAVR - Překladač C/C++ pro AVR.....	43
4.4.3	Programování mikrokontroléru.....	44
4.4.4	Program pro modul.....	45
5	Funkce modulu.....	51

5.1	Režimy modulu.....	51
5.1.1	Režim menu a nastavení.....	51
5.1.2	Režim čítače.....	52
5.1.3	Režim měření rychlosti otáček.....	53
5.1.4	Režim odesílání.....	54
6	Závěr.....	55
7	Seznam použité literatury.....	57
	SEZNAM PŘÍLOH:.....	59
	Přílohy:.....	61
	Příloha 1:.....	61
	Příloha 2:.....	62
	Příloha 2:.....	63
	Příloha 2:.....	64
	Příloha 2:.....	65
	Příloha 2:.....	66
	Příloha 3:.....	67
	Příloha 4:.....	68
	Ovládání:.....	68
	MENU:.....	68
	COUNTER:.....	69
	TACHOMETR:.....	70
	RS232 SENDING LOOP:.....	70
	NASTAVENÍ ČÍSELNÝCH HODNOT:.....	70

1 ÚVOD

Pro řízení pohonu mechanických posunů či natočení, ať už v automatizaci, robotice nebo jiných odvětvích, je třeba znát přesnou polohu nebo rychlost pohonu a to v daných okamžicích. S výjimkou krokových motorů, kde je pootočení dané ovládním motoru, je třeba údaj o poloze motoru zjistit zpětně. Jednou z možností je použít rotační snímač.

Rotační snímače jsou ve své podstatě elektromechanické převodníky, které převádí rotační pohyb na elektrický signál. Tento signál je třeba zpracovávat za běhu pohonu a vyhodnocovat tak údaj o poloze, rychlosti či natočení. Tyto snímače se také dělí na takzvané absolutní rotační snímače (ARC) a inkrementální rotační snímače (IRC). Snímače ARC udávají údaj o poloze pohonu absolutně a od daného počátku. Z tohoto snímače jsme tedy schopni odečíst natočení i když se pohon nepohnul. Snímače IRC udávají pouze relativní přírůstky pootočení od počátku pohybu pohonu.

Při návrhu modulu pro zpracování signálu z inkrementálního snímače je třeba uvážit rychlost snímaného pohybu na jehož základě je signál z čidla měněn a navrhnout vhodný způsob zpracování. Dále také volit hardware k tomuto dostatečně rychlý.

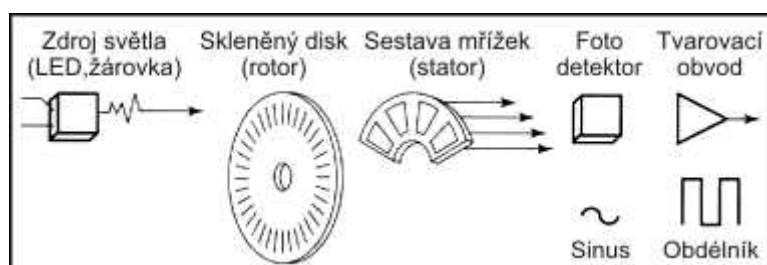
Navrhovaný modul má za úkol zpracovávat signál z inkrementálního snímače a vyhodnocovat relativní natočení snímače popřípadě rychlost otáčení. Modul je také třeba vybavit komunikačním rozhraním, aby zpracované údaje dokázal předat jinému systému.

2 INKREMENTÁLNÍ ROTAČNÍ SNÍMAČ

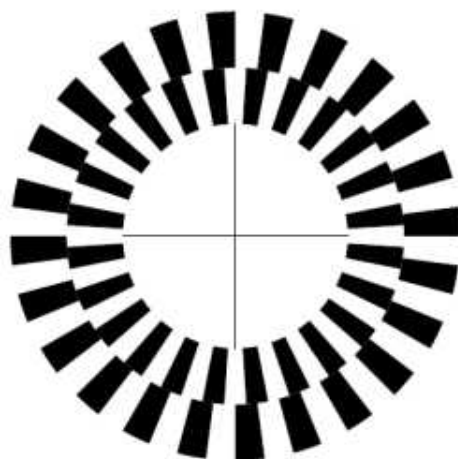
Pro návrh řešení, jak zpracovat signál z inkrementálního snímače je třeba pochopit princip snímače a popsat jeho signál. Dále pak navrhnout jak signál zpracovávat pro další vyhodnocování

2.1 Princip inkrementálního rotačního snímače

Inkrementální snímače pracují na principu optické závory. Rotační pohyb je přenášěn na tenký kotouček (ilustrace na obr. 2), na kterém jsou po obvodu rozmístěny dvě řady okének navzájem posunutých, které propouštějí světlo. Tento kotouček je prosvěcován zdrojem světla, většinou infračervenou LED diodou, dopadajícím na optický senzor. Pohyb kotoučku tedy přerušuje proud světla dopadající na senzor, který pak vytváří elektrický harmonický signál a ten je ještě za pomoci elektroniky zesilován a tvarován. Grafická ilustrace na obr. 1 naznačuje princip funkce snímače.[12]



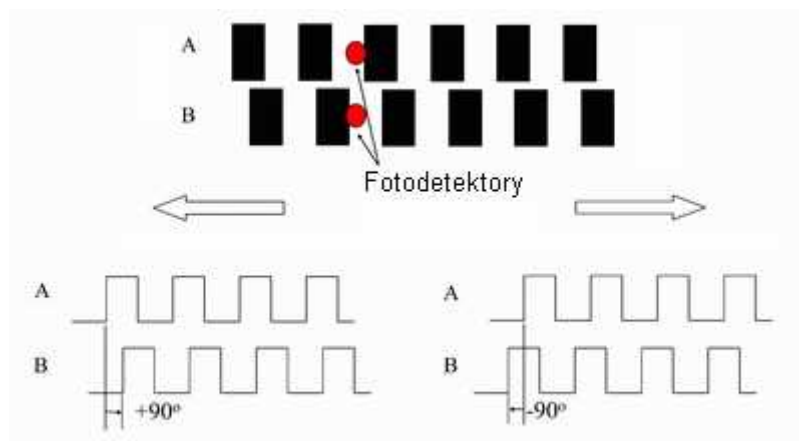
obr. 1 Princip inkrementálního rotačního snímače



obr. 2 Ilustrace optického kotoučku inkrementální snímače

2.2 Signál inkrementálního rotačního snímače

Inkrementální snímače generují dva nebo tři elektrické signály (v závislosti na typu). Základem jsou dva signály nazývané též kanály. Je to kanál A a kanál B. Jak bylo popsáno v principu snímače jedná se o harmonické elektronické impulsy které mají vůči sobě posunutou fázi o 90° . Existence dvou signálů a jejich posuv slouží k určení směru otáčení. Pokud se motor otáčí jedním směrem je kanál B za kanálem A zpožděn o 90° a naopak. Toto zakódování signálu se nazývá Kvadrurní kódování. Jeho ilustrace je na obr. 3



obr. 3 Ilustrace průběhu signálu na kanálech A a B inkrementálního snímače

U některých druhů inkrementálních snímačů je ještě vytvářen jeden signál nazývaný jako kanál I. Signál z tohoto kanálu pak dává pulz při každém celém otočení motoru a je fázově nezávislý na ostatních kanálech A a B. [13]

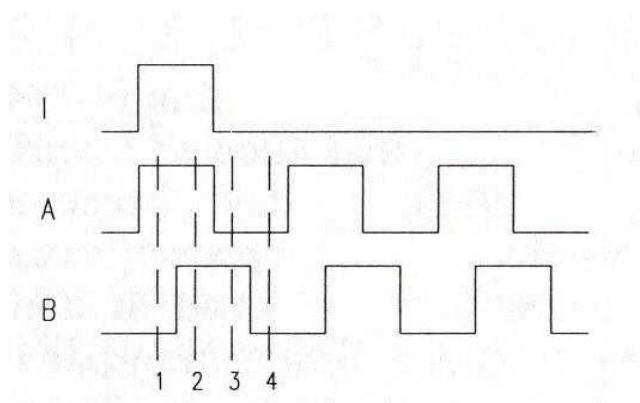
Elektrické výstupy snímačů jsou typu „otevřený kolektor“ a je třeba je vybavit takzvanými „pull-up“ rezistory. Některé snímače již tyto rezistory mohou mít implementovány.

2.3 Dekódování signálu z inkrementálního snímače

Elektrický signál z kanálů A a B (I) sám o poloze nebo rychlosti motoru nic neříká a je třeba jej tedy dekodovat.

Jak vyplývá z principu snímače, každá náběžná hrana signálu udává posunutí o jedno průsvitné okénko vpřed. Hrana sestupná pak posuv na neprůsvitných okénkách. Čítáme-li tedy jednu z těchto hran, získáváme inkrementy polohy motoru.

Počet těchto hran v signálu na jednu otáčku je tedy dán počtem okének v kotoučku. Vezmeme-li v úvahu oba kanály A i B a čítáme jejich náběžné i sestupné hrany dostáváme tedy při každém pootočení o jedno okénko kotoučku 4 načítané hrany. To díky fázovému posunutí kanálů A a B.



obr. 4 Dekódování kanálů A a B

Kvadrurní signál z kanálů A a B je tedy možné převést na počet načítaných impulzů při použití přesnosti 1x, 2x nebo 4x. Počet těchto impulzů nám říká relativní posunutí motoru a zjištění fázového posuvu mezi kanály, směr otáčení. K tomuto účelu se používají takzvané kvadrurní dekodéry.[13]

Při dekodování tohoto signálu hraje klíčovou roli rychlost otáčení motoru. Při vysokých otáčkách motoru mohou dosahovat signály z čidla vysokých frekvencí. Závisí to na přesnosti snímače.

Příklad:

Otáčí-li se motor rychlostí 10000ot./min, použitý senzor má 1000 dílků na otáčku, pak je frekvence signálu na jednom z kanálů přibližně 167KHz.

Výpočet:

$$\frac{10000}{60} \cdot 1000 = 167 [Khz] \quad \frac{[ot. \cdot min^{-1}]}{[s]} \cdot [impuls]$$

Pokud bychom chtěli takovýto signál spolehlivě dekodovat, a to v režimu s přesností 4x, frekvence čítaných pulzů by byla přibližně 670KHz.

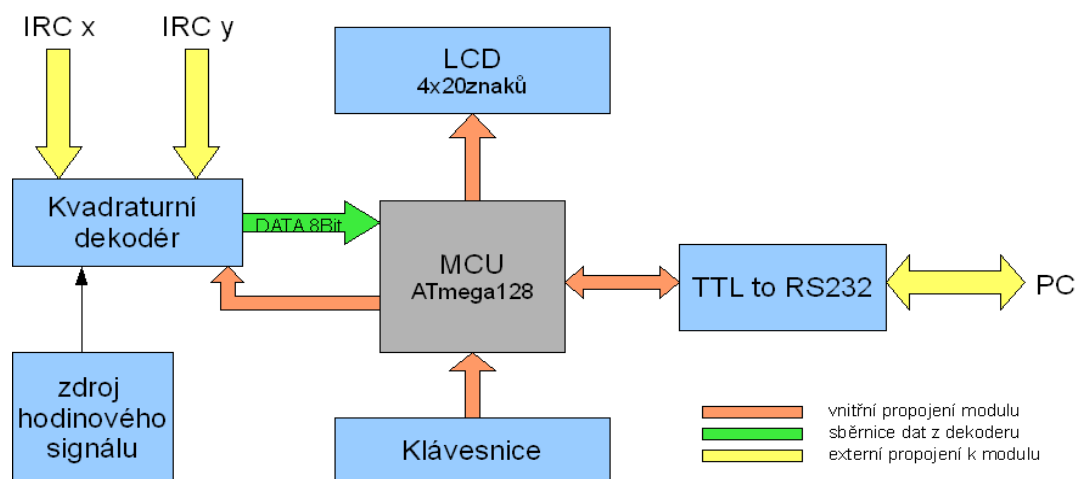
Při aplikaci v řízení, kdy je třeba údaj o poloze získávat v krátkých intervalech řádu milisekund a je třeba kvadrurní signál zároveň dekodovávat a zároveň provádět vyhodnocování pro akční zásah, může být časování kritické.

3 HARDWARE MODULU

3.1 Volba hardware pro modul

Při volbě hardwaru pro modul který má za úkol dekódování signálu z inkrementálního snímače je třeba přihlídnout právě k rychlosti respektive přesnosti zpracování signálu, protože modul má signál nejen dekódovat, ale i zpracovávat a poskytovat zpracované údaje k dalšímu použití. Z těchto důvodů je tedy vhodné dekódování hardwarově oddělit od dalšího zpracování. Zpracované hodnoty je vhodné také zobrazit nebo odeslat z modulu pro další zpracování nadřazeným systémem a v poslední řadě také modul ovládat.

3.2 Blokové schéma modulu

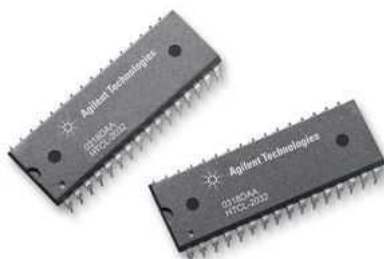


obr. 5 Blokové schéma zapojení modulu

Jednotlivé bloky modulu ukazuje obr. 5 . Bloky reprezentují obvodové části modulu, které sestávají z více součástek. Barevné šipky ilustrují propojení jednotlivých bloků. Vnitřní propoje a 8bitová sběrnice jsou ve skutečnosti spoje na desce modulu, externí propojení jsou pak zakončeny konektory jejíž prostřednictvím se připojují externí zařízení, jako například počítač nebo přímo inkrementální senzory. Blíže jsou bloky popsány v dalším textu.

3.3 Kvadrurní dekodér

Pro dekódování signálu z inkrementálních snímačů existují speciální integrované obvody. Pro účel tohoto modulu byl zvolen integrovaný obvod HCTL-2032 od výrobce Agilent Technologies (obr. 6).



obr. 6 Integrovaný obvod HCTL-2032

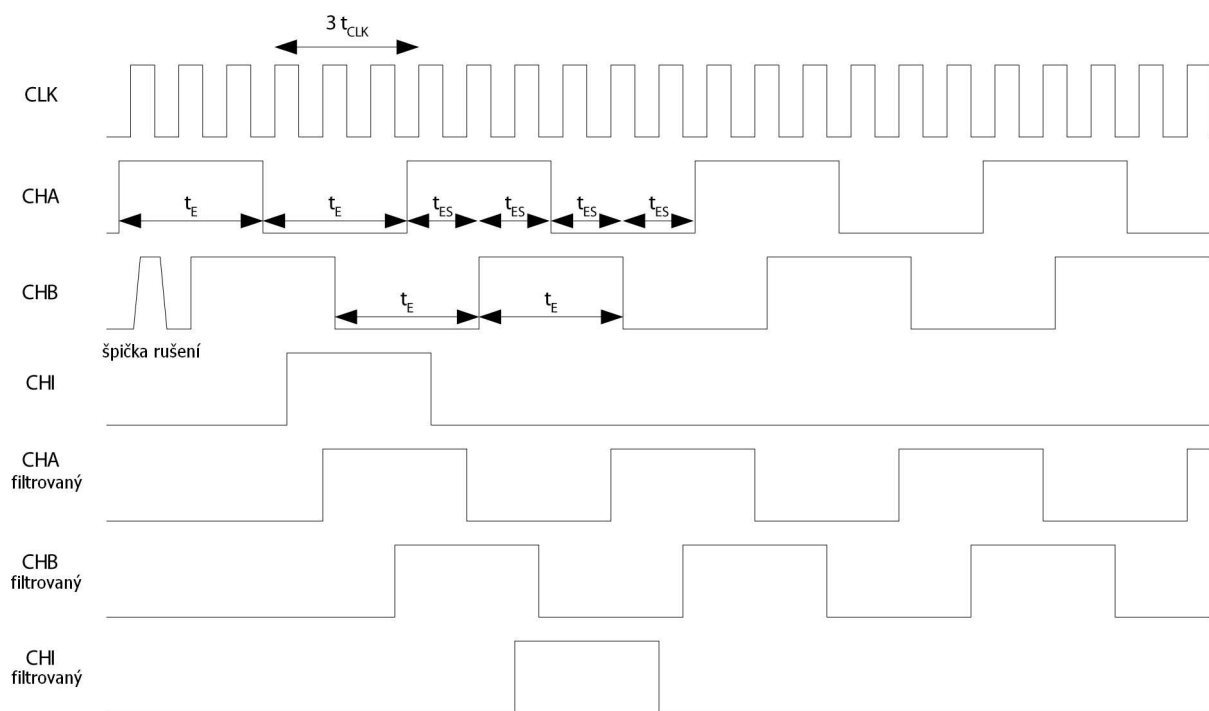
Jedná se o dvouosý kvadrurní dekodér, který dokáže dekódovat signály paralelně ze dvou snímačů. Tento obvod je navržen jako interface k mikrokontroléru, se kterým se propojuje 8bitovou sběrnicí a několika dalšími řídicími signály k ovládní sběrnice, volby přesnosti dekódování a výběru osy.

Pro každý snímač je vybaven 32bitovým čítacím registrem. Tyto registry jsou zdvojeny a odděleny odpojovací logikou, která pak zajistí naplnění dat do výstupních registrů a tím podržení data na sběrnicí pro vyčtení nadřazeným obvodem aniž by docházelo ke ztrátám čítaných údajů. Tímto jsou sníženy nároky na rychlost nadřazeného obvodu. Obvod dále obsahuje filtry proti překmitům a rušení v signálech ze snímačů. Obvody z řady HCTL-2032 je také možné propojovat do kaskády s dalšími čítači nebo registry pro rozšíření kapacity. Činnost obvodu je synchronizována externím zdrojem hodinového signálu a jeho pracovní frekvence může být až 32MHz. [1]

3.3.1 Rychlost dekódování HCTL-2032

Maximální rychlost zpracování signálů ze snímačů závisí na pracovní frekvenci obvodu, tedy na kmitočtu přivedeného ze zdroje hodinového signálu.

Maximální zpracovatelná frekvence signálu ze snímače je pak omezena filtrem překmitů zabudovaného v HCTL-2032.



obr. 7 Časování kanálů A, B, I při průchodu filtrem HCTL-2032

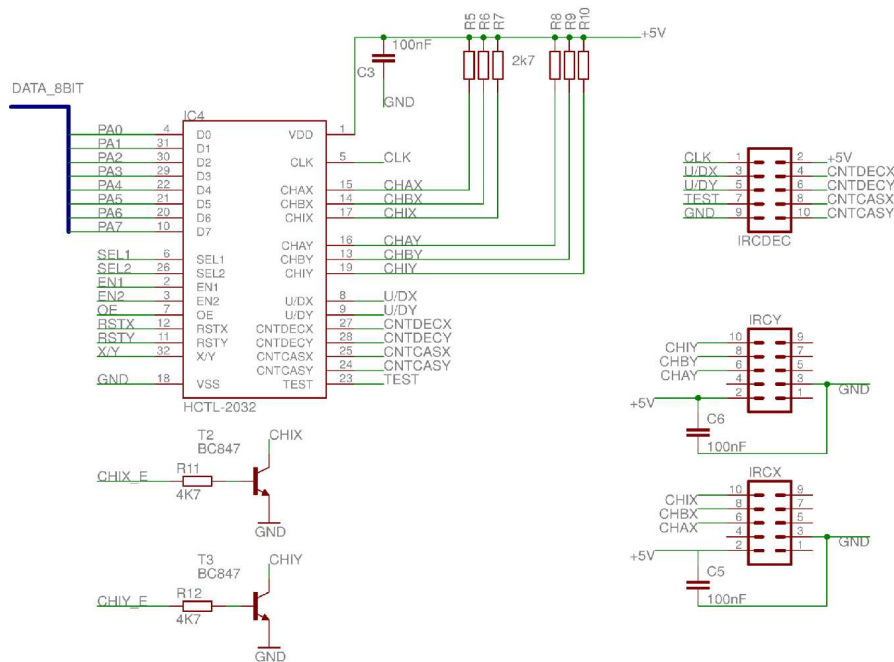
Z obr. 7 vyplývá, že pro správnou funkci filtrování, je třeba aby doba t_E byla delší než 3 periody hodinového signálu.

Pro výpočet maximální zpracovatelné frekvence, považujeme-li signál ze snímače za harmonický, je pak doba t_E půl periodou frekvence signálu z čidla. S maximální pracovní frekvencí 32MHz je pak možno zpracovávat signály větší než 21,4MHz.

výpočet maximální zpracovatelné frekvence:

$$\frac{32 \cdot 10^6}{3} \cdot 2 = 21,4 \cdot 10^6 \text{ Hz}$$

3.3.2 Zapojení HCTL-2032



obr. 8 Zapojení HCTL-2032 v modulu

Zapojení kvadraturního dekodéru HCTL-2032 je na obr. 8 . Toto zapojení je doporučené podle katalogového listu výrobce.

Výstupy z inkrementálních snímačů jsou připojeny přes pull-up rezistory R5-R10. Tyto rezistory zajišťují správné logické úrovně přiváděného signálu. Pokud není signál přiveden nebo je připojen na napětí 5V, je čtena logická 1. Pro čtení logické 0 je nutno signál připojit k zemi.

Režim dekódování se nastavuje kombinací logických signálů přiváděných na vstupy obvodu označené EN1 a EN2. Kombinace ukazuje tabulka na obr. 9 . [1].

		Režim čítání		
EN1	EN2	4x	2x	1x
0	0	Zakázaný stav		
1	0	On		
0	1		On	
1	1			On

obr. 9 Kombinace logických úrovní pro zvolení přesnosti dekódování

Vstupy SEL1,SEL2,OE, X/Y slouží pro ovládání zpoždovací logiky obvodu, která zajišťuje plnění dat na sběrnici DATA_8BIT. Protože obvod obsahuje 32bitové registry pro čítání dekódovaných pulzů je tedy třeba data vyčíst ve čtyřech krocích po 8bitech. Nastavením vstupu OE (output enable) na logickou úroveň 0 se aktivuje zpoždovací logika obvodu a v závislosti na kombinaci logických úrovní na vstupech SEL1 a SEL2 se na sběrnici nastaví požadovaný bajt ze čtyř. Sekvenci čtení bajtů je nutno dodržet podle tabulky na obr. 10 . Logická 0 přivedená na vstup X/Y nastavuje na sběrnici data pro osu x, logická 1 pak pro osu y. Je nutno osu zvolit ještě před zahájením sekvence zpoždovací logiky.

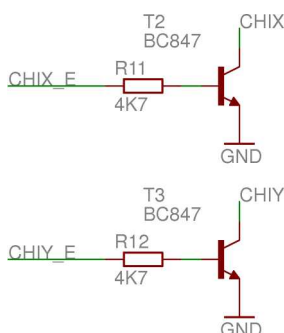
		VYBRANÝ BAJT			
SEL1	SEL2	HORNÍ	2.	3.	SPODNÍ
0	1	D4			
1	1		D3		
0	0			D2	
1	0				D1

obr. 10 Sekvence postupného čtení 32bitového registru

Výstupy U/DX, U/DY, CNTCASX, CNTCASY, CNTDECX, CNTDECY, jsou v obvodu připojeny k dekódovací logice a k vnitřním registrům. Mohou složit k připojení dalších čítačů nebo registrů. Signály CNTDECX a CNTDECY dávají dekódované pulzy v daném režimu pro každou osu a signály U/DX a U/DY směr otáčení na každý dekódovaný pulz. Logická 1 znamená doprava a logická 0 doleva. Na signálech CNTCASX a CNTCASY nastává pulz při přetečení nebo podtečení čítacích registrů dekodéru.

CNTCASX a CNTCASY jsou připojeny k řídicímu mikrokontroléru, aby mohlo být zjištěno přetečení nebo podtečení registrů. Ostatní jsou pak v modulu vyvedeny pouze na externí konektor.

3.3.3 Připojení I-kanálu



obr. 11 Rozpínací obvod I-kanálu inkrementálního snímače

Obvod HCTL-2032 je sice navržen pro čidla které jsou tímto kanálem vybaveny, je tím ale „omezena“ jeho schopnost čítání. Respektive tento signál je v obvodu pevně připojen k signálům RTSX a RSTY, čímž dojde při každé celé obrátce motoru k vynulování čítacích registrů. Takle vlastnost může být v některých případech nechtěná a je tedy vhodné signál z čidla nepoužít a využít tak čítání polohy přes více obrátek.

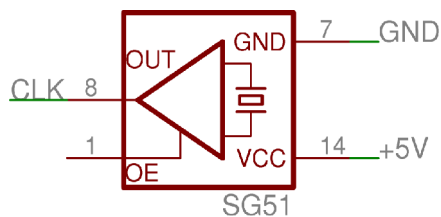
Z těchto důvodů je modul vybaven tranzistory (obr. 11), které při jejich sepnutí signál z I-kanálu čidla trvale srazí na logickou úroveň 0, tedy v podstatě signál odpojí. Báze transistorů je připojena k řídicímu MCU, který tak může chování kanálu nastavit.

Ne každý inkrementální snímač také disponuje I-kanálem. Pokud by byl připojen tento snímač přímo k dekodéru, byla by na vstupu pro I-kanál trvalá logická úroveň 1 zapříčiněná pull-up rezistorem, což by vedlo k tomu, že čítače obvodu HCTL-2032 by byly ve stavu reset a tudíž by nečítaly žádné pulzy přicházející z dekodovací logiky obvodu.

3.4 Zdroj hodinového signálu pro kvadrurní dekodér.

Zdroj hodinového signálu pro kvadrurní dekodér je tvořen krystalovým oscilátorem z řady SG51 s frekvencí 20MHz. Tento obvod je vyráběn v pouzdře DIL14 a do modulu je osazen v patici, a je tedy možné jej vyměnit za podobný s jinou frekvencí.

Jeho zapojení je vidět na obr. 12.



obr. 12 Zapojení krystalového oscilátoru SG51

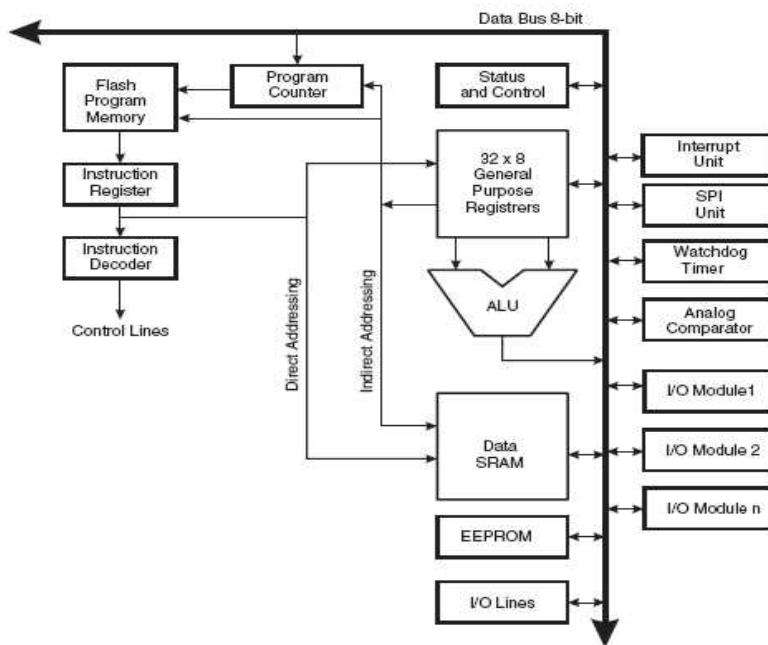
3.5 Mikrokontrolér ATmega128

Srdcem modulu je jednočipový 8bitový mikrokontrolér Atmega128 od výrobce Atmel. Tento mikrokontrolér obsahuje šest 8bitových vstupně výstupních paralelních portů a jeden 6bitový. K těmto portům jsou pak v modulu připojeny jednotlivé bloky modulu, které jsou mikrokontrolérem řízeny.

Jednočipový mikrokontrolér ATmega128 je 8bitový mikroprocesor postavený na platformě RISC. Je založený na harvardské koncepci, kde paměť dat je oddělena od paměti programu. Má v sobě integrovány jednoduché, ale rychle proveditelné instrukce, které dokážou být provedeny v jednom hodinovém cyklu. Jeho výkon dosahuje 16MIPS na 16MHz, což je jeho maximální pracovní frekvence.

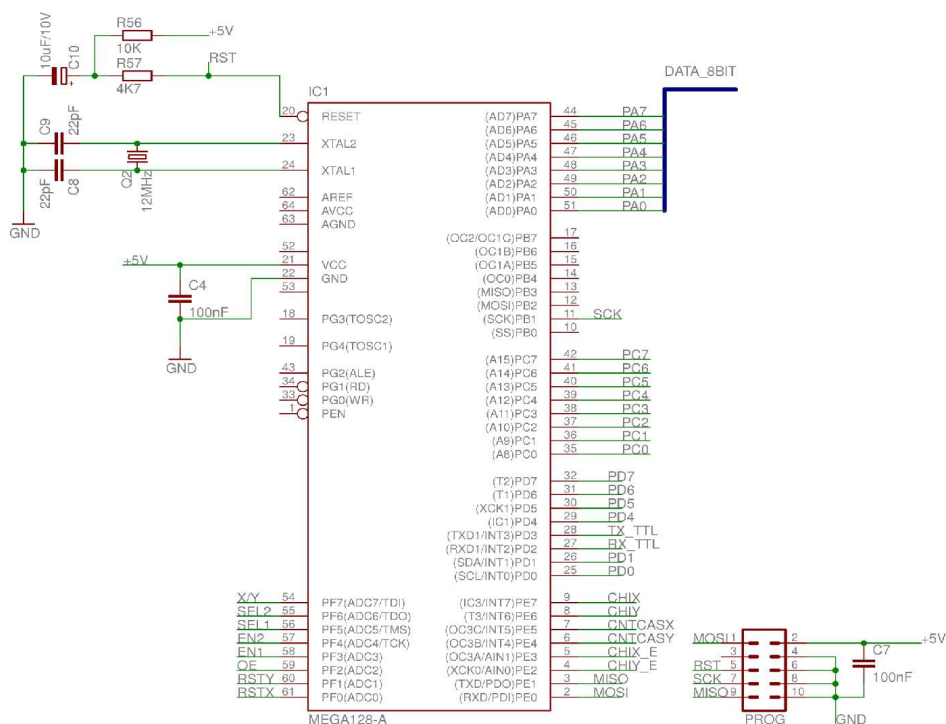
Paměť programu je realizována pamětí typu FLASH a má kapacitu 128kbajtů. Tyto paměti jsou blokově programovatelné a ukládá se do nich právě uživatelem vytvořený program, který je v paměti uchován i po odpojení napájení. Paměť dat je rychlá pracovní operační paměť mikroprocesoru typu SRAM o velikosti 4kbajty. Tato paměť je přístupná pouze z programu procesoru. Kromě paměti programu a dat obsahuje rychlé registry pro přesuny dat v procesoru a také ALU jednotku pro provádění aritmetickologických operací. Pro uchování dat vytvořených za běhu programu slouží 4kylobajty paměti typu EEPROM. Jeho vstupně výstupní porty jsou také vybaveny alternativními vlastnostmi, které jsou softwarově ovladatelné. Jedná se například o AD převodníky, komparátory, generátory PWM, obvody pro komunikaci po sběrnici I²C nebo řadič sériového kanálu USART. Obsahuje také 4 časovače/čítače, pro oměřování intervalů, nebo čítání událostí na portech.

Činnost mikrokontroléru je synchronizována pomocí hodinového signálu. Tento signál je vytvářen obvodem integrovanými přímo v čipu a jeho frekvence je řízena připojeným krystalem[2]. Zjednodušená Architektura mikrokontroléru z řady ATmega je na obrázku obr. 13 .



obr. 13 Zjednodušená architektura mikroprocesorů ATmega

3.5.1 Zapojení řídicího bloku s ATmega128



obr. 14 Zapojení mikrokontroléru ATmega128 v modulu

Řídicí blok s MCU ATmega128 je zapojen dle obrázku obr. 14.

V zapojení je použit pro řízení pracovního kmitočtu MCU krystal 12MHz. Tato hodnota byla zvolena pro jednodušší a přesnější vypočítávání časových intervalů při programování mikrokontroléru. V modulu je krystal osazen do patice a je tedy možno jej případně vyměnit za jiný. Krystal je připojen k mikrokontroléru přes doporučené keramické kondenzátory 22pF dle katalogového zapojení ATmega128.

RC článek tvořený kapacitou C10 10uF a rezistory R56, R57 zajišťují zpoždění a hladký náběh signálu RST, který slouží pro reset mikroprocesoru. V modulu je využit pouze pro programování.

Jelikož je mikroprocesor v modulu osazen jako SMD (surface mounted device) součástka, je nutné jej programovat přímo v zapojení. Příslušné programovací vodiče jsou tedy připojeny k externímu konektoru na desce spojů modulu, pro připojení programátoru. Jsou to signály MOSI, MISO, RST a SCK.

Port PA procesoru je připojen k datové sběrnici kvadrurního dekodéru DATA_8BIT.

Port PF řídí kvadrurní dekodér.

Port PC obsluhuje LCD displej.

K portu PD je částečně připojena tlačítková klávesnice, jako ovládání modulu a zároveň je využita alternativní funkce tohoto portu, která připojuje řadič sériového kanálu na vodiče RX_TTL a TX_TTL. Pomocí těchto vodičů je pak signál z řadiče přiveden do bloku převodníku úrovně TTL-RS232.

K portu PE je připojeno ovládání I-kanálu, přes vodiče CHIX_E a CHIY_E. Po vodičích CHIX a CHIY jsou přenášeny signály z I-kanálů snímačů a jsou přivedeny na alternativní vlastnost portu PE ovládající volání interruptu (přerušení) v procesoru. Tímto způsobem je schopen mikrokontrolér detekovat signál z I-kanálu nezávisle na HCTL-2032, čímž jsou rozšířeny možnosti modulu při zpracování signálů ze snímačů. Vodiče CNTCASX a CNTCASY jsou také připojeny k přerušení. Mikrokontrolér tak detekuje přetečení registrů kvadrurního dekodéru.

3.6 LCD displej

Pro účelnost modulu, který zpracovává signály a vyhodnocuje signály z inkrementálních snímačů je modul osazen LCD displayem. Na displeji mohou být zobrazeny zpracované údaje např. o poloze nebo rychlosti.

Použitý LCD displej je vlastně modul displeje a nese typové označení M2004E-SYL. Provedení modulu displeje je na fotografii obr. 15

Tento modul displeje obsahuje:

- čtyř-řádkový , dvaceti znakový LCD displej
- LED podsvícení
- regulaci kontrastu
- řadič zobrazovače HD44780



obr. 15 Fotografie LCD displeje použitého v modulu

3.6.1 Řadič displeje HD44780

Hlavní součástí displeje je právě řadič HD44780, který zajišťuje zobrazování znaků na displeji. Tento řadič je používán v drtivé většině LCD displejů jako takových.

Komunikace mezi displejem pak probíhá pomocí 8bit nebo 4bitové datové sběrnice a pomocí řídicích signálů pro řízení toku dat na sběrnici. Celé propojení s modulem displeje je pak pomocí 16ti vodičů dle obr. 16 . Vývody řadiče jsou kompatibilní s logikou typu otevřený kolektor.

Vývod	Název	Funkce
1	Vss	GND
2	Vcc	napájení +5V
3	Vee, V0	nastavení kontrastu
4	RS	volba mezi 0 - instrukce, 1 - data
5	R/W	volba mezi 0 - zápis, 1 - čtení
6	E	hodinový vstup
7	DB0	data 0
8	DB1	data 1
9	DB2	data 2
10	DB3	data 3
11	DB4	data 4
12	DB5	data 5
13	DB6	data 6
14	DB7	data 7
15	LED+	anoda podsvětlení
16	LED-	katoda podsvětlení

obr. 16 zapojení LCD displeje s řadičem HD44780

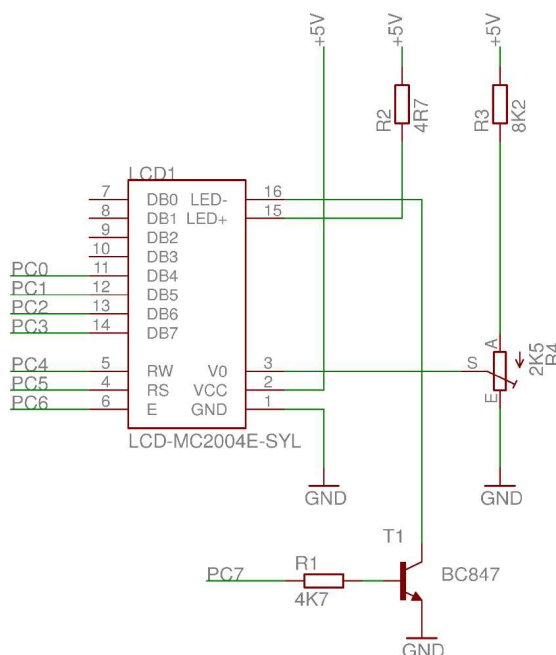
Řadiči můžeme posílat buďto instrukce, jako například umístí kurzor, zapni nebo vypni kurzor, nastav velikost znaku apod., nebo data, která reprezentují znaky zakódované podle ASCII. Pro volbu mezi instrukcemi nebo daty slouží vývod nazvaný RS.

S řadičem je možno komunikovat i obousměrně tedy nejen mu posílat data a instrukce, ale například zjišťovat aktuální polohu kurzoru, nebo právě zobrazené znaky. Režim čtení/zápis nastavuje vývod R/W.

Komunikace s displejem je synchronizována hodinovými pulzy na vstup E.

Vývody DB0-DB7 slouží pro data nebo instrukce posílané displeji nebo z něj. Veškeré instrukce nebo data jsou osmibitová.

3.6.2 Zapojení a komunikace s LCD displejem



obr. 17 Zapojení LCD displeje v modulu

V zapojení LCD modulu na obr. 17 je celý modul zakreslen jako součástka LCD1.

Rezistor R3 a trimr R4 tvoří dělič napětí, který je připojen k vývodu pro ovládání kontrastu. Kontrast displeje je řízen napětím.

Rezistor R2 o velikosti 4,7ohmů připojený na anodu LED podsvícení, omezuje proud tekoucí LED diodami, aby nedošlo k jejich zničení při připojení na napětí. Tranzistor T1 potom rozpojuje obvod podsvícení a je řízen řídicím MCU.

Komunikace s displejem probíhá po čtyřech vodičích. PC0-PC3, které jsou připojeny k mikrokontroléru. Signály na vodičích PC4-PC6 řídí komunikaci s řadičem displeje.

Při použití 4bitové komunikace s řadičem je data nutno posílat a potvrzovat „napůl“. Tedy sekvence je taková, že požadovaná instrukce nebo data, které jsou vždy 8bitová, se rozdělí na dvě 4bitová slova. Na sběrnici se vystaví vyšší 4bity a proběhne potvrzení pulzem na vstup E displeje. Poté se vystaví nižší 4bity a proběhne potvrzení. Vyjímkou je jen první inicializační instrukce, kterou je třeba poslat displeji po zapnutí k napájení. Ta se odešle jako osmibitová, respektive odešle se jen její *horní slovo*. Poté se inicializační instrukce pošle ještě jednou již ve 4bitovém režimu.

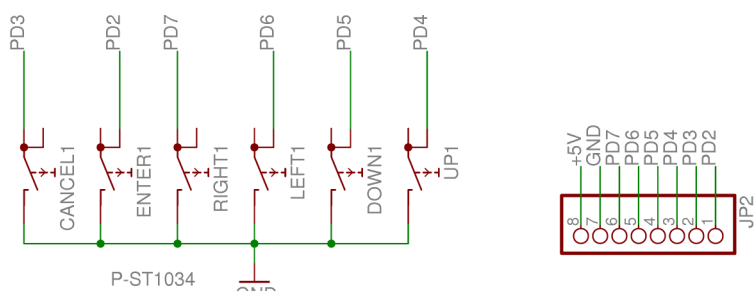
3.7 Klávesnice

Pro ovládání modulu slouží jednoduchá klávesnice sestavená z šesti tlačítek.

Čtyři černá tlačítka sestavena do směrového kříže slouží pro navigaci v menu zobrazovaném na displeji, nebo jsou jim přiřazeny funkce v režimech ve kterých modul právě běží.

Červené a šedé tlačítko slouží k potvrzování nebo stornování.

3.7.1 Zapojení klávesnice



obr. 18 Zapojení klávesnice modulu

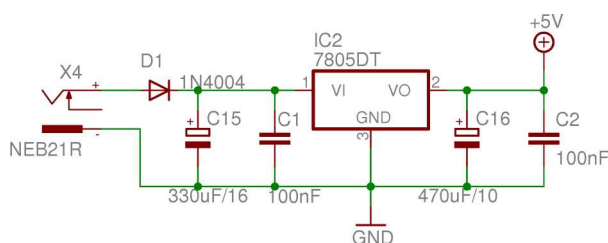
Tlačítka klávesnice jsou připojena k vývodům portu PD řídicího MCU a spínají tyto vývody se zemí modulu. Řídicí MCU má na těchto portech integrované pull-rezistory a detekuje tak logickou úroveň 1 při nestisknutém tlačítku a logickou úroveň 0 při stisknutém tlačítku.

Klávesnice je k modulu připojena přes konektor kvůli rozebiratelnosti modulu.

3.9 Napájení modulu

Modul může být napájen bezpečným napětím v rozmezí 6-30V stejnosměrných. Je tedy možné použít například běžný síťový adaptér. Napájecí konektor je uzpůsoben standardům těchto adaptérů.

3.9.1 Zapojení zdroje modulu



obr. 20 Zapojení stabilizace napětí pro napájení obvodů modulu

Pro správnou funkci obvodů použitých module je třeba stabilizované napětí 5V. Toto je v modulu zajištěno tří nožičkovým stabilizátorem 7805DT s pevným napětím 5V.

Tento stabilizátor má stabilizační úbytek minimálně 1V a maximální dovolené vstupní napětí 30V. Při několikanásobně vyšším vstupním napětí dochází ovšem k disipaci nadbytečné energie do tepla a obvod se může značně zahřívat. V modulu není tento stabilizátor opatřen externím chladičem. [3]

Polovodičová dioda D1 slouží k ochraně proti přepólování.

4 REALIZACE MODULU

Pro realizaci modulu bylo třeba navrhnout schéma zapojení, dimenzovat a zvolit provedení některých součástek a vybrat konektory pro připojení čidel a jiných periférií k modelu. Dále pak s ohledem na velikost vybrat vhodnou krabičku, do které bude modul vestavěn. Návrh desky plošných spojů byl pak podřízen tvarům a rozměrům krabičky.

Kompletní seznam součástek je uveden v *příloze 1*.

Pro návrh schématu, rozmístění součástek a návrh desky plošných spojů byl použit software EAGLE k tomu uzpůsobený.

4.1 Eagle 5.4.0

Software EAGLE vytvořila firma CADSoft. Název tohoto softwaru je zkratkou pro „Easily Applicable Graphical Layout Editor“, což by se dalo volně přeložit jako „Snadno použitelný editor pro navrhování designu“. Slouží tedy pro kreslení schémat elektrických obvodů a následný návrh desky plošných spojů (DPS) z vytvořeného schématu. Eagle obsahuje také knihovny běžných součástek, ze kterých se součástky pak snadno vkládají do návrhu.

Při návrhu byla použita verze EAGLE 5.4.0 Light ve školních laboratořích. Firma CADSoft nabízí tento software také jako freeware s omezením některých parametrů návrhu[7].

Program EAGLE obsahuje několik dílčích modulů pro kompletní zpracování návrhu:

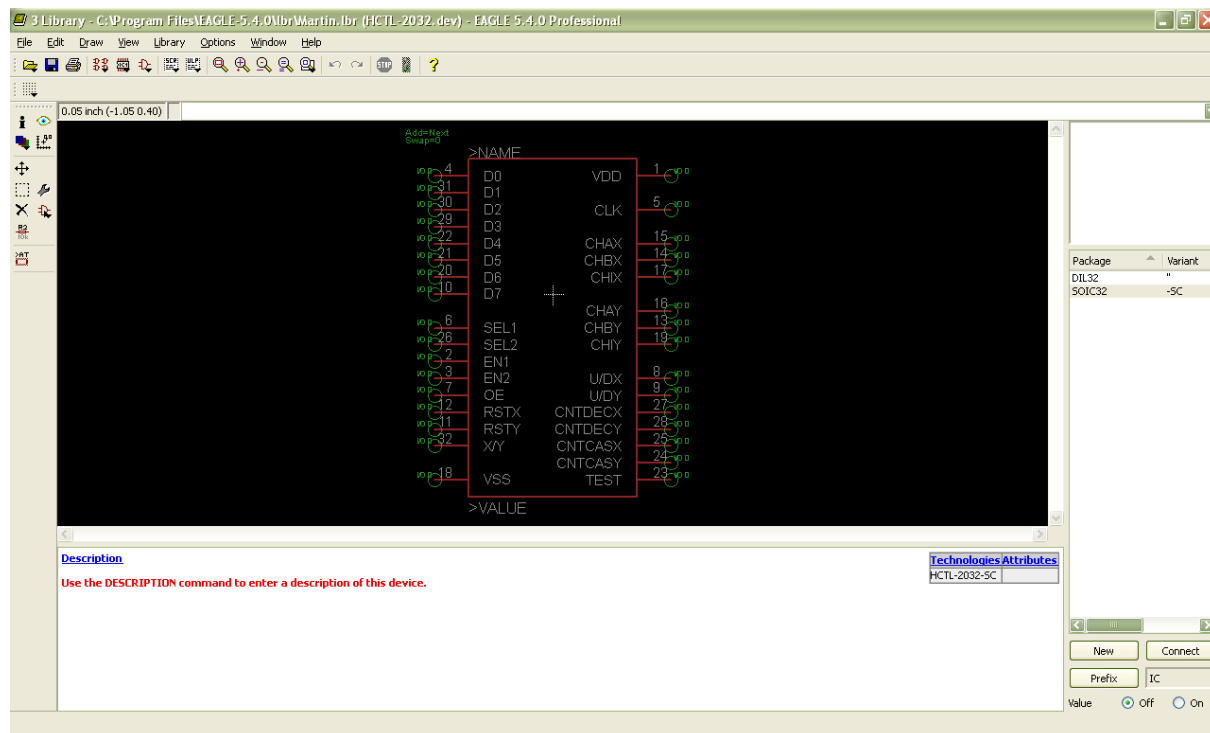
- Ovládací panel
 - slouží pro správu projektů a pro spuštění ostatních modulů
- Editor schémat
 - modul pro návrh elektrických schémat
- Editor plošných spojů
 - modul pro editaci návrhu desky spojů
- Editor knihoven součástek
 - umožňuje editovat stávající knihovny se součástkami nebo definovat do eaglu součástky vlastní
- Autorouter
 - je součástí modulu pro editaci plošných spojů a umí návrh desky vytvářet automaticky na základě uživatelem nadefinovaných parametrů

Při realizaci modulu byly použity všechny výše uvedené moduly, kromě autorouteru, který v použité verzi programu EAGLE 5.4.0 Light není zařazen.

4.1.1 Editor knihoven součástek

Pro sestavení kompletního schématu modulu bylo nejdříve zapotřebí definovat některé součástky do knihoven Eaglu.

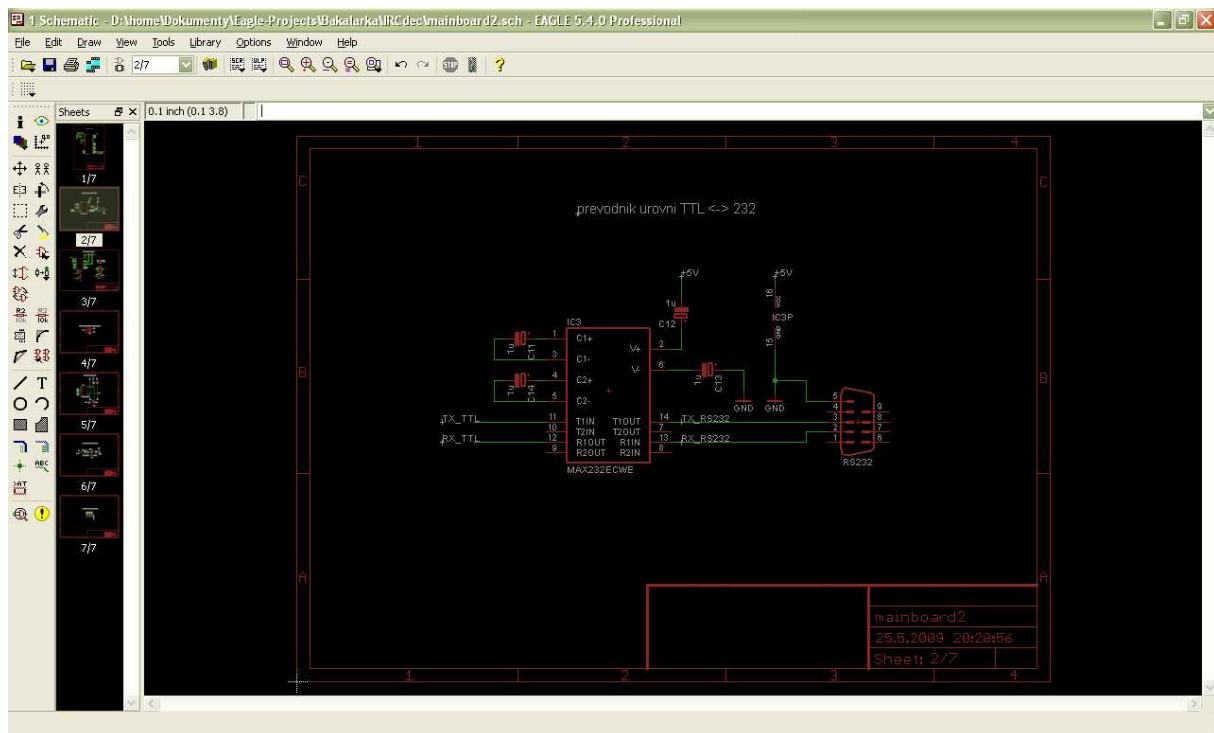
Důležitou součástí modulu je kvadrurní dekodér HCTL-2032. Byl dostupný pouze v provedení pro „klasickou“ montáž součástek do desky, kdy se jeho vývody zaletují přes vrtné otvory skrz DPS. Pouzdro této součástky má 32 vývodů. V tomto provedení je poměrně velký oproti provedení SMD, které ovšem nebylo v době návrhu dostupné. Do knihovny v Eaglu byly tedy nadefinovány k jeho standardní patici DIL32 také popisy jednotlivých vývodů. Podobně byl v Eaglu nadefinován i modul LCD displeje i s jeho rozměry, aby se k nim dalo přihlédnout při návrhu desky. Ukázka prostředí editoru knihoven je na obr. 21 .



obr. 21 Editor knihoven součástek v Eaglu

4.1.2 Návrh schématu

Nejdříve se do návrhu přidají knihovny s potřebnými součástkami a ty se rozmístí do plochy návrhu. Editor sám při vkládání součástek dává součástkám číselná označení, projektant pak jen definuje hodnotu parametru součástky, jako např. kapacitu u kondenzátorů apod. Součástky se pak propojují nástrojem pro kreslení vodičů. Pro přehlednost rozsáhlejších schémat není třeba vodiče natáhnout poctivě od vývodu k vývodu, ale stačí vodič „ustříhnout“ kousek za vývodem a opatřit ho jmenovkou. Tato jmenovka je pak v rámci editoru unikátní a pokud v návrhu existuje více vodičů se stejným jménem, chápe je Eagle jako spojené. Schéma se dá tak rozdělit do několika bloků a tyto bloky pak pohodlně rozmisťovat při tisku. Tímto způsobem byla kreslena veškerá schémata modulu. Ukázka schématu v prostředí Eaglu je na obr. 22. Kompletní schémata jsou v příloze 2.



obr. 22 Editor schémat v Eaglu

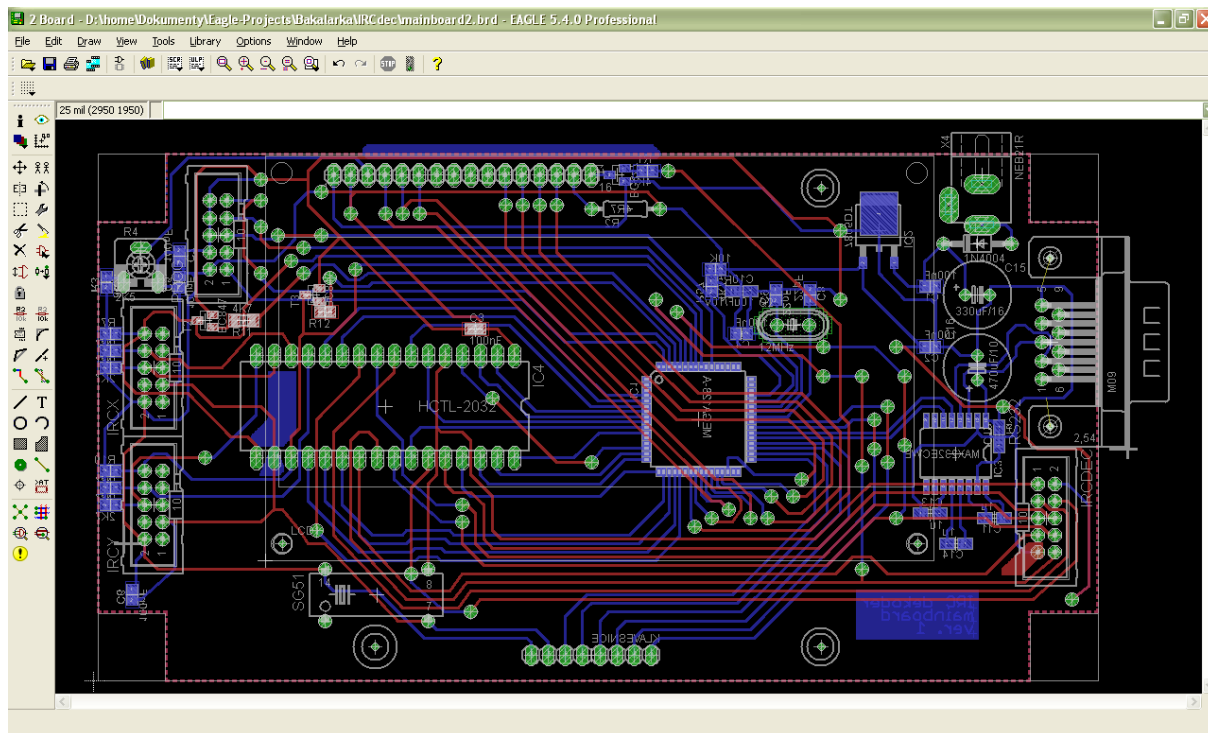
4.1.3 Návrh desky plošných spojů

Návrh desky plošných spojů byla na celém návrhu asi ta nejpracnější část. Je třeba ji udělat pečlivě a důkladně kontrolovat, aby nedošlo k chybám při návrhu cest, zejména kvůli vyrobiteľnosti v laboratorních podmínkách. Editor spojů sice disponuje nástroji pro automatickou kontrolu dle pravidel, které projektant nastaví, ale počítá s podmínkami pro sériovou výrobu desek, kde se dá dosáhnout například větší minimalizace rozměrů. Nebo, což je nejzásadnější, s použitím technologie pro spojování vrstev vícevrstvých desek, tedy takzvaných prokovů. Prokovy lze při sériové výrobě umísťovat např. pod konektory nebo některé součástky typu SMD.

Při návrhu DPS se nejprve nakreslí rozměr desky a poté do něj rozmístí patice součástek, které byli použity ve schématu. Tyto patice Eagle automaticky vkládá do návrhu dle schématu. Patice jsou pak propojeny takzvanými vzdušnými spoji, což jsou ve skutečnosti úsečky naznačující které vývody součástek jsou dle schématu propojeny vodičem a nesou i jejich názvy. Nástrojem pro kreslení cest spojů se pak jednotlivé vývody propojují a projektant rozmísťuje cesty podle svého uvážení, nebo použije autorouter. Editor barevně rozděluje grafické objekty jako jsou, pájecí plošky, obrysy součástek, cesty nebo prokovy do takzvaných hladin podle jejich významu. Každá z těchto hladin má pak definovány vlastnosti právě podle jejího významu. Např. tloušťky spojů, jejich nejmenší dovolené vzdálenosti nebo tvary, rozměry a vrtání u pájecích plošek.

Zapojení modulu bylo rozvrženo na dvě desky plošných spojů. Hlavní desku, na kterou byly umístěny všechny části modulu kromě klávesnice, a desku pro klávesnici. Vzhledem ke složitosti propojení na hlavní desce, byla použita oboustranná deska. Tedy dvě vrstvy spojů propojené prokovy. Jednoduchá klávesnice pak byla navržena na jednostranné desce menších rozměrů. Na obě desky byly přidány konektory pro jejich propojení.

Ukázka z prostředí editoru plošných spojů při návrhu desky je na obr. 23. Navržené desky jsou v příloze 3.



obr. 23 Editor plošných spojů v Eagle

4.2 Výroba desky plošných spojů

Obě desky modulu byly vyrobeny ve školní laboratoři na kuprexitové polotovary, které byly zakoupeny.

Kuprexitová deska je kompozit textilních materiálů zpevněných laminováním a potažený tenkou spojitou vrstvou mědi, na které se pak vytvářejí obrazce plošných spojů.

4.2.1 Technologie výroby

Desky plošných spojů byly vyrobeny za použití foto-litografické technologie. Jedná se o postup, kde je na desku nanesen fotorezist reagující např. na ultrafialové světelné spektrum. Na tento fotorezist se pak přiloží předloha, která nepropustí záření daného spektra a deska se pak vystaví tomuto záření. Osvícený fotorezist změní své fyzikální vlastnosti. Deska se pak vyvolá v chemickém roztoku zvaném „vývojka“. Na desku je tak přenesena předloha takzvaná „maska“, která chrání měď před působením jiného chemického roztoku takzvaného „leptacího roztoku“. Po ponoření do leptacího roztoku se nechráněná měď odleptá a na desce zůstane měď chráněná fotorezistivním lakem, což vytvoří požadované spoje.

Deska se dále očistí od fotorezistu, případně se odvrtnají otvory v desce, odmastí se od nečistot a nanese pájecím lakem, který chrání čistou měď před oxidací a zároveň zlepšuje vlastnosti pájky při letování součástek.

Polotovar desky spojů byl zakoupen s již naneseným fotorezistem.

Použité pomůcky pro výrobu:

- stolní lampa s ultrafialovou výbojkou
- vývojka – 1,5 % NaOH
- leptací roztok – FeCl₃
- odmašťovadlo – aceton, izopropyl
- pájecí lak ve spreji
- vrtačka na plošné spoje
- smirkový papír
- předloha na průsvitné fólii

4.2.2 Předloha

Pro kvalitní výsledek foto-litografické metody je velice podstatnou věcí kvalitní předloha. Tedy, aby předloha v místech spojů spolehlivě nepropustila paprsky UV záření a naopak v ostatních místech byla spolehlivě propustná.

V laboratorních nebo domácích podmínkách se dá jako předloha například použít pauzovací papír nebo různé průhledné fólie do tiskáren. Ovšem tyto nemusí být dostatečně spolehlivé, zejména při použití tenkých cest v návrhu. Nepříjemné je, že pokud předloha selže a nebude dostatečně chránit desku před UV zářením v požadovaných místech, dojde k částečnému, nebo mnohdy i k úplnému vyvolání rezistu ve vývoje a deska je tak znehodnocena. Také může dojít k takzvanému „podleptávání“ při leptání a výsledkem mohou být, v lepším případě jen neostře hrany spojů, a v tom horším, přerušené cesty. V takových případech je třeba výrobu opakovat a ta se tak stává nejen časově náročnější, ale také nákladnější.

Pro zajištění kvalitní předlohy byla předloha navržená v Eaglu exportována do grafické podoby formátu PDF a zadána reklamní společnosti, která přenesla předlohu na osvitový film. Tento film je tenká dokonale průhledná fólie na které je vyvolána tmavá nepropustná vrstva. Podobně jako při vyvolání filmu z fotoaparátu.

4.2.3 Osvit

Před osvitem bylo třeba připravit předlohu. Při použití oboustranné desky bylo třeba zajistit, aby obě strany předlohy pro osvit na desku byly přesně souosé. Předloha byla tedy rozřezána a pomocí lepící pásky byly obě strany předlohy slepeny proti sobě co možná nejpřesněji. Vznikla tak jakási kapsa do které se poté vložila již zastřižená deska přibližných rozměrů. S předlohou se deska zafixovala mezi dvě skla, aby se s předlohou při osvětlování nedalo pohnout.

Osvit proběhl pod UV lampou cca ve vzdálenosti 20cm. Každá strana se tak osvětlovala zvlášť cca 5min.

4.2.4 Vyvolání

Osvícená deska se ponořila do ploché misky s vývojkou. Pomocí pinzety bylo s deskou jemně pohybováno, aby na povrchu desky roztok stálé proudil a omýval tak již vyvolané zbytky osvíceného laku. Po několika minutách na desce začala vystupovat předloha a po úplném vyvolání osvíceného laku byla deska vytažena. Doba cca 10min.

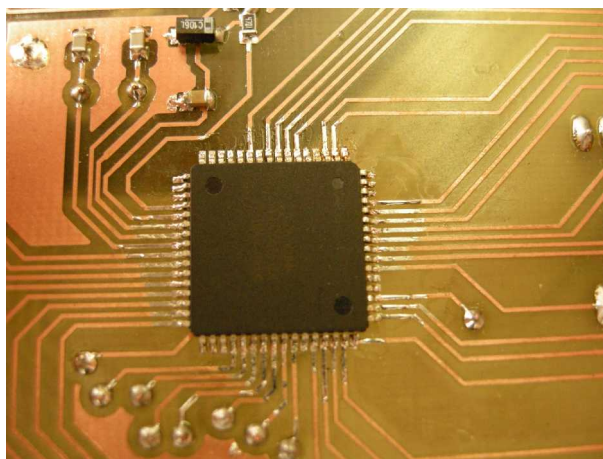
4.2.5 Leptání

Po vyvolání byla deska omyta vodou, aby se odstranily zbytky vyvolaného laku a usušena. V ploché misce byl připraven leptací roztok o teplotě okolí. Účinky tohoto roztoku se zlepšují při zahřátí na vyšší teploty. Roztok je pak „agresivnější“. Při menších vadách na předloze nebo nedokonalosti krytí fotorezistem by tak mohlo dojít k podleptání. Deska byla položena na hladinu roztoku mědi dolů. Tím že deska plavala na hladině bylo zajištěno že odleptané kousky mědi vlivem gravitace padají na dno a při kontrolování rozeznáme co se již vyleptalo a co ne. Každá strana desky se odleptávala přibližně 30minut.

4.2.6 Osazení součástek

Před osazením součástek bylo třeba desku očistit a konzervovat. Po odleptání byla deska omyta vodou od zbytků leptadla, vrstva fotorezistu byla rozpuštěna a odstraněna acetonem. Deska byla také zastřížena do konečného tvaru. Díry pro součástky pak byly odvrtny na stojanové vrtačce. Nerovnosti po odvrtávání a stříhání se začistily smirkovým papírem a jemným smirkovým papírem byl vyleštěn i povrch měděných spojů. Deska se ještě odmastila izopropylem a zakonzervovala pájecím lakem.

Po zaschnutí laku se na desku mohli osadit součástky. Začalo se od prokovů, které jsou tvořeny tenkými drátky. Ty byli prostrčeny vyvrtanými dírami a zapájeny z obou stran. Dále se pak osadily součástky SMD, jako např. mikroprocesor. Nakonec se zaletovali patice a konektory.



obr. 24 Detail osazeného MCU a některých SMD součástek

4.3 Oživení a finální sestavení modulu

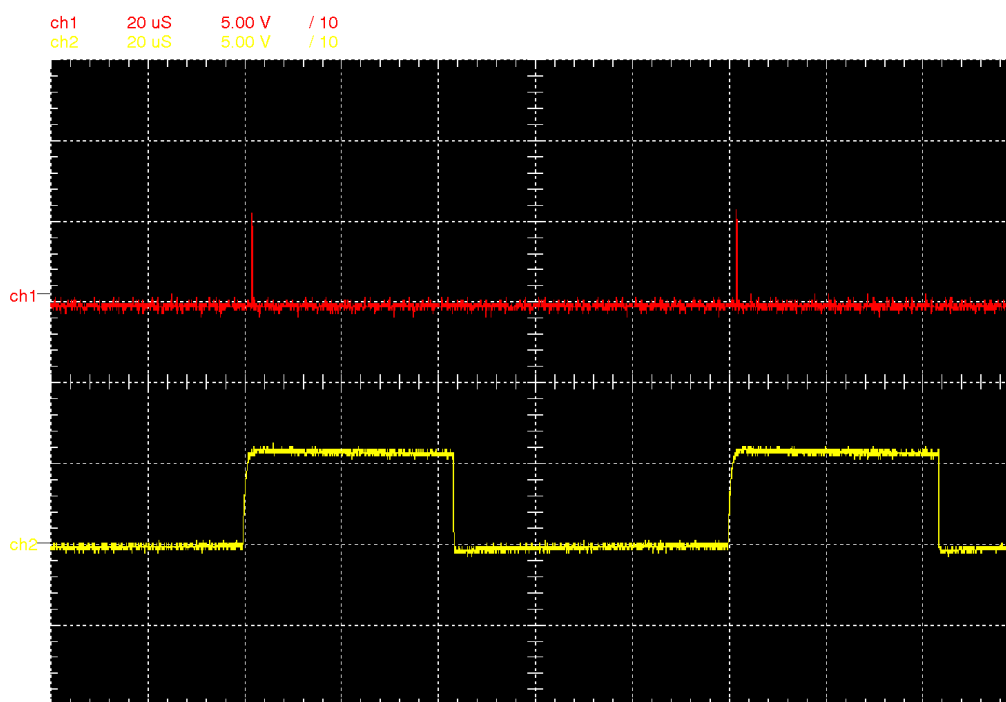
4.3.1 Oživení modulu

Před finálním sestavením modulu bylo třeba otestovat funkčnost zapojení obvodů na desce.

Modul se připojil k napájení a bez osazených součástek do patice bylo na desce proměřeno napájení ze stabilizace obvodu, jestli je přivedeno správně na součástky. Toto měření bylo provedeno pomocí digitálního multimetru.

Dále bylo nutné otestovat správné zapojení mikrokontroléru. Deska se osadila krystalem a přes programovací konektor se deska připojila k programátoru mikrokontroléru. Programátor softwaru detekoval připojený čip a přečetl zatím prázdnou paměť mikrokontroléru. Do patice na desce se vložily ostatní součástky a ještě se pomocí osciloskopu proměřil hodinový signál z krystalového oscilátoru, jestli dává kmitavý signál pro kvadrurní dekodér. Pro otestování funkce kvadrurního dekodéru byl k modulu připojen laboratorní motor s inkrementálním snímačem. Na výstupy dekodéru CNTDEC a ke kanálu A inkrementálního snímače byli připojeny sondy osciloskopu. Motor se snímačem byl připojen k napájení a byl roztočen.

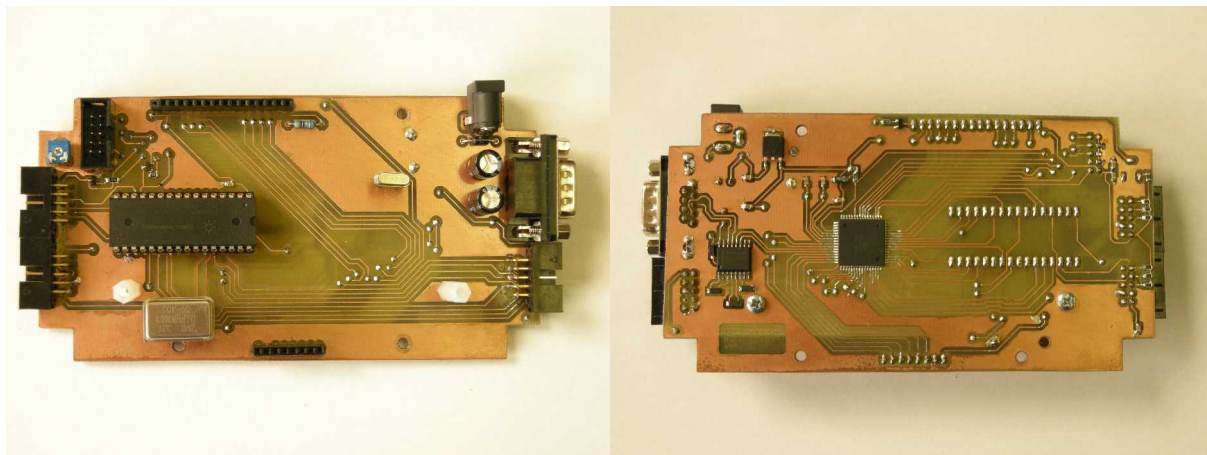
Na obr. 25 je vidět průběh signálu z kanálu A snímače (žlutě) a průběh dekódovaných pulzů (červeně).



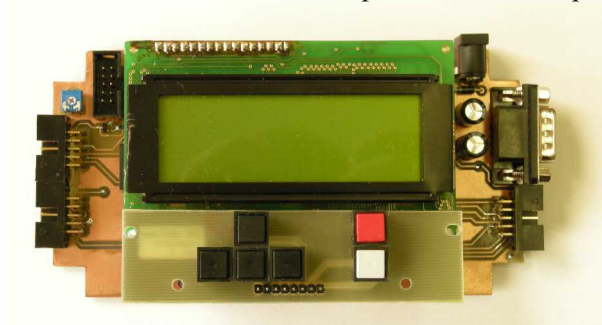
obr. 25 Průběh dekódování signálu na osciloskopu

4.3.2 Sestavení modulu

K sestavení modulu do krabičky bylo třeba jednotlivé desky vymezit a sešroubovat pomocí distančních sloupků a do krabičky vyřezat patřičné otvory. Nakonec byly na krabičku nalepeny popisky vytištěné na lepicí papír. Jednotlivé části modulu a sestavený modul jsou na fotografiích níže.



obr. 26 Osazená hlavní deska modulu – pohled z horní a spodní strany



obr. 27 Sestavené komponenty. Hlavní deska, LCD displej a klávesnice



obr. 28 Sestavený modul v krabičce

4.4 Programování řídicího mikrokontroléru

Aby řídicí mikrokontrolér dokázal číst data z kvadrurního enkodéru, zobrazovat je na displej, reagovat na stisky kláves, nebo posílat data přes rozhraní RS232, bylo nutné naprogramovat obslužný program a pomocí programátoru jej zapsat do čipu mikrokontroléru. Takový program se vytváří na platformě PC a pak je pomocí překladače přeložen do strojového kódu pro jednočipový 8bitový procesor a zapsán do procesoru.

Pro mikrokontrolér ATmega128, který je z řady čipů nazývané AVR, je od výrobce Atmel k těmto účelům zdarma poskytováno vývojové prostředí pro projektování kódu pro tyto procesory AVRStudio.

Pro programování mikrokontroléru byly použity tyto nástroje:

- AVRStudio od výrobce Atmel
- Překladač jazyka C WinAVR pro mikroprocesory
- Programátor PRESTO od firmy ASIX
- Obslužný program UP pro programátor PRESTO.

4.4.1 AVRStudio

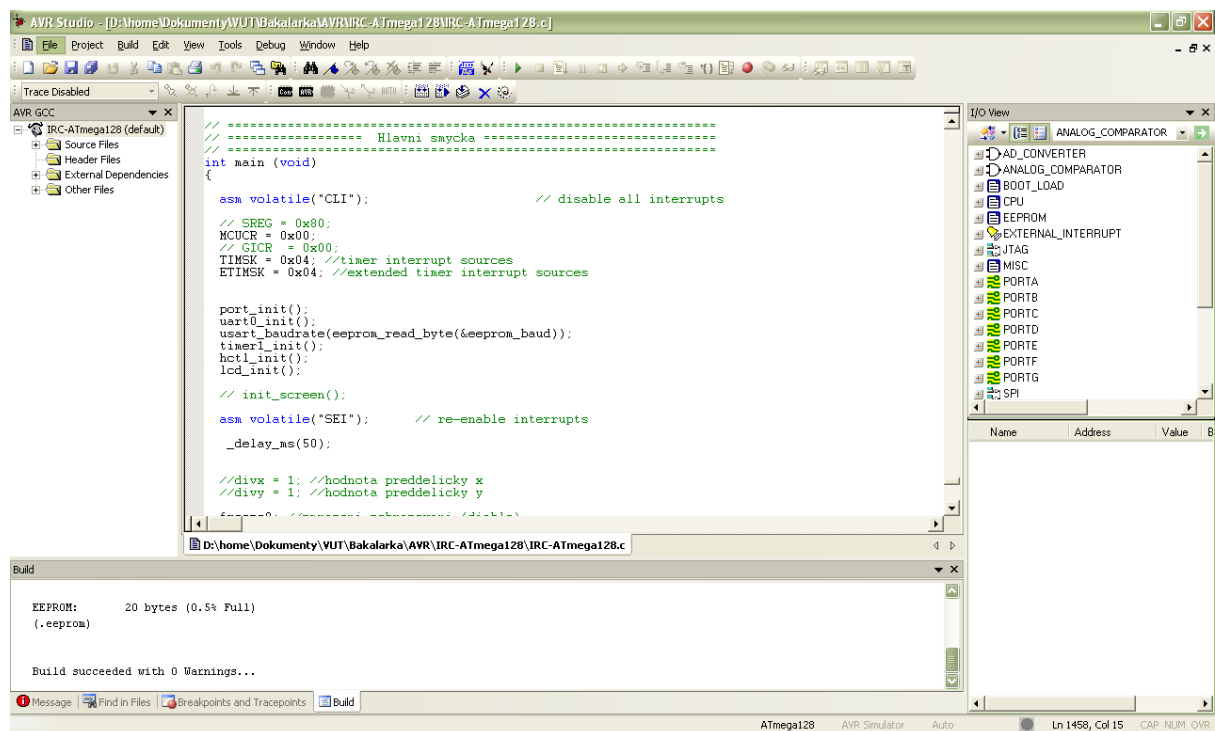
AVRStudio je vývojové prostředí pro mikrokontronroléry z řady AVR od výrobce Atmel. Toto prostředí je od výrobce poskytováno volně.

Prostředí AVRStudia mimo jiné v sobě zahrnuje:

- správu projektů
 - pro snadné a přehledné vytváření rozsáhlých projektů
- simulátor procesorů a debugger
 - simulace běhu programu v mikroprocesoru a možnost ladění programu za běhu takzvaný „debugger“
- integrovaný překladač assembleru do strojového kódu
 - pro vytváření programu za pomoci jazyka symbolických adres takzvaného „assembleru“
- možnost integrace jiných překladačů
 - dává možnost integrovat překladač jiného jazyka ve kterém je možno programy vytvářet

4.4.2 WinAVR - Překladač C/C++ pro AVR

Při programování rozsáhlejších programů do jednočipových procesorů, se stává použití jazyka assembler dosti náročné. S výhodou můžeme využít překladače z vyšších jazyků jako je například jazyk C. Přimo do prostředí AVRstudia je možno integrovat balík překladače jazyka C WinAVR. Tento balík integruje do AVRStudia vývojové pomůcky a knihovny pro jazyk C a C++. Knihovny WinAVR obsahují základní funkce a definice pro práci s mikrokontroléry. Jedná se o otevřený software pro platformu Windows založený na licenci GNU. Kromě integrovaného prostředí do AVRStudia je vybaven i vlastním nezávislým prostředím pro psaní kódu. [8]



obr. 29 Prostředí AVRStudia s WinAVR

Na obr. 29 je ukázka prostředí AVRStudia s integrovaným WinAVR. Pracovní plocha je rozdělena do několika oken:

- vlevo je navigátor mezi jednotlivými soubory projektu
- uprostřed se nachází editor kódu, který barevně odlišuje text podle syntaxe jazyka C.
- vpravo leží okno simulátoru periférií mikroprocesoru
- v dolní části je okno zpráv

4.4.3 Programování mikrokontroléru.

Za použití výše uvedených pomůcek je už možné vytvořit vlastní kód pro mikroprocesor.

V AVRStudiju se vytvoří nový projekt. Napíše se vlastní kód v jazyce C v editoru kódu a přeloží se pomocí nástroje pro překlad v AVRSutidu. Takto vznikne v adresářích projektu soubor s příponou *.HEX, který obsahuje přeložený program do strojového kódu. Tento soubor se pak otevře pomocí programu UP, který obsluhuje programátor, a pomocí něj se program zapíše do procesoru. Po nahrání programu do mikroprocesoru programátor sám provede reset procesoru, čímž se spustí běh programu v procesoru.

Pro vytváření programu pro mikrokontrolér je důležité znát funkci použitých periférií mikroprocesoru a registry které jsou důležité pro jejich programové ovládání.

Porty

Pro obsluhu každého z portů procesoru jsou důležité tři registry. Tyto registry jsou 8bitové a každý bit odpovídá jednomu vývodu portu a jeho chování může být nastaveno individuálně.

- Registr DDR (Data Direction register) – nastavuje směr portu 0 = vstupní, 1 = výstupní
- Registr PORT – reprezentuje vývod portu a ve výstupním režimu se jeho pomocí nastavují logické úrovně na výstupu. Ve vstupním režimu potom zapíná(hodnota 1) nebo vypíná(hodnota 0) integrované pull-up rezistory každého z vývodů.
- Registr PINA – registr přečtených hodnot z portu. Při výstupním režimu je v něm uchována předchozí nastavená hodnota[9].

Čítače/časovače

Čítače/časovače obsahuje mikrokontrolér ATmega 128 čtyři a jsou pojmenované TIMER0 až TIMER3. Tyto čítače/časovač obsahují několik registrů pro ovládání jejich funkce. Pro odměřování času bych zde uvedl pouze výpočet hodnot pro registry odměřování času.

Pro účely modulu byli použity TIMER1 a TIMER3 které jsou 16bitové. Pro nastavení požadovaného času jsou zapotřebí tedy dva 8bitové registry TCNTxH a TCNTxL, kde malé x značí číslo časovače. Pokud je časovač spuštěn, je v každém strojovém cyklu mikroprocesoru jeho hodnota inkrementována. Při přetečení těchto registrů je vyvoláno patřičné přerušení od časovače a může tak být vykonána obsluha tohoto přerušení.

Výpočet hodnoty odměřovaného času vychází z hodnoty pracovní frekvence mikrokontroléru, tedy z hodnoty krystalu použitého v modulu. Tato hodnota je 12MHz. Jeden strojový cyklus tedy odpovídá době jedné periody pracovní frekvence. Mikroprocesor také má logickou předděličku, kterou je možné přednastavit na hodnoty 1, 8, 64, 256 a 1024.[10] Vztah mezi těmito proměnnými by mohl vypadat takto:

$$65536 - \frac{t \cdot f}{p} \quad \frac{[s] \cdot [Hz]}{[]}$$

kde konstanta 65536 je maximální počet cyklů do přetečení, t je požadovaná doba, f je frekvence krystalu a p je hodnota předděličky.

Pro přednastavení doby byl napsán pro modul podprogram timer3_preset(double t), který je součástí celého zdrojového kódu pro modul. Parametrem t této funkce se pak nastavuje požadovaná hodnota v milisekundách.

Přerušení

Vyvolat přerušení znamená, zastavit běh programu pro obsloužení jiného prioritnějšího požadavku[11].

Přerušení v mikrokontroléru může vyvolat celá řada věcí. Pro potřeby modulu je přerušení voláno jen v těchto případech:

- Přerušení od časovače1, který slouží pro ovládání klávesnice
- Přerušení od časovače3, který cyklicky odesílá data přes sériovou linku
- Externí přerušení, při příchodu pulzu z I kanálu čidla (pokud je zapnut)
- Přetečení čítacích registrů kvadrurního dekodéru

USART

Mikrokontrolér má hardwarově integrovány dva řadiče pro sériový přenos dat. USART0 a USART1. V modulu je použit USART1. Pro programovou obsluhu USARTu je třeba nejdříve nastavit parametry přenosu které souvisí se sériovou komunikací, jako je přenosová rychlost, parita, stop bit aj.

Nastavení potřebných registrů bylo provedeno za pomoci demoverze programu ICCAVR7 firmy ImageCraft Creations, který obsahuje nástroje pro výpočet hodnot těchto registrů.

4.4.4 Program pro modul

Program pro řídicí MCU modulu má klíčový význam pro jeho funkci. Musí zajišťovat čtení dat přicházejících z kvadrurního dekodéru, zpracovávat je a zobrazovat na displej nebo odesílat pomocí sériové linky. Dále také obsluhovat požadavky z klávesnice a na jejich základě nastavovat parametry chodu.

Celý program běží v nekonečné smyčce, která provádí dané rutiny jednotlivých podprogramů, které mají za úkol ovládání jednotlivých bloků modulu. Pro sestavení celého kódu bylo třeba řešit komunikaci s jednotlivými bloky modulu postupně a jednotlivé rutiny odladit samostatně.

Funkce jednotlivých částí bude popsána postupně a zdrojové kódy jsou v elektronické příloze.

Ovládání klávesnice

Ke čtení požadavků z klávesnice je použit port D procesoru, respektive některé jeho vývody. Tyto vývody bylo třeba nastavit jako vstupní. Stisk jednotlivých kláves klávesnice potom mění logické stavy na těchto portech.

Pro správnou funkci klávesnice bylo třeba ošetřit možné zákmity při přechodném ději stisku a uvolnění tlačítek klávesnice a také opakovaný stisk klávesy. K tomuto byl použit jeden z časovačů mikrokontroléru.

Ošetření zákmitů probíhá tak, že po prvním stisku klávesy je přečtena hodnota portu, pak následuje krátká prodleva několika milisekund a čtení je zopakováno. Pokud místo stisku tlačítka tlačítka nastane jen krátký zákmit, čtená hodnota před prodlevou a po prodlevě se nebudou shodovat a procesor tak nevyhodnotí stisk klávesy.

Procesor může stisky kláves vyhodnocovat velice rychle a aby nedocházelo k zaregistrování několikanásobného stisku kláves při krátkém stlačení, je třeba po přečtení jednoho stisku čekat určitou dobu než může být zaregistrován další stisk. Toto je ošetřeno právě časovačem 1. Tento časovač je přednastaven na dobu 200ms a spustí se při stisku klávesy. V ten okamžik je čtení kláves zakázáno dokud nedoběhne čas odměřovaný časovačem, po kterém je čtení opět povoleno.

Ovládání klávesnice zajišťují podprogramy:

keyboard()

funkce vrací hodnotu načtených kláves v hexa.

timer1_start()

spouští odměřování doby 200ms

ISR(TIMER1_OVF_vect)

obsluhuje přerušení v procesoru při odměření doby časovače 1 a povolí čtení z klávesnice

Ovládání displeje

Pro ovládání displeje bylo třeba pochopit princip komunikace s jeho řadičem jak bylo popsáno výše.

LCD displej je připojen k celému portu C na řídicím MCU. Komunikace s displejem probíhá jednosměrně, respektive MCU pouze posílá data nebo příkazy řadiči displeje, proto jsou všechny vývody portu C nastaveny jako výstupní.

Podprogramy pro řízení displeje jsou poměrně rozsáhlé a jejich zdrojové kódy jsou tedy napsány v samostatném souboru lcd.c, který je součástí přiloženého zdrojového kódu. Tento zdrojový soubor kromě jiných obsahuje základní funkce pro ovládání displeje:

`lcd_init(void)`

inicializuje displej při prvním zapnutí

`lcd_data(unsigned char data)`

posílá data displeji

`lcd_cmd(unsigned char cmd)`

posílá příkaz displeji

`lcd_cursoron(char blink), lcd_cursoroff(char blink)`

zapnete nebo vypnete kurzoru blikající nebo neblinkající

`lcd_clr()`

smaže displej

`lcd_posstr(const char* strn, unsigned char r, unsigned char c)`

umístí text na souřadnice

`lcd_goto(unsigned char row, unsigned char col)`

přesune kurzor na souřadnice

Řízení obvodu HCTL-2032

Řídící signály obvodu HCTL-2032 jsou připojeny k řídicímu MCU na port F, který je nastaven jako výstupní a datová sběrnice pak k portu A, který je nastaven jako vstupní.

Veškeré podprogramy, které zajišťují komunikaci s tímto obvodem jsou napsány ve zdrojovém souboru `hctl.c`, který je součástí přiloženého zdrojového kódu.

Popis jednotlivých funkcí:

`hctl_init()`

vynuluje čítače obvodu a nastaví režim dekódování na 4x.

`hctl_readbytebyte(void)`

provádí přečtení 8bitové hodnoty na sběrnici obvodu. Čtení se provádí dvakrát po sobě a pokud by bylo neúspěšné je funkce volána rekurzí opakovaně. Funkce poté přečtenou hodnotu vrátí.

`uint32_t hctl_read32(char osa)`

funkce provádí sekvenci postupného vyčtení 32bitového registru čítače obvodu. Při čtení využívá funkci `hctl_readbyte`. Parametr funkce *osa* může nabývat hodnot 0, pro čtení osy x nebo 1 pro čtení osy y.

`hctl_reset(char osa)`

vymaže obsah čítacích registrů obvodu. Parametr *osa* může být 0 pro osu x, 1 pro osu y nebo 2 pro obě osy.

`hctl_stop(char osa), hctl_start(char osa)`

zastaví nebo spustí čítání. Parametr *osa* může být 0 pro osu x, 1 pro osu y nebo 2 pro obě osy.

`hctl_decmode(char mode)`

nastavuje režim čítání. Parametr *mode* může být 1 = přesnost 1x, 2 = přesnost 2x, 4 = přesnost 4x.

`hctl_ichannels(unsigned char i)`

kombinací hodnot parametru *i* připojuje I kanál snímačů na jednotlivých osách. 0 = oboje vypnuto, 1 = x-zapnuto a y-vypnuto, 2 = x-vypnuto a y-zapnuto, 3 = xy-zapnuto

Posílání dat po sériové lince přes převodník TTL-RS232

Mikrokontrolér má hardwarově integrovaný řadič pro sériový přenos dat USART (jak bylo popsáno výše). Po sériové lince modul odesílá hodnotu čítače spolu s údajem, o který ze dvou snímačů se jedná a směrem otáčení každého z nich. Pro pravidelné odesílání těchto dat je použito přerušení časovače 3, čímž je zajištěn běh tohoto odesílání paralelně s během hlavní smyčky programu.

Zdrojové kódy podprogramů jsou umístěny v souboru IRC-ATmega128.c, který je součástí přílohy. Ovládání sériové komunikace zajišťují tyto funkce:

`usart_transmit_string(const char* data)`
odesílá řetězec znaků po sériové lince

`usart_baudrate(char b)`
nastavení přenosové rychlosti dle parametru *b*

`ISR(TIMER3_OVF_vect)`
obsluha časovače 3 zajišťuje pravidelné odesílání dat po sériové lince

5 FUNKCE MODULU

Modul je ovládán pomocí černých tlačítek rozmístěných do směrového kříže a dvou tlačítek červeného a šedého.

Při přivedení napájení k modulu je na displeji zobrazeno hlavní menu, ze kterého se pomocí tlačítek prochází do dalších nastavení nebo se spouští jeho funkce.

Kompletní struktura položek menu, nastavení a podrobný popis ovládání je v *příloze 4*.

5.1 Režimy modulu

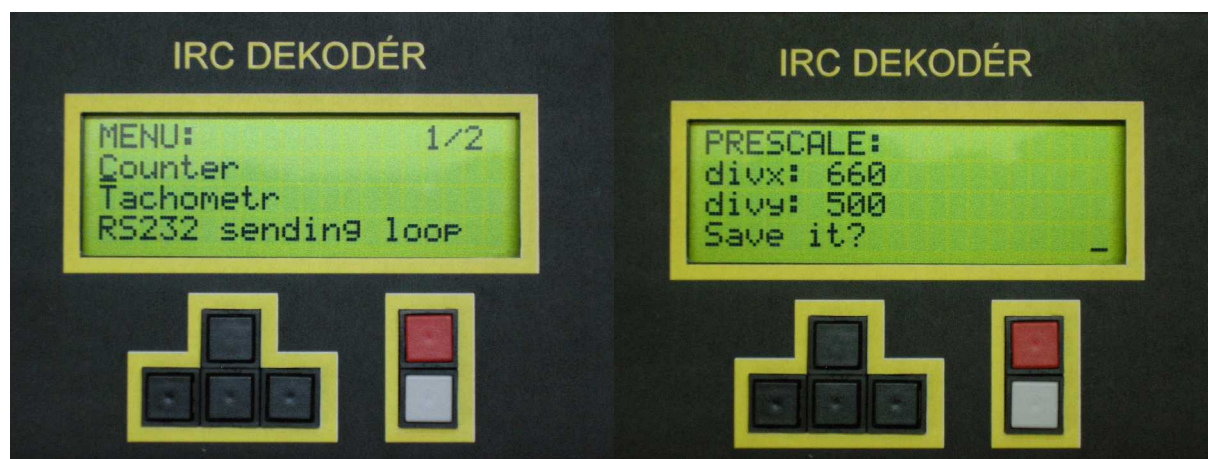
Modul pracuje celkem ve čtyřech režimech. Jsou to režimy:

- režim menu a nastavení
- režim čítače
- režim měření rychlosti otáček
- režim odesílání

5.1.1 Režim menu a nastavení

V tomto režimu modul nedělá nic jiného, než čeká na zadání z klávesnice. Je tak buď zobrazeno některé menu, ve kterém se dá pohybovat klávesami nahoru a dolů nebo nastavovat číselné parametry nebo přejít do jiného režimu.

Tento režim je spuštěn po připojení modulu k napájení.

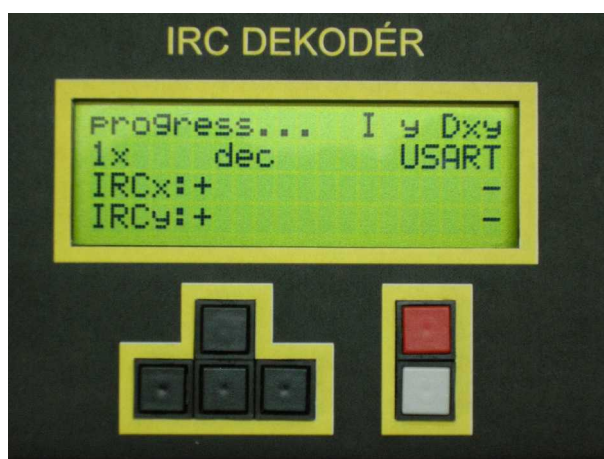


obr. 30 Fotografie modulu v hlavním menu. a v nastavení předděličky

5.1.2 Režim čítače

V tomto režimu spustí modul čítání signálů z inkrementálních snímačů a zobrazuje na displej čítané pulzy a směr otáčení obou připojených snímačů současně. Zobrazení čítané hodnoty lze přepínat do hexadecimální nebo decimální podoby. Tyto hodnoty se také dají zobrazit podělené celočíselnou hodnotou, kterou lze nastavovat pro každý ze snímačů individuálně v režimu nastavení. Na displeji je také zobrazován režim přesnosti který se dá za běhu měnit. Pro každé z čidel se dá také zapnout nebo vypnout I-kanál.

Při zapnutém I-kanálu na některém ze snímačů dochází při otočení snímače o 360° k vynulování čítané hodnoty. Čítač tak čítá pouze do maximálního rozlišení snímače pořád dokola. V tomto režimu je maximální hodnota uložena a je možné ji při pozastaveném zobrazení odečíst. Pokud takto čítáme z neznámého snímače v režimu 1x, který je vybaven I kanálem a přičteme li k čítané hodnotě jedna udává nám tak rozlišení snímače. Zobrazení čítané hodnoty se dá také pozastavit, aby se hodnota při rychlých otáčkách dala odečíst. Z tohoto režimu lze spustit odesílání po sériové lince, kdy modul odesílá čítané údaje při každé změně čítané hodnoty.



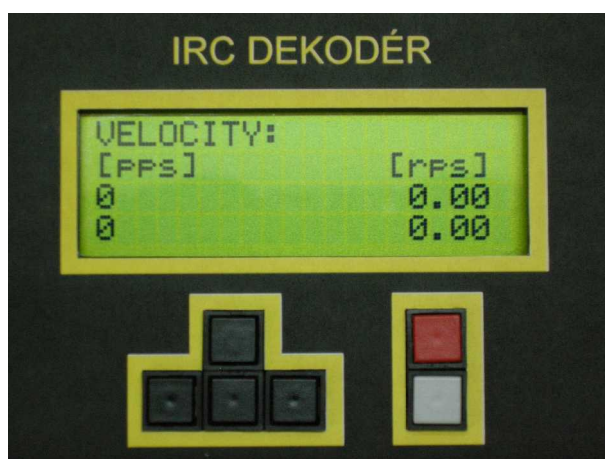
obr. 31 Fotografie modulu v režimu čítače

Na obr. 31 je fotografie modulu v režimu čítače. Symbol I y v horním rohu displeje symbolizuje zapnutý I-kanál pro osu y a symbol Dxy předdělení čítaných hodnot z obou os. Nápis USART znamená, že modul zobrazované hodnoty také posílá po sériové lince.

5.1.3 Režim měření rychlosti otáček

V tomto režimu modul počítá rychlost otáčení snímače v jednotkách rps (rotation per second), tedy v obrátkách za sekundu. Zobrazuje také hodnotu pps (pulse per second), počet pulzů za sekundu. Pro správné měření rychlosti otáček za sekundu je třeba v režimu nastavení nastavit přesnost snímačů.

Výpočet rychlosti je prováděn každých 250ms. Měření probíhá v režimu dekódování 1x.



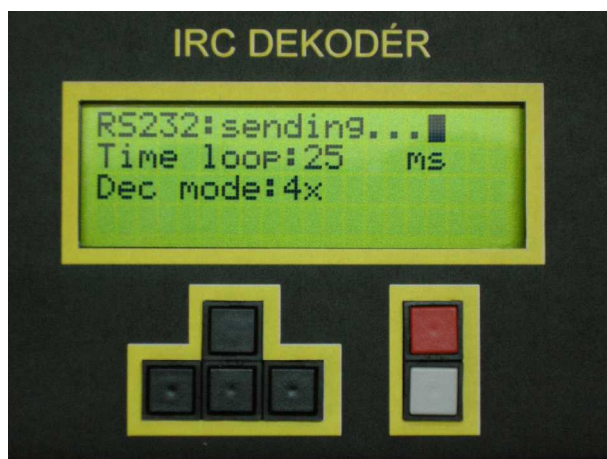
obr. 32 Fotografie modulu v režimu měření rychlosti

5.1.4 Režim odesílání

V tomto režimu modul čítá hodnoty z obou snímačů a v pravidelných intervalech je posílá na sériovou linku. V nastavení tohoto režimu se dají tyto intervaly volit v rozmezí od 10ms do 500ms po kroku 5ms.

Parametry sériového přenosu jsou:

- asynchronní přenos
- žádný stop bit
- žádná parita
- rychlost komunikace v baudech lze nastavit v nastavení modulu



obr. 33 Fotografie modulu v režimu odesílání.

Modul odesílá každých 25ms hodnoty po sériové lince

6 ZÁVĚR

K vytvoření modulu pro zpracování signálu z inkrementálního snímače bylo nutno se seznámit s principem inkrementálních snímačů. Pochopit problematiku zpracování jejich signálů a navrhnout vhodné řešení pro modul. Při návrhu řešení bylo nutné zvážit možnosti jak lze signál dekodovat a nalézt tak vhodný hardware pro realizaci. Zajistit aby zpracovaný signál mohl být využitý pro další zpracování, popřípadě aby mohly být zpracované údaje zobrazovány. Pro realizaci modulu bylo třeba získat potřebné znalosti a zkušenosti se sestavováním elektronických obvodů a jejich navrhování. Dále se seznámit s problematikou programování jednočipových mikrokontrolérů, speciálně pak v jazyce C.

Dekodování signálu z inkrementálního snímače s použitím kvadrurního dekodéru HCTL-2032 se ukázalo jako velice spolehlivé. Dekodér přesně a bez chyby dekoduje signál i při vyšších otáčkách. K modulu byl připojen motor, který se otáčel rychlostí 6700ot./min. Zapojení dekodéru do obvodu není složité a je k tomu zapotřebí minimum součástek. Nevýhodou je snad jen to že zdroj hodinového signálu musí být generován externím oscilátorem, který je třeba pro běh dekodéru buďto sestavit, nebo použít jako hotovou součástku. Vhodnější by bylo, kdyby měl dekodér oscilátor integrován v sobě a řídil by se pouze připojeným krystalem, jak je tomu například u jednočipových mikroprocesorů.

Programování řídicího mikrokontroléru za pomoci jazyka C bylo využito s výhodou. Bylo třeba napsat několik rutin pro obsluhu displeje, sériové linky a v neposlední řadě obsluhu dekodéru. Program se tak stal poměrně rozsáhlý a bylo třeba jej rozdělit do více souborů kvůli přehlednosti. Vznikly tak zvlášť funkce pro obsluhu dekodéru a zvlášť pro obsluhu displeje, které při správném nadefinování vstupů a výstupů mohou být použity i pro jiné aplikace.

Modul se podařilo realizovat a otestovat na laboratorních motorech a snímačích.

7 SEZNAM POUŽITÉ LITERATURY

- [1] HCTL-2032. *Datasheet* [online]. 2008 [cit. 2009-05-17].
Dostupný z WWW: <<http://www.avagotech.com/docs/AV02-0096EN.pdf>>.
- [2] ATmega128. *Datasheet* [online]. 2008 [cit. 2009-05-17].
Dostupný z WWW: <http://www.atmel.com/dyn/products/datasheets.asp?family_id=607#760>.
- [3] 7805DT. *Datasheet* [online]. 2008 [cit. 2009-05-17].
Dostupný z WWW: <http://www.gme.cz/_dokumentace/dokumenty/934/934-051/dsh.934-051.1.pdf>.
- [4] Ovládání znakových LCD s řadičem HD44780. *Návod* [online]. 2007 [cit. 2009-05-17].
Dostupný z WWW: <<http://www.elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatice.html>>.
- [5] MAX232. *Datasheet* [online]. 2008 [cit. 2009-05-17].
Dostupný z WWW: <<http://focus.ti.com/lit/ds/symlink/max232.pdf>>.
- [6] RS232. *Návod* [online]. 2005 [cit. 2009-05-17].
Dostupný z WWW: <<http://hw.cz/rs-232#urovne>>.
- [7] EAGLE. *Česká stránka programu* [online]. 2009 [cit. 2009-05-17].
Dostupný z WWW: <<http://www.eagle.cz/>>.
- [8] WinAVR. *WinAVR-home page* [online]. 2009 [cit. 2009-05-17].
Dostupný z WWW: <<http://winavr.sourceforge.net/>>.
- [9] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR*. [s.l.] : [s.n.], 2003. s. 121.
- [10] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR*. [s.l.] : [s.n.], 2003. s. 215.
- [11] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR*. [s.l.] : [s.n.], 2003. s. 193.
- [12] MAZANEC. *Princip rotačního optického enkodéru pro měření natočení* [online].
[cit. 2009-05-17]. Dostupný z WWW:
<http://jonatan.spse.pilsedu.cz/~mazanec/opticke_ekodery.htm>.
- [13] MAZANEC. *Vyhodnocení signálu z inkrementálního enkodéru* [online].
[cit. 2009-05-17]. Dostupný z WWW:
<http://jonatan.spse.pilsedu.cz/~mazanec/opticke_ekodery.htm>.

SEZNAM PŘÍLOH:

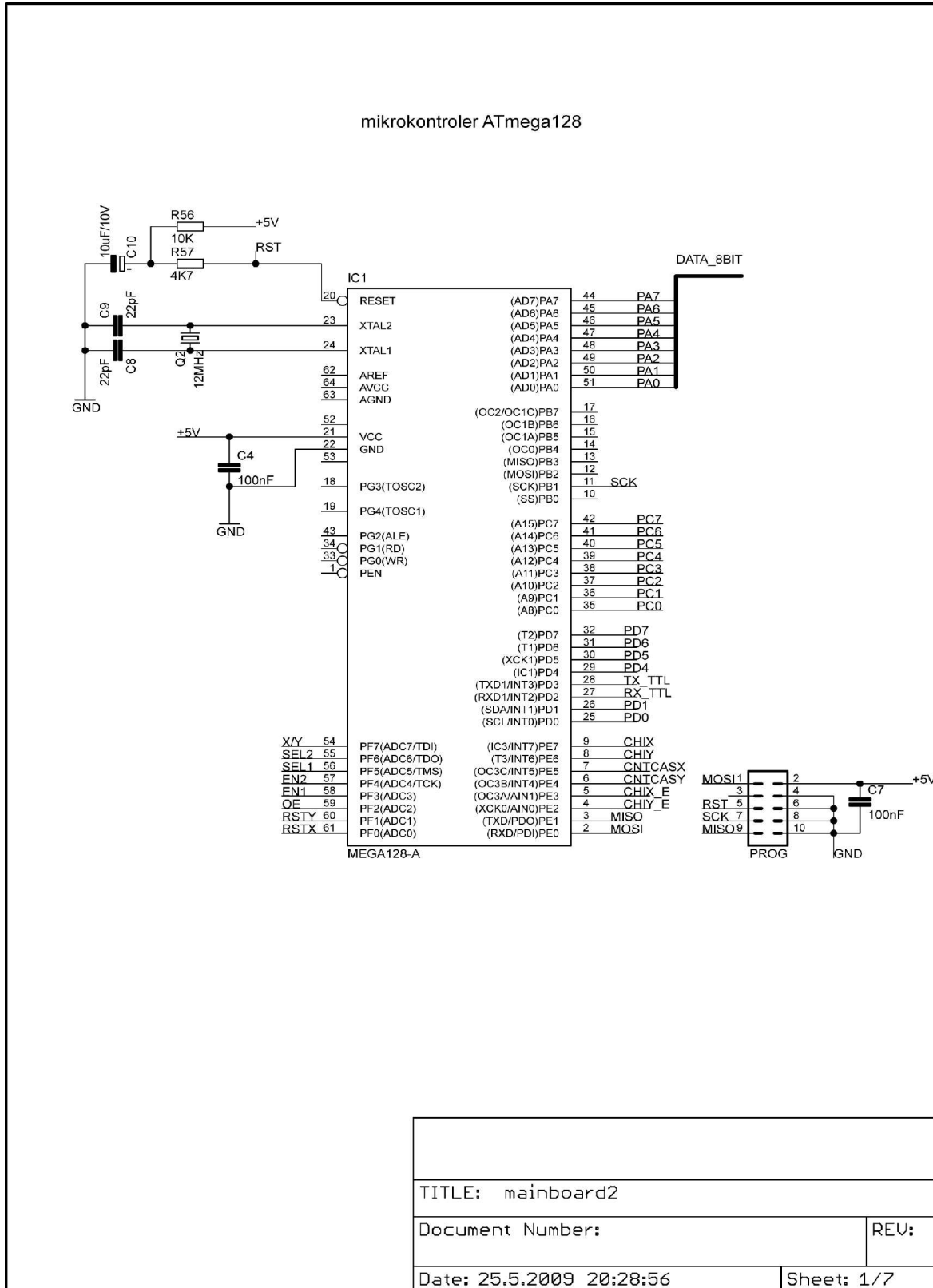
- Příloha 1: Seznam součástí
- Příloha 2: Schémata modulu
- Příloha 3: Desky plošných spojů modulu
- Příloha 4: Návod k ovládní modulu

PŘÍLOHY:

Příloha 1:

Ks	Hodnota	Název součástky	Součástka
1		M09H	RS232
4		ML10	IRCDEC, IRCX, IRCY, PROG
1		PINHD-1X8	KLAVESNICE
1		QO32MHZ	SG51
1	1N4004	1N4004	D1
4	1u	CPOL-EUA/3216-18R	C11, C12, C13, C14
1	2K5	TRIM_EU-CA6V	R4
6	2K7	R-EU_R0805	R5, R6, R7, R8, R9, R10
3	4K7	R-EU_R0805	R1, R12, R57
1	4K7	R-EU_R1206	R11
1	4R7	R-EU_0207/10	R2
1	8K2	R-EU_R0805	R3
1	10K	R-EU_R0805	R56
1	10uF/10V	CPOL-EUA/3216-18R	C10
1	12MHz	CRYTALHC49S	Q2
2	22pF	C-EUC0805	C8, C9
7	100nF	C-EUC0805	C1, C2, C3, C4, C5, C6, C7
1	330uF/16	CPOL-EUE3.5-10	C15
1	470uF/10	CPOL-EUE3.5-10	C16
1	7805DT	7805DT	IC2
3	BC847	BC847	T1, T2, T3
1	HCTL-2032	HCTL-2032	IC4
1		LCD-MC2004E-SYL	LCD1
1	MAX232ECWE	MAX232ECWE	IC3
1	MEGA128-A	MEGA128-A	IC1
1	NEB21R	NEB21R	X4
1		PINHD-1X8	JP2
6	P-ST1034	P-ST1034	CANCEL1, DOWN1, ENTER1, LEFT1, RIGHT1, UP1
1		KP-32	krabice

Příloha 2:



TITLE: mainboard2

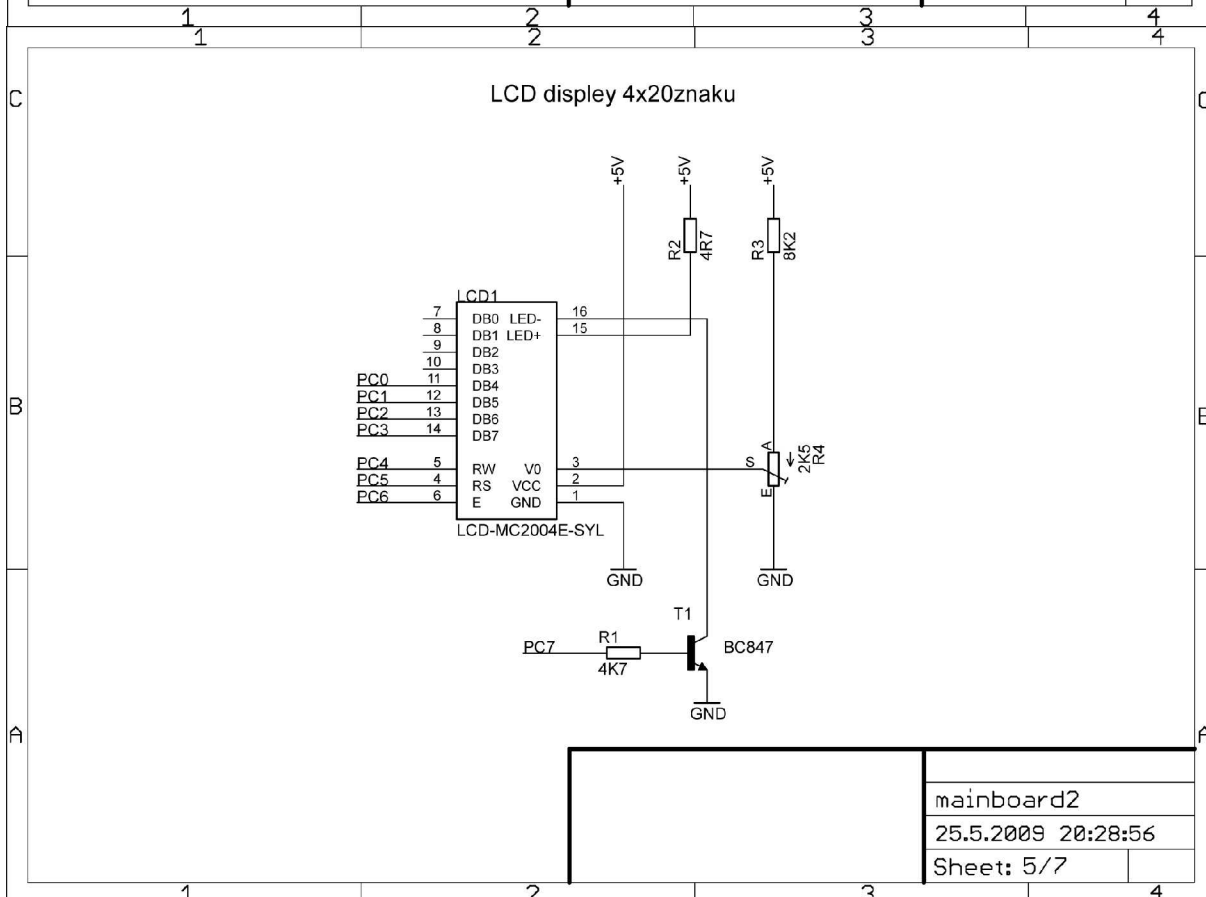
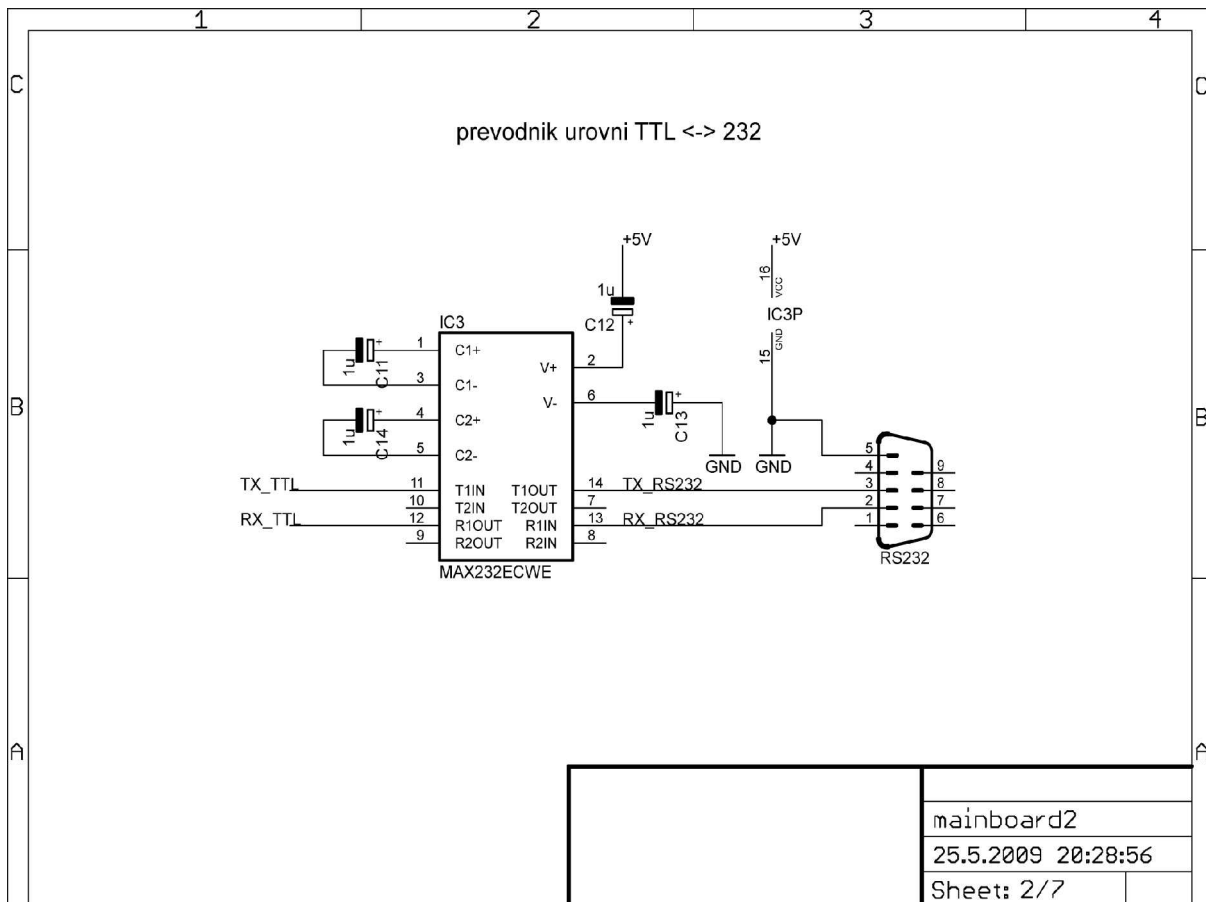
Document Number:

REV:

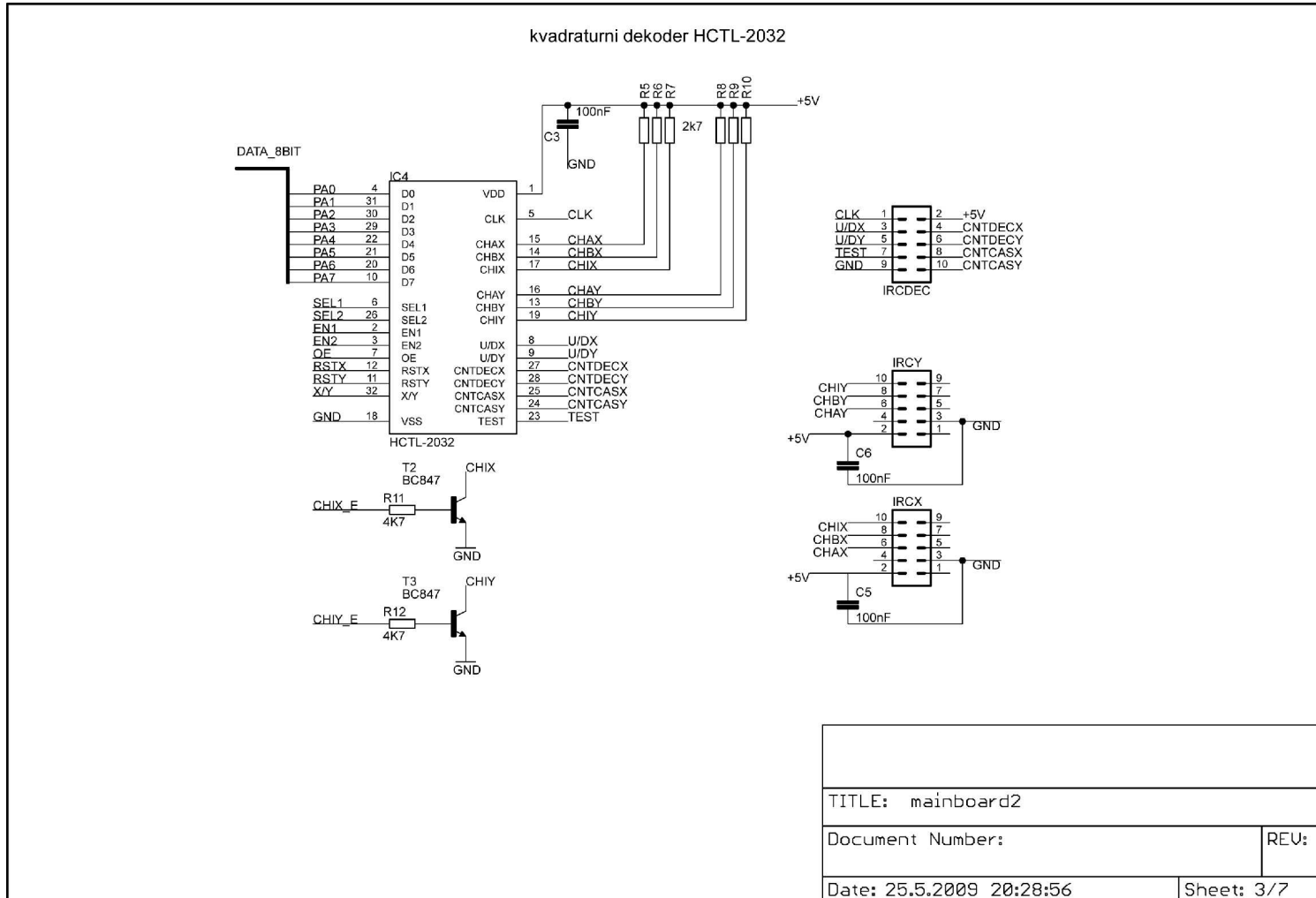
Date: 25.5.2009 20:28:56

Sheet: 1/7

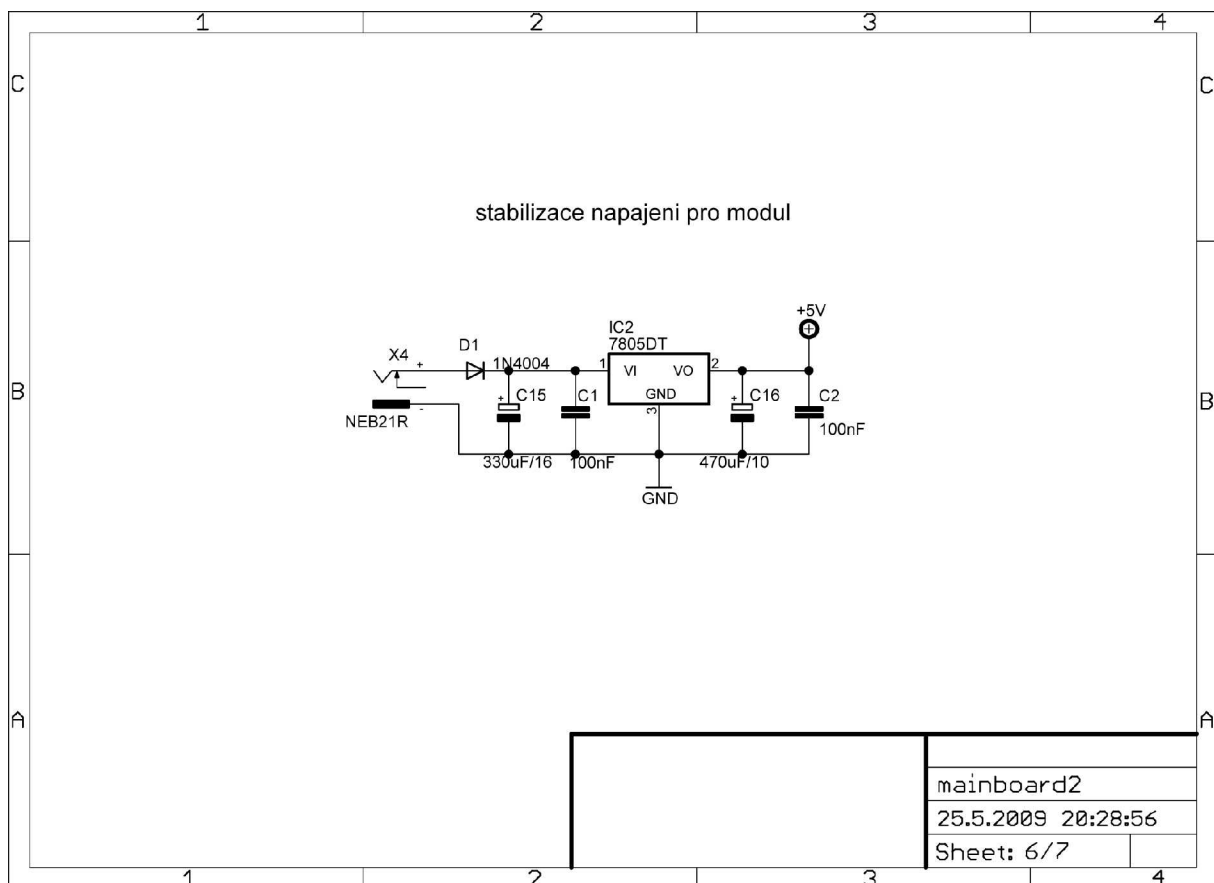
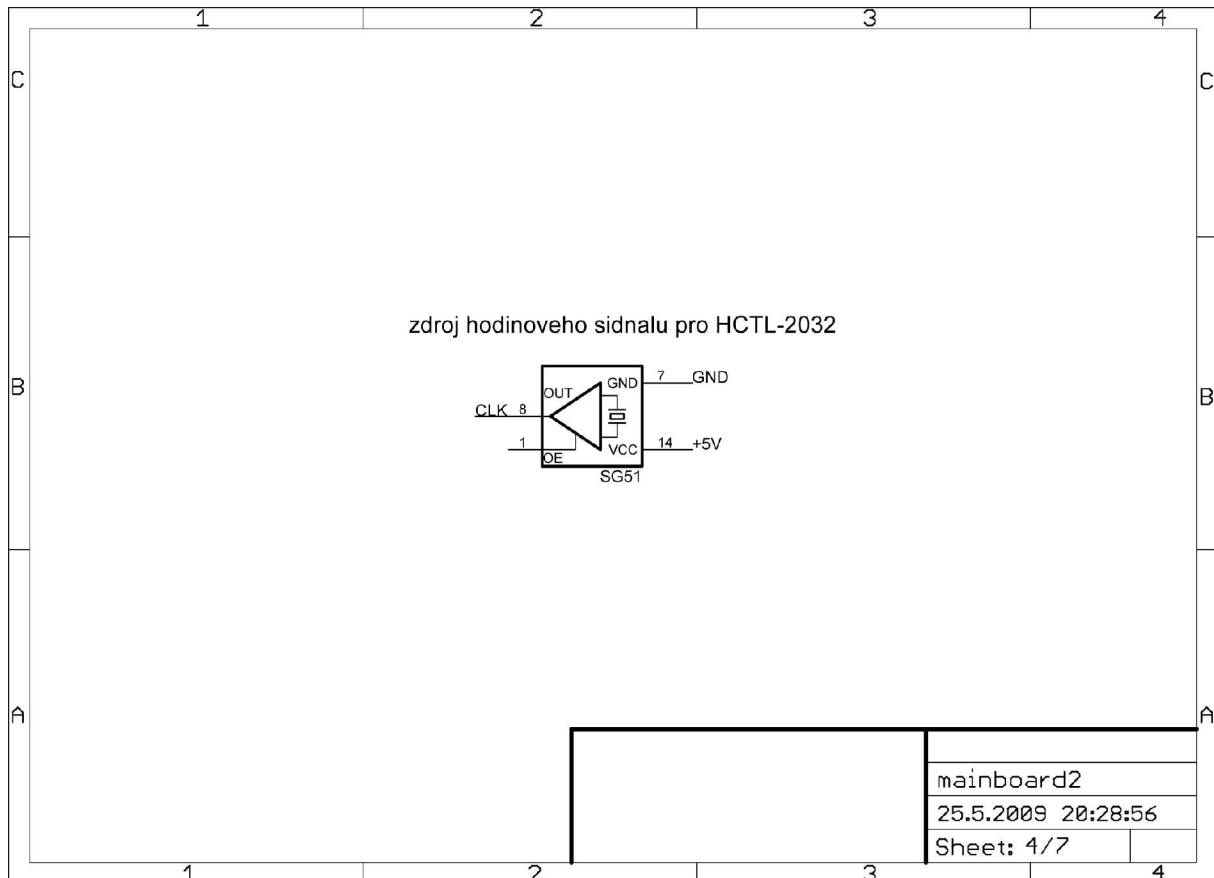
Příloha 2:



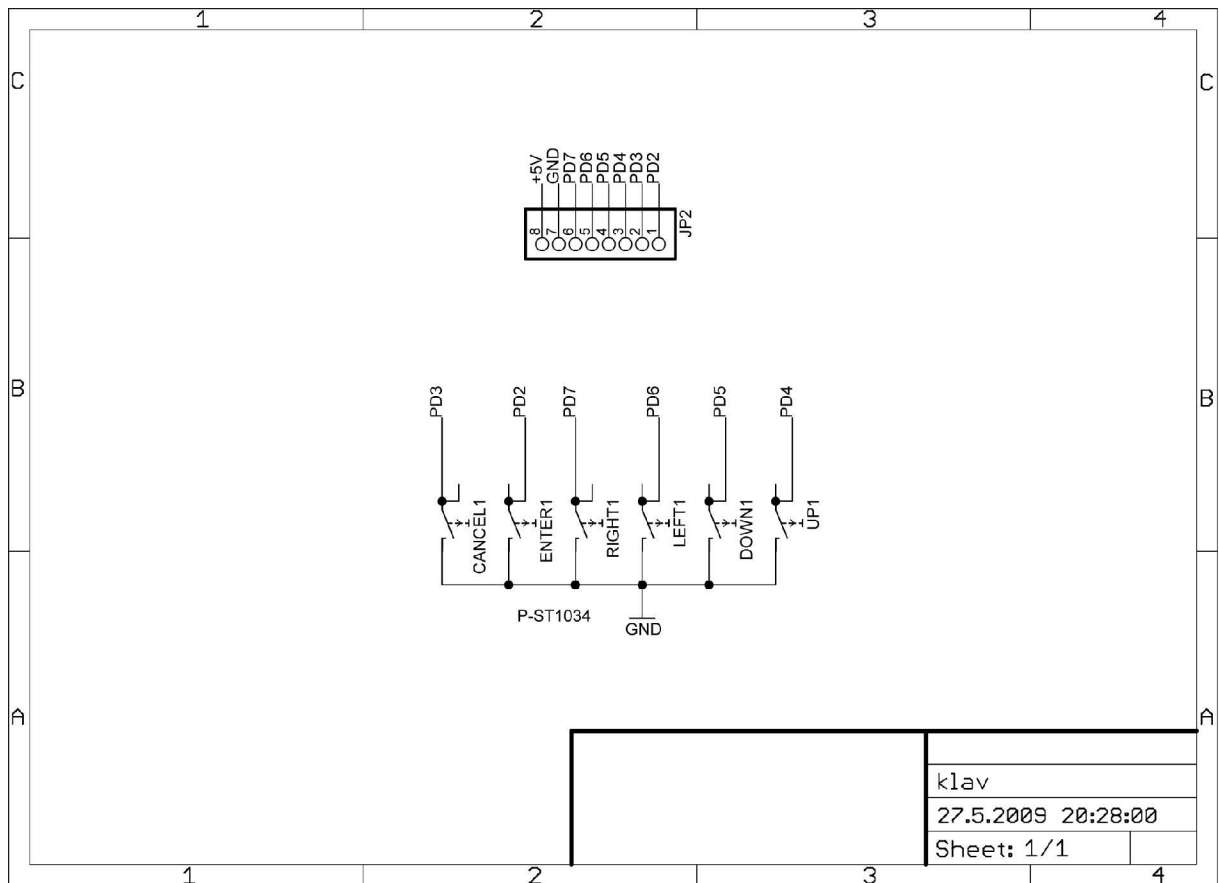
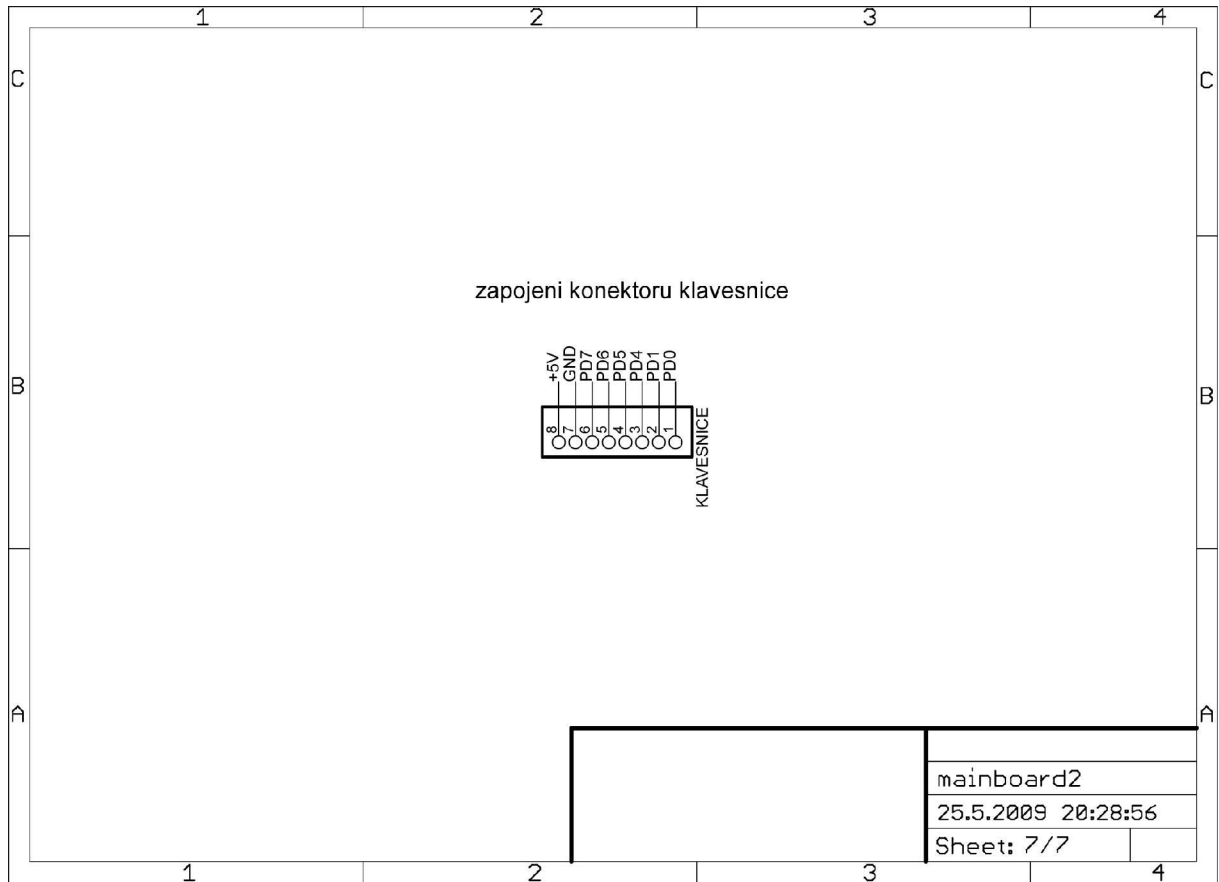
Příloha 2:



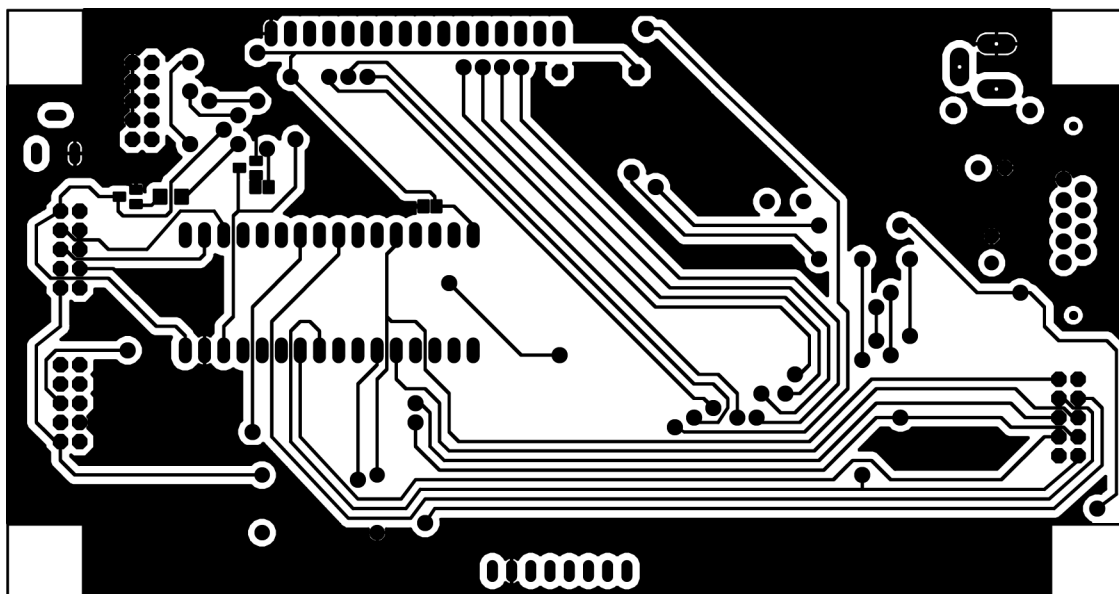
Příloha 2:



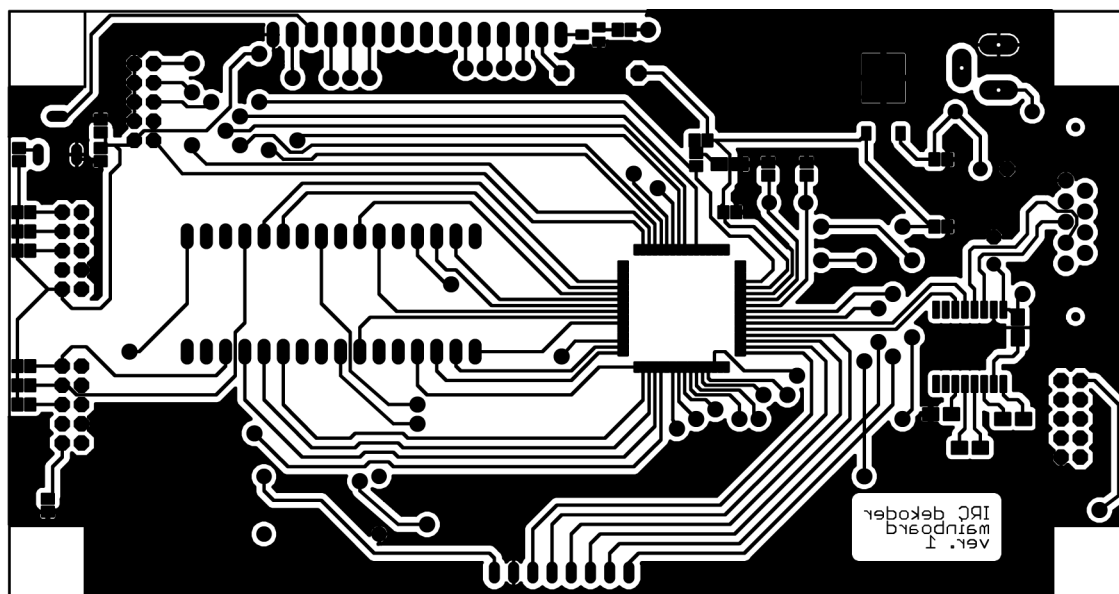
Příloha 2:



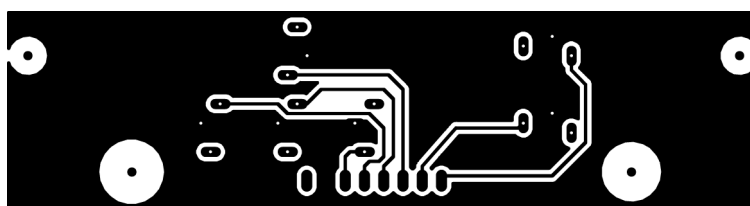
Příloha 3:



Vrchní strana hlavní desky modulu



Spodní strana hlavní desky modulu

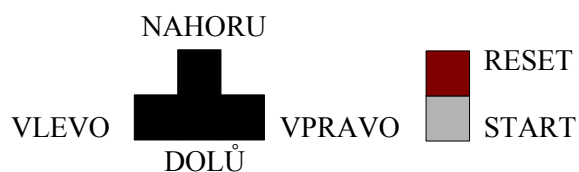


deska klávesnice modulu

Příloha 4:

Ovládání:

Modul se ovládá pomocí šesti tlačítek. Čtyři černá tlačítka tvoří směrový kříž, červené tlačítko RESET a šedé START.



MENU:

Modul obsahuje jednoduché menu ve kterém jsou na řádcích zobrazovány jeho jednotlivé položky. V položkách menu se prochází za pomoci blikajícího kurzoru, který se přesouvá po položkách pomocí kláves NAHORU a DOLŮ. Položky se pak volí klávesou VPRAVO.

Položky menu:

- COUNTER – režim čítání inkrementů
- TACHOMETR – režim měření rychlosti otáček
- RS232 SENDING LOOP – režim odesílání na sériovou linku
- SETTINGS – menu nastavení

Položky settings:

- PRESCALE – nastavení hodnot předděličky pro režim COUNTER
- TACHOMETR – nastavení hodnot pro měření rychlosti otáček
- USART – nastavení parametrů sériové linky a odesílací smyčky

COUNTER:

V tomto režimu jsou na displeji zobrazovány čítané hodnoty z obou os x,y. Okamžitě při přechodu do tohoto režimu modul čítá, a na horním řádku displeje je zobrazen nápis progress.... Na druhém řádku vlevo je naznačen režim čítání a uprostřed nápis dec značící soustavu ve které jsou zobrazovány číselné hodnoty. V tomto režimu se dá klávesou START přecházet z režimu progress... do režimu Freezed, kdy je pozastaveno zobrazování čítaných hodnot, ale modul čítá dál. Vpravo na displeji je indikován směr otáčení. Znaménko + a – před zobrazovanými hodnotami indikují přetečení vnitřních čítačů.

ovládání v režimu progress...

- NAHORU
Zapíná nebo vypíná předdělení hodnot z os x,y. V pravém horním rohu je předdělení indikováno zobrazením písmena „D“ a písmenem příslušné osy. Hodnota dělitele se nastavuje v SETTINGS → PRESCALE
- DOLŮ
Přepíná mezi režimy dekodování 1x, 2x, 4x a zobrazuje je vlevo na druhém řádku.
- VLEVO
Návrat do hlavního menu.
- VPRAVO
Přepíná soustavu, ve které jsou čítané hodnoty zobrazovány.
dec = desítková, hex = hexadecimální
- RESET
Vynuluje čítače obou os.
- START
Přepíná mezi režimy progress... a Freezed.

ovládání v režimu progress...

- NAHORU
Zapíná nebo vypíná posílání zobrazovaných hodnot na sériovou linku.
- DOLŮ
Aktivuje I kanály snímačů. V pravém horním rohu je zobrazeno Písmeno „I“ a písmeno příslušné osy.
- VLEVO
Pokud je aktivní I kanál, přepíná mezi čítanou hodnotou a maximální uloženou načítanou hodnotou.
- VPRAVO
Přepíná soustavu ve které jsou čítané hodnoty zobrazovány. dec = desítková, hex = hexadecimální
- RESET
Pokud je aktivní I kanál a zobrazená maximální hodnota, nuluje ji.

- **START**
Přepíná mezi režimy progress... a Freezed.

Pozn.: Pokud je aktivní I kanál u některé z os, čítač čítá hodnoty do doby, kdy přijde signál z I kanálu. Tím jsou čítače modulu vynulovány a čítání probíhá od nuly. Modul ukládá maximální načtenou hodnotu než se čítače vynulují signálem na I kanálu. U příslušné osy je zobrazeno písmeno „M“.

TACHOMETR:

Při přechodu tohoto režimu modul stále měří rychlost otáčení obou se. K tomuto měření je třeba znát přesnost připojených snímačů a nastavit ji v **SETTINGS** → **TACHOMETR**. Stiskem tlačítka **VLEVO** se vrátí do menu.

RS232 SENDING LOOP:

Modul čítá hodnoty z obou os a posílá je v pravidelných intervalech na sériovou linku. Odesílání se spouští a zastavuje tlačítkem **START**. Při zastaveném odesílání je možno klávesami **NAHORU** a **DOLŮ** měnit interval odesílání, tlačítkem **DOPRAVA** režim čítání a tlačítkem **RESET** nulovat čítání.. Tlačítkem **VLEVO** se lze vrátit do menu pouze při zastaveném čítání.

NASTAVENÍ ČÍSELNÝCH HODNOT:

Nastavování číselných hodnot se týká položek menu **SETTINGS** → **PRESCALE**, **SETTINGS** → **TACHOMETR** a **SETTINGS** → **USART**. Tlačítka **NAHORU**, **DOLŮ** se volí číslice a tlačítka **VLEVO**, **VPRAVO** pozice číslice v čísle. Tlačítkem **START** se potvrzuje zadání a přejde na další řádek. Tlačítkem **RESET** lze hodnotu vynulovat. Po zadání obou hodnot se modul zeptá jestli chcete zadání uložit. Tlačítkem **START** se hodnoty uloží do vnitřní paměti. Tlačítkem **RESET** se zadání neuloží a vrátí se k původním hodnotám. V menu **SETTINGS** → **USART** je ovládání stejné jen hodnoty nelze zadávat pomocí číslic, ale pouze tlačítka **NAHORU**, **DOLŮ** volit z přednastavených hodnot.