

Středoškolská odborná činnost 2006/2007

Obor 9 – Strojírenství, hutnictví,
doprava a průmyslový design

RoboCraner²

Principy mechatroniky

Autoři:

**Kubička Matěj
Šimána František**

Konzultant práce:

Ing. Petr Hlávka

Plzeň, 2007
Plzeňský kraj

RoboCraner²

Principy mechatroniky

- *Scientia est potestias* -
- *Ve věděni je síla* -

Kubička Matěj
Šimána František

2007

Tento projekt by nebylo možné dokončit bez podpory Ing. Petra Hlávky a Ing. Petry Maškové a jejich rad v oblasti strojírenství a elektroniky, firmy Elfetex, která nám zdarma dodala některý elektromateriál, firmy Christ Car Wash, která nám darovala šnekovou převodovku a firmy SEG, díky tomu, že vyrobila některé strojní součásti.

Prohlašujeme tímto, že jsme soutěžní práci vypracovali samostatně pod vedením Ing. Petr Hlávky a uvedli v seznamu literatury veškerou použitou literaturu a další informační zdroje včetně internetu.

V Plzni dne: pondělí 5. března 2007

Matěj Kubička

František Šimána

vlastnoruční podpis autorů

Obsah

Obsah.....	5
Anotace.....	8
1. Úvod.....	9
2. Stručný popis manipulátoru.....	10
3. Strojírenská řešení.....	11
3.1. Popis konstrukce.....	11
3.2. Svislý posuv elektromagnetu.....	11
3.3. Otáčení konstrukce manipulátoru.....	12
3.3.1. Uložení motoru se šnekem.....	12
3.4. Posuv jeřábové kočky.....	13
4. Elektronická řešení.....	14
4.1. Sériová sběrnice I2C (TWI).....	15
4.2 Topologie připojení k počítači PC.....	15
4.3. Práce s maticovou klávesnicí.....	16
4.4. Práce s alfanumerickým displejem.....	17
4.5 Principy řízení otáček motorů.....	18
4.5.1. Pulzně-šířková modulace (PWM).....	18
4.5.2 Pulzní proudový zdroj – chopper.....	19
4.6 Elektromagnet a jeřábní světlo.....	20
4.7 Popis motorů.....	21
4.7.1 Hybridní krokový motor.....	21
4.7.2 Stejnoseměrný motor.....	22
4.8. Elektronické obvody.....	22
4.8.1. Nižší úroveň řízení.....	22
4.8.2. Buzení stejnosměrných motorů.....	23
4.8.3. Buzení krokového motoru.....	24
4.8.4. Rozhraní maticové klávesnice.....	24
4.8.5. Rozhraní LCD displeje.....	25

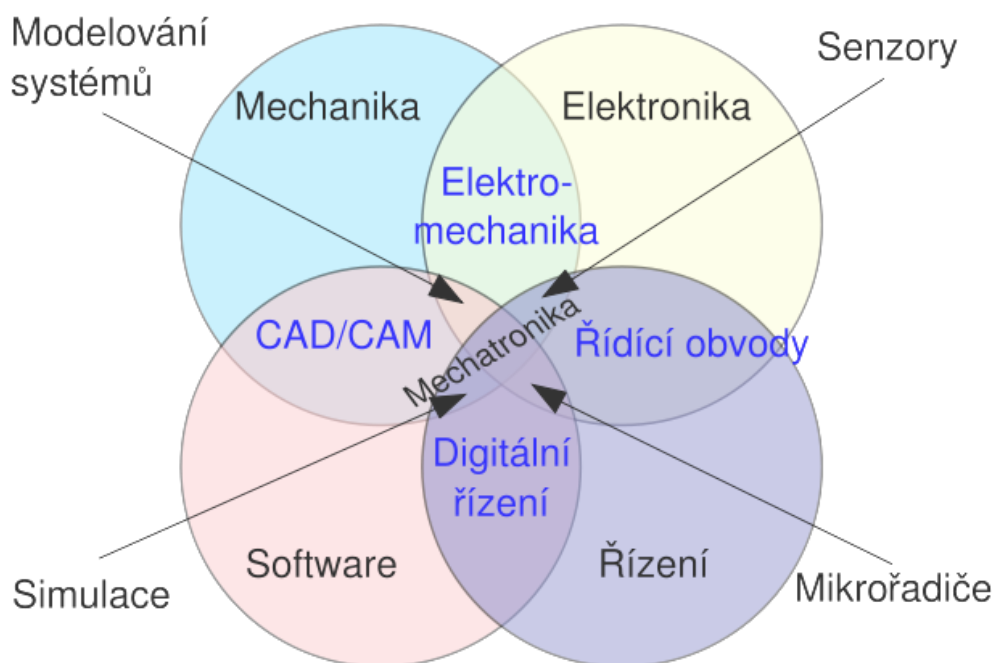
4.8.6. Převodník dat RS-232/I2C.....	26
4.8.7. Převodník RS232C - TTL/CMOS	26
5. Popis softwaru mikrokontrolérů.....	28
5.1. Použité programovací jazyky.....	28
5.1.1. Assembler.....	28
5.1.2. ANSI C.....	28
5.2. Standard RS-232.....	29
5.3. Sběrnice I2C (TWI).....	29
5.3.1. Adresování a přenášení dat na sběrnici.....	30
5.3.2. Módy přenosů.....	30
5.3.3. Používání TWI.....	30
5.3.3.1. Řídící registry TWI.....	31
5.3.3.2. Inicializace master zařízení.....	31
5.3.3.3 Inicializace slave zařízení.....	32
5.3.3.4. Příjem a odesílání dat.....	33
5.4. Komunikační protokoly.....	34
5.4.1. Způsoby přenosu dat.....	34
5.4.2 Adresy zařízení připojených do sběrnice I2C.....	34
5.4.3. Protokol nižší úrovně řízení.....	34
5.4.4. Protokol pro komunikaci s displejem.....	35
5.5. Vnitřní systém manipulátoru.....	36
5.6. Generování PWM signálů.....	36
5.7. Komunikace s řadičem HD44780.....	37
5.8. Čtení stavů klávesnice.....	38
6. Výsledky.....	40
7. Závěr a diskuze.....	41
Příloha A – Softwarové licence.....	42
Příloha B – Slovník použitých výrazů.....	43
Příloha C - Literatura.....	44

Anotace

RoboCraner vznikl jako projekt, který má demonstrovat principy sdružené v moderním oboru zvaném mechatronika. V druhém ročníku jsme navrhli na principu jeřábu první verzi RoboCraneru, která odrážela naše možnosti. RoboCraner druhé verze je po technologické, elektronické a softwarové stránce zcela přepracován a zdokonalen. Poučili jsme se z chyb první verze.

RoboCraner je otočný jeřáb stavebního typu, tzn. na svislé ose vysoké 45 cm je kolmé rameno o délce 53 cm, na kterém se posunuje jeřábová kočka, z které se odvíjí lano s elektromagnetem. Celá konstrukce se otáčí kolem své osy v úhlu 300 stupňů. Pohyb zajišťují dva stejnosměrné a jeden krokový motor. Řídící elektronika na bázi mikrokontrolérů ATMEEL plní naprogramované funkce dle požadavků, které přichází z ovládacích vstupů po sběrnicích RS232C a I²C. Konstrukci je možno ovládat ze tří míst – z počítače PC, programovatelného automatu PLC a vlastního zabudovaného ovládání.

Konstrukce demonstruje použití některých průmyslových integrovaných obvodů, které výkonově řídí a budí stejnosměrné motory pomocí PWM a krokový motor pomocí technologie pulzního proudového zdroje „chopperu“. Mikrokontroléry ATMEEL, modelové řady MegaAVR, demonstrují možnosti programování ve dvou hlavních jazycích pro mikrokontroléry - Assembleru a ANSI C.



Motto: *Scientia est potestias*

Ve věděni je síla

1. Úvod

- Motivace
 - Účel této práce
 - Souhrn zjištěných faktů na poli mechatroniky
-

V dnešní době počítačů a digitální elektroniky, je naše generace velmi silně vystavena jejím vlivům. Mnoho našich vrstevníků umí programovat, opravovat počítače, či má vlastní internetové stránky.

Naším cílem je prezentace využití mechatronických principů zejména z elektronické a částečně strojírenské části (například převody A/D, D/A, P/S, S/P, ale i také PWM, USART (RS-232), TWI nebo ze strojírenské části samosvorný šnekový převod na otáčení konstrukce, či uložení motoru se šnekem). Toto mechatronické zařízení budeme dále prezentovat na soutěžích, případně jej naše škola využije jako prezentativní subjekt na Dnu otevřených dveří.

Na poli mechatroniky (robotiky) bylo již zjištěno mnoho a přitom stále málo. V 18. stol. jsou známé mechanické napodobeniny člověka – androidy, švýcarských mistrů Pierra a Henryho Drozů. Jejich automat – písař, byl schopen psát perem několik vět a velmi dobře napodoboval člověka. V roce 1949 byl zahájen výzkum číslicově řízených strojů, roku 1961 byl dán do provozu první průmyslový robot UNIMATE od firmy General Motors. V r. 1964 byly otevřeny laboratoře umělé inteligence na Massachusetts Institute of Technology (M.I.T.). V poslední době je nasazení robotů v průmyslové praxi stále vyšší (v roce 2003 se jedná o cca 90000 prům. robotů, na rozdíl od roku 1998, kdy jich bylo „jen“ 67000).

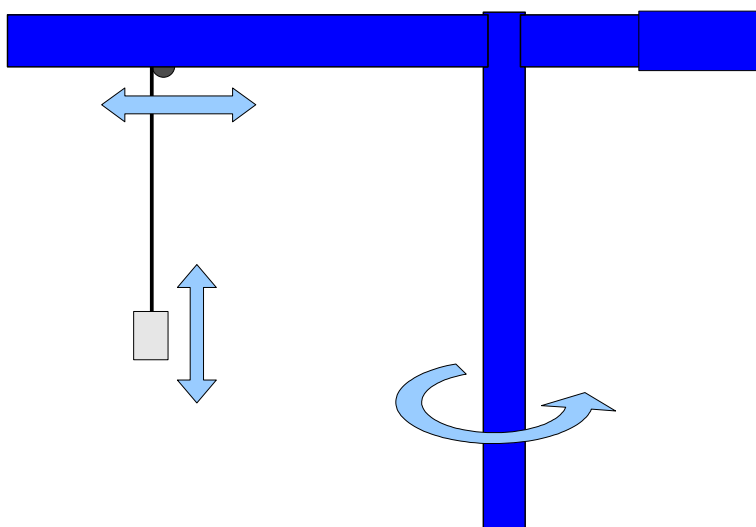
Naší práci jsme na internetu vytvořili vlastní stránky na doméně III. řádu nazvané <http://robocraner.digitalbrain.cz>, na této adrese můžete nalézt tuto dokumentaci, další fotografie, atd..



2. Stručný popis manipulátoru

- Stručný popis manipulátoru
- Efektory a sensorika
- Způsoby ovládání manipulátoru

Manipulátor je „jeřábového“ vzhledu. Jeřábovým vzhledem je myšlena konstrukce nosného těla a ramene, která se otáčí v rozsahu 300°. Stejně jako každý jiný jeřáb, i tento je schopen přenést předmět z feromagnetického materiálu z místa A do místa B. Předmět je přidržován pomocí magnetického pole cívky. Cívkou vertikálně pohybuje naviják ovládaný krokovým motorem. Další pohyby zajišťují dva stejnosměrné komutátorové motorky. Manipulátor ovládáme z počítače PC. Jeho rozhraní představuje aplikaci, napsanou v jazyce C#. Manipulátor je připojen k počítači přes sériovou linku RS-232C.



Akčními členy (efektory) jsou dva stejnosměrné komutátorové motorky a jeden hybridní krokový motor. Výkon stejnosměrných motorků regulujeme pomocí 8-bitové pulzně šířkové modulace (PWM). Krokový motor řídíme bipolárně pomocí dvou „H“ můstků, se zabudovaným proudovým zdrojem typu „chopper“. Dva senzory hlídají koncové polohy jeřábové kočky, a jeden senzor detekuje výchozí pozici šnekového kola.

3. Strojírenská řešení

- Shrnutí nejzajímavějších řešení na konstrukci manipulátoru
- Popis konstrukce
- Šnekový převod

3.1. Popis konstrukce

Na dřevotřískovou podkladovou desku 80x80 cm je přes podložky přišroubována závitová tyč M12, která je osou konstrukce jeřábu. Na této závitové tyči jsou umístěna dvě kuličková ložiska zajištěná kontramatkami, na která je nasazena ocelová trubka a vedle ní spojená další menší trubka z leštěného duralu pro kabeláž. Konstrukce se otáčí v úhlu 300° (60° tvoří tzv. „nulový úhel“). Ve spodní části je na ocelové trubce šnekový převod o poměru 1/60 (otáčející celou konstrukci). Na šnekovém kole je připevněn držák krokového motoru s navijákem pro vertikální posun elektromagnetu a případného nákladu (krokový motor má i dostatečný přídržný moment, takže není problém držet těleso tzv. „ve vzduchu“). Šnekový převod je otáčen stejnosměrným komutátorovým motorem s maximální rychlostí 297 ot/min

Na svislé ocelové trubce, která tvoří tělo jeřábu je ve výšce 45 cm umístěno rameno dlouhé 50 cm, které je tvořeno z jeklu 40x20 mm. Jeřábová kočka se uvnitř ramene se pohybuje v rozmezí 380 mm, tato vzdálenost je omezena dvěma taktilními senzory. Pohyb kočky zajišťuje závitová tyč M8, která tvoří převod pohybu otáčivého na posuvný. Otáčivý pohyb zajišťuje stejnosměrný motor.

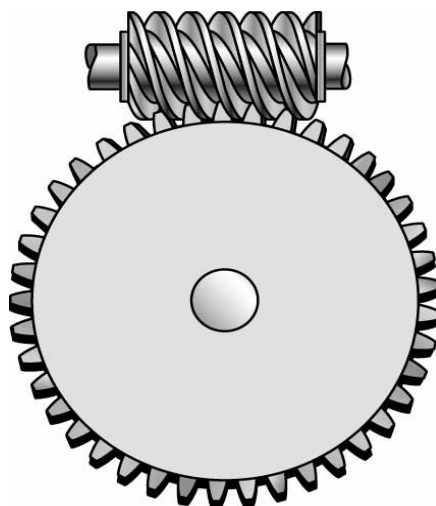
Manipulaci s nákladem zajišťuje elektromagnet, který je zavěšen přímo na dvou vodičovým napájecím kabelem, který se navíjí na navijáku.

3.2. Svislý posuv elektromagnetu

Z důvodu použití elektromagnetu jako háku, je přivedeno napájení pomocí kabelu, který je zároveň navíjecím lanem. Kabel je plochá dvojlinka 2 x 0,8 s maximální proudovou zatížitelností 5A. S ohledem na tvar kabelu je použit speciální naviják, který má optimální vnitřní šířku pro použitý vodič. Naviják tvoří dvě části, které do sebe zapadají, sešroubovávají se k sobě čtyřmi šrouby M3. Výhodou navijáku je velký průměr největších závitů, zajišťujících vysokou obvodovou rychlost navíjení a odvíjení kabelu (Obvod posledního a největšího závitu je roven 150mm, tím tvoří 1/3 rozsahu posunu. Naviják potřebuje přibližně 3 otáčky pro navinutí resp. odvinutí v maximálním rozsahu). Kabel spojuje elektromagnet s elektronickým výkonovým zesilovačem

3.3. Otáčení konstrukce manipulátoru

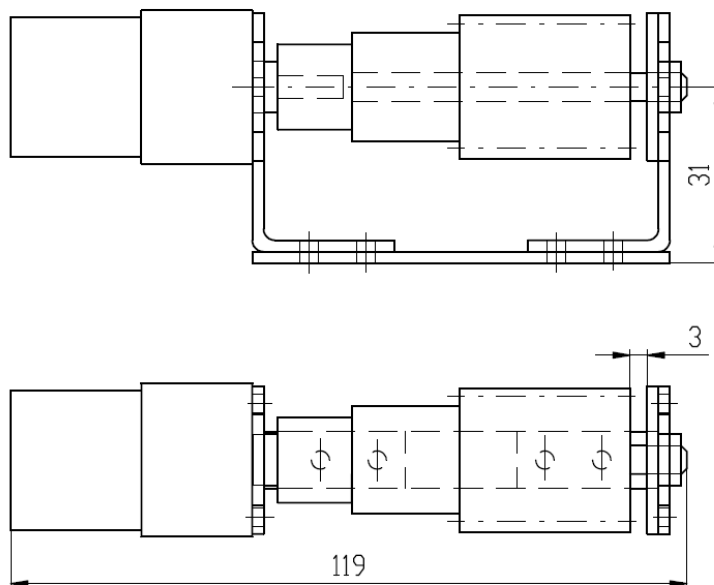
Použili jsme šnekovou převodovku, kterou nám zdarma dodala firma Christ Car Wash. Jedná se o převod, při kterém jsou osy otáčení na sebe kolmé a zuby hnaného kola zapadají do závitů hnací části. Tento převod je samosvorný, toho jsme s výhodou využili pro zamezení samovolného, nebo násilného otáčení konstrukce.



Academy Artworks

3.3.1. Uložení motoru se šnekem

Šneková převodovka je poháněna motorem, který je konstrukčně menší a lehčí než samotný šnek. Z důvodu, aby nedocházelo k přetěžování motoru na jeho ose otáčení, je motor se šnekem uchycen v uložení s kluzným ložiskem. Uchycení je realizováno dvěma „L” profily proti sobě. Z jedné strany je motor, připevněný na trn, který prochází do kluzného ložiska v druhém držáku (viz obrázek). Použitý motor je stejnosměrný komutátorový s 290ti ot/min.



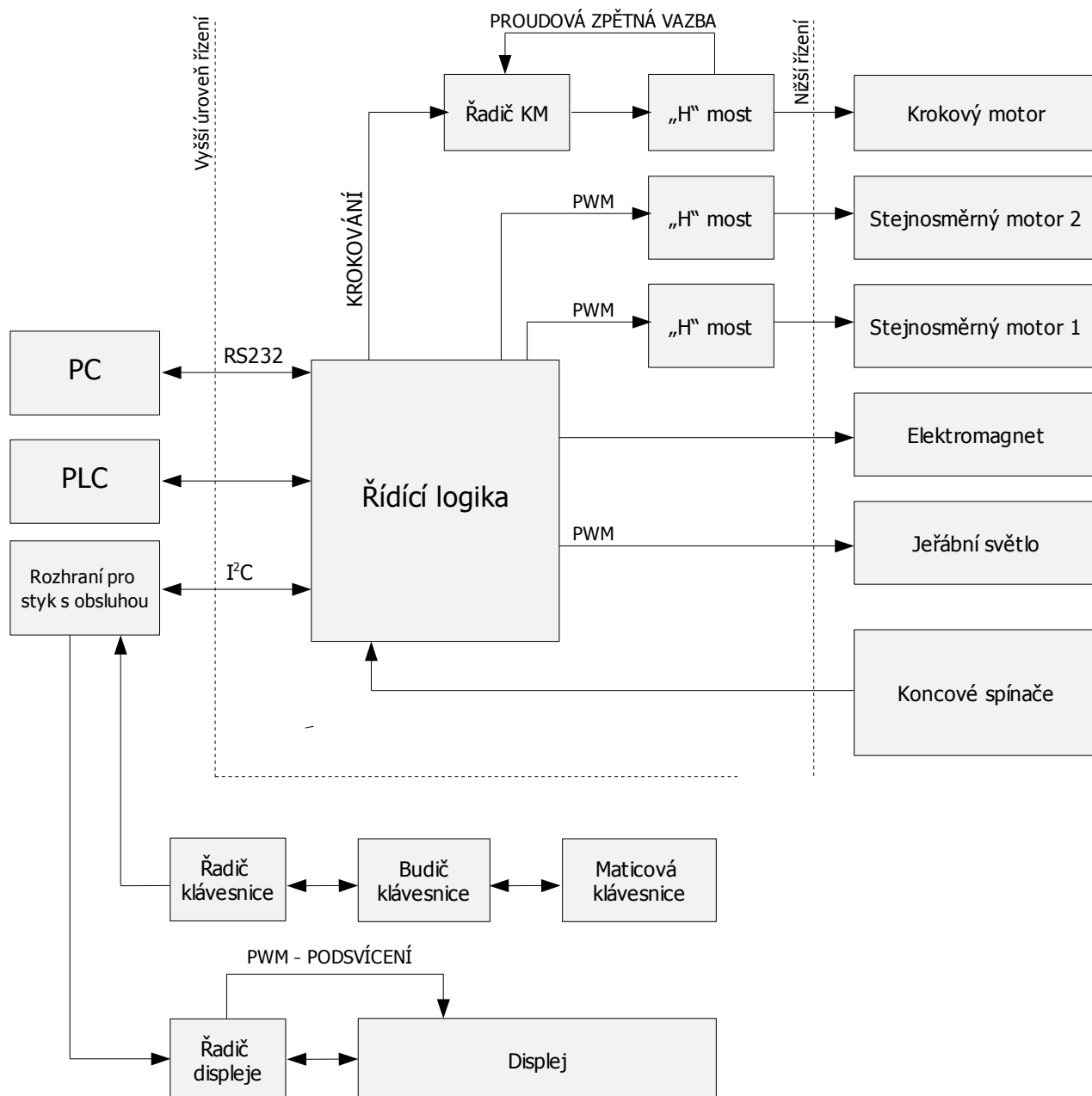
3.4. Posuv jeřábové kočky

Horizontální posuv od a ke středu otáčení konstrukce je vyřešen silonovou jeřábovou kočkou, umístěnou na závitové tyči M8. Tím dosahujeme převodu z pohybu otáčivého na posuvný a převodu rychlosti, který je závislý na stoupání použité tyče (v milimetrech). Z důvodu zabránění prohýbání závitové tyče, které by způsobovalo „poskakování“ a narážení do ramene, je kočka vedena v podelných kolejnicích tvaru „L“. Tímto se dosahuje nulového svislého namáhání závitové tyče (vlastní tíhou elektromagnetu) a zároveň plynulého chodu motoru. Kolejnice jsou pro jednoduchou montáž a demontáž našroubovatelné z vnější strany ramene. Stejnoseměrný motor, který pohání závitovou tyč, má dostatečný krouticí moment a dostatečně vysoké otáčky pro posun kočky. Dráha kočky je 380 mm a je omezena dvěma koncovými spínači.

4. Elektronická řešení

Elektronika manipulátoru je založena na mikroprocesorové technice. Jednočipové mikrokontroléry využíváme pro komunikaci po sběrnících I²C (TWI) a RS232c, pro modulaci pulzně šířkovou (PWM), vytvoření rozhraní styku s obsluhou (displej, klávesnice), atd..

Plošné spoje jsou realizovány z části součástkami s povrchovou montáží (SMD).

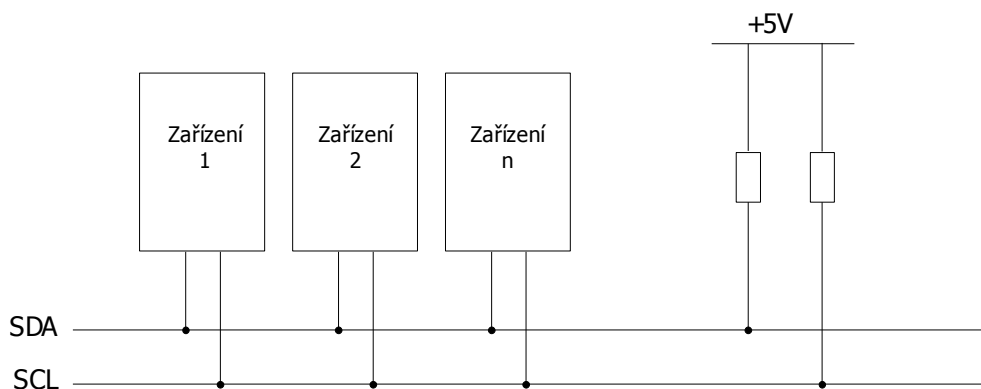


Blokové schéma zapojení elektroniky manipulátoru

4.1. Sériová sběrnice I²C (TWI)

Pro komunikaci mezi mikrokontroléry uvnitř manipulátoru využíváme ekvivalent sériové sběrnice I²C (IIC; InterIC) - sběrnici TWI (Two-wire interface). Na sběrnici „I²C bus“ vlastní patent firma PHILIPS, proto vzniklo několik nepatentovaných ekvivalentů této sběrnice, které jsou mnohem levnější, než při používání elektronických součástek se zabudovanou sběrnicí I²C.

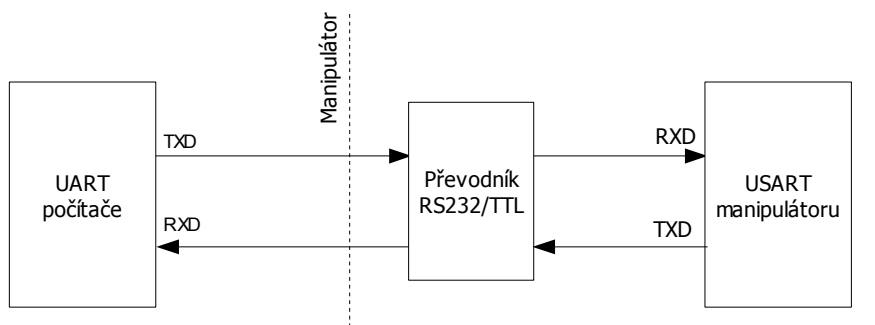
I²C je synchronní dvou vodičová sběrnice s přenosovou rychlostí do kmitočtu 100kHz. Používá hodinovou linku SCL (Serial Clock) a datovou linku SDA (Serial Data), je adresovatelná (7 bitů) a připojená zařízení rozděluje na typ Slave (ovládaný) a typ Master (ovládající). Detaily o průběhu datové komunikace najdete v kapitole 5.3 - Sběrnice I²C (TWI).



Propojení sběrnice I²C (TWI)

4.2 Topologie připojení k počítači PC

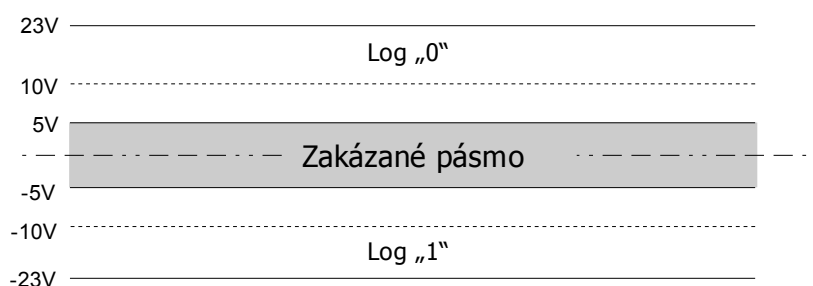
Standart RS232 (resp. jeho poslední verze RS232C z roku 1969, nazývaná také sériový port, COM port) se dodnes používá k propojení počítače a dalších zařízení. Jeho výhodou oproti sběrnici USB je, že není tak komplexní. RS232 definuje jen jakým způsobem posílat data, nezajímá se již o vlastní obsah dat. Způsob posílání dat a komunikační protokol na vyšších vrstvách je popsán v kapitole 5.4. Komunikační protokoly.



Propojení dvou zařízení pomocí RS232

UART je asynchronní přijímač/vysílač dat určený pro RS232. Dva UART moduly jsou mezi sebou propojeny křížem (vysílač – přijímač), data vyslaná jedním zařízením po svém TXD (transmission data) výstupu budou přijata druhým UART modulem na jeho vstupu RXD (receive data). USART je funkčně schodný s UARTem, ale je navíc schopný synchronní třívodičové komunikace (přidáním hodinové linky).

Standard RS232 používá pro přenos odolný proti rušením bipolární napětí $\pm 5\text{--}23\text{V}$, tyto hladiny musí být pro mikrokontrolér upraveny do hladin TTL/CMOS. Převod zajišťuje speciální elektronický obvod, viz kapitola 4.8.7. Převodník RS232 - TTL/CMOS



Napětíové průběhy na sběrnici RS232

4.3. Práce s maticovou klávesnicí

Zabudované ovládání využívá maticovou klávesnici o velikosti 4x3 znaky, informace z této klávesnice jsou čteny jednočipovým mikrokontrolérem ATtiny26, tento mikrokontrolér má dostatečný počet vstupně/výstupních bran pro čtení stavu klávesnice, a výstupní čtyřbitovou hodnotu. (podrobnější popis viz. kapitola 4.8.4. Rozhraní maticové klávesnice)

Klávesnice se dělí na inteligentní, vracející hodnoty přímo na datové sběrnici, na tlačítkové, kde je připojeno každé jednotlivé tlačítko a maticové, kde každé tlačítko propojuje kontakt mezi řádkem a sloupcem. V našem případě se jedná o maticovou klávesnici, ze které jsme pomocí mikrokontroléru udělali inteligentní.



Klávesnice mají při stisknutí tlačítka různý přechodový odpor většinou v rozsahu 30-100Ω, proto je na výstupech klávesnice zabudovaný tranzistorový budící obvod, který převede stav tlačítek na logickou hodnotu.

Zákmity tlačítek jsou ošetřeny softwarově, pravděpodobně nejpoužívanějším algoritmem několika měření, viz kapitola 5.7. - Čtení stavů klávesnice

4.4. Práce s alfanumerickým displejem

Displeje se u nás objevily koncem 80 let. Tehdy se považovaly za zázrak techniky. Dnes jsou LCD zobrazovací displeje všude okolo nás. Setkáváme se s nimi při koupi jízdenky na autobus nebo vlak, v telefonních automatech, na parkovištích, ve faxech atd. Každý displej má v sobě speciální integrovaný obvod - řadič, který ovládá celý displej a komunikuje s okolím. Standartem se stal řadič HD44780 od firmy HITACHI. Všichni výrobci používají tento řadič, nebo jeho ekvivalent. Alfanumerický LCD displej se vyrábí ve více variantách - 8x2, 16x1, 16x2, 16x4, 20x2, 20x4, 24x2, 24x4, 40x2, 40x4 znaků. Moduly se vyrábí s podsvícením LED nebo výbojkou.

Jádrem řadiče jsou dvě RAM paměti - CGRAM a DDRAM. CGRAM (Character Graphics RAM) je osmibajtová paměť pro uložení osmi vlastních znaků (nejčastěji se používá pro znaky české abecedy), DDRAM (Display Data RAM) obsahuje vlastní text na displeji. Řadič s okolím komunikuje po osmibitové, nebo čtyřbitové paralelní sběrnici a používá tři speciální signály (RS, E, R/W). Signál RS (Register Select) určuje jestli chceme zapisovat/číst z CGRAM, nebo DDRAM. Signál R/W se používá pro definici operace s registrem (čtení/zápis). Signál E (Enable) povoluje čtení datové sběrnice.

Popis vývodů řadiče:

Vývod	Název	Popis
1.	VSS	Napájení (GND)
2.	VDD	Napájení (+5V)
3.	VO	Kontrast
4.	RS	Register Select
5.	R/W	Read/Write
6.	E	Enable
7.	DB0	Datová sběrnice (nejvyšší bit)
8.	DB1	Datová sběrnice
9.	DB2	Datová sběrnice
10.	DB3	Datová sběrnice
11.	DB4	Datová sběrnice
12.	DB5	Datová sběrnice
13.	DB6	Datová sběrnice
14.	DB7	Datová sběrnice (nejnižší bit)
15.	ANODE	Anoda LED diod podsvícení
16.	CATHODE	Katoda LED diod podsvícení

Pro komunikaci s displejem je třeba 6 až 11 vodičů (v závislosti na potřebných možnostech a rychlosti komunikace). Po připojení napájení je nutno nastavit šířku komunikační sběrnice (4, nebo 8 bitů/vodičů) a poté displej inicializovat nastavením hlavně směru psaní, velikosti znaků, pozice kurzoru, blikání kurzoru (více viz [21], nebo [28] a [29]). Každý znak je definován jako matice 5x8, nebo 8x8 pixelů. V tabulce můžeme vidět zapojení vývodů řadiče.

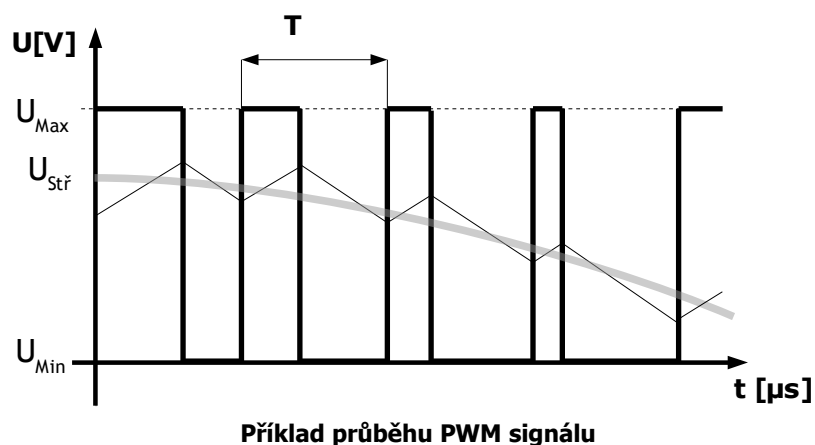
Displej manipulátoru má rozměr 20x4 znaků s zeleným LED podsvícením. Je používána osmibitová (rychlá) komunikační datová sběrnice. Displej je nastaven pro psaní zleva, velikost znaků je 5x8 pixelů, bez kurzoru.

4.5 Principy řízení otáček motorů

Jedním ze základních cílů tohoto projektu je prezentace moderních principů regulace otáček a výkonu na různých typech elektromotorů. Jeřáb využívá dva typy motorů – stejnosměrný komutátorový a krokový motor. Tyto motory jsou velmi odlišné a vyžadují jiné principy řízení i regulace. Použili jsme je tedy pro demonstraci dvou odlišných způsobů řízení. Zvolili jsme pro oba typy motorů moderní řešení na vysoké technické i technologické úrovni.

4.5.1. Pulzně-šířková modulace (PWM)

Pulzně-šířková modulace (anglicky Pulse Width Modulation – PWM) je změna poměru šířky pulzu, k době trvání periody výstupního napětí, který nazýváme střídou „s“. Tato změna se projeví na změně střední hodnoty výstupního napětí a tím například na rychlosti otáčení motoru, nebo na jasu světla LED diody.



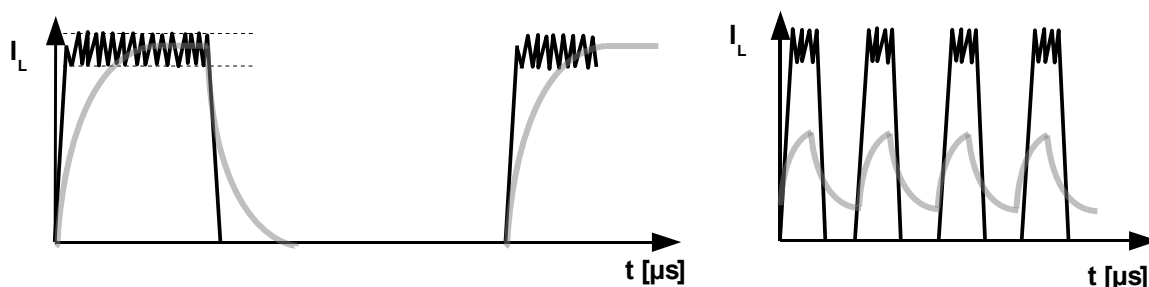
Graf znázorňuje průběh popsaného PWM signálu, perioda T určuje konstantní kmitočet. Jak je z grafu vidět, změnou střídy se mění střední hodnota napětí.

Tento princip využíváme při regulaci otáček stejnosměrných motorů. Pro prezentaci možností PWM, má software zabudovaný parametr nastavující dobu zpoždění přechodu z původních otáček do otáček nových. Detailní popis PWM přechodů se nachází v kapitole 5.6. - Generování PWM signálů, detaily o plošném spoji, který zajišťuje buzení dvou stejnosměrných motorů, ovládaných PWM signálem najdeme v kapitole 4.8.2. – Buzení stejnosměrných motorů

4.5.2 Pulzní proudový zdroj – chopper

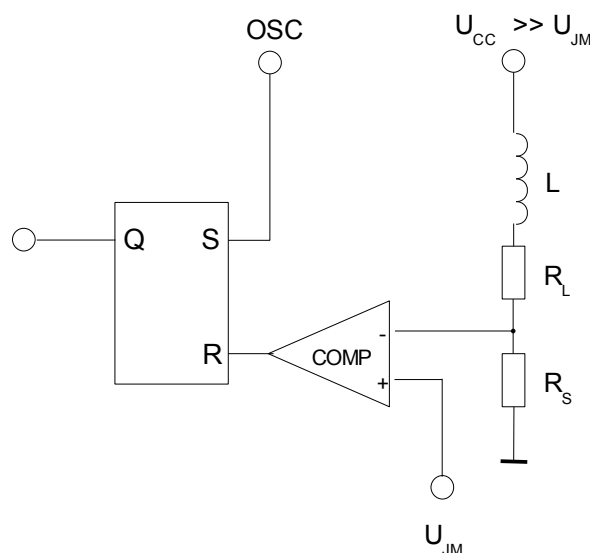
Dle [3] je proudový zdroj nejideálnější obvod sloužící k buzení krokových motorů (KM). Hlavním důvodem poklesu momentové charakteristiky u nejjednoduššího buzení napětovým zdrojem je nízká strmost proudu protékajícího fází (princip KM viz kapitola 4.7.1. - Hybridní krokový motor), která je úměrná časové konstantě, způsobené vlastním odporem indukčnosti R_L indukčností L . (více například v [3], nebo [17]). Chopper potlačuje vliv časové konstanty buzením napětím řádově vyšším, než je jmenovité napětí motoru. Aby při překročení jmenovitého napětí nedošlo k poškození motoru, chopper (neboli „odsekávač“) odpojí napájení vinutí motoru. Toto se děje se pomocí komparátoru, na jehož jeden vstup je přivedeno napětí úměrné proudu do motoru. Ve chvíli kdy je napájení odpojeno, zatěžovací proud i napětí na fázi začne klesat, a poté komparátor znovu sepne. Tento jev se na grafu projevuje mírným „poskakováním“ napětí v sepnutém stavu fáze. Známý obvod L297 [24] v kombinaci s budičem (například obvody L298 – viz [25], nebo L293 – viz [26]) zajišťuje rozhraní mezi mikrokontrolérem a KM, včetně regulace zatěžovacího proudu pomocí chopperu.

Jak je z grafu vidět, při vyšším krokovacím kmitočtu má základní způsob buzení – z napětového zdroje problém s časovou konstantou, což způsobí klesání momentové charakteristiky a snížení rychlosti otáčení rotoru.



Porovnávání strmosti náběhu zatěžovacího proudu, pomocí buzení typu chopper (černě) a buzení z napětového zdroje. Graf vlevo znázorňuje průběh při nízkém krokovacím kmitočtu, graf vpravo při vysokém kmitočtu.

Pro regulaci proudu protékajícího vinutím je použit klopný obvod typu RS, kde na nastavovací vstup je připojen oscilátor a na resetovací vstup komparátor porovnávající pomocí snímacího odporu R_s jmenovité napětí U_{JM} s napětím U_{CC} , které protéká vinutím L dané fáze. Rychlost strmosti náběhu proudu ve vinutí je určena časovou konstantou, způsobené vlastním odporem a indukčností vinutí R_L a indukčností L . Pokud napětí na vinutí je větší než napětí jmenovité U_{JM} , komparátor sepne a vyresetuje klopný obvod, který odpojí zdroj proudu od vinutí. Klopný obvod zůstane resetovaný do doby, než oscilátor vytvoří náběžnou hranu, poté se celý proces opakuje.



Zjednodušené schéma vnitřního zapojení chopperu

Tento způsob buzení využíváme pro získání lepších rychlostních i momentových charakteristik z KM, který má na ose připevněn naviják kabelu elektromagnetu. Krokový motor je zapojen bipolárně, polarita je měněna na „H“ mostech v obvodu L298.

4.6 Elektromagnet a jeřábní světlo

Každý jeřáb ke své funkci potřebuje zařízení, které přenáší břemeno. Manipulátor využívá elektromagnet (detaily provedení, viz. kapitola 3.2. - Svislý posuv elektromagnetu), který je schopen přenášet kovové předměty s maximální vahou do 300g. Jelikož při přenášení břemene je jakákoliv modulace výkonu nevyužitelná a spínání elektromagnetu probíhá běžně v delších časových intervalech, je pro spínání použito relé, se spínacím napětím 5V. Relé spíná 12V pro elektromagnet.

Jeřábové světlo tvoří sada modrých vícesvítivých (2000-2500 mCd) LED diod. Tyto diody jsou tranzistorově buzeny, a jejich svítivost je regulována za pomoci PWM signálu. Z hlediska firmwaru není možné měnit svítivost LED diod z jakéhokoliv ze tří ovládacích rozhraní, ale při zapínání/vypínání se modré světlo odráží na podkladové desce a vytváří obraz přechodového děje. Jedná se o nejprůzračnější prezentaci možností regulace pomocí PWM.

4.7 Popis motorů

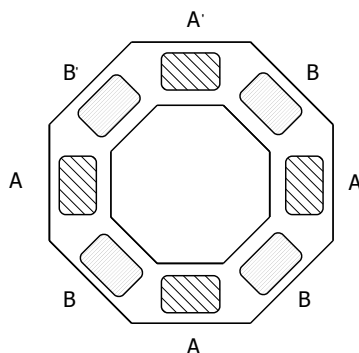
4.7.1 Hybridní krokový motor

Krokové motory se dělí na unipolární, bipolární, hybridní, lineární hybridní, lineární reluktanční a s pasivním rotorem (viz [3]). Dnes se převážně využívají dvoufázové KM, ale první typ krokového motoru (patentován ve Velké Británii r. 1919) používal 3 fáze. V dnešní době existují i pětifázové krokové motory, které si patentovala německá firma Berger-Lahr.

Prvně krokové motory využívalo od roku 1920 námořnictvo Velké Británie pro navigaci torpéd. Dalších 40 let KM nenašly jiné uplatnění, až v 60. letech s rozvojem číslicové techniky a vyvinutím prvního 4-bitového mikrokontroléru (Intel, 1971), začínaly nacházet uplatnění v aplikacích s přesným pozicováním (např. průmyslové roboty). Dnes se KM využívají zejména v těchto aplikacích:

- Tiskárny
- NC, CNC stroje
- Pohony průmyslových robotů, manipulátorů
- Polohování hlaviček pevných disků
- Akademická činnost (výzkum, publikace)

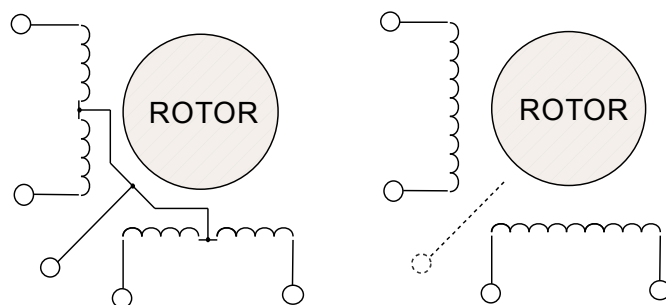
Hybridní krokový motor (dvoufázový) je tvořen dvěmi základními částmi – rotorem a statorem. Stator je tvořen osmi pólovými nástavci. Na nástavcích se střídá vinutí první a druhé fáze, každý nástavec je dále rozdělen na 5 zubů (není na obrázku).



Způsob propojení fází pólových nástavců; každé šrafování představuje navzájem propojená vinutí (fáze A, nebo B).

Rotor motoru je tvořen hřídelí z nemagnetické oceli, na které jsou nalisovány dva pólové nástavce složené z plechů. Mezi těmito nástavci je axiálně polarizovaný permanentní magnet. Pólové nástavce jsou dále rozděleny na 50 zubů o stejné šířce jako statorové.

Hybridní KM je možné zapojit unipolárně, i bipolárně. U unipolárního zapojení přivedeme na střed vinutí napájení a uzemňujeme 4 fáze v daném pořadí dle způsobu krokování. V bipolárním zapojení měníme pomocí dvou „H“ mostů polaritu dané fáze, také v závislosti na způsobu krokování.



Způsoby zapojení fází hybridního krokového motoru. Vlevo unipolární, vpravo bipolární.

Existují 3 základní způsoby krokování motoru – půlkrokování, krokování a dvojkrokování. Jejich realizace je velmi jednoduchá, záleží jen na způsobu přepínání fází řídicí logiky (viz [22]).

4.7.2 Stejnosměrný motor

Stejnosměrný motor s permanentním magnetem je často využívaným typem motoru, díky příznivému poměru cena/výkon a výkon/hmotnost. Výhodami tohoto motoru je jednoduché řízení otáček a dostupnost. Nevýhodou je hlavně poměrně složité a drahé polohové řízení (v porovnání s KM) a díky komutátoru (omezuje maximální proud a má nároky na údržbu) je tento motor také zdrojem elektromagnetického rušení, některé starší typy těchto motorů nesplňují ekologické požadavky pro normu EMC (elektromagnetická kompatibilita).

Stejnosměrný motor obvykle pracuje na relativně vysokých otáčkách, při nízkém momentu. Tento problém je řešen převodovkami, většinou v poměru od 1:2 do 1:100. Změnou polarity stejnosměrného motoru docílíme změny směru otáčení. Změnou napětí pak změnu rychlosti otáčení.

Pro řízení otáček stejnosměrného motoru je z energetických důvodů výhodné použít napájecí obvod s PWM (viz kapitola 4.5.1. - Pulzně-šířková modulace). Tím se dá dosáhnout vhodného „dávkování“ energie do vinutí motoru, aniž by vznikaly ztráty na spínacích prvcích

4.8. Elektronické obvody

Následujících 8 podkapitol popisuje nejdůležitější elektronické obvody, které manipulátor využívá. Zajišťují komunikaci, buzení a zejména řízení.

4.8.1. Nižší úroveň řízení

Nižší úroveň řízení zajišťuje vykonání příkazů přijatých po sériové sběrnici TWI, nebo RS232 (konektory SV2 a SV1). Jádrem zapojení je mikrokontrolér Atmega8, který ovládá všechny akční členy a snímá stav koncových snímačů. Konektor SV4 se připojuje k budiči krokového motoru (viz. kapitola 4.8.3. Buzení krokového motoru). Konektor SV5 k budičům stejnosměrných motorů (viz. kapitola 4.8.2. Buzení stejnosměrných motorů). Tranzistor T1 ve spínacím režimu spíná LED diody jeřábního světla a tranzistor T2 (také ve spínacím režimu) spíná cívku relé. Na vývody konektoru SV3

připojujeme koncové snímače. Posuvný registr HCF4094 vytváří další „imaginární“ výstupní bránu mikrokontroléru, pro signalizaci stavu konstrukce.

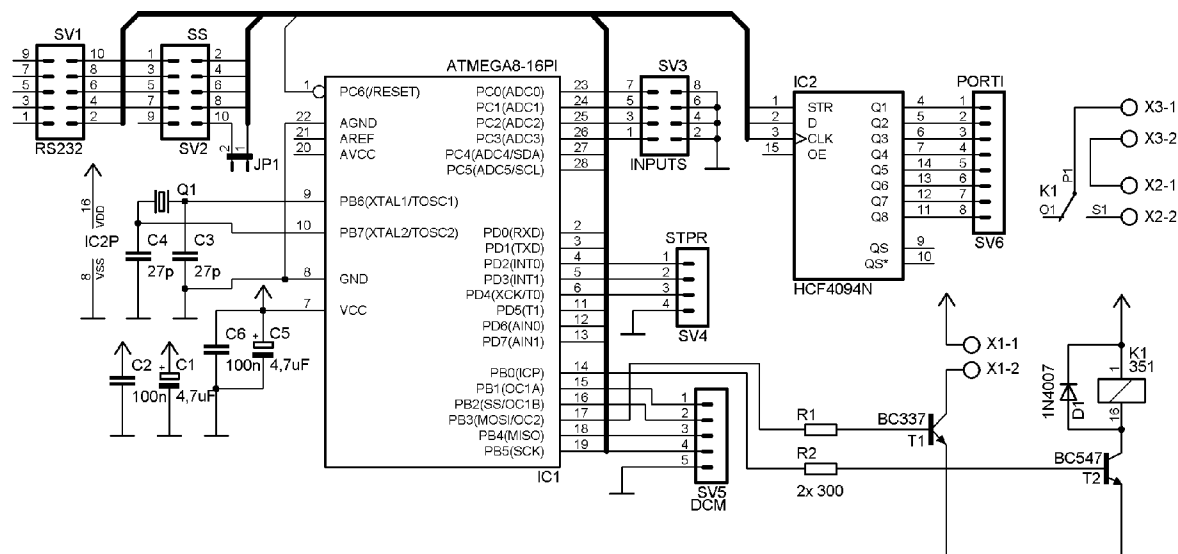


Schéma nižší úrovně řízení

4.8.2. Buzení stejnosměrných motorů

Tento obvod zajišťuje buzení obou stejnosměrných motorů. Jádrem zapojení je obvod L298, dvojitý „H“ most, s maximálním zatěžovacím proudem můstku 2A. Budič je galvanicky oddělen od řídicí logiky optočleny, s dostatečným mezním kmitočtem, pro přenášení PWM signálu. Obvod L298 má zabudovanou logiku pro obracení polarit na „H“ mostu (v závislosti na vstupech INPUTx), proto bylo nutné přidat do obvodu regulátor 7805 v pouzdře TO92, který zajistí napájecí napětí této logiky. Silové výstupy jsou ošetřeny vratnými pojistkami PolySwitch na 1,85A a Shottkyho diodami ($t_R < 200\text{ns}$), které chrání motory před napětovými špičkami vytvářejícími se na vinutí rotoru.

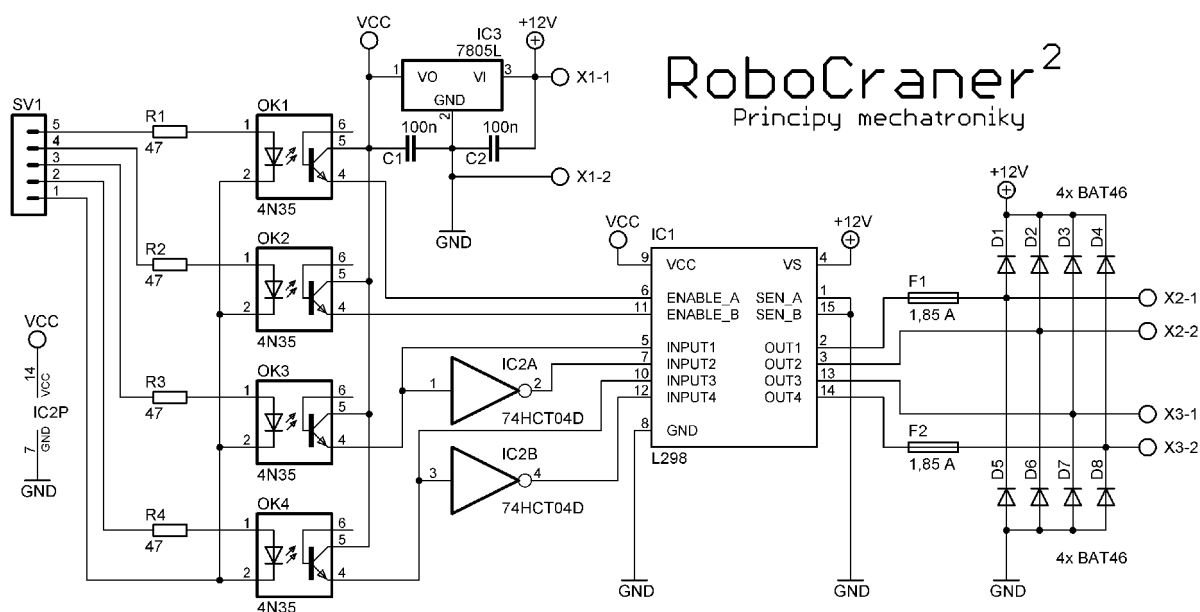


Schéma zapojení budiče stejnosměrných motorů

Budič jednoho motoru je ovládán dvěma signály – povolení (vstup ENABLE) a směr otáčení (vstupy INPUT). Budicí obvod má pro ovládání polarit (směru otáčení) vyvedeny dva vstupy – pro každou polaritu jeden, proto je první vstup neinvertovaný a druhý invertovaný, pomocí invertoru 74HCT04 (tento obvod je letován na DPS ze strany spojů, v provedení SMD).

4.8.3. Buzení krokového motoru

Tento elektronický obvod je budičem jednoho krokového motoru. Má stejně jako budič stejnosměrných motorů galvanicky oddělené vstupy, řídicí logiku zajišťuje obvod L297, který realizuje s několika externími součástkami buzení pomocí proudového zdroje typu chopper. Výstupy z obvodu L297 jsou přivedeny do budiče L298, dvojitého „H“ mostu s maximálním proudovým zatížením 2A. KM je zapojen bipolárně na výstupy „H“ mostů, vinutí fází jsou chráněna Schottkyho diodami proti napěťovým špičkám a proudový odběr je omezen vratnou pojistkou PolySwitch.

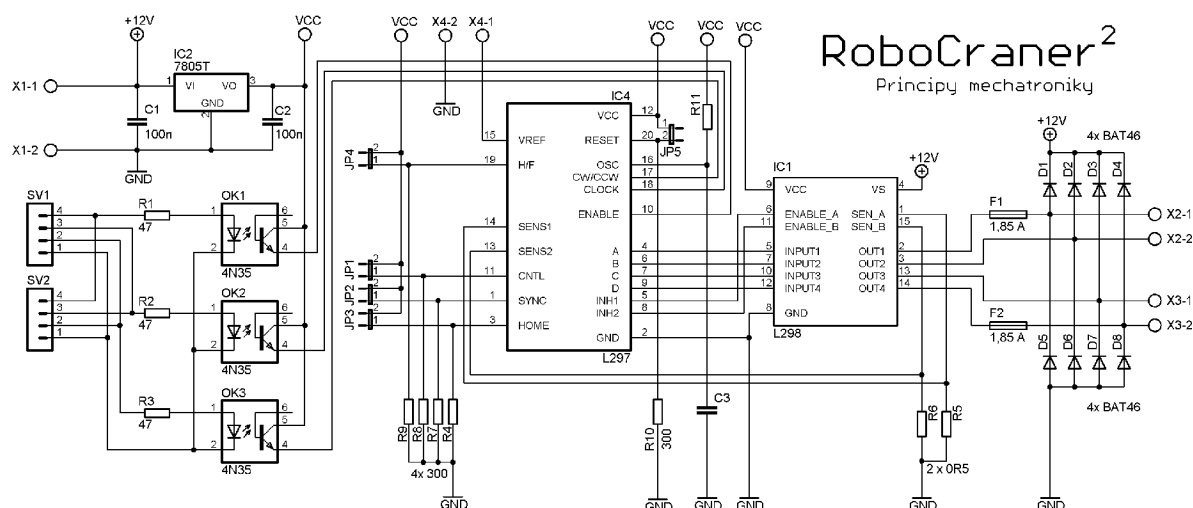


Schéma zapojení budiče krokového motoru

Obvod má tři vstupy – KROK, SMĚR, POVOLENÍ. Kmitočet na vstupu KROK určuje rychlost otáčení (krokování) motoru. Vstup SMĚR definuje smysl otáčení a vstup POVOLENÍ (enable) se používá k zapnutí/vypnutí fází motoru.

4.8.4. Rozhraní maticové klávesnice

Tento obvod zajišťuje buzení a čtení dat z maticové klávesnice, které vrací v binárním tvaru. Mikrokontrolér budí vždy jednu ze tří řádků a hledá odezvu na jednotlivých sloupcích. Součástky na DPS jsou v provedení povrchové montáže (SMD). Obsahuje LED signalizaci stisknutého tlačítka.

Celý plošný spoj je rozměrově shodný s použitou klávesnicí, je možné jej našroubovat přímo pod klávesnici, za pomoci distančních sloupků. Vstupní konektor (KYB) tvoří pinové zdíčky, do kterých se zastrčí klávesnice. Na výstupní konektor (OUT) je připojen binární výstup, přerušovací pin a je jím přivedeno napájení modulu.

4.8.6. Převodník dat RS-232/I²C

Tento obvod realizuje pomyslnou vstupní bránu počítače do systému manipulátoru. Je obousměrným převodníkem datového toku mezi sběrnicemi RS-232 a I²C. Tyto sběrnice používají rozdílné logické úrovně, komunikují jinými rychlostmi a jejich topologie je také rozdílná.

Obvod signalizuje stav převodu na LED diodách OK a ERR a stav přenosu oběma směry (LED diody RX a TX). V řešení je použit mikrokontrolér AtMega8, dalo by se říci, že je to až přespříliš velký „kalibr“ pro toto řešení, ale tento obvod má velmi výkoné periférie, zejména moduly UART (RS-232) a TWI (I²C), které nám umožní výrazně zjednodušit program mikrokontroléru.

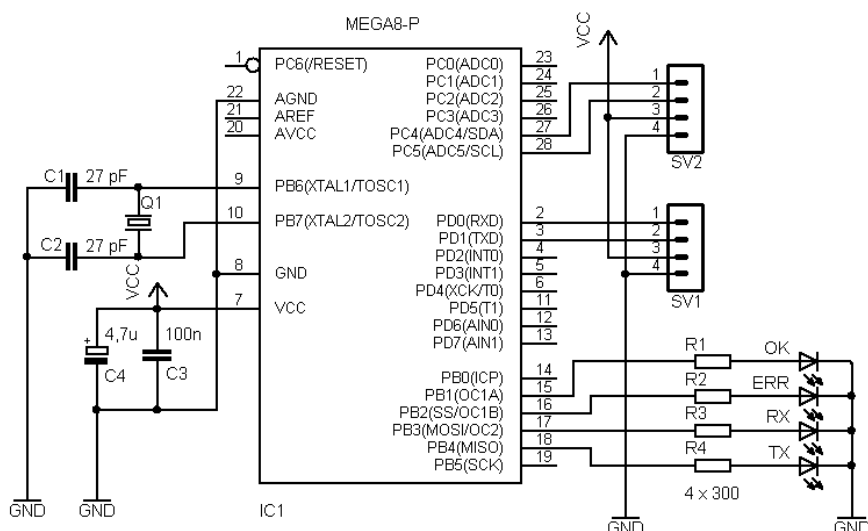


Schéma převodníku RS-232/I²C

4.8.7. Převodník RS232C - TTL/CMOS

Tento převodník vytváří rozhraní mezi počítačem a mikrokontrolérem. Napěťové úrovně protokolu RS232 jsou bipolární ±5-23V, úrovně zpracovávající mikrokontrolér jsou standardní TTL/CMOS. Plošný spoj je vyroben z větší části v provedení povrchové montáže (SMD).

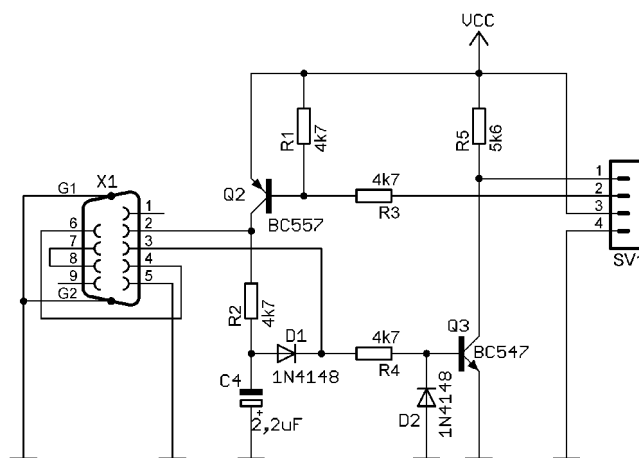


Schéma převodníku

Tento převodník je zajímavý svým provedením. Jedná se o levnější alternativu k notoricky známému zapojení s obvodem MAX232. Celý obvod se skládá z několika málo součástek.

5. Popis softwaru mikrokontrolérů

5.1. Použité programovací jazyky

Pro prezentaci různých způsobů programování a algoritmizace jsme se rozhodli napsat software mikrokontrolérů ve dvou různých jazycích – v assembleru a ANSI C.

5.1.1. Assembler

Assembler, neboli jazyk symbolických adres je programovací jazyk velice blízký strojovému kódu. Assembler je programovací jazyk nejnižší úrovně, tvoří jej pouze mnemotechnické symboly, které přímo odpovídají strojovému kódu. Tyto symboly se mnohem lépe zapamatovávají, než strojový kód v binární, nebo šestnáctkové podobě. Například dva bajty strojového kódu o hodnotě **1011000001100001**_[2] (**0xB161**_[16]) odpovídají instrukci **MOV A, 61h** procesoru Intel 80386 (počítače řady 386).

Procesor každého počítače i mikrokontroléru má svoji vlastní instrukční sadu, například tradiční mikrokontrolér Intel 8051 používal assembler s instrukční sadou ASM51. Manipulátor je osazen mikrokontroléry firmy ATMEL, řady AVR. Assembler těchto mikrokontrolérů se nazývá AvrAssembler.

Příklad zdrojového kódu v AvrAssembleru (Realizace běžícího světla):

```
.equ A = r16                ; Definice zástupného jméno registru

START: ser DDRB              ; Brána B (reg. PORTB) jako výstupní
      sbi PORTB, 0           ; Nastavit sedmý bit registru PORTB
LOOP:  in A, PORTB           ; Načtení hodnoty PORTB do reg. A
      rol A                  ; Rolovat bity v registru A doleva
      out PORTB, A           ; Nakopírování hodnoty A do registru
                                PORTB, tím posune světlo.
      rcall DELAY            ; Zavolá spožďovací rutinu
      rjmp LOOP              ; Skočí na řádku s návěstím LOOP (Celý
                                proces se bude opakovat)
```

5.1.2. ANSI C

C je programovací jazyk, který vyvinuli Ken Thompson a Dennis Ritchie pro potřeby operačního systému Unix, v současné době je to jeden z nejpopulárnějších jazyků. Vzhledem k jeho nízké úrovni abstrakce (zobecnění) se nejčastěji používá pro psaní systémového softwaru pro mikrokontroléry a pro časově náročné aplikace na osobních počítačích.

Zdrojový kód C je mnohem čitelnější než zdrojový kód assembleru, jeho nevýhodou je, že i když je nízkourovňový, neposkytuje zdaleka nulovou úroveň zobecnění, jak je tomu u assembleru. Programy psané v jazyce C jsou obvykle na rozdíl od assembleru strukturované, nepoužívají instrukce, ale příkazy. Každý příkaz je při kompilaci kompilátorem přepsán na soubor instrukcí strojového jazyka. Strojový jazyk se dá zpětně vyjádřit mnemotechnickými symboly – tedy assemblerem. V závislosti na

kvalitě C kompilátoru je výsledný program dlouhý (v bajtech) a rychlý (v počtu hodinových cyklů). Rychlost výsledného kódu má vliv například u některých časovacích algoritmů, velikost zkompilovaného programu může vytvářet problémy u mikrokontrolérů s malou pamětí ROM pro program.

Příklad zdrojového kódu v ANSI-C (Realizace běžícího světla):

```
void main(void)
{
    unsigned int i;           // Deklarace proměnné i
    DDRB = 0xFF;             // Bránu B (PORTB) nastavit na výstup

    while(1)                 // Nekonečný cyklus
    {
        PORTB = 0x01;        // Nastavení počáteční hodnoty
        for(i=8; i>0; i-=1)  // Cyklus - 8x opakovat
        {
            PORTB = PORTB<<1; // Násobení bitů v PORTB dvěma
            Delay();           // Spožďovací rutina
        }
    }
}
```

5.2. Standard RS-232

Standard RS-232 (resp. jeho poslední varianta „C“ z roku 1969; nazývaný také sériový port, nebo COM port) se používá jako komunikační rozhraní osobních počítačů a dalšího zařízení. Na rozdíl od modernějšího USB, standart RS-232 pouze definuje jak přenést určitou sekvenci bitů a nezabývá se již vyššími vrstvami řízení. V referenčním modelu ISO/OSI tak představuje jen nejnižší fyzickou vrstvu komunikace.

5.3. Sběrnice I²C (TWI)

Sběrnice I²C je standartem pro komunikaci mezi mikrokontroléry a dalšími speciálními součástkami, které jsou schopny komunikovat po této sběrnici. Vyvinula a patentovala ji firma PHILIPS. Kvůli patentům dnes existují další sběrnice kompatibilní s I²C, jednou takovou je i TWI (Two-Wire Serial Interface), kterou používají mikrokontroléry ATMEL.

I²C i TWI je synchronní dvouvodičová sběrnice, která má v protokolu zabudované adresování a dělení na zařízení typu Master a Slave. Master je zařízení, které začíná a ukončuje komunikaci, generuje hodinový signál, adresuje Slave zařízení a definuje směr toku informace. Master může zapisovat, nebo číst Slave zařízení, směr toku dat určuje nejnižší bit (R/W) prvního poslaného bajtu, sedm vyšších bitů tohoto bajtu je adresa dotazovaného Slave zařízení. Vždy když vysílající zařízení (transmitter) odešle jeden bajt, receiver odpoví tzv. kvitovacím (ACK) bitem.

5.3.1. Adresování a přenášení dat na sběrnici

TWI využívá dva datové vodiče - SDA (serial data) a SCL (serial clock). SDA pro data, SCL pro hodinový signál, který generuje Master. Existují dva speciální stavy na sběrnici: START a STOP. Tyto stavy určují začátek a konec datového paketu. Po odeslání START sekvence vždy následuje bajt složený takto:

ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	R/W
-------	-------	-------	-------	-------	-------	-------	-----

ADDR6:0 je adresa volaného Slave zařízení (na sběrnici je možné připojit až 127 různých zařízení). Poslední bit R/W je tzv. směrový bit a určuje, zda si Master přeje uložit data do Slave zařízení, nebo si přeje od Slave data přijímat.

Pro odeslání určitého počtu bajtů je nutné nejdříve vygenerovat START podmínku, odeslat adresu a R/W bit (tím se definuje směr komunikace a určí adresované Slave zařízení). Pokud Slave zařízení potvrdilo přijetí (odkviťovalo ACK bitem), tak odeslat první datový byte, počkat na potvrzení, odeslat druhý datový byte a analogicky pokračovat až do odeslání celé informace.

5.3.2. Módy přenosů

Existují 4 druhy přenosů a každý z nich má své vlastní status kódy (status kód je informace o průběhu odesílání/přijímání dat na sběrnici). Pokud směrový bit R/W, má hodnotu WRITE, tak Master i Slave zůstanou v původních módech (Master Transmitter a Slave Receiver). Pokud směrový bit má hodnotu READ, Master přejde do módu Master Receiver a Slave do Slave Transmitter.

Směrový bit	Popis
WRITE	Master zařízení říká, že si přeje uložit do Slave zařízení informaci, takovou informaci může být například rychlost otáčení motoru, který tento Slave ovládá.
READ	Master zařízení říká, že potřebuje od Slave zařízení informaci zjistit, například stav koncového spínače.

5.3.3. Používání TWI

Na základě teoretických znalostí o sběrnici byl vytvořen jednoduchý ovládací software v jazyce AvrAssembler. Nejedná se o žádný komplexní nástroj, který by zabíral velké množství místa a dodával vysokou úroveň zobecnění. Tyto rutiny jsou spíše „kuchařkou“ jak rychle a jednoduše psát programy, které TWI sběrnice využívají pro komunikaci s okolím.

5.3.3.1. Řídící registry TWI

Popis registrů, které ovládají hardwarový modul TWI

Název Registru	Popis
TWI Control Register (TWCR)	Řídící registr TWI, v tomto registru se provádí vlastní povolení modulu, nastavuje se příznak přerušení a upravují se další možnosti přenosu informace.
TWI Bit-Rate Register (TWBR)	Hodnota tohoto registru přímo určuje přenosovou rychlost. Viz kapitola 5.3.3.2. - Inicializace master zařízení
TWI Status Register (TWSR)	Vyšších pět bitů tohoto registru obsahuje indikaci stavu jednotky TWI, dva nejnižší bity obsahují nastavení děličky generovaného hodinového signálu.
TWI Data Register (TWDR)	Vstupně/výstupní datový registr. Pokud mikrokontrolér přijme data, uloží je do tohoto registru. Pokud program mikrokontroléru uloží data do tohoto registru, budou odeslána. Jelikož dle topologie nemůže být sběrnice TWI duplexní, stačí jen tento jediný registr pro odesílání/přijímání dat (či adresy).
TWI Address Register (TWAR)	Tento registr používají jen slave zařízení. Sedm vyšších bitů určuje vlastní adresu, poslední bit TWGCE určuje, zda má jednočip reagovat na General Call (na adresu 0x00).

5.3.3.2. Inicializace master zařízení

Master potřebuje pro svou inicializaci definovat přenosovou rychlost sběrnice a nastavení vektorů přerušení TWI modulu. Kmitočet hodinového signálu se nastavuje uložením konstanty do registru TWBR (viz. vzorec). TWI jednotka také disponuje děličkou hodinového signálu, její nastavení se provádí v registru TWSR, bity TWPS1 a TWPS0 (detailní informace můžete najít v katalogových listech mikrokontrolérů ATMEL, které mají zabudovanou podporu TWI, např. ATmega8 - [19])

Inicializace TWI jako masteru

```
.org 0x000 rjmp START           ; Resetovací vektor přerušení
.org 0x011 rjmp TWI_INT         ; TWI vektor přerušení

;   Nastavení rychlosti přenosu
.equ fCPU = 16000000           ; Kmitočet krystalu MCU [Hz]
.equ fTWI = 20000              ; Přenosová rychlost [Hz]
```

```

; Vzorec pro přepočet rychlosti sběrnice
.equ TWIpomer = ((fCPU/(8*(fTWI)))-2)

START: ldi r16, LOW(RAMEND)           ; Nastavení zásobníku
      out SPL, r16
      ldi r16, HIGH(RAMEND)
      out SPH, r16
      ldi r16, TWIpomer               ; Nastavení rychlosti
      out TWBR, r16

      sei                             ; Globální povolení přerušení

```

Přepočet frekvence hodinového kmitočtu:

$$SCL\ kmitočet = \frac{Kmitočet\ MPU}{16 + 2(TWBR) * 4^{TWPS}}$$

Kmitočet MPU musí být minimálně 16x vyšší, než je kmitočet přenosové rychlosti a hodnota v registru TWBR by měla být větší než 10, pokud je menší, master může generovat nekorektní výstup na linkách SDA i SCL.

5.3.3.3 Inicializace slave zařízení

Inicializace TWI jako slave probíhá podobně jako u masteru, uloží se vlastní adresa do registru TWAR, spustí systém přerušení a spustí se logika detekce adresace (vyvolává přerušení když zařízení adresované masterem je shodné s hodnotou v registru TWAR)

Inicializace TWI jako slave

```

.equ ADDRESS 0b11111110           ; Adresa Slave zařízení

.org 0x000 rjmp START             ; Resetovací vektor přerušení
.org 0x011 rjmp TWI_INT           ; TWI vektor přerušení

START: ldi r16, LOW(RAMEND)        ; Nastavení zásobníku
      out SPL, r16
      ldi r16, HIGH(RAMEND)
      out SPH, r16
      ldi r16, ADDRESS             ; Nastavení vlastní adresy
      out TWAR, r16

; Spuštění TWI jednotky, a povolení TWI přerušení
ldi r16, (1<<TWINT) | (1<<TWEA) | (1<<TWEN) | (1<<TWIE)
out TWCR, r16

; Globální povolení přerušení
sei

```

5.3.3.4. Příjem a odesílání dat

Pro odeslání jednoho bytu (či adresy) musíme nejdříve daný bajt uložit do registru TWDR, počkat na plné odeslání a poté vyhodnotit vrácený status kód. Status kódy jsou různé jak pro odesílání adresy, tak i pro data v různých módech. Status kódy se dají nalézt v katalogovém listu použitého mikrokontroléru.

Odeslání jednoho bajtu pomocí TWI

```
out TWDR, r16 ; Uložit byte pro odeslání
ldi r16, (1<<TWINT) | (1<<TWEN) ; Vynulování přerušení
out TWCR, r16

; Počkání na odeslání bytu a získání kvitovacího bitu
TWIWAIT:
in r16, TWCR
sbrs r16, TWINT ; dokončen přenos?
rjmp TWIWAIT_BT ; Přenos nebyl dokončen,
; skoč na návěští TWIWAIT

in r16, TWSR ; Přenos hotov, načti TWSR
andi r16, 0xF8 ; Získání status kódu
cpi r16, MT_DATA_ACK ; Vše proběhlo v pořádku?
brne TWI_ERROR ; Neproběhlo, skočit na
; chybovou rutinu

ret
```

Rutina pro příjem jednoho bytu je dosti podobná odeslání. Hardware vyvolá přerušeni o přijetí bytu. V přerušeni tento byte přijmeme, poté potvrdíme příjem druhému zařízení, a nakonec vyhodnotíme status kódy. Pokud status kód odpovídá datovému bajtu, začne se zpracovávat (v závislosti na úloze programu)

Příjem jednoho bajtu pomocí TWI

```
TWI_INT:
in r19, SREG ; Záloha status registru
cli ; Vypnutí systému přerušeni
in r17, TWDR ; Přijetí bajtu
ldi r18, (1<<TWINT) | (1<<TWEA) | (1<<TWEN) | (1<<TWIE)
out TWCR, r18 ; Potvrzení přijetí
in r18, TWSR ; Načtení status kódu
andi r18, 0xF8 ; Získání status kódu
cpi r18, 0x80 ; Jedná se o data?
brne END ; Nejedná, skočit na konec

; ... ; Rutina pro zpracování přijatých dat

END:
out SREG, r19 ; Aktualizace status registru
reti
```


5.4. Komunikační protokoly

Programy mikrokontrolérů mají vytvořené komunikační protokoly, které zajišťují pochopení požadavku „Master“ zařízení ve sběrnici I²C.

5.4.1. Způsoby přenosu dat

Vždy první přijatý datový bajt je příkaz, příkazem je myšlena hodnota od 0x00 do 0xFF, která přesně určí požadavek nadřazeného zařízení. Existují 3 druhy příkazů:

Bezparametrový příkaz	Přijatý bajt přesně specifikuje požadavek nadřazeného zařízení (například příkaz zapínající elektromagnet jeřábu, není nutné ho nijak specifikovat)
Parametrový příkaz	Přijatý bajt specifikuje co je od zařízení požadováno, ale je nutné dodat specifikující informaci o provedení (například požadavek změnit rychlost otáčení motoru, zde je parametrem nová rychlost)
Hromadný přenos	U tohoto typu příkazu mluvíme již o datovém přenosu, první bajt po přijetí příkazu definuje rozměr datového paketu (max 255 bajtů) a poté již následuje vlastní odesílání jednotlivých datových bajtů.

Pokud je příkaz větší než 0xEF, jedná se o hromadný přenos, pokud je instrukce v rozmezí 0xA0 – 0xEF, jedná se o parametrový příkaz a pokud je hodnota příkazu pod 0x9F jedná se o bezparametrový příkaz.

5.4.2 Adresy zařízení připojených do sběrnice I²C

Do sběrnice I²C jsou zapojeny tři zařízení - počítač (přes převodník), nižší úroveň řízení a rozhraní pro styk s obsluhou, následující tabulka definuje jejich sedmibitové adresy.

Zařízení	Adresa
Převodník RS-232/I ² C (Počítač PC)	0x02 (0b0000001x)
Nižší úroveň řízení	0x04 (0b0000010x)
Rozhraní pro styk s obsluhou	0x08 (0b0000100x)

5.4.3. Protokol nižší úrovně řízení

Nižší úroveň řízení pracuje s konstrukcí manipulátoru. Do sady příkazů, které tento program zpracovává, patří:

- Regulace otáček stejnosměrných motorů
- Regulace otáček krokového motoru
- Zapínání/Vypínání jeřábního světla a elektromagnetu
- Zpracovávání stavu senzoriky

Úplná sada příkazů nižší úrovně řízení:

Směr	Příkaz	Popis
Příjem	0x10	Tento příkaz vysílá ovládací program při připojování k manipulátoru. Program MCU tento pozdrav vrátí v podobě příkazu 0x20.
	0x41	Zapnutí elektromagnetu manipulátoru
	0x31	Zapnutí jeřábního světla
	0x42	Vypnutí elektromagnetu
	0x32	Vypnutí jeřábního světla
	0xA1 (+Parametr)	Zapnutí stejnosměrného motoru pro otáčení konstrukcí. Prvních 7 bitů parametru určuje rychlost otáčení, nejnižší bit určuje směr.
	0xA2 (+Parametr)	Zapnutí stejnosměrného motoru pro posuv kočky. Prvních 7 bitů parametru určuje rychlost otáčení, nejnižší bit určuje směr.
	0xA3 (+Parametr)	Zapíná motor navijáku. Prvních 7 bitů parametru určuje rychlost otáčení, nejnižší bit určuje směr.
Vysílání	0x20	Odpověď na pozdrav od počítače
	0x6-	Odeslání stavu senzorů manipulátoru (nižší 4 bity určují stavy)

5.4.4. Protokol pro komunikaci s displejem

Současný software pro komunikaci s řadičem displeje dovoluje zapisovat a číst data zobrazená na displeji, ukládat vlastní znaky a navíc komunikovat s řadičem klávesnice. Rozhraní je připojeno do systému manipulátoru na sběrnici I²C jako slave zařízení.

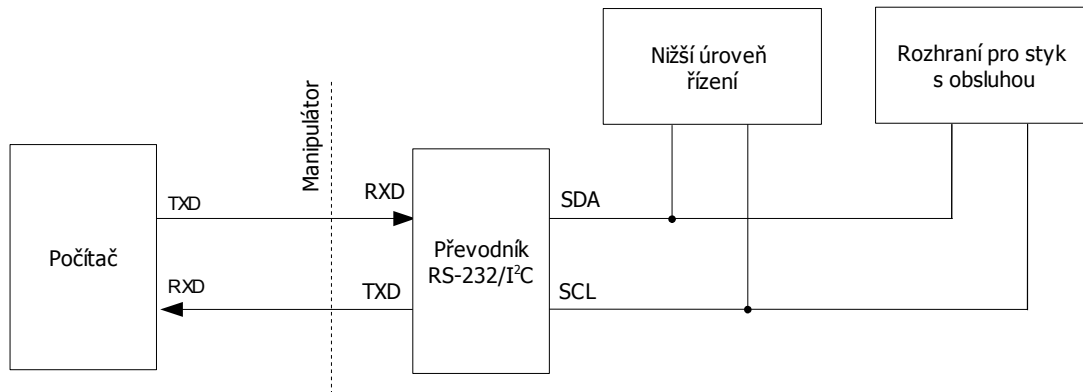
Způsob přenosu dodatečných dat jednotlivých příkazů je uveden v kapitole 5.4.1. Způsoby přenosu dat

Úplná sada příkazů rozhraní pro styk s obsluhou:

Směr	Příkaz	Popis
Příjem	0xF0	Vypisuje data na aktuální pozici kurzoru na displeji, využívá hromadný přenos.
	0xF2	Ukládá uživatelsky definovaný znak, první bajt datového paketu definuje pozici znaku v CGRAM paměti řadiče (max hodnota 8), následuje 8 datových bajtů se znakem. využívá hromadný přenos.
	0xA0	Nastavuje X pozici kurzoru na displeji
	0xA1	Nastavuje Y pozici kurzoru na displeji
	0xA2	Nastavuje podsvícení displeje (0x00 – min, 0xFF - max)
	0x10	Zapíná kurzor
	0x11	Vypíná kurzor
Vysílání	0x6-	Vrací stav klávesnice (nižší 4 bity; pokud od posledního dotazu nedojde ke změně, zařízení neodpoví)
	0xF1	Čte všechna data z paměti displeje, využívá hromadný přenos.
	0xF3	Čte všechny uživatelské znaky v uložení v paměti CGRAM

5.5. Vnitřní systém manipulátoru

Tento systém tvoří komunikační síť mezi jednotlivými částmi manipulátoru. Nižší úroveň řízení a rozhraní pro styk s obsluhou (displej + klávesnice) pracují v módu I²C „slave“, Převodník RS232/I²C tvoří jakousi vstupní bránu počítače do systému, počítač je ovládající, tudíž převodník pracuje v režimu I²C „master“.



Blokové schéma propojení manipulátoru

5.6. Generování PWM signálů

Pro generování PWM signálu používáme zabudované časovače mikrokontroléru, které mohou pracovat v osmibitovém „Fast PWM“ módu. Mikrokontrolér poskytuje ideální úroveň zobecnění tím, že stačí inicializovat časovač, a poté už jen nastavujete hodnotu středy PWM signálu (Viz kapitola 4.5.1. - Pulzně-šířková modulace).

Příklad inicializace časovače a generování PWM:

```
; Inicializace PWM kanálu 1 (pro jeřábní světlo)
ldi r16, (1<<WGM21) | (1<<WGM20) | (1<<COM21) | (1<<CS20)
out TCCR2, r16

; Inicializace PWM kanálu 2 (Motory 1 a 2)
ldi r16, (1<<COM1A1) | (1<<COM1B1) | (1<<WGM10)
out TCCR1A, r16
ldi r16, (1<<WGM12) | (1<<CS10)
out TCCR1B, r16

; Střední hodnotu napětí na PWM kanálu 1 nastavit na 50%
ldi r16, 0x7F
out OCR2, r16

; Střední hodnotu napětí na PWM kanálu 2 nastavit na minimum
ldi r16, 0x00
out OCR1AL, r16

; Střední hodnotu napětí na PWM kanálu 3 nastavit na maximum
ldi r16, 0xFF
out OCR1BL, r16
```

Pomocí instrukcí:

```
ldi r16, (1<<WGM21) | (1<<WGM20) | (1<<COM21) | (1<<CS20)
out TCCR2, r16
```

Spustíme časovač nastavením bitu CS20 pro zdroj hodinových impulsů z MPU mikrokontroléru. Pomocí bitů WGMx nastavujeme časovač jako generátor „Fast PWM“ signálu a nakonec bit COM21 zapne neinvertovaný PWM signál. Poté jen nastavujeme hodnotu střidy do registru, který se porovnává s aktuální hodnotou daného časovače, pokud výsledek porovnání je kladný, stav na výstupu časovače se překlápí. Když dojde k přetečení časovače, stav výstupu se opět překlápí, a celý cyklus se opakuje (doba periody je přibližně $4 \cdot 10^{-5} = 25\text{kHz}$).

5.7. Komunikace s řadičem HD44780

Většina alfanumerických displejů v dnešní době používá řadič firmy HITACHI HD44780, nebo jeho obdobu (Více o zapojení tohoto řadiče - viz. kapitola 4.4 Práce s alfanumerickým displejem).

Přes datovou sběrnici (DB7:0) lze na displej zapisovat a zároveň z něj i číst v závislosti na stavu linky R/W. Před zapojením s processorem musíme zvolit datovou komunikaci. Ta může být čtyřbitová (DB7-DB4), nebo osmibitová (DB7-DB0). Po připojení se musí displej inicializovat nastavením směru psaní, posunu řádku, pozice a blikání kurzoru.

Pokud použijeme osmibitovou komunikaci, zapíšeme všech 8 bitů a potvrdíme impulsem na lince E (enable). Pokud máme nedostatek vývodů na mikrokontroléru, použijeme čtyřbitovou sběrnici. Při odesílání musíme nejdříve zapsat vyšší 4 bity, ty potvrdit a poté zapsat nižší 4 bity a znovu potvrdit. Čtyřbitová sběrnice je tedy 2x pomalejší, jelikož se celý proces musí 2x opakovat.

HIGH-ORDER 4 BIT	LOW- ORDER 4 BIT	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111	
xxxx0000	CG RAM (1)					0	Q	P	`	P	-	2	E	x	p
xxxx0001	(2)					!	1	A	Q	a	q	7	4	ä	q
xxxx0010	(3)					"	2	B	R	b	r	"	ı	ı	p
xxxx0011	(4)					#	3	C	S	c	s	#	ı	ı	e
xxxx0100	(5)					\$	4	D	T	d	t	\$	ı	ı	p
xxx0101	(6)					%	5	E	U	e	u	%	ı	ı	ü
xxx0110	(7)					&	6	F	V	f	v	&	ı	ı	p
xxxx0111	(8)					'	7	G	W	w	7	'	ı	ı	q
xxxx1000	(1)					<	8	H	X	x	x	<	ı	ı	ı
xxxx1001	(2)					>	9	I	Y	y	9	>	ı	ı	y
xxxx1010	(3)					*	:	J	Z	j	z	*	:	ı	j
xxxx1011	(4)					+	;	K	[k	[+	;	ı	ı
xxxx1100	(5)					,	<	L	ı	ı	ı	,	<	ı	ı
xxxx1101	(6)					=	=	M]	m	=	=	ı	ı	ı
xxxx1110	(7)					.	>	N	^	n	^	.	>	ı	ı
xxxx1111	(8)					/	?	O	_	o	_	/	?	ı	ı

Znaková sada displeje s řadičem HD44780

Řadič má dvě RAM paměti CGRAM a DDRAM. CGRAM (Character Graphics RAM) je osmibajtová paměť pro uložení vlastních znaků, DDRAM (Display Data RAM) obsahuje zobrazovaný text. Každá pozice na displeji má svou vlastní adresu v paměti DDRAM, kde je uložen bajt definující znak ze znakové sady (viz. obrázek). Rozložení znaků v paměti DDRAM je přímo závislé na použitém displeji (resp. na jeho rozměrech). Manipulátor využívá displeje s rozměry 20x4 znaků, proto rozložení znaků v paměti DDRAM je následující:

1. řádek	00h	01h	...	0Fh	10h	11h	12h	13h
2. řádek	40h	41h	...	0Fh	50h	51h	52h	53h
3. řádek	14h	15h	...	23h	24h	25h	26h	27h
4. řádek	54h	55h	...	63h	64h	65h	66h	67h

Pro vypsání znaku na třetí pozici druhého řádku (adresa DDRAM 42h), za předpokladu že máme displej inicializovaný, musíme nejdříve nastavit RS = R/W = 0. Poté po sběrnici odeslat bajt ve tvaru: 11000010b = 82h (předepsaný tvar 1XXXXXXX, kde nižších 7 bitů představuje adresu v DDRAM). Tím nastavíme ukazatel v paměti DDRAM na hledanou třetí pozici druhého řádku (adresa 42h). Nyní odešleme číslo znaku ze znakové sady a přijatá hodnota se uloží na pozici ukazatele. Pokud vše proběhlo v pořádku, na displeji se daný znak objeví.

Existují i další příkazy, podobné ukládání znaků do DDRAM, ty najdete například v [28], nebo [21].

5.8. Čtení stavů klávesnice

Algoritmus, který čte data z maticové klávesnice tvoří určitý logický problém, způsob jeho řešení má úzkou spojitost s kreativitou a výkonností vývojáře, který je autorem zdrojových kódů. Jak již bylo uvedeno v úvodu, účelem tohoto projektu je prezentovat principy, proto jsme si kladly za cíl vytvořit kvalitní a univerzální zdrojové kódy pro práci s různými druhy maticových klávesnic. Zdrojové kódy klávesnice jsou volně ke stažení pod lincencí BSD (GNU GPL kompatibilní) na internetových stránkách tohoto projektu. Tyto kódy jsou napsány zcela univerzálně, lze je použít pro jakoukoliv maticovou klávesnici do velikosti 8x8 (znaky x řádky).

Pro zprovoznění programu s jiným typem mikrokontroléru, nebo klávesnicí je nutné jen upravit několik konstant. Jde zejména o nastavení rozměrů klávesnice a používané kontakty na bránách. Také je možné nastavit počet měření algoritmu pro odhalování zámků (viz níže). Výstup měření klávesnice je paralelní čtyřbitovou binární hodnotou, tyto hodnoty se dají nastavit také.

Nastavení klávesnice (hodnoty jsou smyšlené, pro demonstrativní účely):

```

; DŮLEŽITÉ: Tyto konstanty určují pin portu, který je bitově
; nejníže. Program si zabere potřebný počet pinů od
; zadaného výše (v závislosti na rozměrech klávesnice).
.equ RW1 = PC0           ; Vstup stavu řádek
.equ C1 = PD5            ; Výstup na sloupce
.equ O1 = PD0            ; Paralelní výstup stavu

```

```

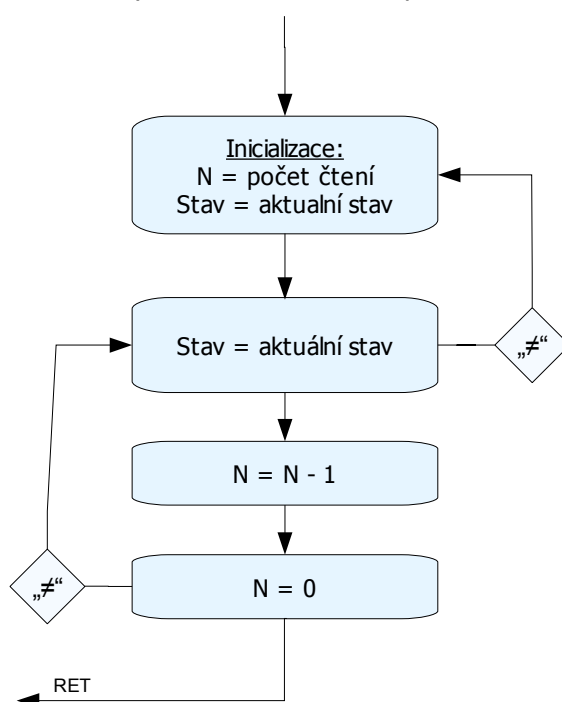
;   Nastavení rozměrů klávesnice
.equ KeyboardRows = 4           ; Počet sloupců klávesnice
.equ KeyboardCols = 3           ; Počet řádek klávesnice

;   Nastavení citlivosti detekce zákmitů
.equ AntiGlim = 10

```

Používání těchto klávesnic není zdaleka bezproblémové, každá klávesa má svůj přechodový odpor a při stisku dochází k tzv. zákmitům. Přechodový odpor je většinou v rozsahu 10-300Ω, má často za následek chybu při snímání stavu stisku. Zákmit je definován jako přechodný stav klávesy před jejím plným stiskem, trvá typicky 16-20 ms, mikrokontrolér může tento přechodný stav chybně vyhodnotit jako stisk (čím vyšší kmitočet MPU, tím větší má mikrokontrolér i rozlišovací schopnost).

Ošetření zákmitů na klávesnici probíhá pomocí jednoho z nejznámějších algoritmů – pomocí N měření. S ohledem na typickou dobu trvání zákmitu, přečteme stav vstupu N krát a pokud se stav za tu dobu (minimálně 30ms) nezmění, vyloučíme zákmitu na vstupu.



Blokové schéma algoritmu pro čtení klávesnice

6. Výsledky

Výsledkem naší práce je prezentace principů mechatroniky. V elektronice jde například o regulaci otáček stejnosměrných motorů pomocí pulzně-šířkové modulace (PWM) a regulaci výkonu krokového motoru pomocí pulzního proudového zdroje - „chopperu“, komunikaci po sběrnících RS232C a I²C, atd.. V programování jsme prezentovali rozdíly dvou nepoužívanějších programovacích jazyků pro mikrokontroléry, poukázali na použitá softwarová řešení s důrazem na využití zabudovaných periférií mikrokontrolérů.

7. Závěr a diskuze

Výroba strojních součástí a sestavování konstrukce nám přinesla mnoho pozitivních i negativních zkušeností. I když byl jeřáb pečlivě nakreslen v programu AutoCad, jeho stavba přinesla nové zkušenosti.

Naším cílem je prezentovat principy regulace otáček a výkonu různých typů motorů, proto jsme přišli s řešeními na vysoké technické i technologické úrovni, které se mohou rovnat s průmyslovými standardy. Mezi další řešení na vysoké technologické úrovni se řadí například komunikace po sběrnících RS232C a I²C, nebo komunikace s řadičem alfanumerického displeje.

Abychom prezentovali výhody a nevýhody různých programovacích jazyků, při psaní softwaru jsme se rozhodli psát programy pro mikrokontroléry ve dvou jazycích – Assembleru a ANSI C. Zdrojové kódy pro mikrokontroléry mají celkově kolem jednoho tisíce řádek. Zdrojové kódy řídicí aplikace jsou napsány v plně objektovém vysokoúrovňovém platformním jazyce.

Příloha A – Softwarové licence

Jsme autory všech programů ve formě binární i zdrojového kódu. Zdrojové kódy pro ovládací program manipulátoru z PC a vybrané zdrojové kódy programů mikrokontrolérů jsou publikovány pod veřejnou licencí BSD (kompatibilní s GNU GPL). Tato licence je jednou z nejsvobodnějších. Umožňuje volné šíření licencovaného obsahu, přičemž vyžaduje pouze uvedení autora a informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo.

Prakticky jsme vybrali tuto licenci proto, že ji může kdykoliv a kdokoliv svobodně využívat v freewarově i sharewarové licencovaných dílech se zárukou, že dílo nevydá za své.

Neoficiální český překlad licence:

(rok a jméno autora jsou smyšlené, slouží pro demonstrativní účely)

Copyright © 2007, Tomáš Novák. Všechna práva vyhrazena.

Redistribuce a použití zdrojových i binárních forem díla, v původním i upravovaném tvaru, jsou povoleny za následujících podmínek:

- *Šířený zdrojový kód musí obsahovat výše uvedenou informaci o copyrightu, tento seznam podmínek a níže uvedené zřeknutí se odpovědnosti.*
- *Šířený binární tvar musí nést výše uvedenou informaci o copyrightu, tento seznam podmínek a níže uvedené zřeknutí se odpovědnosti ve své dokumentaci a/nebo dalších poskytovaných materiálech.*
- *Ani jméno vlastníka práv, ani jména přispěvatelů nemohou být použita při podpoře nebo právních aktech souvisejících s produkty odvozenými z tohoto software bez výslovného písemného povolení.*

TENTO SOFTWARE JE POSKYTOVÁN DRŽITELEM LICENCE A JEHO PŘÍSPĚVATELI „JAK STOJÍ A LEŽÍ“ A JAKÉKOLIV VÝSLOVNÉ NEBO PŘEDPOKLÁDANÉ ZÁRUKY VČETNĚ, ALE NEJEN, PŘEDPOKLÁDANÝCH OBCHODNÍCH ZÁRUK A ZÁRUKY VHODNOSTI PRO JAKÝKOLIV ÚČEL JSOU POPŘENY. DRŽITEL, ANI PŘÍSPĚVATELÉ NEBUDOU V ŽÁDNÉM PŘÍPADĚ ODPOVĚDNI ZA JAKÉKOLIV PŘÍMÉ, NEPŘÍMÉ, NÁHODNÉ, ZVLÁŠTNÍ, PŘÍKLADNÉ NEBO VYPLÝVAJÍCÍ ŠKODY (VČETNĚ, ALE NEJEN, ŠKOD VZNIKLYCH NARUŠENÍM DODÁVEK ZBOŽÍ NEBO SLUŽEB; ZTRÁTOU POUŽITELNOSTI, DAT NEBO ZISKŮ; NEBO PŘERUŠENÍM OBCHODNÍ ČINNOSTI) JAKKOLIV ZPŮSOBENÉ NA ZÁKLADĚ JAKÉKOLIV TEORIE O ZODPOVĚDNOSTI, AŽ UŽ PLYNOUCÍ Z JINÉHO SMLUVNÍHO VZTAHU, URČITÉ ZODPOVĚDNOSTI NEBO PŘEČINU (VČETNĚ NEDBALOSTI) NA JAKÉMKOLIV ZPŮSOBU POUŽITÍ TOHOTO SOFTWARE, I V PŘÍPADĚ, ŽE DRŽITEL PRÁV BYL UPOZORNĚN NA MOŽNOST TAKOVÝCH ŠKOD.

Příloha B – Slovník použitých výrazů

MCU	– <i>Micro-Controller Unit</i>
MPU	– <i>Micro-Processing Unit</i>
I²C	– <i>Inter-Integrated Circuit</i>
TWI	– <i>Two Wire Interface</i>
SPI	– <i>Serial Peripheral Interface</i>
RS232C	– <i>EIA RS-232C</i>
UART	– <i>Universal Asynchronous Receiver/Transmitter</i>
USART	– <i>Universal Synchronous and Asynchronous Receiver/Transmitter</i>
ISP	– <i>In-System Programming</i>
DPS	– <i>Deska Plošných Spojů</i>
SMD	– <i>Surface Mounted Device</i>
LED	– <i>Light Emitting Diode</i>

Příloha C - Literatura

- [1] Kainka, B., Berndt, H.-J. Využití rozhraní PC pod Windows, nakladatelství HEL Ostrava, 2000, 151s., ISBN 80-86167-13-5
- [2] Kainka, B., Elektronika s podporou PC, nakladatelství HEL Ostrava, 2004, 184s., ISBN 80-86167-22-4
- [3] Novák P., Mobilní roboty, nakladatelství BEN Praha, 2005, 247s, ISBN 80-7300-141-1
- [4] Kesi, J., Elektronika I – Analogová technika, nakladatelství BEN Praha, 2006, 144s., ISBN 80-7300-143-8
- [5] Kesi, J., Elektronika III – Číslicová technika, nakladatelství BEN Praha, 2005, 112s., ISBN 80-7300-182-9
- [6] Maťátko, J., Elektronika, IDEA Servis Praha, 2005, 327s., ISBN 1866-067-05
- [7] Matoušek, D., Práce s mikrokontroléry ATMEL, 2. díl, nakladatelství BEN Praha, 2002, 304s., ISBN 80-7300-066-0
- [8] Matoušek, D., Práce s mikrokontroléry ATMEL, 5. díl, nakladatelství BEN Praha, 2006, 200s., ISBN 80-7300-205-1
- [9] Matoušek, D., Práce s inteligentními displeji LCD, 1. díl, nakladatelství BEN Praha, 2006, 224s., ISBN 80-7300-121-7
- [10] Pinker, J., Mikroprocesory a mikropočítače, nakladatelství BEN Praha, 2006, 160s., ISBN 80-7300-110-1
- [11] Juránek A., Hrabovský M., nakladatelství BEN Praha, 2005, 192s., ISBN 80-7300-177-2
- [12] Šmejkal, L., Martinásková, M., PLC a automatizace, nakladatelství BEN Praha, 1999, 224s., ISBN 80-86056-58-9
- [13] Kainka, B., USB – Měření, řízení a regulace pomocí sběrnice USB, nakladatelství BEN Praha, 2003, 248s., ISBN 80-7300-073-3
- [14] Matoušek, D., USB prakticky s obvody FTDI, nakladatelství BEN Praha, 2003, 272s., ISBN 80-7300-103-9
- [15] Váňa, V., Mikrokontroléry ATMEL AVR: Assembler, nakladatelství BEN Praha, 2003, 144s., ISBN 80-7300-093-8
- [16] Mičkal, K., Technická Mechanika I, INFORMATORIUM Praha, 1997, 212s.
- [17] Vlach J., Řízení a Vizualizace technologických procesů, nakladatelství BEN Praha, 1999, 159s., ISBN 80-86056-66-X
- [18] <http://atmel.com/>
Výrobce mikrokontrolérů AVR (anglicky, 17.3.2007)
- [19] http://atmel.com/dyn/resources/prod_documents/doc2486.pdf
Katalogový list mikrokontroléru ATmega8 od Firmy ATMEL (anglicky, 17.3.2007)
- [20] <http://robocraner.digitalbrain.cz>
Domovská stránka tohoto projektu (17.3.2007)
- [21] <http://www.cmail.cz/doveda/lcd/index.htm>
Popis nejpoužívanějšího řadiče alfanumerických LCD displejů (17.3.2007)
- [22] <http://www.elektrorevue.cz/clanky/05043/>
Popis krokových motorů (17.3.2007)
- [23] <http://en.wikipedia.org/wiki/RS-232>
Popis komunikačního protokolu RS232 (anglicky, 17.3.2007)
- [24] <http://micromouse.cannock.ac.uk/motors/L297.pdf>
Katalogový list L297 (anglicky, 17.3.2007)

- [25] <http://www.st.com/stonline/products/literature/ds/1773.pdf>
Katalogový list L298 (anglicky, 17.3.2007)
- [26] <http://www.st.com/stonline/books/pdf/docs/1328.pdf>
Katalogový list L293 (anglicky, 17.3.2007)
- [27] http://www1.jaycar.com.au/images_uploaded/CD4094BC.PDF
Katalogový list posuvného registru 4094 (anglicky, 17.3.2007)
- [28] <http://www.cmail.cz/doveda/lcd/hd44780u.pdf>
Katalogový list řadiče HD44780 (anglicky, 17.3.2007)
- [29] <http://www.elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatice.html>
Popis řadiče HD44780 (17.3.2007)
- [30] http://en.wikipedia.org/wiki/Surface-mount_technology
Popis technologie SMD (anglicky, 17.3.2007)
- [31] <http://www.lynxmotion.com/images/data/ghm06.pdf>
Katalogový list motoru použitého pro otáčení konstrukce (anglicky, 17.3.2007)
- [32] <http://www.lynxmotion.com/images/data/ghm07.pdf>
Katalogový list motoru použitého pro posuv kočky (anglicky, 17.3.2007)