

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program : B2612 – Elektrotechnika a informatika

Studijní obor : 2612R011 – Elektronické informační a řídicí systémy

Aplikace mikroprocesoru řady 8051 pro automatické krmítko

Application of chip microprocessor series 8051 for automatic haunt

Bakalářská práce

Autor:

Vedoucí práce:

Konzultant:

David Musil

Ing. Josef Grosman

Ing. Tomáš Martinec

UNIVERZITNÍ KNIHOVNA
TECHNICKÉ UNIVERZITY V LIBERCI



3146088058

V Liberci 1. 2. 2007

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Katedra: softwarového inženýrství

Akademický rok: 2005/2006

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jméno a příjmení: **David Musil**

studijní program: B 2612 – Elektrotechnika a informatika

obor: 2612R011 – Elektronické informační a řídicí systémy

Vedoucí katedry Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto bakalářskou práci:

Název tématu: **Aplikace mikroprocesoru řady 8051 pro automatické krmitko**

Zásady pro vypracování:

1. Seznamte se s monolitickými mikropočítači řady 8051
2. Navrhnete mikropočítačový systém s vybraným procesorem, který bude splňovat funkci dávkování potravy pro akvarijní rybičky:
 - pro zobrazení vyberte LCD displej, nebo vícemístný segmentový displej
 - dávkování realizujte pomocí krokového motorku
 - zvolte vhodné prvky pro nastavování hodnot
3. Návrh realizujte sestavením funkčního vzorku včetně programového vybavení

+CD

V 5/04 Mb

450, 14 a. přel.

KS1/
EIRS

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

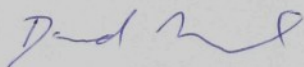
Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum 5.1.2007

Podpis



Poděkování

Na tomto místě bych rád poděkoval všem, kteří mi pomáhali a podporovali mě v práci zde popisované. Jmenovitě panu Ing. Josefu Grosmanovi, svému vedoucímu, za rady a vedení této práce, jenž přestože směřoval mou činnost ke konkrétnímu cíli, zachoval mi dostatečnou tvůrčí svobodu k uplatnění vlastních myšlenek a postupů. Dále panu Ing. Vladimíru Vikovi za rady při práci s programováním mikroprocesoru řady 8051. Také bych chtěl poděkovat svým rodičům za podporu, které se mi od nich dostávalo po celou dobu studia a v nemenší řadě též své přítelkyni za toleranci při vývoji a realizaci BP.

Abstrakt

Tato práce se zabývá návrhem a konstrukcí Automatického krmítka pro akvarijní ryby.

Skládá se ze dvou částí. V první je popsán návrh konstrukce a realizace automatické krmíčky. Ve druhé části je popsán software pro řízení vestavěného systému.

Jádrem automatického krmítka je mikroprocesor AT89C2051. Komunikace s uživatelem je zajištěna pomocí 16-místného LCD displeje a čtyř programovacích tlačítek. Aplikace krmení do nádrže je realizována pomocí krokového motoru Tandon.

Abstract

This work is about construction of Automatic Haunt for Fish.

It consists of two parts: First is about design, construction and realization of automatic haunt. Second part describes software which controls internal system.

The core of automatic haunt is a microprocessor AT89C2051. Communication with user is supported through four functional buttons and LCD display with 16 positions for numbers. Application of haunting to aquarium is realized with step motor Tandon.

Obsah

PROHLÁŠENÍ	3
PODĚKOVÁNÍ	4
ABSTRAKT	5
OBSAH	6
ÚVOD	8
1. AKVARISTIKA	8
2. KRMENÍ	8
AUTOMATICKÁ KRMIČKA RYB.....	9
3. POUŽITÝ HARDWARE.....	9
3.1. Mikrořadič - AT89C2051	9
3.1.1. Charakteristika	9
3.1.2. Flash paměť	9
3.1.3. RAM	10
3.1.4. Porty	10
3.1.5. Čítače / časovače	10
3.1.6. Přerušení	11
3.1.7. Sériový kanál	11
3.1.8. Analogový komparátor	11
3.1.9. Idle a power down mód	11
3.2. LCD displej - CM160100SRNNA-01.....	13
3.2.1. Charakteristika	13
3.2.2. Zapojení vývodů	13
3.2.3. Komunikace s modulem	13
3.2.4. Zobrazení znaků	14
3.2.5. Definování uživatelských znaků	14
3.2.6. Tabulka instrukcí	15
3.2.7. Popis instrukcí	16
3.2.8. Funkce resetu	18
3.3. Krokový motor – TANDON KP4M2-203.....	20
3.3.1. Technické parametry.....	20
3.3.2. Princip funkce.....	20
3.3.3. Bipolární řízení.....	20
3.3.4. Čtyřtaktní způsob ovládání.....	21
4. POUŽITÝ SOFTWARE	22
4.1. Textový editor	22
4.2. Překladač ASM 51	22
4.3. PAtmel X	22
4.4. Eagle 4.16.....	23
4.4.1. Tvorba schématu.....	23
4.4.2. Tvorba desky plošného spoje.....	23
5. MECHANICKÁ KONSTRUKCE	25
5.1. Možnosti řešení	25
5.1.1. S otočným válcem.....	25

5.1.2.	<i>S Archimédovým šroubem</i>	26
5.1.3.	<i>S posuvným pístem</i>	27
5.2.	Deska plošných spojů	28
5.3.	Krabice.....	30
5.4.	Krycí fólie předního panelu	30
5.5.	Aplikátor krmiva	31
6.	ŘÍDÍCÍ JEDNOTKA	32
6.1.	Úvod	32
6.2.	Popis zapojení řídicí jednotky.....	33
6.3.	Seznam součástek.....	35
6.4.	Ochranné prvky	36
7.	PROGRAM V ASEMLERU	38
7.1.	Obecný popis programu	38
7.2.	Struktura a ovládání programu.....	38
7.3.	Architektura programu.....	40
	MOŽNÁ ZLEPŠENÍ	44
	ZÁVĚR	44
	SEZNAM POUŽITÉ LITERATURY	45
	OBSAH PŘÍLOŽENÉHO CD-ROM	45
	PŘÍLOHA 9.01 – Znaky uložené v ROM paměti LCD displeje	46
	PŘÍLOHA 9.02 – Schéma řídicí jednotky	47
	PŘÍLOHA 9.03 – Foto osazené DPS	48
	PŘÍLOHA 9.04 – Foto krabice s předním panelem	49
	PŘÍLOHA 9.05 – Foto kompletního aplikátoru s motorem	50
	PŘÍLOHA 9.06 – Výkres krabice	51
	PŘÍLOHA 9.07 – Výkres aplikátoru a hřídele	52
	PŘÍLOHA 9.08 – Zdrojový kód programu	53

Úvod

1. Akvaristika

Akvaristika patří mezi krásné záliby a mnozí se k ní obracejí s touhou mít ve své blízkosti alespoň malý kousek živé přírody. Správně zařízené, udržované a prosperující akvárium, vhodně začleněné do bytového interiéru, působí jako nádherný, živý obraz, který se stane ozdobou našeho domova.

2. Krmení

Krmení a výživa akvariijních ryb patří k jednomu ze základních předpokladů jejich úspěšného chovu. Mezi základní pravidla při krmení našich ryb patří především zajištění co největší různorodosti a rozmanitosti podávaného krmiva. Druh a velikost krmiva má odpovídat chovaným rybám. Je důležité ryby nepřekrmovat. Ve svém přirozeném prostředí ryby nemají neustále takovou hojnost potravy jako v akváriu a k jejímu získání pak vydávají mnohem více energie a času než ryby chované v zajetí.

Převážná většina ryb, vhodných do společného akvária, jsou všežravci. Podáváme jim živou, rostlinou i umělou potravu. Dospělé ryby postačí krmit jednou denně, případně 2x denně v menším množství. Dáváme jim jen tolik krmiva, kolik stačí spotřebovat do 5 - 10 minut.

Při dlouhodobém podávání nadměrného množství potravy dochází ke ztloustnutí ryb. Přebytké, nespotebované krmivo se v akváriu kazí a rozkládá spolu s výkaly ryb. Odpadní látky poté zatěžují vodu. Může dojít k zakalení, přemnožení řas, až k narušení biologické rovnováhy, kdy při nedostatku kyslíku pak hrozí i úhyn ryb.

Automatická krmička pro aplikaci krmení je schopna ušetřit značnou část lidské práce. Zajistit pravidelnost krmení a snížit spotřebu krmiva, díky přesnějšímu dávkování a minimálním ztrátám oproti ručnímu krmení. Zajímavé je také posouzení vlivu automatického krmení na rychlost růstu ryb a jeho srovnání s ručním krmením.

Automatická krmička ryb

3. Použitý hardware

3.1. Mikrořadič - AT89C2051

3.1.1. Charakteristika

Jde o klon známého mikrořadiče řady 8051, původně vyráběného firmou Intel, v provedení DIL20. Výrobcem je firma Atmel. Tento obvod je instrukčně shodný s původním obvodem 8051. V porovnání s ním má menší počet vstupních a výstupních linek a zároveň je obohacen o několik dalších vlastností. Zde je přehledný seznam vlastností mikrořadiče.

- ❖ 2KB Flash paměti pro kód
- ❖ 128 bytů RAM, 128 bytů registrů spec.funkcí (SFR) z toho 128 bitů v bitově adresovatelné části paměti
- ❖ 15 vstupně / výstupních linek (2 porty po 8 a 7 bitech)
- ❖ dva 16 bitové čítače / časovače
- ❖ 5 zdrojů přerušení
- ❖ programovatelný plně duplexní sériový kanál (UART)
- ❖ analogový komparátor

Mikrořadič lze provozovat při napětích v rozsahu 2.7 – 6V s taktovací frekvencí 0 – 24MHz. Obvod má možnost být uveden do dvou typů režimů se sníženým příkonem (tzv.Idle a Power down mód). Porty mikrořadiče jsou navrženy tak, že jimi lze přímo budit zařízení s odběrem do 20mA (např.LED).

3.1.2. Flash paměť

Jinak zvaná PEROM (Programmable and Erasable Read Only Memory) je elektricky programovatelná a mazatelná paměť pro uživatelský program. Výrobce zaručuje 1000 cyklů zápis / mazání a udržení dat 10 let. Tato paměť

umožňuje velice snadnou a rychlou změnu uživatelského programu. Celý cyklus (tj. mazání a nový zápis + ověření) netrvá více než několik desítek sekund.

3.1.3. RAM

Paměť pro čtení a zápis (Random Access Memory). Procesor má 128 x 8 bitů této vnitřní paměti pro data programu (v této části je realizován i zásobník, 4 sady po osmi registrech R0 až R7 a 128 přímo adresovatelných bitů). Dalších 128 bytů je využito pro tzv. registry speciálních funkcí, které umožňují ovládání různých periferních a řídicích obvodů mikrořadiče.

3.1.4. Porty

Mikrořadič má celkem 15 vstupně / výstupních linek. Je-li do odpovídajícího registru portu zapsána logická 1, lze tento využít jako vstupní. Některé z těchto linek plní zároveň jiné funkce, jako např. sériový vstup a výstup, vstupy analogového komparátoru a zdroje vnějších přerušení. Kromě vstupů P1.0 a P1.1 (vstupy analogového komparátoru) mají všechny ostatní vstupy vnitřní pull-up odpory.

3.1.5. Čítače / časovače

Jde o dvojici šestnáctibitových registrů, které mohou fungovat v režimu čítačů vnějších událostí nebo v režimu časovačů. Při provozu v režimu čítače událostí se obsah registru zvyšuje, při každém přechodu z logické 0 do 1 na vstupu T0 (T1). Maximální rychlost přičítání je rovna 1/24 frekvence oscilátoru. Ve funkci časovače lze použít jeden ze čtyř možných režimů.

- ❖ režim 0 – osmibitový čítač s předřazeným děličem 32
- ❖ režim 1 – šestnáctibitový čítač
- ❖ režim 2 – osmibitový čítač s automatickým novým naplněním
- ❖ režim 3 – čítač 0 se chová jako dva oddělené osmibitové čítače, čítač 1 při přepnutí do tohoto režimu uchovává načítanou hodnotu

Rychlost přičítání v těchto režimech je vždy rovna $1/12$ frekvence oscilátoru (v režimu 0 samozřejmě ještě vydělená 32).

Časovač 1 lze také využít při sériovém přenosu jako generátor přenosové rychlosti.

3.1.6. *Přerušení*

U tohoto procesoru je k dispozici pět zdrojů přerušení, z toho dvě vnější. Vnější přerušení lze vyvolat od vstupů INTO a INT1 buď logickou úrovní nebo její změnou. Vnitřní přerušení vznikají od přetečení časovačů T0, T1 a od sériového kanálu.

3.1.7. *Sériový kanál*

Jde o plně duplexní sériový port, který umožňuje současné vysílání i přijímání znaků přes výstup TXD a vstup RXD. Na rozdíl od sériového rozhraní RS - 232C jsou napěťové úrovně kompatibilní s TTL logikou a je proto nutno při vzájemném spojování těchto rozhraní vytvořit vhodný interface pro převod napěťových úrovní.

3.1.8. *Analogový komparátor*

Jde o specialitu, která na standardním mikrořadiči 8051 není realizována. Vstupy tohoto komparátoru jsou vyvedeny na nejnižší dva bity portu P1 a výstup z komparátoru je vnitřně zaveden na 6.bit portu P3, který tudíž není vyveden ven z pouzdra mikrořadiče. Kvůli tomuto komparátoru též nejsou vnitřně posíleny porty P1.0 a P1.1. Komparátor je schopen zpracovávat pouze kladná vstupní napětí v rozsahu 0V až do V_{CC} (napájecí napětí).

3.1.9. *Idle a power down mód*

V režimu Idle (čekání) se hodinový signál odpojí od procesoru, ale nadále časuje obvody přerušení, časovačů a sériového portu. V tomto režimu se zachovává úplný stav mikrořadiče. Tento režim je možno ukončit buď přerušením nebo resetem obvodu.

3.2 V režimu Power down (se sníženým napájením) se úplně zastaví oscilátor a s ním i všechny funkce. V tomto režimu se zachovává obsah SFR registrů, vnitřní RAM a portů. Přitom lze snížit napájecí napětí a tím ještě více minimalizovat odběr. Jediným ukončením tohoto stavu je reset, který znovu definuje obsah SFR, ale zachovává obsah paměti RAM.

Toto byl pouze velmi stručný popis možností tohoto mikrořadiče. Podrobnější popis lze nalézt například v [1] nebo [2].

3.2. LCD displej - CM160100SRNNA-01

3.2.1. Charakteristika

Každý displej má v sobě speciální integrovaný obvod - řadič, který ovládá celý displej a komunikuje s okolím. Standardem se stal řadič HD44780 nebo jeho ekvivalent. Proto je jedno od jakého výrobce displej je. Pro dosažení plné kompatibility displeje, je zapojení přírodního konektoru stejné.

Displej se vyrábí v několika variantách 8x2, 16x1, 16x2, 16x4, 20x2, 20x4, 24x2, 24x4, 40x2, 40x4, které se liší počtem sloupců a řádků. Moduly se vyrábí s podsvícením LED nebo výbojkou.

Pro Automatickou krmičku ryb byl zvolen LCD displej 16x1. K zobrazování hodin ve tvaru hh:mm:vv je potřeba 8 znaků a pro informační popisky sejevilo dalších 8 znaků jako plně dostačující.

3.2.2. Zapojení vývodů

Číslo vývodu	Označení	Popis
1	V _{SS}	napájení GND
2	V _{DD}	napájení +5V
3	V _O	Kontrast 0 – 5V
4	RS	Register Select (0=instrukce, 1=data)
5	R / W	Read / Write (0=zápis, 1=čtení)
6	E	Enable (1, hrana z 1 -> do 0)
7	DB0	Data Bus 0
8	DB1	Data Bus 1
9	DB2	Data Bus 2
10	DB3	Data Bus 3
11	DB4	Data Bus 4
12	DB5	Data Bus 5
13	DB6	Data Bus 6
14	DB7	Data Bus 7
15	A	podsvícení - anoda LED
16	K	podsvícení - katoda LED

3.2.3. Komunikace s modulem

Pro komunikaci s displejem je potřeba minimálně 6 vodičů až maximálně 11 vodičů a napájení +5V. Před zapojením s procesorem si musíme zvolit

datovou komunikaci. Komunikace s modulem může být po 8 - bitové sběrnici (DB0 - DB7) nebo 4 - bitové sběrnici (DB4 - DB7).

Po zapnutí procesoru a displeje je nutné provést základní inicializaci displeje. Nastaví se počet bitů datové komunikace (8 nebo 4 bity), směr psaní znaků, posun řádku, pozice kurzoru a blikání kurzoru. Po této inicializaci je displej připraven na komunikaci.

- ❖ 8-bitová sběrnice (DB0 - DB7). Pošle se 8 bitů a zapíšou se Enablem
- ❖ 4-bitová sběrnice (DB4 - DB7). Tuto komunikaci je vhodné použít při nedostatku vývodů na procesoru. Komunikace bude 2x pomalejší, protože se data posílají nadvakrát. Nejříve se pošlou vyšší 4 bity a zapíšou se Enablem a pak nižší 4 bity a zapíšou se Enablem. Nižší 4 bity (DB0 - DB3) na displeji je potřeba spojit se zemí.

3.2.4. Zobrazení znaků

Každý znak je zobrazován jako matice 5x8 bodů. Definice jednotlivých znaků je uložena napevno ve vnitřní paměti ROM. Přehled znaků v *příloze 9.01*. Dále je možno zobrazit kurzor, nastavit blikání kurzoru, definovat posouvání zobrazených znaků, smazat displej, zobrazovat na konkrétní pozici apod.

V tabulce jsou uvedeny pozice v DDRAM (číslo je v hexadecimálním tvaru), na kterých jsou uloženy znaky pro zobrazení na řádku.

Pozice LCD	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Adresa DDRAM	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

3.2.5. Definování uživatelských znaků

Prvních 8 znaků (znak 0 - 7) je možno nadefinovat uživatelem třeba jako české znaky nebo různé grafické symboly podle vlastního přání. Pro češtinu je osm znaků málo, proto se znaky můžou měnit v generátoru znaků před jejich zobrazením. Tento postup přepisování znaků nám přepíše i zobrazený znak (0 - 7) na displeji za nově nadefinovaný. Velikost znaku, který zapisujeme do paměti je 8x8 pixel, ale zobrazí se jen 5x8 (šířka x výška).

3.2.6. Tabulka instrukcí

Instrukce	Signál										Čas
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
Smazat displej	0	0	0	0	0	0	0	0	0	1	1,64ms
Návrat na pozici 0	0	0	0	0	0	0	0	0	1	*	1,64ms
Nastavení módu	0	0	0	0	0	0	0	1	I/D	S	40μs
Kontrola displeje zap / vyp	0	0	0	0	0	0	1	D	C	B	40μs
Posun kurzoru, displeje	0	0	0	0	0	1	S/C	R/L	*	*	40μs
Nastavení funkce displeje	0	0	0	0	1	DL	N	F	*	*	40μs
Nastavení adresy CGRAM	0	0	0	1	CGRAM adresa						40μs
Nastavení adresy DDRAM	0	0	1	DDRAM adresa							40μs
Čtení příznaku Busy Flag a adresy	0	1	BF	CGRAM / DDRAM adresa							0μs
Zápis dat do CGRAM nebo DDRAM	1	0	zápis dat								40μs
Čtení dat z CGRAM nebo DDRAM	1	1	čtení dat								40μs

Vysvětlivky :

- DDRAM - Display Data RAM. (paměť zobrazovaných znaků)
- CGRAM - Character Generator RAM. (paměť pro uložení vlastního znaku)
- * - Na hodnotě nezáleží.

Název bitu	Popis	
I/D	0 - Snížení	1 - Zvýšení
S	0 - Není posun displeje	1 - Posun displeje při zápisu dat
D	0 - Vypnout displej	1 - Zapnout displej
C	0 - Vypnout kurzor	1 - Zapnout kurzor
B	0 - Vypnout blikání kurzoru	1 - Zapnout blikání kurzoru
S/C	0 - Posun kurzoru	1 - Posun displeje
R/L	0 - Posun doleva	1 - Posun doprava
DL	0 - 4bitová komunikace	1 - 8bitová komunikace
N	0 - 1 řádek	1 - 2 řádky
F	0 - 5x8 bodů	1 - 5x10 bodů
BF	0 - Operace je ukončena	1 - Operace probíhá

3.2.7. *Popis instrukcí*

❖ Smazat Displej (Clear Display)

Smaže displej. Zapiše kód 20H (kód 20H je prázdný znak) do všech adres DDRAM. Potom nastaví DDRAM adresu 0 adresového čítače a návrat displeje k jeho originálnímu stavu, jestliže byl posunutý. Jinými slovy, displej zmizí a kurzor nebo blikání směřuje displej doleva (v 1.řádce 2 - řádkového displeje). To také nastaví I / D na 1 (zvýšení) v nastavovacím módu. S se v módu nastavení nezmění.

❖ Vrátit se na začátek (Return Home)

Vrátí se na začátek. Nastaví adresu DDRAM na 0 a vrátí displej do jeho originálního stavu, jestliže byl posunut. Nezmění se obsah DDRAM. Kurzor nebo blikání směřuje displej doleva (v 1.řádce 2 - řádkového displeje).

❖ Vlož nastavení módu (Entry Mode Set)

I / D: Zvětší (I / D = 1) nebo zmenší (I / D = 0) adresu DDRAM o 1, když kód znaku je zapsán do nebo čten z DDRAM. Kurzor nebo blikající pohyb doprava zvětšen o 1 a doleva zmenšený o 1. Stejně platí při zápisu a čtení z CGRAM.

S: Posune celý displej o jeden znak doprava (I / D = 0) nebo doleva (I / D = 1), když S je 1. Displej se neposune, jestliže S je 0. Jestliže S je 1, bude se zdát jako by se kurzor nepohyboval, ale displej pracuje. Displej se neposune, když čte z DDRAM. Také psaní do nebo čtení z vnější CGRAM displej neposune.

❖ Displej zap / vyp (Display on / off control)

D: Displej zapnut když D je 1 a vypnut když D je 0. Když je vypnut, data displeje zůstanou v DDRAM, ale můžou být zobrazena okamžitě tím, že D je 1.

-
- C: Kurzor je zobrazen když C je 1 a nezobrazen když C je 0. Jestliže kurzor zmizí, funkce I / D nebo jiné specifikace se nebudou měnit během zápisu dat displeje. Kurzor je zobrazen pomocí 5 teček v 8 - mé řádce pro 5x8 bodů a v 11-té řádce pro 5x10 bodů.
- B: Znak ukázaný kurzorem bliká, když B je 1. Blikání je zobrazováno jako přepínání mezi celým prázdným místem a zobrazovaným znakem rychlostí 409,6 ms.

❖ Posun kurzoru nebo displeje (Cursor or display shift)

Posun kurzoru nebo displeje změní místo kurzoru nebo displeje doprava nebo doleva bez psaní nebo čtení zobrazovaných dat. Tato funkce opravuje nebo hledá na displeji. Na 2.řádek se kurzor přemístí z 1.řádky, když bude přes 40 - té číslo 1 řady. První a druhá řádka displeje se posouvá zároveň. Když zobrazovaná data jsou posouvána opakovaně, každá řádka se pohybuje jenom vodorovně. Z druhé řádky se neposune do první.

❖ Nastavení funkce (Function set)

- DL: Nastaví délku datového rozhraní. Data jsou poslaná nebo přijatá 8-bitově (DB7 - DB0) když DL je 1 a 4 - bitově (DB7 - DB4) když DL 0. Při vybrání 4 - bitové délky, data musí být poslána nebo přijata dvakrát.
- N: Nastaví počet řádků displeje.
- F: Nastaví velikost znaku.

❖ Nastavení adresy CGRAM (Set CGRAM address)

Nastavením adresy CGRAM dá CGRAM binární adresu AAAAAA do adresového čítače. Data jsou pak zapsána nebo čtena od MPU pro CGRAM.

❖ Nastavení adresy DDRAM (Set DDRAM address)

Nastavením adresy DDRAM dá DDRAM binární adresu AAAAAAA do adresového čítače. Data jsou pak zapsána nebo čtena od MPU pro DDRAM. Nicméně, když N je 0 (1 - řádkový displej), AAAAAAA může být 00H - 4FH. Když N je 1 (2 - řádkový displej), AAAAAAA může být 00H - 27H pro 1 řádek a 40H - 67H pro 2 řádek.

❖ Číst busy flag a adresy (Read busy flag & address)

Čtení busy flag a čtení adresy busy flag (BF) ukazuje, že systém vnitřně pracuje na předchozí přijaté instrukci. Jestliže BF je 1, vnitřní operace je v běhu. Příští instrukce nebude přijata do okamžiku než BF je 0. Kontrolovat BF stav před příštím zápisem operace. Ve stejný čas, hodnota adresový čítač v binární AAAAAAA je čten ven. Tento adresový čítač je použitý u obou CG a DDRAM adresy a jeho hodnota je předurčená předchozí instrukcí.

❖ Zapiš data do CG nebo DDRAM (Write data to CG or DDRAM)

Zapisuje data do DDRAM nebo CGRAM.

❖ Čti data z CG nebo DDRAM (Read data from CG or DDRAM)

Čte data z CGRAM nebo DDRAM.

3.2.8. *Funkce resetu*

Vnitřní reset se automaticky inicializuje po zapnutí řadiče HD44780 :

❖ Smazání displeje

❖ Nastavení funkce :

DL = 1 8 - bitový interface

N = 0 1 - řádkový displej

F = 0 velikost znaku 5x8 bodů

❖ Kontrola displeje zap/vyp :

D = 0 Displej : vyp

C = 0 Kursor : vyp

B = 0 Blikání kurzoru : vyp

❖ Nastavení módu :

I / D = 1 zvýšení

S = 0 posun displeje : vyp

Po zapnutí napájení a provedení vnitřního resetu se musí provést inicializace displeje. Nastaví se 8-bitová nebo 4-bitová komunikace a základní uživatelské nastavení. V následující tabulce je vypsána inicializace po jednotlivých instrukcích.

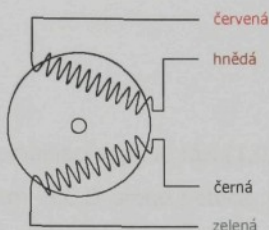
Inicializace pro 4 – bitovou komunikaci										Poznámka
Zapnutí napájení (Power On)										
čekej minimálně 15ms po náběhu V _{CC} nad 4,5V										Nelze testovat BF, nastaven 8-bitový interface
RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	1	1	-	-	-	-	
čekej minimálně 4,1ms										Nelze testovat BF, nastaven 8-bitový interface
RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	1	1	-	-	-	-	
čekej minimálně 100μs										Nelze testovat BF, nastaven 8-bitový interface
RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	1	1	-	-	-	-	
RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Čekej na BF=0 nebo 40μs, nastaven 4-bitový interface
0	0	0	0	1	0	-	-	-	-	
RS	R/W	D7	D6	D5	D4	D7	D6	D5	D4	2-řádky, font znaků 5x8
0	0	0	0	1	0	N(1)	F(0)	-	-	
0	0	0	0	0	0	1	0	0	0	Displej, kurzor, blikání vyp.
0	0	0	0	0	0	0	0	0	1	Smazat displej
0	0	0	0	0	0	0	1	I/D(1)	S(0)	Posun kurzoru do prava
0	0	0	0	0	0	1	1	0	0	Zapnout displej
Konec inicializace										

Toto byl jednoduchý popis možností tohoto LCD displeje. Podrobnější popis lze nalézt například ve [3].

3.3. Krokový motor – TANDON KP4M2-203

3.3.1. Technické parametry

❖ Počet fází :	2
❖ Odpor mezi fázemi :	37,5 Ohm
❖ Proud mezi fázemi :	0,15 A
❖ Napětí :	12 V
❖ Počet kroků :	280
❖ Úhel jednoho kroku :	1,8°
❖ Hmotnost :	270g



3.3.2. Princip funkce

Základním principem KM je pohyb rotoru o jeden krok. Krok je definován jako mechanická odezva rotoru KM na jeden řídicí impuls řídicí jednotky, při níž vykoná rotor pohyb z výchozí magnetické klidové polohy do nejbližší magnetické klidové polohy. Změna polohy se dosahuje změnou napájení vinutí jednotlivých fází statoru. Po sepnutí určité fáze se rotor snaží natočit tak, aby výsledný magnetický odpor byl minimální. U nezatíženého motoru se tedy sesouhlasí poloha zubů statoru a rotoru. V této poloze má motor nulový statický vazební moment a při vychýlení vnější zátěží moment stroje narůstá. Maximální hodnota statického vazebního momentu odpovídá natočení o čtvrtinu kroku. Jedno mechanické otočení hřídele KM o 360° představuje určitý počet kroků, jejichž počet je dán konstrukcí motoru a způsobem řízení.

3.3.3. Bipolární řízení

Při bipolárním řízení prochází proud vždy dvěma protilehlými cívkami. Ty jsou zapojené tak, že mají navzájem opačně orientované magnetické pole. Motor v tomto režimu poskytuje větší krouticí moment, ovšem za cenu vyšší spotřeby. Pro řízení jsou zapotřebí 2 H - můstky: pro každou větev jeden. To ve výsledku znamená jednak složitost zapojení a větší počet kontrolních linek (jejich počet lze zredukovat pomocí přídavné logiky).

V dnešní době je mnoho firem, které řeší tento problém integrováním všech požadovaných subsystémů na jeden čip. Existuje několik druhů integrovaných obvodů např. L293D a L298 od firmy SGS Thomson nebo MPC1710A od firmy Motorola. Jako výkonový obvod pro řízení krokového motoru byl vybrán L293D. Viz kapitola 6.2.

3.3.4. Čtyřtaktní způsob ovládání

U čtyřtaktního způsobu ovládání je první kombinací sepnutí fází (1,0,0,0), což odpovídá stavu, kdy fáze A teče proud a fázemi B,C,D proud neteče. Rotor se natočí tak, aby zaujal stav, kdy je minimální magnetický odpor (Minimální reluktance magnetického obvodu). Proti statorovým pólům, které jsou buzeny (pod fází A), se nastaví nejbližší póly rotoru do souhlasné polohy. Říkáme, že rotor je v magnetické klidové poloze. V této poloze je moment působící na rotor nulový. Sousední póly statoru se nekryjí s póly rotoru. Pod protilehlým statorovým pólem (také buzeným) se rotorový zub kryje se statorovým. Jestliže na vstup komutátoru přivedeme řídicí impuls, dojde ke změně kombinace sepnutí fází (0,1,0,0) tj. bude buzena pouze fáze B. Rotor se pootočí a zaujme novou magnetickou klidovou polohu, která opět odpovídá minimálnímu magnetickému odporu. Pro další kombinaci sepnutí fází (0,0,1,0) bude buzena pouze fáze C a rotor se opět pootočí o jeden krok do nové magnetické klidové polohy. Dalším řídicím impulsem dojde ke změně fází na (0,0,0,1) a k dalšímu pootočení rotoru o jeden krok. Následujícím řídicím impulsem se nastaví kombinace buzení fází (1,0,0,0), která odpovídá výchozí situaci a celý děj se cyklicky opakuje. Pohyb uvažovaného krokového motoru je vždy o $1,8^\circ$. Časový průběh buzení fází čtyřfázového KM v závislosti na řídicím signálu. Při tomto buzení je v každé periodě řídicího signálu magnetické pole buzeno jedno ze čtyř fází vinutí.

Podrobnější informace o krokových motorech najdete v [4] nebo [5].

4. Použitý software

4.1. Textový editor

Pro psaní programu byl použit obyčejný textový editor integrovaný ve Windows. Dokument byl uložen ve tvaru *.asm kvůli pozdější kompilaci.

4.2. Překladač ASM 51

Tento program se používá pro přeložení programu z formátu *.asm. Zkontroluje nám možný výskyt syntaktických chyb. Po přeložení z programovacího jazyka (assembler) do strojového kódu vytvoří soubor s příponou *.hex, který obsahuje kód programu v hexadecimálním tvaru.

Pro překlad kódu existuje mnoho programů, byl zvolen ASM 51. Je velmi přehledný a jeho ovládání je jednoduché.

4.3. PAtmel X

Programátor umožňuje všechny potřebné operace s podporovanými obvody, tj. čtení, zápis, mazání a verifikaci datové i programové paměti a nastavování lock i fuse bitů mikropočítače.

Ovládací program je na obsluhu velmi jednoduchý. Nejprve otevřeme příslušný HEX soubor, nastavíme typ mikrořadiče a stiskem příslušného tlačítka zahájíme operaci. K dispozici jsou volby:

- ❖ Reload - znovunačtení programovaného souboru (naposledy používaného)
- ❖ Read - načtení programu z mikroprocesoru
- ❖ Write - zápis programu do mikroprocesoru
- ❖ Verify - porovnání obsahu programovaného souboru s obsahem paměti mikroprocesoru
- ❖ Program! - kompletní proces programování, tedy výmaz paměti mikroprocesoru, zápis programu a následná verifikace
- ❖ Erase - vymazání paměti mikroprocesoru

Detailní návod k programátoru a k software najdete v [6].

4.4. Eagle 4.16

4.4.1. *Tvorba schématu*

Program Eagle je určen pro tvorbu schémat. Schéma je potom podkladem pro tvorbu plošného spoje.

Knihovny programu Eagle jsou velmi obsáhlé a velmi obtížně se součástky vyhledávají. I přes možnou aktualizaci jsem si musel některé součástky vytvořit sám.

Jednou z výhod tohoto návrhového systému je možnost použití labelů na propojovacích vodičích, aniž by bylo nutno tyto vodiče vést mezi příslušnými piny. Schéma se pak výrazně zpřehlední.

Po vytvoření schématu program umožňuje vygenerování výrobních podkladů, jako je seznam součástek, netlist atd. Více o programu v [7]

4.4.2. *Tvorba desky plošného spoje*

Pomocí tohoto programu je vytvořena konkrétní podoba desky plošného spoje. Vzhledem k ekonomickému hledisku byla zvolena dvouvrstvá deska se snahou o co nejmenší počet prokovů (vodivých spojení mezi vrstvami).

Po automatickém načtení, při splnění podmínky (ve schématu jsou ke všem součástkám přiřazena pouzdra), dojde k zobrazení pouzder všech součástek a vodivých čar mezi jednotlivými vývody tak, jak bylo vytvořeno ve schématu. Obrys desky byl nakreslen podle daných rozměrů ve vrstvě 21tPlace. A do tohoto obrysu byly umístěny konstrukční díry pro uchycení ke krabicí.

Nyní nastala téměř nejnáročnější práce. Rozmístit součástky do desky tak, aby při následném routování vodičů byli spoje co nejkratší a nemusely se hledat příliš složité cesty s množstvím prokovů.

Po rozmístění součástek se začali vytvářet vodivé spoje mezi jednotlivými součástkami. V programu Eagle lze využít Autorouter. Tato možnost byla ihned zamítnuta. Autorouter, zabudovaný ve verzi, který byl k dispozici, není tak dokonalý, aby vytvořil nekomplikovaný motiv. Spoje jsou příliš dlouhé s množstvím prokovů.

Velmi důležitým faktorem při návrhu byla volba šířky spojů. Pro nevýkonové signály postačila šířka spoje 16 mils (0,4mm), pro napájecí a výkonové spoje byla použita šířka 25 mils (0,6mm). Vše bylo routováno ručně. Návrh byl sice časově mnohem náročnější, ale spoje jsou pak mnohem přehlednější a bez chyb.

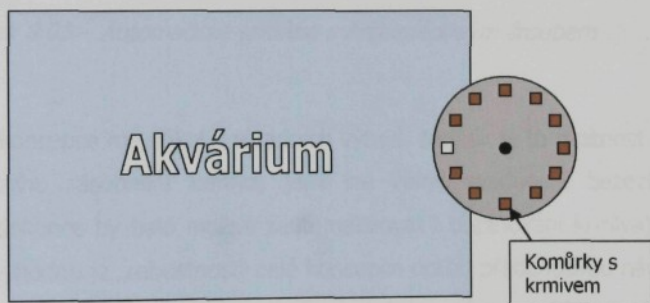


5. Mechanická konstrukce

5.1. Možnosti řešení

5.1.1. S otočným válcem

Na počátku návrhových prací bylo třeba vymyslet vhodnou mechanickou konstrukci krmícího zařízení. Možných řešení existuje mnoho. Po prozkoumání sériově vyráběných akvaristických zařízení se ukázalo, že s oblibou využívají struktury načrtnuté na *obrázku 8.02*. Zásobník s krmivem je tvořen válcem, otočným kolem své osy. V tomto válci jsou podélné komůrky naplněné krmivem (jedna komůrka představuje jednu dávku). Dolní podstava válce se otáčí po neprostupné podložce, ve které je pouze v jednom místě (nad akváriem) umístěn otvor pro vysypávání krmiva.



Obrázek 8.02 – Automatické krmítko s otočným válcem

Toto řešení má své výhody, které spočívají v jednoduchosti konstrukce a také v oddělitelnosti různých druhů krmiv.

Nevýhodou je, že obvod daného válce je omezen počtem komůrek pro krmení. A také při velmi malé zrnitosti krmiva by se mohlo stávat, že do mezery mezi otočným válcem s krmivem a napevno umístěnou podstavou se budou dostávat zrníčka krmiva, které mohou vést i k zaseknutí otočného válce.

5.1.2. S Archimédovým šroubem

Toto zařízení je konstruováno na principu jednoho z nejstarších známých čerpadel - Archimédův šroub. Znázorněno na *obrázku 8.03*. Krmivo, které je v zásobníku nad trubicí, propadává na vodorovně uložený šnekový mechanismus, namotaný kolem hřídele. Převrácené krmivo je v kapsách tvořených závity. Otáčením šneku nebo hřídele dochází k posuvu krmiva a následnému sypaní do akvária.



Obrázek 8.03 – Automatické krmítko s Archimédovým šroubem

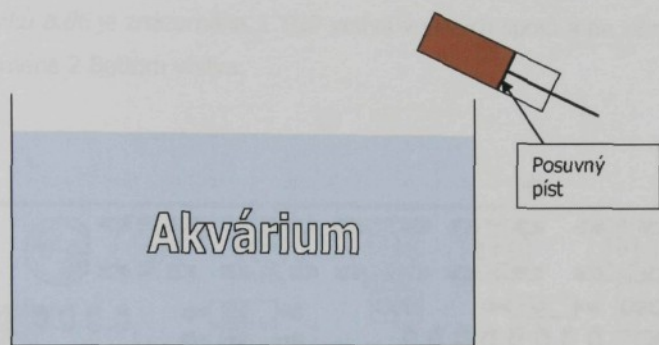
Tato koncepce má několik zásadních výhod. Jednak je to možnost použití velkokapacitního zásobníku krmiva, jenž lze velmi snadno a bezetrátově doplňovat (dokonce by bylo možné zautomatizovat i doplňování krmiva). Další nespornou výhodou je „robustnost“ celé koncepce oproti předchozímu návrhu.

Ovšem i tento koncept má své stinné stránky. Dávkování krmiva pomocí Archimédova šroubu není příliš přesné, zvláště tehdy, pokud šroub v trubce není dostatečně těsný. Dále je tu značná výrobní složitost a tím i cena zařízení – vyrobít velmi přesný Archimédův šroub není snadná ani levná záležitost. Ukázalo se, že použití sériově vyráběných zařízení tohoto typu není dost dobře možné. Např. mlýnek na maso zdaleka nemá dostatečně těsný šroub. Proto na přesné dávkování krmiva není vhodný. Také si musíme uvědomit, že v závislosti na použitém krmivu může dojít k jeho znehodnocení tím, že šroub bude krmivo drtit.

Tento způsob po drobné modifikaci byl na automatickou krmičku použit.

5.1.3. *S posuvným pístem*

Společným jmenovatelem všech předchozích zařízení je propadávání krmiva systémem shora dolů, a to pouze za pomoci gravitace. Tento systém není příliš robustní, protože krmivo bývá nezdřídka lepkavé a mnohdy se někde uchytí a něco ucpe, takže potom nepropadává, jak by mělo. U této konstrukce je krmivo vytlačováno zdola nahoru, místo abychom spoléhali na jeho propadávání shora dolů. Podle této myšlenky vzniknul poslední zde představovaný způsob aplikace krmení. Princip znázorněn na *obrázku 8.04*.



Obrázek 8.04 – Automatické krmítko s posuvným pístem

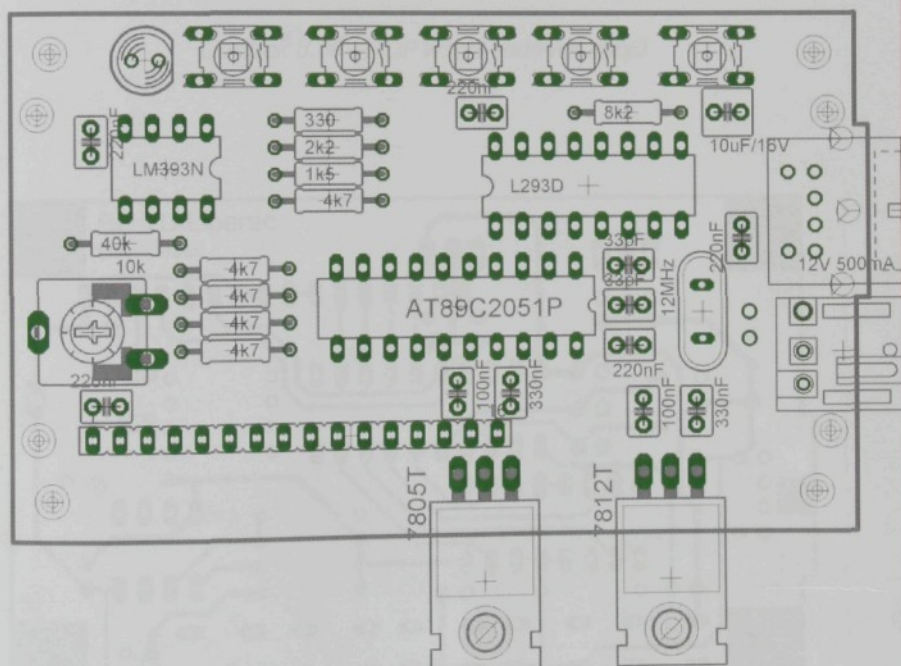
Systém sestává z nádoby (kontejneru), ve které je krmivo. Dno této nádoby je pohyblivé (je tvořeno pístem, který se může touto nádobou volně pohybovat) a horní konec nádoby je otevřený. Tato nádoba je upevněna nad akváriem, míří šikmo vzhůru, tak, aby krmivo nemohlo samovolně vypadávat. Pokud má zařízení krmit, vytlačí příslušnou dávku krmiva pohybem pístu vzhůru. Velikost této dávky je odměřena velikostí (délkou) posuvu pístu.

Hlavní výhodou této koncepce je její nepopíratelná robustnost – krmivo se nemá kde zachytit a co ucpat. Další výhodou je možnost přesného dávkování velmi malých objemů krmiva nezávisle na jeho tvaru a granularitě.

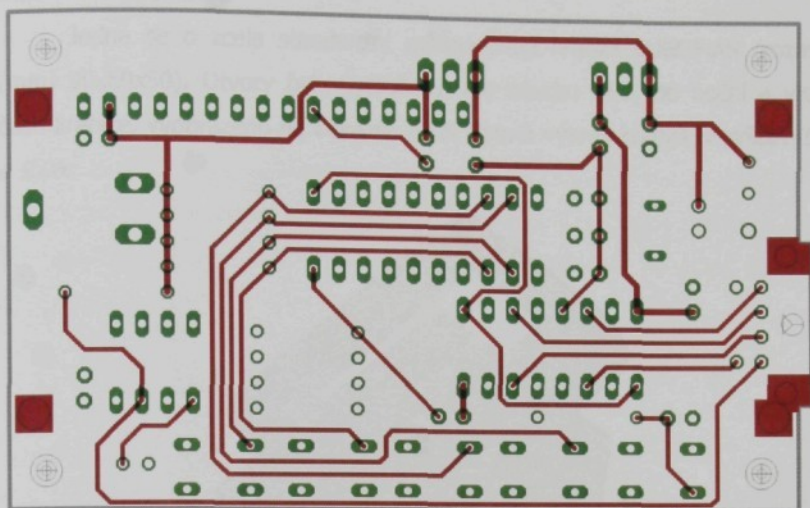
Zařízení má také několik nevýhod. Především je to jeho omezená kapacita a obtížnější systém doplňování krmiva než například u krmičky s Archimédovým šroubem. Další nevýhodou je nutnost získávání informace o poloze pístu v nádobě. Je také nutná k tomu, abychom věděli, kolik krmiva ještě v krmičce zbývá.

5.2. Deska plošných spojů

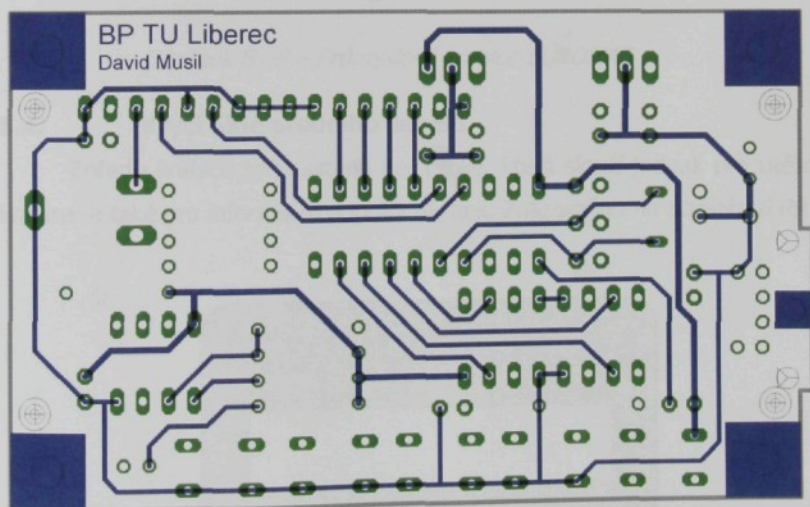
Při návrhu DPS pro Automatické krmítko bylo nutno dodržet rozměry, které byly dány velikostí, již zakoupené, průmyslové krabice. Přední panel s displejem je umístěn vodorovně, ve výšce 9,5mm nad základní deskou. Základní deska je přišroubována ke krabici pomocí čtyř šroubků velikosti M3, ve výšce 30mm nad dnem krabice. Z tohoto důvodu jsou všechny součástky a hlavní elektronické obvody umístěny zdola DPS viz *obrázek 8.05*. Jen konektor pro připojení displeje a tlačítka pro ovládání jsou umístěny na horní TOP vrstvě. Na *obrázku 8.06* je znázorněna 1 TOP vrstva vodivých spojů a na *obrázku 8.07* je znázorněna 2 Bottom vrstva.



Obrázek 8.05 – Rozmístění součástek na DPS



Obrázek 8.06 – TOP vrstva vodičů spojů

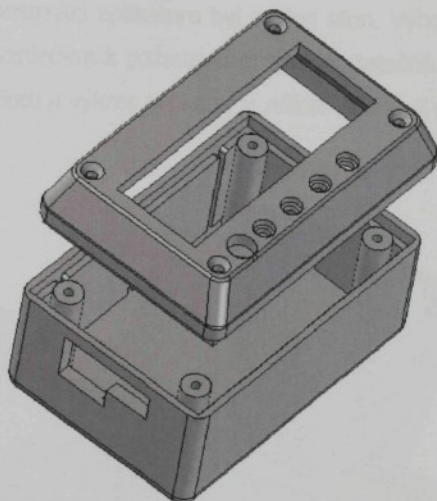


Obrázek 8.07 – Bottom vrstva vodičů spojů (vrchní)

V příloze 9.03 je z obou stran vyfocena osazená deska plošných spojů.

5.3. Krabice

Jedná se o zcela standardní průmyslovou krabici EUROMAS (rozměry [mm] 90x60x50). Otvory byly vytvořeny, dle návrhu DPS, do boční a vrchní části krabice, vyobrazeno na *obrázku 8.08*. Foto a výkres krabice v *příloze 9.04* a *9.06*.



Obrázek 8.08 – Průmyslová krabice EUROMAS

5.4. Krycí fólie předního panelu

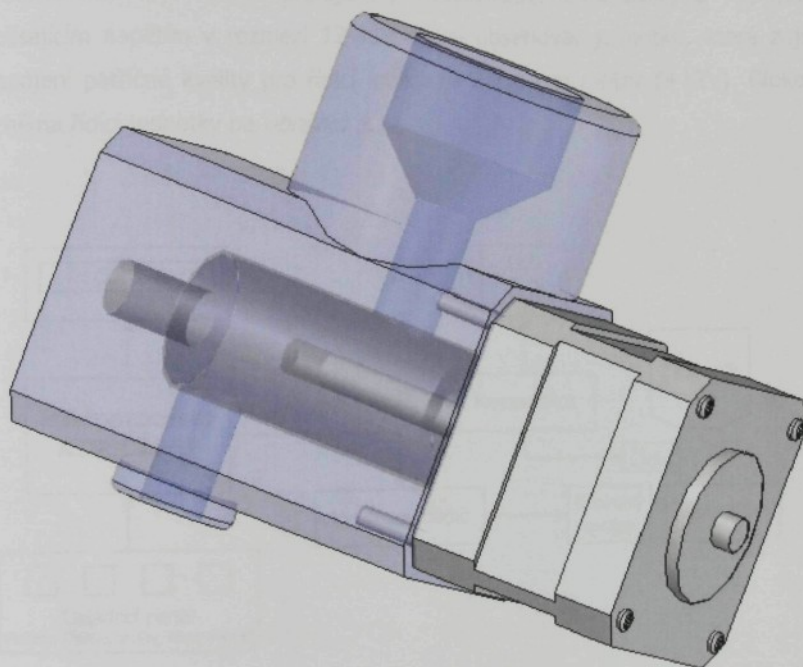
Zpředu krabice je umístěna krycí fólie, která slouží jednak pro utěsnění krabice, a také pro informační popisky tlačítek. Znáznorněno na *obrázku 8.09*.



Obrázek 8.09 – Krycí fólie na přední panel

5.5. Aplikátor krmiva

Jako mechanická konstrukce aplikátoru krmiva byla zvolena varianta s Archimédovým šroubem, která je podrobně popsána v kapitole 5.1.2. Rozměry byly dány velikostí krokového motoru. Pro posun krmiva byla namísto Archimédova šroubu zvolena, trochu poupravená, hřídel s trapézovým závitem. Materiálem pro konstrukci aplikátoru byl zvolen silon. Výhodou oproti železu je váha, pevnost je vzhledem k požadavkům také dostatečná. Realizace přibližena na obrázku 8.10. Foto a výkres aplikátor v příloze 9.05 a 9.07.



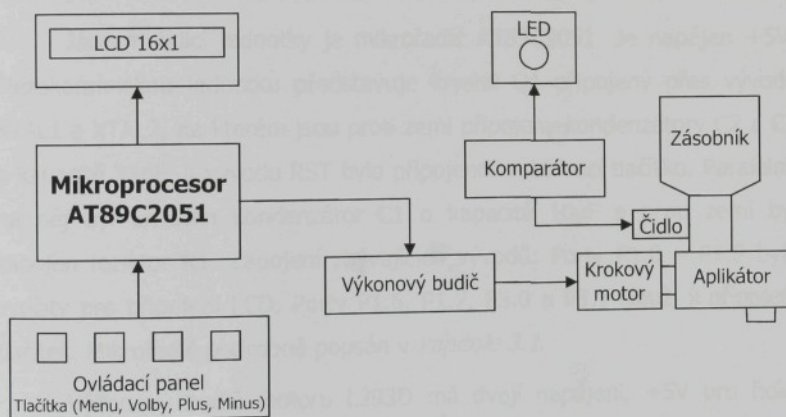
Obrázek 8.10 – Aplikátor krmiva pro Automatickou krmičku ryb

Světle modrá	-	aplikátor krmiva pro automatickou krmičku
Tmavě šedá	-	hřídel s poupraveným trapézovým závitem
Světle šedá	-	krokový motor TANDON kapitola 3.3.

6. Řídící jednotka

6.1. Úvod

Hlavní částí řídicí jednotky je mikrořadič, který zajišťuje všechny „vyšší“ funkce. Krokový motor je připojen přes výkonový budič, který zajišťuje jeho silové buzení. Jeho vstupy jsou připojeny na řídicí mikrořadič a jeho výstupy na buzený motor. Dalším blokem je komparátor, který při nedostatku krmení rozsvěcí signalizační diodu. Poslední částí ve schématu je blok napájení. Protože krmičku musí být možno připojit i k nestabilizovanému zdroji a napájet ji kolísajícím napětím v rozmezí 12-32V, musí obsahovat jednotku, která zajistí napájení patřičné kvality pro řídicí logiku (+5V) a pro motor (+12V). Blokové schéma řídicí jednotky na *obrázku 8.11*.



Obrázek 8.11 – Blokové schéma řídicí jednotky

6.2. Popis zapojení řídicí jednotky

V této kapitole je detailně popisováno zapojení řídicí jednotky, jejíž schéma je v *příloze 9.02*. Zapojení je maximálně jednoduché a účelné, ale přesto schopné zajistit všechny požadované činnosti a obsahující všechny nezbytné ochranné prvky tak, aby byla zajištěna jeho spolehlivá funkce.

Zařízení je napájeno ze síťového adaptéru Panasonic, který dodává stejnosměrných 12V / 500 mA se souosým konektorem s + uvnitř.

Úpravu vstupního napájecího napětí zajišťují obvody IC1 a IC2. Jedná se o standardní napěťové stabilizátory 7805 a 7812. Na vstupu mají proti zemi připojeny kondenzátory C4 a C5 (C6 a C7) o kapacitě 100nF a 330nF, sloužící k vyhlazení vstupního napětí. Kondenzátory C8 až C12 jsou umístěny mezi napájecími vývody jednotlivých integrovaných obvodů a pravidelně rozmístěny mezi vodiči napájecího vedení. Funkce těchto kondenzátorů je detailněji popsána v *kapitole 6.4*.

Jádrem řídicí jednotky je mikrořadič AT89C2051. Je napájen +5V. Piezokeramickou jednotku představuje krystal Q1 připojený přes vývody XTAL1 a XTAL2, na kterém jsou proti zemi připojeny kondenzátory C2 a C3 o kapacitě 33pF. K vývodu RST bylo připojeno resetovací tlačítko. Paralelně na něj byl umístěn kondenzátor C1 o kapacitě 10μF a proti zemi byl zapojen rezistor R1. Zapojení zbývajících vývodů: Porty P1.0 – P1.5 byly využity pro připojení LCD. Porty P1.6, P1.7, P3.0 a P3.1 slouží k připojení tlačítek. Mikrořadič podrobně popsán v *kapitole 3.1*.

Výkonový budič motoru L293D má dvojí napájení, +5V pro řídicí logiku a +12V pro výkonovou část. Jeho čtyři vstupy jsou připojeny na piny P3.2 – P3.5 řídicího procesoru. Jejich stav kopírují výkonové výstupy Out1 - Out4. Vstupy označené jako En1 a En2 slouží k nastavení výstupů Out1 - Out4 do třetího stavu. Jsou ovládány společně výstupem procesoru P3.7. Pokud motor stojí, nastaví procesor výstupy obvodu IC4 do třetího stavu, aby vinutími motoru zbytečně netekl proud.

Vyhodnocení množství krmení je prováděno pomocí OZ IC5A, který je zapojen jako komparátor. Je napájen nesymetricky +5V. Porovnává

vstupní signál s napětím na invertujícím vstupu. Napětí je přivedeno pomocí děliče z rezistorů R7 a R8 o odporech 1k50Ohmů a 4k70Ohmů. Rezistor R9 slouží k nastavení pracovního bodu fototranzistoru. Na výstupu komparátoru je připojena LED dioda D1.

LED dioda D1 slouží k indikaci stavu krmení v zásobníku. Pokud se dostane stav krmné směsi pod 10% začne svítit.

Fototranzistor není na schématu znázorněn, je připojen přes konektor MOTOR. Je umístěn na dně zásobníku. Při nedostatku krmiva na fototranzistor dopadá světelné záření a na jeho vývodech vzniká napětí. To je pak dále vyhodnocováno pomocí komparátoru.

LCD displej se stará o komunikaci mezi mikrořadičem a uživatelem. Displej je připojen na +5V, kontrast je řízen pomocí rezistoru R1 o odporu 10kOhm. Jeho podrobnější popis naleznete v *kapitole 3.2*.

Ovládací tlačítka MENU, VOLBA, PLUS a MINUS slouží k ručnímu řízení krmičky. Tlačítka jsou vybavena zdvihacími rezistory R3 a R6 o odporu 4k70Ohmů. Tlačítko RESET slouží k resetování mikrořadiče.

6.3. Seznam součástek

❖ IC obvody

IC1	7805T	-	stabilizátor napětí 5V
IC2	7812T	-	stabilizátor napětí 12V
IC3	AT89C2051	-	mikrořadič
IC4	L293D	-	výkonový budič
IC5	LM393N	-	operační zesilovač

❖ Rezistory

R1	Trimr	-	10k Ω
R2		-	8k2 Ω
R3 – R6 a R8		-	4k7 Ω
R7		-	1k5 Ω
R9		-	40k Ω
R10		-	2k2 Ω
R11		-	330 Ω

❖ Kondenzátory

C1		-	10 μ F/16V
C2 a C3		-	33pF
C4 a C6		-	330nF
C5 a C7		-	100nF
C8 – C12		-	220nF

❖ Fototranzistor

T1	BP103
----	-------

❖ Dioda

D1		-	LED červená, Ø5mm
D2	1N4007	-	usměrňovací dioda

❖ Krystaly

Q1		-	krystal 12,288Mhz
----	--	---	-------------------

❖ Konektory

NAPÁJENÍ		-	standardní souosý konektor, Ø 2,5mm
MOTOR		-	Mini Din YAMAICHI (6 pinů)
LCD		-	16-ti pinová lišta

❖ Tlačítka

T1 – T5		-	mikrospínač
---------	--	---	-------------

❖ Motor

TANDOM		-	krokový motor
--------	--	---	---------------

6.4. Ochranné prvky

Řídící jednotka automatické krmičky může být mnohdy vystavena různým rušivým vlivům. Proto je vybavena i prvky, které nemají přímý vliv na její funkci, ale zajišťují její ochranu proti těmto vlivům. V této části jsou podrobně popisovány jejich funkce.

První ochranný prvek se nachází hned na vstupu napájení. Je jím dioda D2, která zaručuje, že nedojde k poškození zařízení, pokud bude omylem připojeno napájecí napětí opačné polarity.

Za ochranné prvky lze považovat i vyhlazovací kondenzátory C4 a C6. Tyto kondenzátory vyhlazují průběh napětí na vstupech stabilizátorů IC1 a IC2. Jejich funkce je důležitá zejména v případě, že je k napájení zařízení použit nestabilizovaný zdroj.

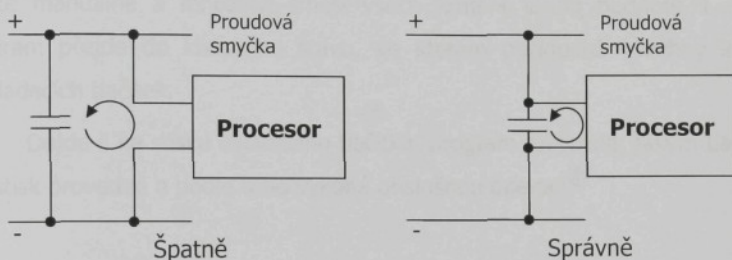
Důležitou funkci mají tzv. blokovací kondenzátory C8 – C12. Jsou důležité zvláště tehdy, když napájecí zdroj není dostatečně „tvrdý,“ tj. vykazuje nadměrný pokles napětí při zvětšení odběru. Vysvětleme si blíže jejich funkci.

Nahlédneme-li letmo do datasheetu některého z použitých integrovaných obvodů, například mikrořadiče AT89C2051, můžeme snadno dospět k mylnému názoru, že k jeho napájení postačí i relativně „měkký“ zdroj napětí, vždyť jeho proudový odběr je pouhých 2,8 mA. Nenechme se ale mýlit tímto velmi malým údajem. Hodnota uváděná v datasheetu je hodnota průměrná. AT89C2051 a mikrořadiče obecně nemají v čase konstantní proudový odběr. Časový průběh jejich napájecího proudu spíše připomíná dirackovy impulsy. Proudový odběr je korelovaný s hodinovým signálem a nejvyšších hodnot dosahuje na jeho hranách. Tyto špičkové hodnoty mohou dosahovat řádově až několika set miliampér. Proudový odběr potom ovšem velice rychle spadne téměř k nule.

Pokud napájecí zdroj není dostatečně „tvrdý,“ může jeho napětí v čase špičkového odběru značně poklesnout. To může snadno způsobit „zamrznutí“ mikrořadiče. Tento neduh vyřeší blokovací kondenzátor, zapojený mezi napájecí vývody mikrořadiče, jak je naznačeno na *obrázku 8.12*. Tento kondenzátor se nabije až na úroveň napájecího napětí v době, kdy je odběr mikrořadiče zanedbatelný. Když pak odběr prudce vzroste a napájecí napětí začne klesat,

počne se kondenzátor vybíjet a zabrání tak jeho prudkému poklesu. Výsledkem je „vyhlazený“ průběh napájecího napětí.

Důležité je, aby blokovací kondenzátor byl fyzicky co nejbližší napájecím vývodům a aby jeho přívody byly co možná nejkratší. Tím zmenšíme odpor přívodů a zpomalíme tak vybíjení kondenzátoru. Blokovací kondenzátor nesmí mít příliš velkou kapacitu, aby se stačil nabít mezi jednotlivými proudovými špičkami. Jeho kapacita typicky bývá 100-220 nF.



Obrázek 8.12 – Zapojení blokovacího kondenzátoru

Dalším ochranným prvkem je chladič pro stabilizátor napětí. Chladič je důležitým prvkem sloužící k odvádění ztrátového tepla do okolí.

$$\mathcal{Q}_j = \mathcal{Q}_a + (R_{\theta i} + R_{\theta e}) \times P_{\text{tot}}$$

$\mathcal{Q}_j, \mathcal{Q}_a$ - teplota přechodu, okolí

P_{tot} - maximální výkonová ztráta

$R_{\theta i}, R_{\theta e}$ - tepelný odpor interní, externí



Obrázek 8.13 – Chladič pro stabilizátor

Po výpočtu byl zvolen chladič na obrázku 8.13 s externím tepelným odporem $R_e = 21^\circ\text{C}/\text{W}$.

7. Program v ASEMBLERU

7.1. Obecný popis programu

Začněme nejprve obecným popisem hlavních funkcí programu. Nejobecněji můžeme jeho funkci shrnout například takto:

Po resetu (zapnutí) řídicí jednotky se program nastaví do „start“ polohy. Hodiny jsou nastaveny na výchozí hodnotu (00:00:00), krmení lze uskutečnit pouze manuálně a množství směsi všech krmení je na hodnotě 1. Potom program přejde do klidového stavu, ve kterém poslouchá všechny impulzy z ovládacích tlačítek.

Dojde-li ke stisku ovládacího tlačítka, program rozpozná, jakým tlačítkem byl stisk proveden a podle toho vykoná příslušnou operaci.

7.2. Struktura a ovládání programu

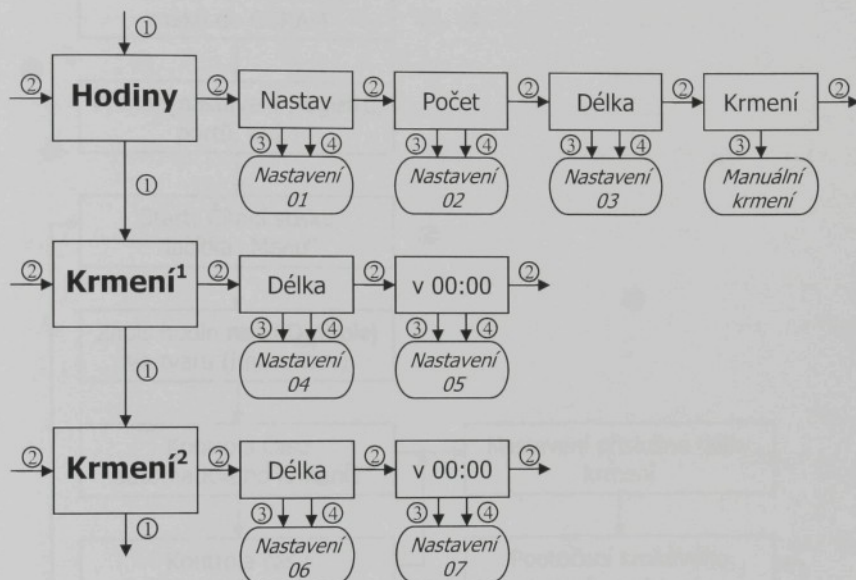
Program je rozdělen na dvě části, jedna je „Menu“. Uvnitř které se pomocí tlačítka „Menu“ pohybujeme mezi Hodinami, Krmením¹ a Krmením². Druhé je „Volba Menu“. Tu řídíme pomocí tlačítka „Volba“. Ta slouží pro pohyb uvnitř jednotlivých sekcí Menu, tj. k nastavení hodin, počtu krmení, manuálního krmení a jeho délky. U krmení 1 nebo 2 slouží tlačítko k nastavení času krmení a jeho příslušné délky. Tlačítko popsané jako „Plus“ potřebujeme k nárůstu hodnot o jedničku, naopak tlačítko s popiskem „Minus“ je k snížení příslušné hodnoty o jedničku.

U všech čtyř tlačítek je program cyklický. Tj. nemůžeme se opětovným stiskem tlačítka dostat do nějakého nedefinovaného stavu.

Při běhu motoru dochází k „zmrazení ovládání“ z důvodů časové náročnosti tohoto cyklu. Běh motoru je znázorněn běžící šipkou v pravé části první poloviny displeje.

V levé polovině displeje se zobrazují informace týkající se nastavování displeje a volby režimu programu. V pravé polovině displeje jsou nastálo zobrazeny hodiny ve formátu (hodiny:minuty:vteřiny).

Na následném diagramu, *obrázek 8.14*, je znázorněn pohyb mezi jednotlivými režimy programu. Hodnota uvedená v kroužku je číslo tlačítka, bráno v pořadí (1 - „Menu“, 2 - „Volba“, 3 - „Plus“, 4 - „Minus“). Text v obdélníku je znázorněn na displeji a text v elipse naznačuje, co se provádí při stisku příslušného tlačítka.



Obrázek 8.14 – Diagram pohybu v menu programu

Vysvětlivky :

Nastavení 01 - nastavení hodin, čítáno po minutách

Nastavení 02 - nastavení počtu krmení

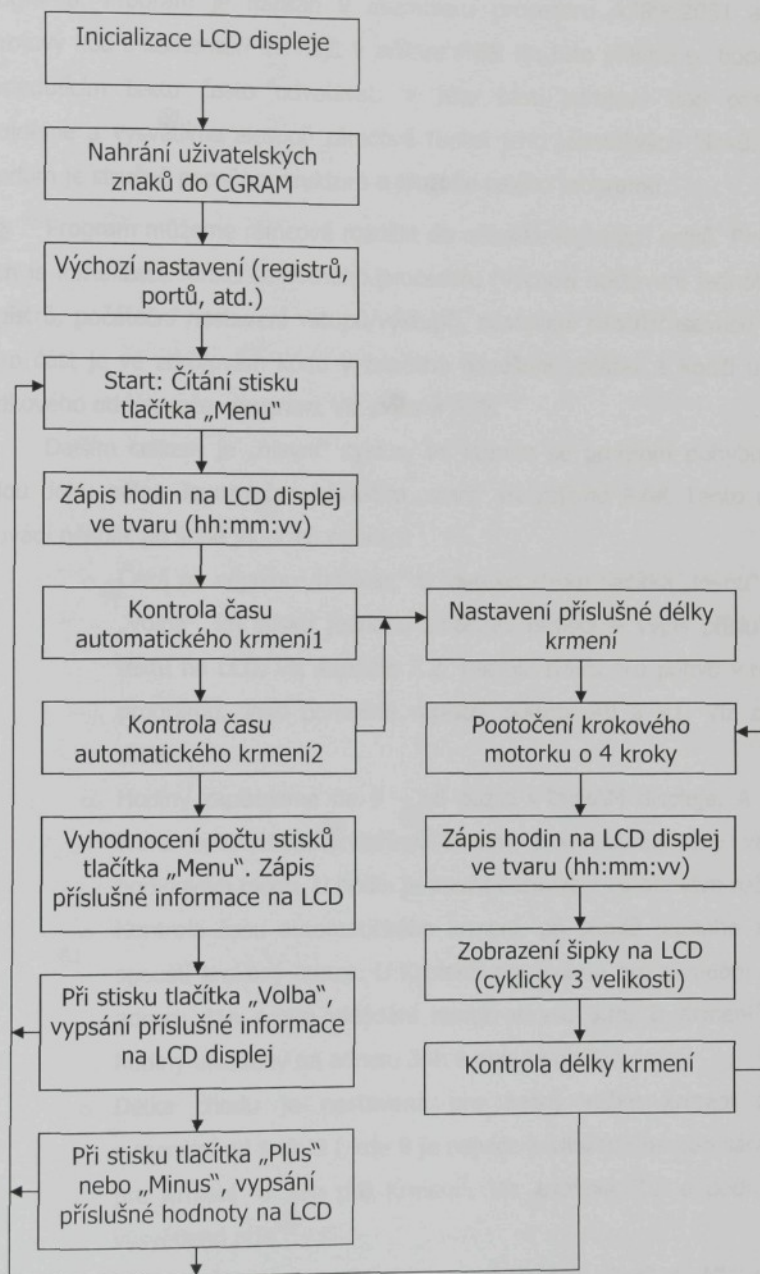
- 0 - krmí pouze manuálně (stisk tlačítka)
- 1 - krmí manuálně + Krmení¹ (automatické krmení)
- 2 - krmí manuálně + Krmení¹ + Krmení²

Nastavení 03 - nastavení množství krmení v rozmezí 1 - 9 (9 – nejvíce)

Nastavení 04 a 06 - nastavení množství krmení pro automatická krmení

Nastavení 05 a 07 - nastavení času automatického krmení, čítáno po minutách

7.3. Architektura programu



Obrázek 8.15 – Blokové schéma větvení programu

Nyní již můžeme přistoupit k podrobnějšímu vysvětlení funkce řídicího programu. Program je napsán v assembleru procesoru AT89C2051 a jeho zdrojový kód s komentáři lze najít v *příloze 9.08*. Na tuto přílohu se budeme v následujícím textu často odvolávat. V této části zdrojový kód postupně projdeme a vysvětlíme alespoň rámcově funkci jeho jednotlivých bloků. Ještě předtím je stručně popsána struktura a filozofie celého programu.

Program můžeme rámcově rozdělit do několika logických celků. Prvním z nich je inicializace funkcí samotného procesoru (výchozí nastavení jednotlivých registrů, počáteční nastavení vstupů/výstupů, nastavení čítačů/časovačů atd.). Tato část je ve zdrojovém kódu vyznačena návěštím začátek a končí u celorádkového oddělovače *program*. Viz *příloha 9.08*.

Dalším celkem je „hlavní“ cyklus, ve kterém se program pohybuje po celou dobu běhu. Je označen návěštím „start“ viz *příloha 9.08*. Tento cyklus provádí několik po sobě jdoucích operací:

- Čeká na nějakou „událost,“ tj. detekci stisku tlačítka „Menu“ nebo „Volba“. Při stisku jednoho z tlačítek následuje výpis příslušného textu na LCD, viz *kapitola 7.2*. Tlačítka slouží pro pohyb v režimu programu. Jako pomocné registry použity R0 a R1. Viz *příloha 9.08*.
- Hodiny zapisujeme na 9 – 16 pozici v DDRAM displeje. A to ve tvaru hodiny:minuty:vteřiny. Hodiny jsou zobrazovány ve 24-hodinovém módu. U hodin je použit časovač v 16-bitovém režimu.
- Kontrola času automatického krmení, při shodě jednoho z časů spouští krokový motor. U Krmení¹ používáme pro ukládání hodin adresu 31h a pro ukládání minut adresu 32h. U Krmení² jsou hodiny ukládány na adresu 34h a minuty na 35h.
- Délka chodu je nastavena pro každý režim krmení zvlášť v rozmezí od 1 do 9 (kde 9 je nejvíce). Uložena na adresách 30h pro Krmení¹ a 33h pro Krmení². Viz *kapitola 7.2*. a podrobněji vysvětleno níže.
- Nastavení hodin provádíme pomocí tlačítek „Plus“ a „Minus“. Při stisku tlačítka „Plus“ zvýšíme hodnotu o 1 a při stisku tlačítka

„Minus“ naopak. Velikost hodin se ukládá do registru R4, minut do R3 a vteřin do R2.

- Čas pro automatické krmení se nastavuje ve tvaru hh:mm. Pomocí tlačítek „Plus“ nebo „Mínus“ nastavíme čas na požadovanou hodnotu. Adresa pro ukládání časů krmení viz výše. Krmení probíhá vždy, když se shoduje čas (tj. hodiny i minuty) a vteřiny jsou rovny nule.
- Nastavení počtu krmení. Lze nastavit v rozmezí od 0 – 2. Blíže vysvětleno v kapitole 7.2. Adresa pro uložení počtu krmení je 28h.
- Nastavení délky krmení přímo souvisí s počtem otáček krokového motoru při jednotlivých krmeních. Při jednom průchodu programem se motor pootočí o čtyři kroky. Vzorec pro výpočet počtu kroků : $\text{počet kroků} = [(\text{délka} \times 27) + \text{délka}] \times 4$
Adresa pro uložení počtu kroků (délka krmení v krocích) je 38h. Při chodu motoru se na sedmé pozici na displeji zobrazuje běžící šipka směrem vzhůru. Postupně zmenšující šipka je uložena na adresách 00h, 01h a 02h v CGRam displeje. Vzhledem k časové náročnosti délky krmení je nutné, do cyklu motoru, zavést zapisování hodin. Časová prodleva mezi jednotlivými kroky je zvolena 50ms. Krokový motor a jeho řízení viz kapitola 3.3.

Poslední logický celek je tvořen skupinou podprogramů, vykonávajících různé podúlohy. V následujících několika odstavcích je stručně popisována jejich funkce.

- Podprogram nastav (za celořádkovým oddělovačem *nastavení a oživení*) se volá hned na začátku běhu programu a slouží k inicializaci LCD displeje. Podrobně popsáno v kapitole 3.2.5.
- Podprogram naspis se též volá hned na začátku běhu programu za celořádkovým oddělovačem *zápis znaků do CGRam + nastavení písma*. Slouží k nahrání uživatelských znaků do paměti CGRam u LCD displeje. Podrobně popsáno v kapitole 3.2.8. U rutiny využíváme direktivy DB, která ukládá do programové paměti osmibitové hodnoty, které jsou dány vyhodnocením položek

seznamu výrazů. Po následném načítání položek seznamu, postupně ukládáme do paměti CGRam. U ukládání používáme registr R7.

- Rutina zobtext slouží k zobrazování příslušného textu na LCD displej. Jednotlivé texty jsou uloženy pod návěstími text1 až text8.
- Rutina zobraz se volá z cyklu programu velmi často. Slouží k postupnému zápisu na LCD displej. Nejdříve se zapisují horní čtyři bity a pak dolní. Toto je z důvodu čtyřbitové komunikace mezi mikrořadičem a LCD displejem. Viz kapitola 3.2.3.
- Rutina začínající návěstím příkaz, je obdobná jako rutina zobraz. S tím rozdílem, že nejde o rutinu sloužící k zobrazení znaků na LCD, ale o vstup příkazu. Tj. nastavování pozice v DDRAM na kterou se má zapisovat. Rutina příkaz je také použita u inicializace displeje.
- Rutina zapis a rozklad je velmi důležitá pro zápis hodin, minut a vteřin na LCD displej. Je napsána za celořádkovým oddělovačem *zobrazení rozloženého čísla na displej*. Slouží k rozkladu vstupující hodnoty na desítky a jednotky. Desítky jsou uloženy do registru A a jednotky do registru B.
- Podprogram začínající návěstím cas slouží k nastavení časovače. Rutina také kontroluje, aby nedošlo k přetečení vteřin a minut přes 60 a hodin přes 24. Časovač je opatřen smyčkou, která pomocí instrukce DJNZ čítá do 1 vteřiny. ($50\text{ms} \times 20 = 1\text{s}$)
- Poslední část podprogramů následuje za celořádkovým oddělovačem *zpožd'ovací smyčky*. Jde o programové zpožd'ovací smyčky nezbytné pro jednotlivé části programu.

Zdrojový kód s komentáři všech popisovaných rutin lze najít v příloze 9.08.

Možná zlepšení

Automatická krmička pro akvarijní ryby by se mohla dále rozšířit o ještě jeden aplikátor krmiva, tzn. ještě jeden motor pro jeho ovládání. Dále by bylo dobré ovládat přesný čas spínání jednotlivých periférií akvária. Např. osvětlení, filtrace, topení nebo vzduchování. Z důvodů těchto rozšíření by se již musel volit jiný druh mikrořadiče nebo stávající rozšířit o expandér portů. Také by se dalo přemýšlet o možné propojenosti s PC, např. přes rozhraní RS 232.

Závěr

V této bakalářské práci jsem navrhnul a realizoval hardware a software pro Automatickou krmičku ryb. Zařízení má za úkol automatické krmení ryb v akváriu. Zařízení jsem pojmenoval Automatické krmítko.

Po nastudování teorie o monolitickém mikrořadiči řady 8051 jsem navrhnul jednotlivé funkční bloky zařízení. V této fázi vývoje byl kladen důraz na co nejjednodušší konstrukci a zároveň spolehlivou funkci.

Dle mého návrhu byla vyrobena deska plošných spojů, kterou jsem osadil součástkami. Tuto desku jsem uchytil v průmyslové krabici a dále navrhl a vyrobil přední panel. Výroba aplikátoru byla o něco složitější, ale s drobnou pomocí se také podařilo.

Funkčnost zařízení byla ověřena u středně velkého akvária a nebyl shledán žádný vážnější nedostatek.

Seznam použité literatury

- [1] Skalický, P.: Mikroprocesory řady 8051, BEN, Praha 2002
- [2] Popis mikroprocesoru a jeho periférií [online] www.dhservis.cz/popis8051.htm
- [3] Připojení LCD displeje k 8051 [online] www.dhservis.cz/lcd.htm
- [4] Ing.Pavel Rydlo : Krokové motory a jejich řízení, PDF, Liberec 2000
- [5] Krokové motory [online] <http://robotika.cz/articles/steppers/cs>
- [6] Programátor PATmel 3 [online] www.volny.cz/d72/
- [7] Informace o programu EAGLE [online] www.eagle.cz/info.htm
- [8] Vedral J., Fischer J.: Elektronické obvody pro měřicí techniku. Praha: vydavatelství ČVUT, 1999
- [9] Ing. Milan Kolář, CSc. : Přednáškové listy z předmětu AEL
- [10] Prof. Ing. Ondřej Novák, CSc. : Přednáškové listy z předmětu CIE
- [11] Datasheety k jednotlivým obvodům [online] www.ges.cz
- [12] Informace o akvaristice [online] www.akvarista.cz/web/

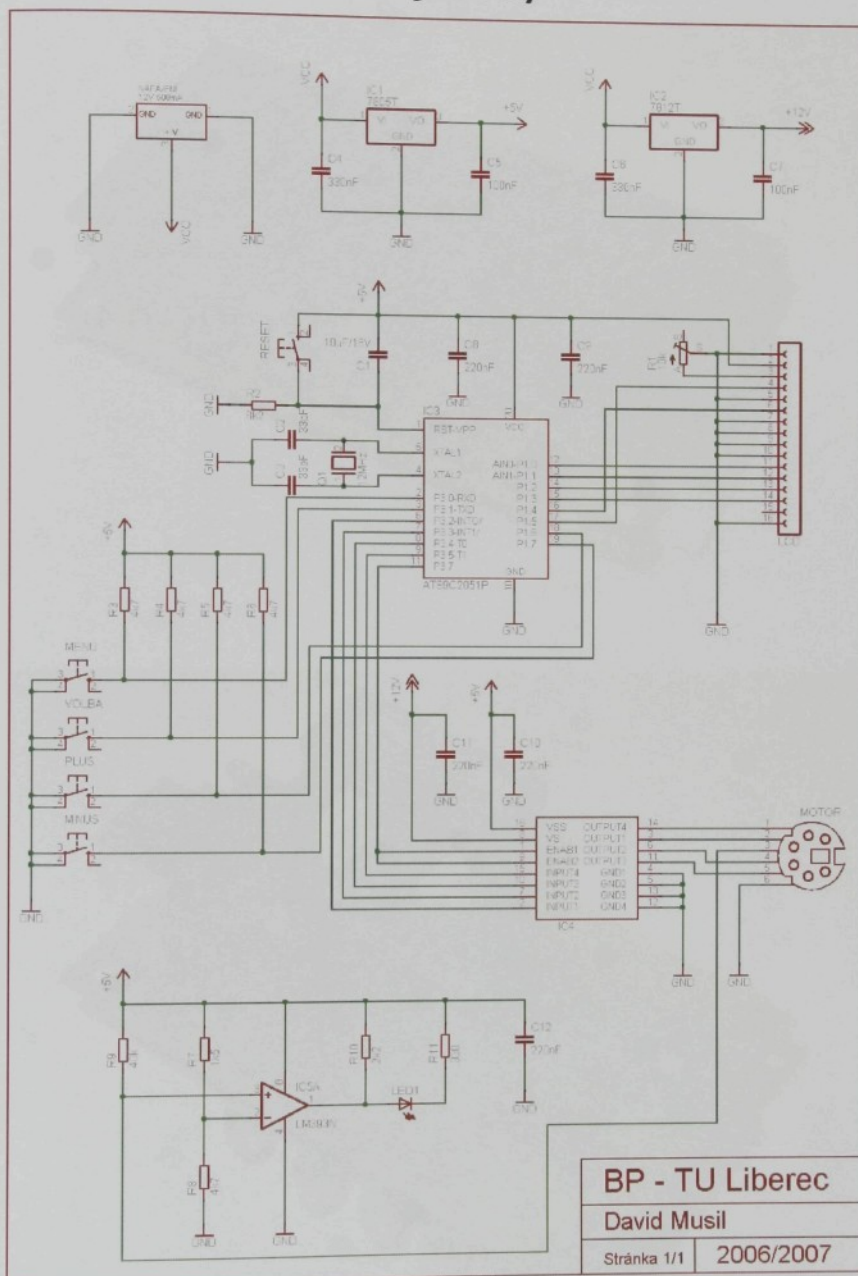
Obsah přiloženého CD-ROM

Příložený CD-ROM obsahuje tuto práci ve formátu doc včetně příloh. Zdrojový kód programu (adresář „program“). Schéma řídicí jednotky ve formátu sch, návrh desky plošných spojů ve formátu brd a nově vytvořené knihovny ve formátu lbr. Vše (adresář „eagle“). Výkresy kompletního aplikátoru a průmyslové krabice ve formátech asm, cfg, dft (adresář „výkres“). Kresleno v programu Solid Edge V16. Obrázky jednotlivých součástí ve formátu jpg (adresář „obrázky“). Nakonec všechny datasheety součástek, zmiňovaných v textu (adresář „datasheet“).

Příloha 9.01 – Znak

Horní 4bity Dolní 4bity	0x 0000	2x 0010	3x 0011	4x 0100	5x 0101	6x 0110	7x 0111	Ax 1010	Bx 1011	Cx 1100	Dx 1101	Ex 1110	Fx 1111
x0 xxxx0000	↑		0	a	P	`	P		—	9	E	o	p
x1 xxxx0001	↑	!	1	A	Q	a	9	a	7	7	4	a	q
x2 xxxx0010	▲	"	2	B	R	b	r	"	イ	ツ	×	β	θ
x3 xxxx0011		#	3	C	S	c	s	┘	ウ	テ	ε	ω	
x4 xxxx0100		\$	4	D	T	d	t	、	工	ト	μ	ω	
x5 xxxx0101		%	5	E	U	e	u	・	才	十	1	ε	0
x6 xxxx0110		&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
x7 xxxx0111		'	7	G	W	g	w	フ	チ	ズ	ラ	q	π
x8 xxxx1000		(	8	H	X	h	x	イ	ク	本	リ	フ	X
x9 xxxx1001		)	9	I	Y	i	y	お	ト	ル	レ	ウ	
xA xxxx1010		*	:	J	Z	j	z	エ	コ	ノ	ク	i	〒
xB xxxx1011		+	;	K	L	k	l	オ	サ	セ	コ	*	ス
xC xxxx1100		,	<	L	¥	l	l	ホ	ニ	フ	ワ	Φ	π
xD xxxx1101		—	=	M	I	m	i	エ	ズ	ハ	シ	ト	÷
xE xxxx1110		。	>	N	^	n	^	ヨ	セ	ホ	レ	ン	
xF xxxx1111		/	?	0	_	o	+	ッ	リ	マ	"	ö	■

Příloha 9.02 – Schéma řídicí jednotky



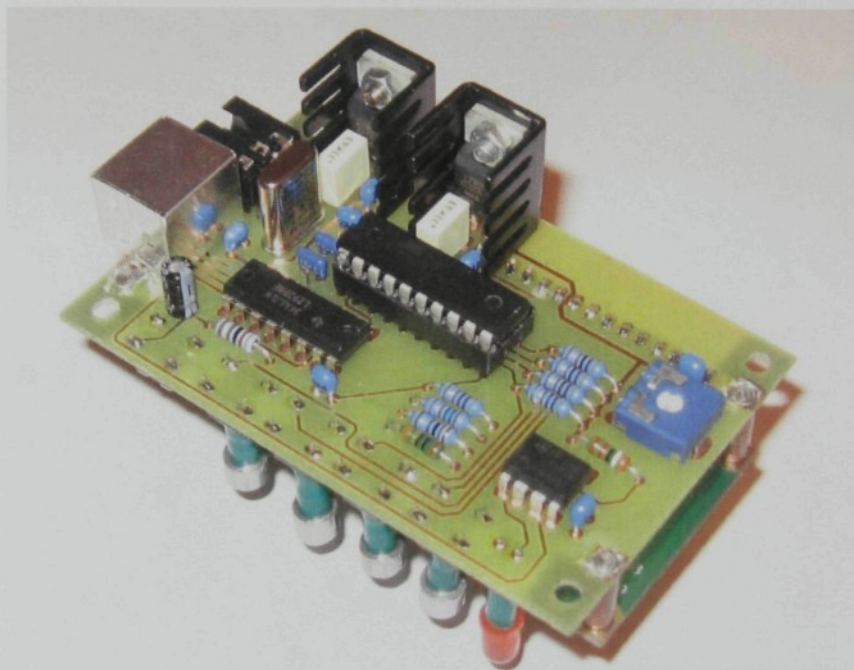
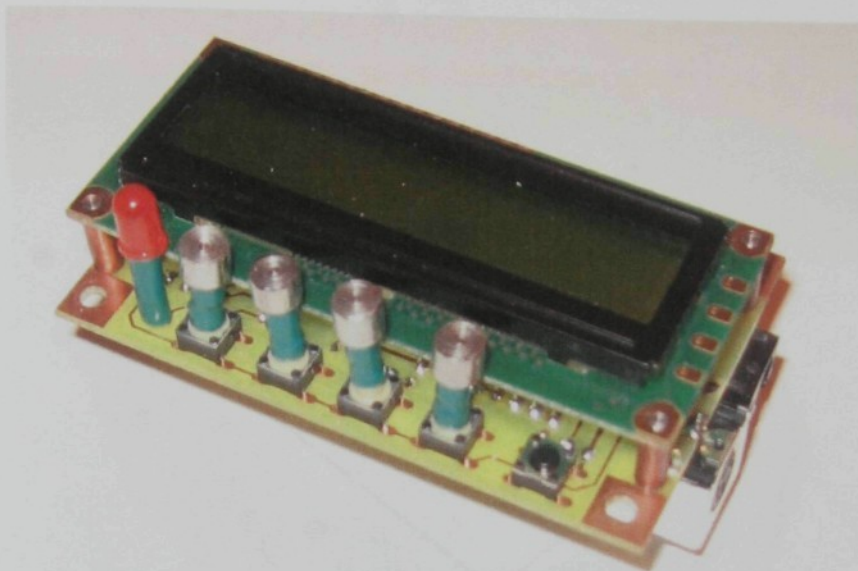
BP - TU Liberec

David Musil

Stránka 1/1

2006/2007

Příloha 9.03 – Foto osazené DPS



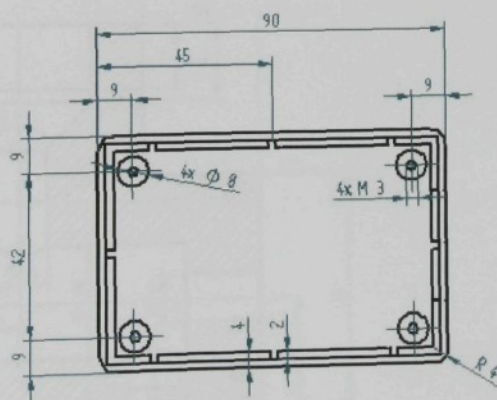
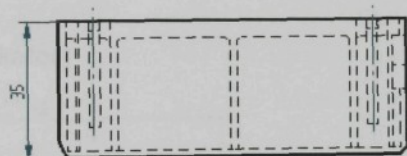
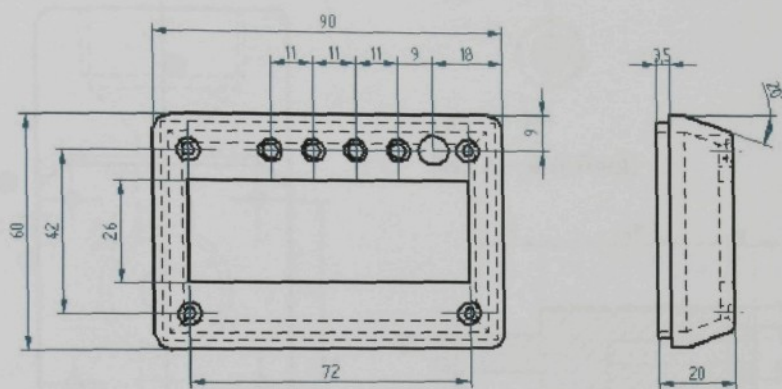
Příloha 9.04 – Foto krabice s předním panelem



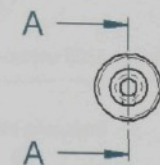
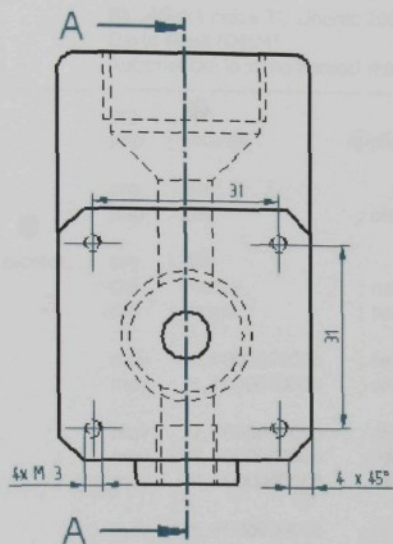
Příloha 9.05 – Foto kompletního aplikátoru s motorem



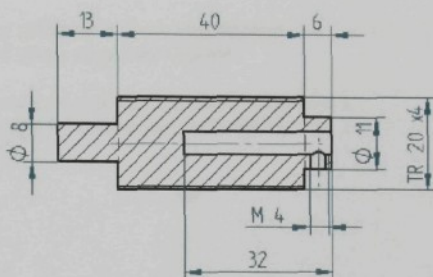
Příloha 9.06 – Výkres krabice



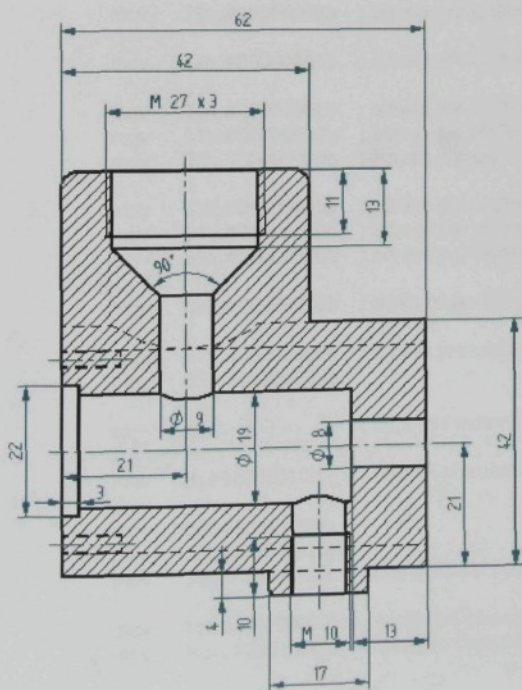
Příloha 9.07 – Výkres aplikátoru a hřídele



Řez A – A (hřidel)



Řez A – A (aplikátor)



Příloha 9.08 – Zdrojový kód programu

```

;-----
;      Bakalářská práce TU Liberec 2006/2007
;      David Musil (D@M)
;      Automatické krmítko pomocí mikroprocesoru 8051 (použit At89c2051)
;-----
      org    00h
      jmp    zacatek          ; přeskočení přerušení

      org    0bh
      jmp    cas              ; obsluha přerušení

zacatek:  org    40h
          call   nastav        ; nastavení potřebných parametrů displeje
          call   naspis        ; nastavení písma do paměti CGRAM

          mov    r0,#00000000b ; registr použit na zobrazování režimu programu 1
          mov    r1,#00000000b ; registr použit na zobrazování režimu programu 2

          mov    r2,#00000000b ; sekundy
          mov    r3,#00000000b ; minuty
          mov    r4,#00000000b ; hodiny

          mov    r5,#00000000b ; přerušení
          mov    r6,#00000000b ; přerušení
          mov    r7,#00000000b

          mov    28h,#00000000b ; adresa pro uložení počtu krmení
          mov    29h,#10000000b ; adresa pro uložení polohy zapisovaného znaku na LCD
          mov    2ah,#00000001b ; adresa pro uložení délky krmení 0

          mov    30h,#00000001b ; adresa pro uložení délky krmení 1
          mov    31h,#00000000b ; adresa pro uložení času krmení 1 (hodiny)
          mov    32h,#00000000b ; adresa pro uložení času krmení 1 (minuty)

          mov    33h,#00000001b ; adresa pro uložení délky krmení 2
          mov    34h,#00000000b ; adresa pro uložení času krmení 2 (hodiny)
          mov    35h,#00000000b ; adresa pro uložení času krmení 2 (minuty)

          mov    36h,#00000000b ; délka textového řetězce

          mov    37h,#00000000b ; adresa pro uložení šipky
          mov    38h,#00000000b

          clr    p3.7          ; odpojení motoru

          mov    ie,#10000010b ; nastaví přerušení

          mov    th0,#00111000b
          mov    tl0,#00000101b
          mov    39h,#00010100b ; přerušení do sekund

          mov    tmod,#00000001b ; nastavení časovače
          mov    tcon,#00010000b ; spuštění časovače

```

-----program-----

```
start:  jb    p3.0,prog      ; tlačítko použito na posun v režimu programu 1
        call  zpo100ms
        call  zpo100ms
        inc   r0
        mov   r1,#00000000b
```

-----zobrazení hodin-----

```
prog:   mov   a,#11000000b  ; nastavení pozice v DDRam( 9 - 16 znak LCD )
        call  prikaz

        mov   a,r4          ; zápis hodin
        call  zapis

        mov   a,#00111010b  ; dvojtečka
        call  zobraz

        mov   a,r3          ; zápis minut
        call  zapis

        mov   a,#00111010b  ; dvojtečka
        call  zobraz

        mov   a,r2          ; zápis sekund
        call  zapis

        call  zpo50ms
```

-----kontroly časů krmení-----

```
        mov   a,28h
        cjne  a,#2d,kon0

        cjne  r2,#1d,kon1    ; kontrola času krmení 2
        mov   a,r3
        cjne  a,35h,kon1
        mov   a,r4
        cjne  a,34h,kon1
        jmp   motor2        ; při schodě časů, pootočí motorem

kon0:   mov   a,28h
        cjne  a,#1d,hodiny

kon1:   cjne  r2,#1d,hodiny    ; kontrola času krmení 1
        mov   a,r3
        cjne  a,32h,hodiny
        mov   a,r4
        cjne  a,31h,hodiny
        jmp   motor1        ; při schodě časů, pootočí motorem
```

```

;-----volba režimu programu-----

hodiny:    cjne    r0,#0d,hodiny1
           cjne    r1,#0d,htlac1
           mov     36h,#8d
           mov     dptr,#text1
           call    zobtext

htlac1:    jb      p3.1,hnastav      ; tlačítko použito na posun v režimu programu 2
           call    zpo100ms
           call    zpo100ms
           inc     r1

hnastav:   cjne    r1,#1d,hpocet
           mov     36h,#8d
           mov     dptr,#text4
           call    zobtext

           jb      p1.7,hpocetp      ; nastavení na požadovanou hodnotu, odečítání po min.
           call    zpo100ms
           cjne    r3,#00d,hpd1      ; jsou-li minuty větší než 00, tak nedělej nic
           mov     r3,#59d           ; jsou-li rovny 00, tak pokračuje
           cjne    r4,#00d,hpd2      ; jsou-li hodiny větší než 00, tak nedělej nic
           mov     r4,#23d           ; jsou-li rovny 00, tak pokračuje
           jmp     hpocetp

hpd2:      dec     r4                ; dekrementace hodin
           jmp     hpocetp

hpd1:      dec     r3                ; dekrementace minut

hpocetp:   jb      p1.6,hpocet      ; nastavení na požadovanou hodnotu, čítání po min.
           call    zpo100ms
           inc     r3                ; inkrementace minut
           cjne    r3,#60d,hpocet    ; jsou-li minuty menší než 60, tak nedělej nic
           mov     r3,#00d           ; jsou-li rovny 60, tak pokračuje
           inc     r4                ; inkrementace hodin
           cjne    r4,#24d,hpocet    ; jsou-li hodiny menší než 24, tak nedělej nic
           mov     r4,#00d           ; jsou-li rovny 24, tak pokračuje

hpocet:    cjne    r1,#2d,hdelka
           call    shp

hdelka:    cjne    r1,#3d,hkrmeni
           call    shd

hkrmeni:   cjne    r1,#4d,hkonec
           mov     36h,#8d
           mov     dptr,#text7
           call    zobtext

           jb      p1.6,hkonec
           jmp     motor0
           call    zpo100ms

hkonec:    cjne    r1,#5d,hodiny1
           mov     r1,#00000000b

hodiny1:   cjne    r0,#1d,hodiny2
           cjne    r1,#0d,h1tlac1

```

	mov	36h,#8d	
	mov	dptr,#text2	
	call	zobtext	
h1tlac1:	jb	p3.1,h1delka	; tlačítko použito na posun v režimu programu 2
	call	zpo100ms	
	call	zpo100ms	
	inc	r1	
h1delka:	cjne	r1,#1d,h1krmv	
	call	sh1d	
h1krmv:	cjne	r1,#2d,h1konec	
	mov	36h,#2d	
	mov	dptr,#text8	
	call	zobtext	
	mov	a,31h	; zápis hodin
	call	zapis	
	mov	a,#00111010b	; dvojtečka
	call	zobraz	
	mov	a,32h	; zápis minut
	call	zapis	
	mov	a,#' '	
	call	zobraz	
	jb	p1.7,h1krt1	; nastavování hodin u krmení ¹ na požadovanou hodnotu
	call	zpo100ms	
	mov	a,32h	
	cjne	a,#00d,h1krth	; jsou-li minuty větší než 00, tak nedělej nic
	mov	32h,#59d	; jsou-li rovny 00, tak pokračuje
	mov	a,31h	
	cjne	a,#00d,h1krth	; jsou-li hodiny větší než 00, tak nedělej nic
	mov	31h,#23d	; jsou-li rovny 00, tak pokračuje
	jmp	h1krt1	
h1krth:	dec	31h	; dekrementace hodin
	jmp	h1krt1	
h1krthm:	dec	32h	; dekrementace minut
h1krt1:	jb	p1.6,h1konec	
	inc	32h	; inkrementace minut
	call	zpo100ms	
	mov	a,32h	
	cjne	a,#60d,h1konec	; jsou-li minuty menší než 60, tak nedělej nic
	mov	32h,#00d	; jsou-li rovny 60, tak pokračuje
	inc	31h	; inkrementace hodin
	mov	a,31h	
	cjne	a,#24d,h1konec	; jsou-li hodiny menší než 24, tak nedělej nic
	mov	31h,#00d	
h1konec:	cjne	r1,#3d,hodiny2	
	mov	r1,#00000000b	
hodiny2:	cjne	r0,#2d,konec	
	cjne	r1,#0d,h2tlac1	
	mov	36h,#8d	
	mov	dptr,#text3	
	call	zobtext	

h2tlac1:	jb call call inc	p3.1,h2delka zpo100ms zpo100ms r1	; tlačítko použito na posun v režimu programu 2
h2delka:	cjne call	r1,#1d,h2krmv sh2d	
h2krmv:	cjne mov mov call mov call mov call mov call mov call	r1,#2d,h2konec 36h,#2d dptr,#text8 zobtext a,34h zapis a,#00111010b zobraz a,35h zapis a,#' zobraz	 ; zápis hodin ; dvojtečka ; zápis minut
	jb call mov cjne mov mov cjne mov jmp	p1.7,h2krt1 zpo100ms a,35h a,#00d,h2krtm 35h,#59d a,34h a,#00d,h2krth 34h,#23d h2krt1	; nastavování hodin u krmení ² na požadovanou hodnotu ; jsou-li minuty větší než 00, tak nedělej nic ; jsou-li rovny 00, tak pokračuje ; jsou-li hodiny větší než 00, tak nedělej nic ; jsou-li rovny 00, tak pokračuje
h2krth:	dec jmp	34h h2krt1	; dekrementace hodin
h2krtm:	dec	35h	; dekrementace minut
h2krt1:	jb inc call mov cjne mov inc mov cjne mov	p1.6,h2konec 35h zpo100ms a,35h a,#60d,h2konec 35h,#00d 34h a,34h a,#24d,h2konec 34h,#00d	 ; inkrementace minut ; jsou-li minuty menší než 60, tak nedělej nic ; jsou-li rovny 60, tak pokračuje ; inkrementace hodin ; jsou-li hodiny menší než 24, tak nedělej nic
h2konec:	cjne mov	r1,#3d,konec r1,#00000000b	
konec:	cjne mov	r0,#3d,skok r0,#00000000b	
skok:	jmp	start	

-----nastavení počtu krmení-----

```
shp:    mov    36h,#6d
        mov    dptr,#text5
        call   zobtext
        call   kshpdat1

        jb     p1.7,kshpp      ; nastavování počtu krmení 0-2
        call   zpo100ms
        call   kshpdat
        cjne   a,#00d,kshpde
        mov    28h,#3d
kshpde: dec    28h
```

```
kshpp:  jb     p1.6,kshp      ; nastavování počtu krmení 0-2
        call   zpo100ms
        inc    28h
        call   kshpdat
        cjne   a,#3d,kshp
        mov    28h,#0d
```

```
kshp:   ret
```

```
kshpdat: mov    a,#10000110b    ; nastavení pozice v DDRam ( 1 - 8 znak LCD)
        call   prikaz
kshpdat1: mov    a,28h
        add    a,#48d
        call   zobraz
        mov    a,#' '
        call   zobraz
        mov    a,28h

        ret
```

-----nastavení délky krmení - 0-----

```
shd:    mov    36h,#6d
        mov    dptr,#text6
        call   zobtext
        call   shdpda1

        jb     p1.7,shdp      ; nastavování délky krmení 0-9
        call   zpo100ms
        call   shdpdat
        cjne   a,#01d,shdpde
        mov    2ah,#10d
shdpde: dec    2ah

shdp:   jb     p1.6,kshd      ; nastavování délky krmení 0-9
        call   zpo100ms
        inc    2ah
        call   shdpdat
        cjne   a,#10d,kshd
        mov    2ah,#1d

kshd:   ret
```

```

shdpdat:  mov    a,#10000110b    ; nastavení pozice v DDRam ( 1 - 8 znak LCD)
          call   prikaz
shdpda1:  mov    a,2ah
          add    a,#48d
          call   zobraz
          mov    a,'# '
          call   zobraz
          mov    a,2ah

          ret

```

;------nastavení délky krmení - 1 -----;

```

sh1d:     mov    36h,#6d
          mov    dptr,#text6
          call   zobtext
          call   sh1dpda1

          jb     p1.7,sh1dp        ; nastavování délky krmení 0-9
          call   zpo100ms
          call   sh1dpdat
          cjne   a,#01d,sh1dpde
          mov    30h,#10d
sh1dpde:  dec     30h

sh1dp:    jb     p1.6,ksh1d        ; nastavování délky krmení 0-9
          call   zpo100ms
          inc     30h
          call   sh1dpdat
          cjne   a,#10d,ksh1d
          mov    30h,#1d

ksh1d:    ret

```

```

sh1dpdat: mov    a,#10000110b    ; nastavení pozice v DDRam ( 1 - 8 znak LCD)
          call   prikaz
sh1dpda1: mov    a,30h
          add    a,#48d
          call   zobraz
          mov    a,'# '
          call   zobraz
          mov    a,30h

          ret

```

;------nastavení délky krmení - 2 -----;

```

sh2d:     mov    36h,#6d
          mov    dptr,#text6
          call   zobtext
          call   sh2dpda1

          jb     p1.7,sh2dp        ; nastavování délky krmení 0-9
          call   zpo100ms
          call   sh2dpdat
          cjne   a,#01d,sh2dpde
          mov    33h,#10d
sh2dpde:  dec     33h

```