

Dynamicky se měnící reklama  
Dynamically changing advertisement

Jiří Růčka

---

Bakalářská práce  
2008



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2007/2008

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří RŮČKA**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
  
Téma práce: **Dynamicky se měnící reklama**

Zásady pro vypracování:

1. Seznamte se mikrokontrolérem Atmel 8051.
2. Navrhněte soustavu, která bude zobrazovat reklamní nápisy na listech točící se vrtule.
3. Navrhněte ve zmiňovaném kontroléru řídicí program.
4. Zařízení prakticky realizujte a ověřte jeho činnost.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Bastian, P.: Praktická elektrotechnika. Europa – Sobotáles, Brno, 2004.
2. HORST, J.: Informační a telekomunikační technika. Praha, BEN, 2004.
3. HRUŠKA, F.: Technické prostředky automatizace IV. UTB ve Zlíně, 2001.
4. HRABOVSKÝ, M.: Eagle pro začátečníky. Praha, BEN, 2007.
5. [www.intel.com](http://www.intel.com)

Vedoucí bakalářské práce:

**Mgr. Milan Adámek, Ph.D.**

Ústav elektrotechniky a měření

Datum zadání bakalářské práce:

**20. února 2008**

Termín odevzdání bakalářské práce:

**5. května 2008**

Ve Zlíně dne 20. února 2008



prof. Ing. Vladimír Vašek, CSc.

*děkan*



doc. Ing. Ivan Zelinka

*ředitel ústavu*

## **ABSTRAKT**

Cílem této bakalářské práce je seznámit se s mikrokontrolérem ATMEL architektury 8051 a navrhnout a prakticky zrealizovat soustavu, která bude zobrazovat na listech točící se vrtule reklamní text. V první části je obecně popsána problematika mikropočítače ATMEL AT89C2051, jeho způsoby programování a některé jeho užitečné funkce. Ve druhé části je pak popsána konstrukce vlastního modelu soustavy, popis použitého programátoru mikrokontrolérů ATMEL a řídicí program pro mikrokontrolér AT89C2051.

Klíčová slova: mikrokontrolér, ATMEL, 89C2051, reklama, vrtule

## **ABSTRACT**

The main aim of this bachelor thesis is to familiarize with microcontroller ATMEL 8051 architecture and design and create system, which can display advertisement text on a rotating propeller. The first part describes features of microcontroller AT89C2051, method of programming this microcontroller and finally some of its useful functions. In the second part is described construction of the system, description of chip programmer used for the current type of microcontroller ATMEL and control program for microcontroller AT89C2051.

Keywords: microcontroller, ATMEL, 89C2051, advertisement, propeller

Rád bych poděkoval vedoucímu bakalářské práce kterým je doc. Mgr. Milan Adámek, Ph.D. za odborné vedení, rady, připomínky a pomoc v průběhu řešení této bakalářské práce.

*Motto:*

*“ Odborník nemůže rozumět všemu, je omezován svým oborem. Kdo není v ničem odborníkem, není ničím omezován. “*

Gabriel Laub

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně 15.5.2008

.....  
Podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>8</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 MIKROKONTROLÉR AT89C2051</b> .....	<b>11</b>
1.1 POPIS PINŮ .....	12
1.2 CHARAKTERISTIKY OSCILÁTORU .....	14
1.3 REGISTRY SPECIÁLNÍCH FUNKCÍ .....	15
1.4 OMEZNÍ URČITÝCH INSTRUKCÍ .....	16
1.4.1 Skokové instrukce .....	16
1.5 PROGRAM MEMORY LOCK BITS .....	17
1.6 IDLE MÓD .....	17
1.7 POWER-DOWN MÓD .....	17
<b>2 PROGRAMOVÁNÍ FLASH PAMĚTI</b> .....	<b>19</b>
2.1 VNITŘNÍ ČÍTAČ ADRES .....	19
2.2 PROGRAMOVACÍ ALGORITMUS .....	19
2.3 DATA POLLING .....	20
2.4 READY/BUSY .....	20
2.5 OVĚŘENÍ PROGRAMU .....	20
2.6 VYMAZÁNÍ ČIPU .....	21
2.7 ČTENÍ VÝROBNÍCH BYTŮ .....	21
<b>3 VYBRANÉ FUNKCE AT89C2051</b> .....	<b>22</b>
3.1 ČASOVAČE A ČÍTAČE .....	22
3.2 REGISTRY PRO ŘÍZENÍ ČASOVAČE/ČÍTAČE .....	22
3.2.1 TMOD (Timer Mod) .....	22
3.2.2 TCON (Timer Control) .....	24
3.3 PŘERUŠOVACÍ SYSTÉM .....	24
3.4 REGISTR IE (INTERRUPT ENABLE) .....	25
3.5 VEKTORY PŘERUŠENÍ .....	26
3.6 ÚROVNĚ PRIORITY PŘERUŠENÍ .....	27
<b>II PRAKTICKÁ ČÁST</b> .....	<b>28</b>
<b>4 HARDWAROVÁ ČÁST</b> .....	<b>29</b>

4.1	VRTULE .....	30
4.2	VÝROBA PLOŠNÉHO SPOJE .....	32
4.3	PROGRAMÁTOR PATMEL3 .....	33
<b>5</b>	<b>PROGRAMOVÁ ČÁST .....</b>	<b>37</b>
5.1	PROGRAMOVACÍ PROSTŘENÍ CONTEXT A PATMEL3 .....	37
5.1.1	ConTEXT .....	37
5.1.2	PAtmel3 .....	38
5.2	ZOBRAZENÍ PÍSMEN .....	39
5.3	NASTAVENÍ ČASOVAČŮ .....	40
5.4	PŘERUŠENÍ A JEHO OBSLUŽNÉ RUTINY .....	41
5.5	HLAVNÍ SMYČKA PROGRAMU A TABULKY TEXTU .....	42
	<b>ZÁVĚR .....</b>	<b>44</b>
	<b>ZÁVĚR V ANGLIČTINĚ .....</b>	<b>45</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>46</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>47</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>48</b>
	<b>SEZNAM TABULEK .....</b>	<b>49</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>50</b>

## ÚVOD

Nejdůmyslnější stroj, který člověk doposud vymyslel, je počítač. Mikropočítač je určitým druhem počítače. Počítače se neustále zdokonalují; zvyšuje se jejich rychlost a výpočtová mohutnost, zjednodušuje se programování, zvyšuje se jejich spolehlivost a snižuje se cena. Podstatnější a závažnější je skutečnost, že se postupně učíme počítače racionálně využívat. Doposud se s úspěchem použily v desítkách tisících aplikací.

V začáteční fázi průmyslové revoluce koncem minulého století byl v závodě jediný zdroj mechanické energie (parní stroj). Z tohoto zdroje se rozváděla mechanická energie k jednotlivým strojům. Jediná porucha ve zdroji měla za následek zastavení práce v celém podniku. Když se začaly vyrábět laciné motory, jednotlivé stroje začaly mít své zdroje mechanické energie. Podobný trend vývoje můžeme pozorovat u počítačů. Po roku 1960 (začátek levných minipočítačů) a zejména po roce 1972 (začátek mikropočítačů) se často v podniku místo jediného centrálního počítače začíná prosazovat na řešení problémů větší počet menších, nezávislých počítačů. V dnešní době se stále více používá nezávislých počítačů spojených v síti a většinou zastřešených hlavním počítačem. Pojem malý počítač se nemůže ztotožňovat s málo výkonným počítačem. Můžeme říci, že mikroprocesory a mikropočítače způsobily větší kvalitativní změny v elektrotechnice než tranzistory a později integrované obvody.

Největšího rozšíření z hlediska aplikací doznaly mikrokontroléry dvou významných výrobců mikroprocesorů - firem Motorola a Intel (podobně jako v technice stolních mikropočítačů). Spektrum výrobců jednočipových mikropočítačů je však velmi široké, mimo již jmenované firmy jsou to například americké Texas Instruments, National Semiconductors, ATMEL, evropské Philips, Siemens, japonské OKI, NEC, Toshiba, Hitachi apod. Charakteristické pro historický vývoj mikrokontroléru však je, že se zde neprojevuje snaha o drastické zvýšení rychlosti a kapacity paměti jako u osobního počítače. Je to dáno zejména faktem, že nasazením řídicího mikropočítače v dané aplikaci musí dojít k podstatnému zjednodušení výroby a tím ke snížení ceny výsledného výrobku. Stačí-li proto například k vykonání specifického řídicího programu operační paměť 64 bytů, není výrobce ničím motivován, aby použil mikrokontrolér s pamětí 128 bytů, protože by tím jen zdražil výsledný výrobek. Obecně platí, že se pro aplikaci použije ten nejjednodušší mikrokontrolér, který pro danou aplikaci vyhovuje. Tato motivace je proto také důvodem k tomu, že výrobci jednočipových

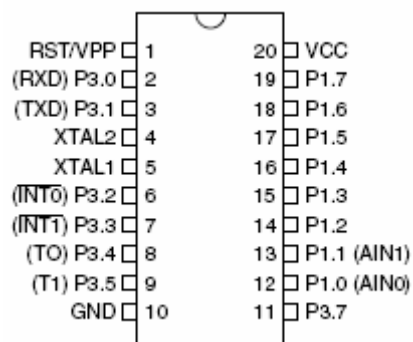


mikropočítačů nabízejí velmi široké spektrum konfigurací mikrokontrolérů, lišící se i drobnými změnami v kapacitě paměti (RAM i ROM), počtů vstupů/výstupů, použitelných periférií apod. Přitom záměr výrobce aplikace je co nejjednodušší (a tím nejlevněji vyrobitelná a nejspolehlivější) sestava potřebného řídicího hardware, nejlépe složená pouze z vlastního mikrokontroléru.

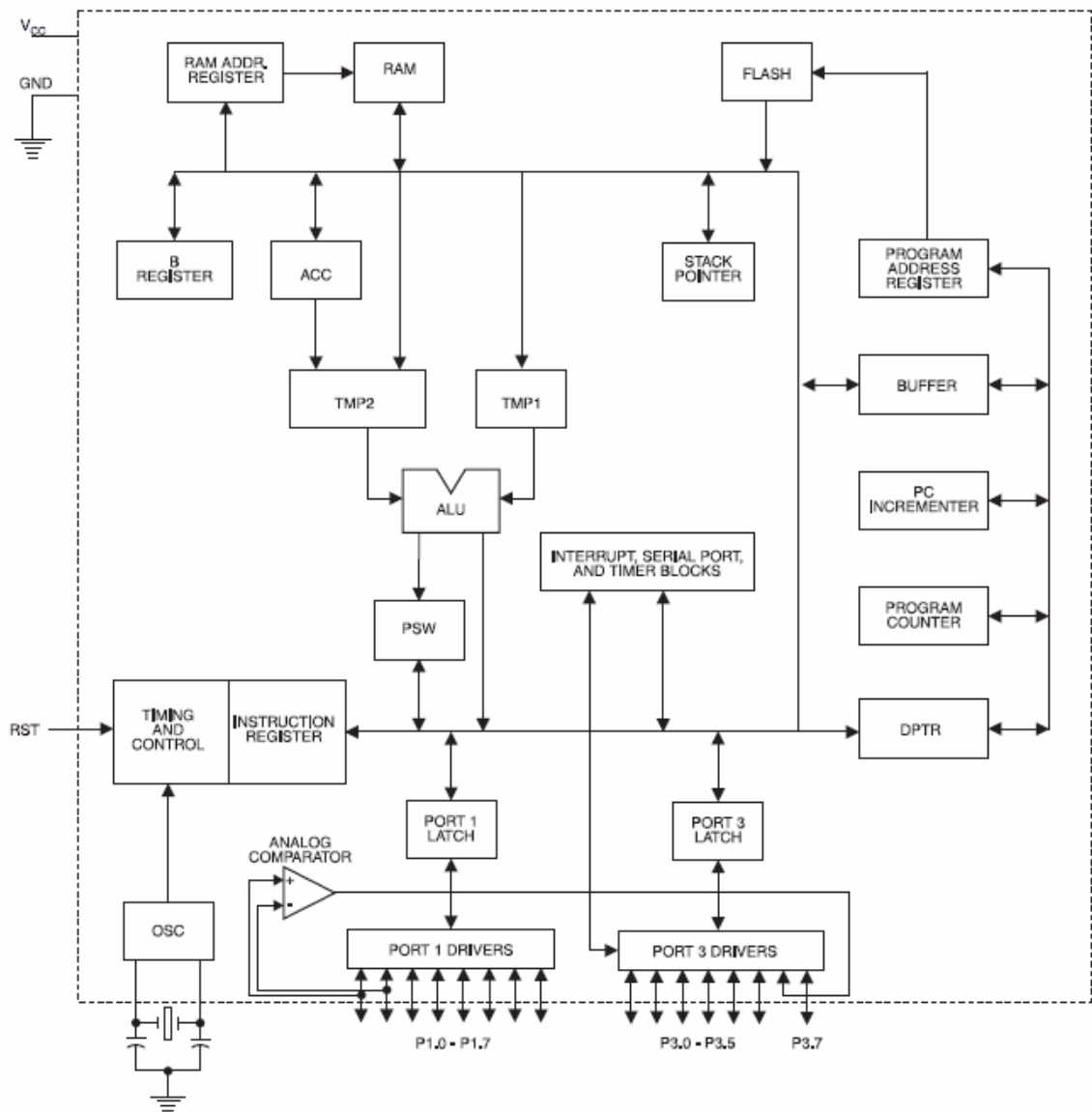
## I. TEORETICKÁ ČÁST

## 1 MIKROKONTROLÉR AT89C2051

AT89C2051 je nízkonapěťový multifunkční 8-bitový CMOS mikrokontrolér s 2KB programovatelné FLASH paměti (PEROM - Programmable and Erasable Read-Only Memory) a jeho instrukční sada je kompatibilní s průmyslovým standardem MCS-51 firmy INTEL. Kombinací 8-bitového CPU s FLASH pamětí na jediném čipu, AT89C2051 poskytuje vysoce flexibilní a cenově efektivní řešení pro mnohé aplikace. AT89C2051 obsahuje tyto základní části: 2KB Flash paměti, 128 bytů paměti RAM, 15 vstupních/výstupních portů, dva 16-bitové časovače/čítače, pět dvouprioritních vektorů přerušení, plně duplexní sériový port, přesný analogový komparátor a oscilátorové a hodinové obvody a podporuje dva softwarové úsporné režimy. Idle Mode zataví CPU, zatímco RAM, časovače/čítače, sériový port a přerušovací systém pokračuje v činnosti. Power-Down Mode uchovává obsah paměti RAM, ale přerušuje činnost oscilátoru, čímž zastaví ostatní funkce čipu, dokud nebude mikrokontrolér resetován. [3] [2]



Obr. č 1. Rozložení pinů mikrokontroléru AT89C2051



Obr. č 2. Bloková schéma mikrokontroléru AT89C2051

## 1.1 Popis pinů

**VCC** – Napájecí napětí. Rozsah napájecího napětí je závislý na frekvenci krystalu. Pro frekvenci 12MHz je rozsah napájení 2,7V – 6V a pro frekvenci 24Mhz 4V – 6V.

**GND** – Uzemění.

**Port 1** – Port 1 je 8-bitový obousměrný vstupní/výstupní port. Piny P1.2 až P1.7 obsahují vnitřní pull-up. P1.0 a P1.1 vyžadují externí pull-up. P1.0 a P1.1 také slouží jako kladný vstup (AIN0) a negativní vstup (AIN1) integrovaného přesného analogového komparátoru. Výstupy portu 1 dodávají proud 20mA a LED displej může být buzen přímo. Pokud je na

pin portu přivedena log 1, může být použit jako vstup. Pokud jsou piny P1.2 až P1.7 použity jako vstupy a jsou uzeměny, budou dodávat proud (IIL) díky vnitřnímu pull-up. Port 1 také přijímá data kódu během Flash programování a ověřování.

**Port 3** – Piny Portu 3 P3.0 až P3.5 a P3.7 jsou obousměrné vstupní/výstupní piny s vnitřním pull-up. P3.6 je integrovaný výstup vnitřního komparátoru a není přístupný obvyklým způsobem jako ostatní piny. Port 3 dodává 20mA. Pokud je na pin portu přivedena log 1, může být použit jako vstup. Pokud jsou piny Portu 3 použity jako vstupy a jsou uzeměny, budou dodávat proud (IIL) díky vnitřnímu pull-up. Piny Portu 3 mají také alternativní použití: [3]

*Tab. č 1. Alternativní použití pinů portu P3*

<b>Pin portu</b>	<b>Alternativní funkce</b>
P3.0	RDX (vstup seriové linky)
P3.1	TXD (výstup seriové linky)
P3.2	INT0 (vnější přerušení 0)
P3.3	INT1 (vnější přerušení 1)
P3.4	T0 (vnější časovač 0)
P3.5	T1 (vnější časovač 1)

Port 3 přijímá kontrolní signály během Flash programování a ověřování.

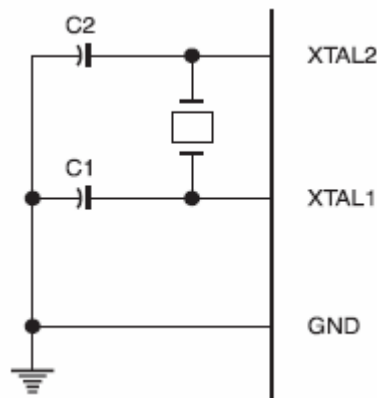
**RST** – Vstup resetu. Jakmile je na RST vstup přivedena log 1, všechny vstupní/výstupní piny jsou nastaveny na log1. Přivedení log 1 na RST vstup po dobu delší jak dva strojové cykly resetuje celý mikrokontrolér. Každý strojový cyklus trvá 12 impulsů oscilátoru.

**XTAL1** – Vstup do invertujícího oscilátoru a vstup do hodinového obvodu.

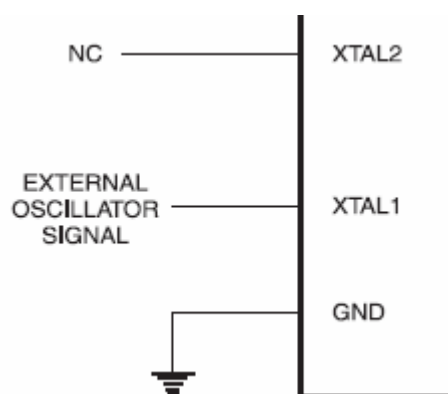
**XTAL2** – Výstup z invertujícího oscilátoru.

## 1.2 Charakteristiky Oscilátoru

XTAL1 a XTAL2 jsou vstup a výstup invertujícího zesilovače, který může být použit jako vnitřní oscilátor (Obr. č 3). Použit může být jak krystal tak keramický oscilátor. Hodnoty kondenzátorů C1 a C2 jsou  $30\text{pF} \pm 10\text{ pF}$  pro krystal a  $40\text{ pF} \pm 10\text{ pF}$  pro keramický rezonátor. Mikrokontrolér může být řízen taky externím hodinovým signálem tak, že XTAL2 zůstane nezapojený, zatímco na XTAL1 je přiveden hodinový signál (Obr. č 4). Nejsou zde žádné požadavky na předepsaný cyklus externího hodinového signálu, protože vstup do vnitřního hodinového obvodu je realizován přes dva klopné obvody, avšak minimální a maximální napětí musí být sledováno.[3]



Obr. č 3. Zapojení vnitřního oscilátoru



Obr. č 4. Zapojení externího hodinového signálu

### 1.3 Registry speciálních funkcí

Mapa vnitřního paměťového prostoru, nazývaného Registry speciálních funkcí (SFR - Special Function Register) je zobrazena v tabulce č.2. Ne všechny adresy jsou obsazeny a neobsazené adresy nemusí být vůbec implementovány v čipu. Čtením z těchto adres získáme náhodné data a pokus o zápis má neurčitý efekt. Uživatelský program by neměl zapisovat do těchto neobsazených adres, protože tyto adresy mohou být využity v novějších verzích mikrokontroléru. V tomto případě resetované nebo neaktivní hodnoty nových bitů budou vždy 0.[3]

Tab. č 2. Mapa Registru speciálních funkcí a implicitní hodnoty

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XXX00000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0XX00000							0AFH
0A0H								0A7H
98H	SCON 00000000	SBUF XXXXXXXX X						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				87H

## 1.4 Omezení určitých instrukcí

AT89C2051 je ekonomicky a cenově efektivní člen rostoucí skupiny mikrokontrolérů ATMEL. Obsahuje 2KB programové paměti Flash a je plně kompatibilní s architekturou MCS-51 a lze ho programovat instrukční sadou pro MCS-51. Přesto je při programování potřeba mít napaměti několik omezení instrukcí. Všechny instrukce používající skok jsou omezeny tak, že jejich cílová adresa skoku musí být v rozsahu fyzické paměti programu, což jsou 2KB. Toto omezení si hlídá programátor sám. Například: LJMP 7E0H je platná instrukce pro AT89C2051 (s 2KB paměti), zatímco LJMP 900H už není.[3]

### 1.4.1 Skokové instrukce

**LCALL, LJMP, ACALL, AJMP, SJMP, JMP a@A+DPTR** - Tyto nepodmíněné skokové instrukce budou provedeny korektně, pokud má programátor na paměti, že cílová adresa skoku musí fyzicky adresovatelná (adresy 00H až 7FFH pro 89C2051). Překročení limitu velikosti fyzické paměti má za následek neurčité chování programu.

**CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ** - Pro tyto podmíněné skokové instrukce platí stejná pravidla jako na nepodmíněné. Opět, překročení limitu velikosti fyzické paměti způsobí nevypočitatelný příkaz. V aplikacích využívající přerušení jsou běžné adresy začátku obsluhy přerušení architektury 80C51 uchráněny.

**MOVX instrukce, Datová paměť** - AT89C2051 obsahuje 128 bytů interní datové paměti. Tím je velikost zásobníku omezena na 128 bytů ze zbývajících paměti RAM. Nejsou podporovány externí datová paměť ani paměť programu. Proto se v programu nemůže vyskytovat žádná instrukce MOVX [...]. Typický assembler 80C51 bude stále obsahovat instrukce, přestože jsou napsány v rozporu s omezeními zmíněných výše. Proto programátor musí znát fyzické vlastnosti a omezení použitého mikrokontroléru a přizpůsobit instrukce odpovídajícím způsobem. [3]



## 1.5 Program Memory Lock Bits

Čip obsahuje dva lock-bity, které mohou být naprogramované (P) nebo nemusí být naprogramované (U) pro využití dalších možností popsaných v tabulce 3. [3]

Tab. č 3. Módy Lock-bitových zámků

Lock-bity		Typ zámků
LB1	LB2	
U	U	Žádné možnosti
P	U	Další programování Flash deaktivováno
P	P	Další programování Flash a ověření deaktivováno

## 1.6 Idle mód

V Idle módu se CPU uvede do spánku, zatímco všechny periferie zůstávají aktivní. Mód je ovládaný softwarově. Obsah vnitřní paměti RAM a speciální registry zůstávají nezměněny během tohoto módu. Idle mód je ukončen kterýmkoliv povoleným přerušením nebo hardwarovým resetem. P1.0 a P1.1 by měly být nastaveny na „1“, pokud není použit externí pull-up nebo „0“ pokud použitý je. Jakmile je Idle mód ukončen hardwarovým resetem, mikrokontrolér pokračuje v místě programu, kde přerušil činnost po dobu dvou strojových cyklů, než převezme kontrolu resetovací rutina. Během této události hardware čipu bání přístupu do vnitřní RAM, avšak nebrání přístupu na piny portů. Aby se v průběhu resetu zabránilo možnosti nečekaného zápisu na pin portu, instrukce, která následuje po instrukci volající Idle mód, by neměla zapisovat na port, nebo do externí paměti.[3]

## 1.7 Power-down mód

V Power-down módu je zastaven oscilátor a instrukce, která volá Power-down mód je poslední vykonaná instrukce. Vnitřní paměť RAM a speciální registry uchovávají hodnoty, dokud není Power-down mód ukončen. Jediné možné ukončení Power-down módu je hardwarový reset. Reset znovu nastaví speciální registry, ale vnitřní paměť RAM nezmění.

Reset by neměl být proveden před tím, než se napájecí napětí obnoví na svou běžnou provozní hodnotu a musí být držena dostatečně dlouho, aby se oscilátor znovu rozběhl a stabilizoval. P1.0 a P1.1 by měly být nastaveny na „0“ ,pokud není použit externí pull-up nebo „1“ pokud použitý je.[3]

## 2 PROGRAMOVÁNÍ FLASH PAMĚTI

AT89C2051 je dodáváný s 2KB PREROM prázdnou pamětí programu (obsah = FFH) a připravenou pro programování. Paměť se programuje po jednom bitu. Jakmile je paměť naprogramována, tak při opětovném programování zaplněných bytů je potřeba tyto byty elektricky vymazat.[3]

### 2.1 Vnitřní Čítač Adres

AT89C2051 obsahuje vnitřní PEROM adresový čítač, který je vždy nastaven na 00H při náběžné hraně resetovacího signálu a je vylepšen o přivedení kladného impulsu na pin XTAL1. [3]

### 2.2 Programovací Algoritmus

K programování AT89C2051 je doporučena tato sekvence:

1. Power-up sekvence:
  - připojení napájení mezi piny VCC a GND
  - uzemnění pinů RST a XTAL1
2. Nastavení pinů RST a P3.2 do log.1.
3. Přivedení správné kombinace log.1 a log.0 na piny P3.3, P3.4, P3.5, P3.7 k nastavení jedné z programovacích operací, zobrazených v tabulce.

Programování a ověření paměti:

4. Přivedení prvního byte programu na adrese 000H na piny P1.0 až P1.7.
5. Zvýšení napětí RST na 12V k aktivování programování.
6. Přivenení jednoho impulsu na P3.2 zapíše byte z P1 do paměti. Zapsání trvá obvykle 1,2ms.
7. Snížení napětí RST z 12V na log.1 a nastavení příslušné kombinace pinů P3.3 až P3.7 k ověření uložených dat. Výstuní data jsou na P1.
8. Přivedení impulsu na XTAL1 (zvýšení adresového čítače o 1) a přivedení dalších dat na P1.

9. Opakování kroků 6 až 8 dokud nebude uložen celý program.

10. Power-off sekvence:

- nastavení XTAL1 na log.0
- nastavení RST na log.0
- odpojení napájecího napětí

### 2.3 Data Polling

AT89C2051 umožňuje Data Polling k indikaci konce zapisovacího cyklu. Během zapisovacího cyklu jakýkoliv pokus o přečtení posledního zapsaného bytu má za následek poslání zapsaných dat na P1. Jakmile je zapisovací cyklus dokončen, data jsou platná na všech výstupech a může začít další cyklus. Data Polling může začít kdykoliv po inicializaci zapisovacího cyklu.[3]

### 2.4 Ready/Busy

Aktuální stav programování může být také monitorován výstupním signálem RDY/BSY. Pokud se během programování pin P3.1 se změní na log.0 po změně úrovně na pinu P3.2 na log.1 procesor je zaneprázdněn (BUSY). Pokud se P3.1 během programování znovu změní na log.1 procesor je připraven (READY).[3]

### 2.5 Ověření programu

Pokud lock-bity nebyly naprogramovány, data programu mohou být zpětně přečteny zkrz port a následně ověřeny následujícím způsobem:

1. Resetování vnitřního adresového čítače na adresu 000H změnou pinu RST z log.0 na log.1
2. Přivedení řídicího signálu pro čtení programu a následné přečtení z portu P1.
3. Přivedení impulsu na XTAL1 (zvýšení adresového čítače o 1)
4. Čtení dalšího bytu dat z portu P1.
5. Opakování kroků 3 a 4 dokud není přečtena celá paměť.

Lock-bity nemohou být ověřeny přímo.

## 2.6 Vymazání čipu

Celá paměť PEROM (2KB) a dva lock-bity může být elektricky vymazána příslušnou kombinací řídicích signálů a přivedení log.0 na port P3.2 po dobu 10ms. Během procesu mazání je paměť zaplněna log.1 a musí být provedena, aby mohl být zaplněný byte být znovu naprogramován.[3]

## 2.7 Čtení výrobních bytů

Výrobní byty mohou být přečteny stejným způsobem, jako standardní ověřování adres 000H, 001H a 003H s tím rozdílem, že na piny P3.5 a P3.7 musí být přivedena log.0. Přečtené hodnoty mohou být například následující:[3]

(000H) = 1EH značí, že čip byl vyroben firmou ATMEL

(001H) = 21H značí typ 89C2051

### 3 VYBRANÉ FUNKCE AT89C2051

#### 3.1 Časovače a čítače

Časovače a čítače (C/T) jsou 16-ti bitové čítací registry (přijde-li na ně impuls, zvýší svou hodnotu o 1). Každý může pracovat jako čítač, nebo časovač. Rozdíl je v tom, jaké vstupní signály zpracovávají. Časovač čítá vnitřní signál s periodou rovnou době trvání strojového cyklu (při  $f = 24\text{MHz}$  je to  $0,5\mu\text{s}$ ) čítač čítá vnější impulsy (čítač T0 ze vstupu P3.4 a T1 ze vstupu P3.5). Časovač/čítač se inkrementuje při každém impulsu a po přetečení přes hodnotu 65535 se nastaví příznakový bit TF0 nebo TF1 (Timer Fact), případně se vyvolá přerušení. Časovač/čítač T0 a T1 se každý skládá ze dvou 8-bitových registrů TLx a THx (x – číslo časovače/čítače).[4]

#### 3.2 Registry pro řízení časovače/čítače

##### 3.2.1 TMOD (Timer Mod)



Obr. č 5. Registr TMOD

Z obrázku č. 5 je patrné, že bity b7-b4 náležejí časovači/čítači 1 a bity b3-b0 časovači/čítači 0.

**GATE** – Pokud má bit hodnotu 0, tak běh C/T se povoluje pouze bitem TR. Pokud má hodnotu 1, tak běh C/T se povoluje logickým součinem bitu TR a příslušného signálu přerušení.

**C/T** – Pokud je nastaven na log.0, pak registr C/T pracuje v režimu časovače (čítá vnitřní strojové cykly). Ve stavu log.1 pracuje v režimu čítače (čítá vnější impulsy).

**M1, M0** – Slouží k nastavení režimů C/T.

Tab. č 4. Tabulka režimů

M1	M0	Režim
0	0	Režim 0
0	1	Režim 1
1	0	Režim 2
1	1	Režim 3

Režimy 0, 1 a 2 jsou pro oba C/T shodné a režim 3 je rozdílný.

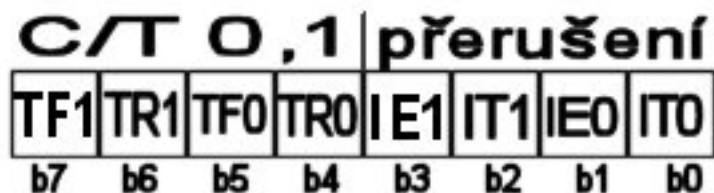
Režim 0 funguje jako 13-bitový C/T. Je tvořen nižšími 5 bity v registru TLx a registru THx. Nejvyšší 3 bity z TLx se zanedbávají. Důvodem existence tohoto režimu je zpětná kompatibilita s předchozími typy mikrokontrolérů. Princip ovládání a čítání je shodný jako v režimu 1.[4]

Režim 1 pracuje jako 16-bitový C/T, který je tvořen registry TLx a THx. Je-li povoleno čítání (bitem TR či vnějším vstupem), inkrementuje se jeho hodnota s každým dalším impulsem. Dojde-li k přetečení z nejvyšší hodnoty (1111111111111111B) na 0, automaticky se nastaví bit TFx. Pro opětovné vytváření dalšího časového zpoždění, je nutné vždy po přetečení časovač znovu naplnit.[4]

V režimu 2 C/T funguje jako 8-bitový s automatickým plněním. 8-bitový registr TLx čítá vzestupně a při přetečení nastaví příslušný bit TFx, ale zároveň se znovu automaticky naplní obsahem registru THx. Ten je nutno naplnit a nastavit předem programově. Tento režim se nejčastěji používá k časování seriového kanálu.[4]

Režim 3 je různý pro C/T 1 s pro C/T 0. Z TL0 a z TH0 se stanou dva oddělené 8-bitové čítače. TL0 je ovládán standardně řídicími bity pro C/T 0 a při přetečení nastaví bit TF0. Časovač tvořený TH0 je ovládán bitem TR1 a pracuje pouze jako časovač. Při přetečení nastaví bit TF1. C/T 1 nečítá (nepracuje) a uchovává si načítanou hodnotu z ostatních režimů. V tomto režimu jsou tedy k dispozici tři nezávislé C/T. C/T 0 v režimu 3 a C/T 1 lze zastavit nebo spustit přechodem z nebo do režimu 3., Tento režim se používá například na generování signálu pro časování seriové linky, nebo jej lze využít v jiné aplikaci nevyžadující přerušování.[4]

### 3.2.2 TCON (Timer Control)



Obr. č 6. Registr TCON

**TF0, TF1 (Timer Flag)** – Nastaví se do log.1 vždy, když časovač přeteče z 65535 na 0. Testováním tohoto bitu je možné zjistit, kdy časovač doběhl a přetekl. Je-li potřeba tento bit použít znovu, musí být bit TF nulován ručně (programem). Bit TF je nulován automaticky pouze tehdy, je-li časovač obsluhován přerušením.

**TR0, TR1 (Timer Run)** – Řídící bit, který umožňuje spuštění časovače.

**IE0, IE1 (Interrupt Enable)** – Příznakový bit externího přerušení.

**IT0, IT1** – V případě log.0 je externí přerušení vvoláno logickou úrovní 0 na vstupu, v případě log. 1 je externí přerušení vvoláno sestupnou hranou.

### 3.3 Přerušovací systém

Při normální činnosti provádí procesor instrukce podle programu a to rychlostí danou hodinovým kmitočtem. Vně procesoru probíhají děje asynchroně vzhledem k tomuto hodinovému kmitočtu a synchronizovat je lze dvěma způsoby:

1. Použití programového čekacího cyklu, který zajistí reakci procesoru na vnější děj. Reakce může být opožděná, dle délky smyčky.
2. Přerušovací systém. Procesor je vybaven vstupy, přes které přijímá žádosti o přerušení běžícího programu. Je-li přerušení povoleno, a zároveň je žádáno, procesor dokončí právě vykovávanou instrukci, uloží adresu následující instrukce a následně vykoná podprogram, který obslouží daný požadavek na přerušení. Po provedení tohoto podprogramu se vrátí k vykonávání původního programu na uložené adrese.[4]



Tab. č 5. Zdroje přerušení

Zdroje přerušení	
INT0	vnější přerušení 0 (P3.2)
INT1	vnější přerušení 1 (P3.3)
TF0	přerušení od časovače 0
TF1	přerušení od časovače 1
TI	přerušení při vysílání seriového kanálu
RI	přerušení při přijímání seriového kanálu

U vnějších zdrojů přerušení lze určit, za jakých okolností bude přerušení vyvoláno. To lze nastavit v registru TCON bity b3-b0. Přerušení od C/T je vyvoláno vastavením příznakových bitů TF0 a TF1. Je-li příslušné přerušení povoleno, příznakový bit se nuluje automaticky hardwarově podobně jako bit IE. Přerušení od seriového kanálu je generováno logickým součtem bitů RI a TI. Oba tyto bity se musí nulovat programově, jinak by nebylo možné určit skutečný zdroj přerušení.[4]

### 3.4 Registr IE (Interrupt Enable)

Každý ze zdrojů přerušení lze individuálně zakázat nebo povolit, ale lze také zakázat všechna přerušení najednou. K tomu slouží registr IE.



Obr. č 7. Registr IE

Pokud je bit registru nastaven na hodnotu log. 1, je příslušné přerušení povoleno, je-li v log. 0, přerušení je zakázáno.

**EA (Enable All)** – Povolení/zakázání všech přerušení.

**ES (Enable Seriól)** – Povolení/zakázání přerušení od seriového kanálu.

**ET0, ET1 (Enable Timer)** - Povolení/zakázání přerušení od časovače/čítače.

**EX0, EX1 (Enable External)** - Povolení/zakázání přerušení vnějšího zdroje přerušení.

### 3.5 Vektory přerušení

Aby bylo možno určit který podprogram se bude vykonávat, je-li vyvoláno některé z přerušení, jsou výrobcem předem stanoveny adresy v paměti programu, na které se mikrokontrolér odkáže, je-li vyvoláno příslušné přerušení. Tyto adresy se nazývají vektory přerušení.[4]

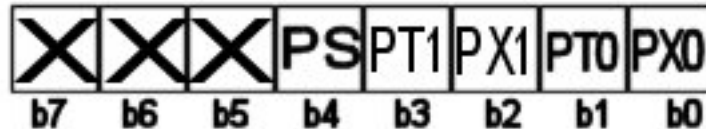
*Tab. č 6. Tabulka vektorů přerušení*

<b>Zdroj</b>	<b>Adresa vektoru přerušení</b>
IE0	03H
TF0	0BH
IE1	13H
TF1	1BH
RI, TI	13H

Program, který obsluhuje příslušné přerušení, musí být ukončen instrukcí RETI. Požadovanou instrukci (první je obvykle skok na místo, kde se podprogram nachází) lze umístit přesně na adresu příslušného vektoru pomocí direktivy ORG. [6]

### 3.6 Úrovně priority přerušeni

Každému zdroji přerušeni lze přiřadit jednu ze dvou úrovní priority a to pomocí registru IP. Hodnota bitu v registru IP 0 znamená nižší prioritu přerušeni a hodnota 1 vyšší.



Obr. č 8. Registr IP

Bity b7-b5 jsou neobsazeny. Bit PS určuje prioritu seriového kanálu, bity PT1 a PT0 slouží k nastavení priority časovače/čítače 1 a časovače/čítače 0 a bity PX1 a PX0 úrovně priority vnějšího přerušeni. Každý zdroj přerušeni lze samostatně naprogramovat a přiřadit mu vyšší prioritu. Všechny zdroje přerušeni mají implicitně nastavenou nižší prioritu. Při obsluze více požadavků najednou může dojít k následujícím situacím. Obslužný program přerušeni s nižší prioritou může být přerušen požadavkem na obsluhu přerušeni s vyšší prioritou. Přijdou-li současně dva požadavky, které mají stejnou prioritu, výběr se provede podle zakódované vnitřní výběrové sekvence: nejprve IE0, TF0, IE1, TF1 a nakonec RI/TI.[4]

## **II. PRAKTICKÁ ČÁST**

## 4 HARDWAROVÁ ČÁST

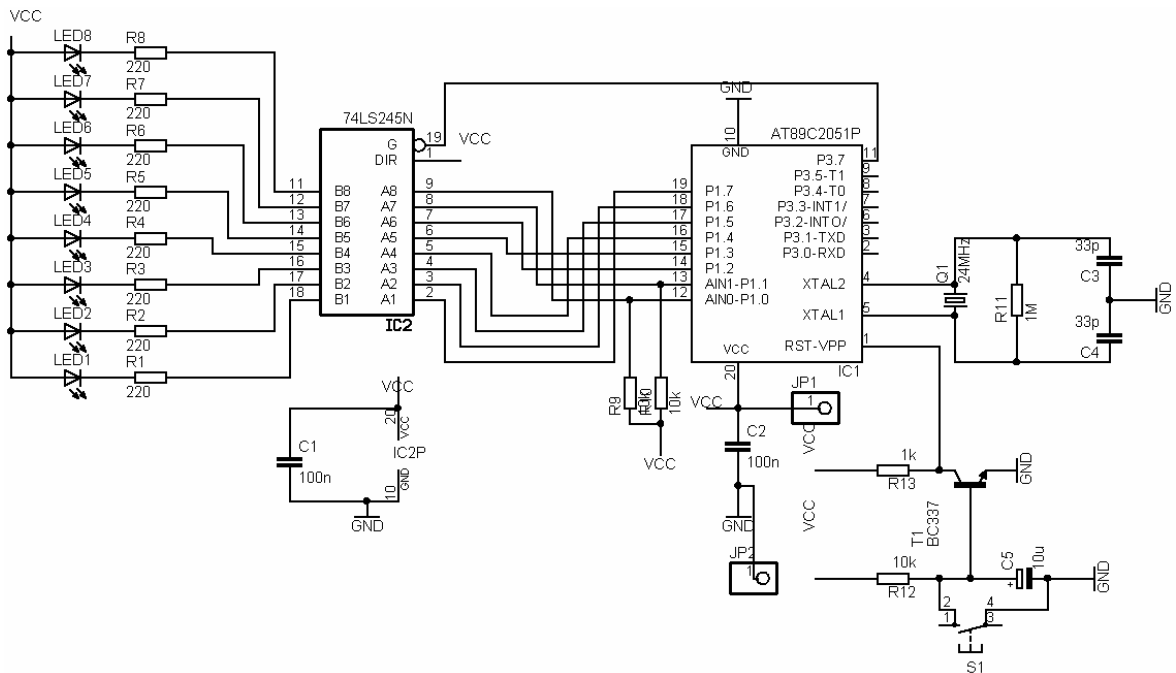
Harwarová část není náročná, jelikož samotný mikrokontrolér má většinu potřebných částí integrovaných. Přesto je kromě samotného mikrokontroléru na desce plošného spoje dalších 20 součástek.

Návrh desky plošného spoje probíhal v programu EAGLE což je uživatelsky přívětivý a výkonný nástroj pro návrh desek plošných spojů (DPS). Název EAGLE je zkratka, pocházející z původního názvu Easily Applicable Graphical Layout Editor. Návrhový systém se skládá ze tří hlavních modulů a to schématický editor, editor plošného spoje a autorouter. [1]

Vlastnosti:

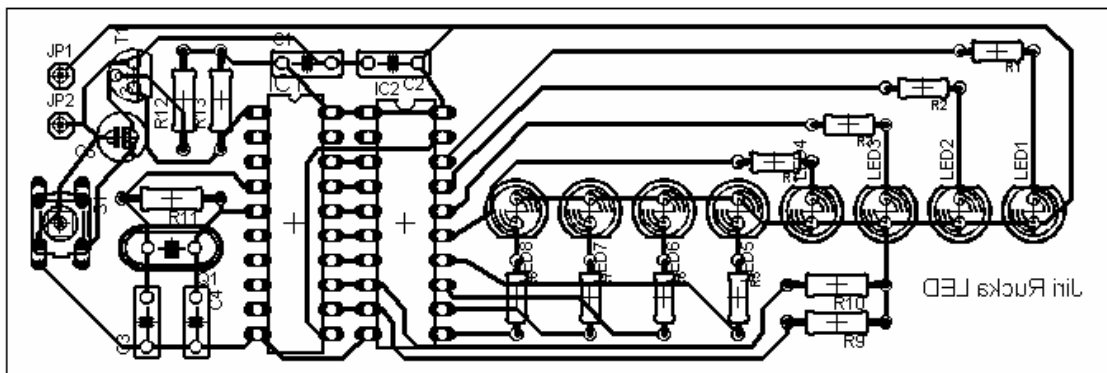
- Jednoduchý přechod mezi editory
- Žádná hardwarová ochrana
- Výkonný uživatelský jazyk
- Integrovaný textový editor
- Snadné vytváření vlastních součástek v editoru knihoven
- Kontrola elektrických návrhových pravidel
- Dostupný pro operační systémy Windows, Linux a MAC
- Není náročný na hardware počítače

## 4.1 Vrtule



Obr. č 9. Schema zapojení vrtule

Základem je mikrokontrolér AT89C2051. Na jeho piny XTAL1 a XTAL2 je připojen krystal s frekvencí 24MHz. Kondenzátory C4 a C3 by měly být podle katalogu 30pF. Použité kondenzátory mají hodnotu 33pF (řada E12) a tudíž jsou v toleranci 10pF udávané katalogem ATMEL. Reset mikrokontroléru je prováděn přes RC člen, tvořený rezistorem R12 a kondenzátorem C5 a tranzistorem T1. Kondenzátory C1 a C2, připojené na napájení integrovaných obvodů, jsou filtrační a slouží k odfiltrování vysokých frekvencí. Port 1 je celý přiveden na vstup integrovaného obvodu 74LS245N, což je budič LED diod. Piny P1.0 a P1.1 nemají vnitřní pull-up rezistory, proto jsou k nim připojeny externí pull-up rezistory R9 a R10. Na výstup budiče už jsou připojeny, přes ochranné rezistory R1-R8, katody LED diod.



Obr. č 10. Osazená deska plošného spoje

Modul LED diod je připojen na port P1 v opačném pořadí (LED 8 je připojena na pin P1.0). Ze zapojení je zřejmé, že LED diody se rozsvěčují přivedením logické 0 na výstup portu. Pin P3.7 je připojen k řídicímu vstupu budiče LED diod, který je aktivován přivedením logické 0 na výstup pinu portu.

Tab. č 7. Rozpiska součástek

T1	BC337
IC1	AT89C2051 – 24PU
IC2	74LS245N
LED1,LED8	L1503GD
Q1	24MHz
R1-R8	220Ω
R9,R10,R12	10kΩ
R11	1MΩ
R13	1kΩ
C1,C2	100nF
C3,C4	33pF
C5	10μF/10V
S1	tlačítko TM043
patice IO	DIL20

## 4.2 Výroba plošného spoje

Nejprve je plošný spoj vytiskut stranou spojů na folii pro laserové tiskárny (měla by odolávat teplotám kolem 250°C) s dobrou propustností UV záření. Dále je folie přiložena k desce a osvívena UV zářením. Černé místa (vodivé cesty) zůstanou neosvíceny a nakopak bílá místabudou osvícena a odleptána. valita výsledného plošného spoje je závislá na mnoha faktorech, např. správná volba fólie, rozlišení tiskárny, intenzita UV – záření a kvalita fotocitlivého plošného spoje. Vyrábí se desky plošných spojů s nanesenou vrstvou, nebo musí být tato vrstva doplněna nástřikem. Mědí obvrstvenou desku je třeba zbavit nečistot a mastnoty. Poté za sníženého osvětlení a pokojové teploty nanést fotocitlivou vrstvu. Tato vrstva je vytvrzena po 3 – 5 hodinách, správně vytvrzený fotocitlivý lak má zelenou barvu a tím je připraven k nasvícení. Fotocitlivý lak po svém vytvrzení reaguje na UV – záření, proto bude zapotřebí zdroj tohoto záření, například rtuťová výbojka. Vzdálenost desky plošných spojů od zdroje UV záření je ovlivněna materiály přes které musí záření pronikat. Na fotocitlivé vrstvě desky je umístěna fólie s obrazcem spojů a je fixována sklem, které zajišťuje, aby nedocházelo ke zkroucení fólie a tím ke snížení kvality plošného spoje. Doba osvitu desky je závislá na nanesené vrstvě fotocitlivého laku. V běžných podmínkách je doba osvitu 3 – 5 minut ze vzdálenosti 20 – 30 cm pro jednu stranu plošného spoje. Nasvícená deska musí být vyvolána a vyleptána. Na vyvolání desky plošných spojů používáme 2% roztok hydroxidu sodného s vodou do kterého je deska vložena na dobu přibližně 2 minut. Roztok hydroxidu sodného reaguje s nasvícenými místy plošného spoje a vymývá je. Na desce zůstává jen neosvícená část. Po vyvolání plošného spoje musí být deska opláchnuta destilovanou vodou a vysušena. Následuje leptání, které musí být pozvolné. Existují dvě varianty leptání plošných spojů. První variantou, pomalejší a také bezpečnější je za pomoci 40% roztoku chloridu železitého, nebo v roztoku připraveného z 770ml destilované vody, 200ml 30% kyseliny chlorovodíkové a 30ml 30% peroxidu vodíku. Druhá varianta je však nebezpečná, protože při leptání uniká do ovzduší chlór, který naleptává sliznici a tím škodí lidskému zdraví. Vysušená deska je vložena do roztoku, kde jsou odleptány všechny osvícená místa. Poté musí být deska opláchnuta vodou. Na vyleptaném plošném spoji zůstává fotocitlivý lak, který odstraníme lihem nebo acetonem a desku ošetříme proti oxidaci a znečištění ochranným lakem.



### 4.3 Programátor PATMEL3

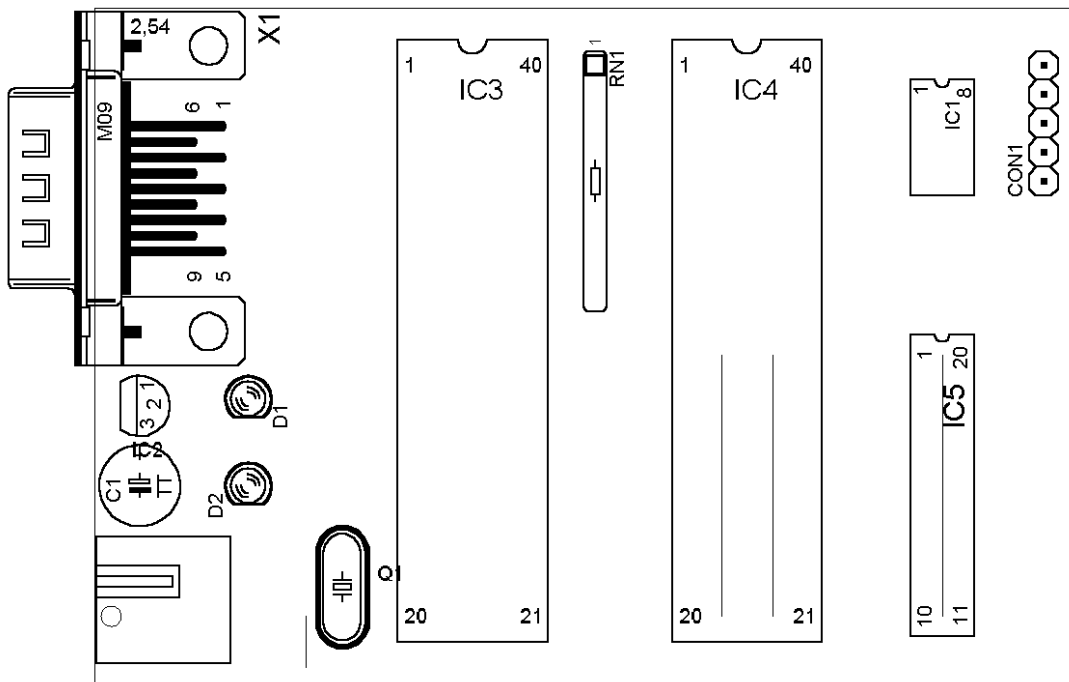
K programování mikrokontroléru AT86C2051 jsem použil programátor PAtmel3 od Ing. Davida Hofra, který je vylepšenou verzí programátoru PAtmel II, podporující většinu procesorů řady 51 a AVR firmy Atmel. Zapojení umožňuje všechny potřebné operace s podporovanými obvody, tj. čtení, zápis, mazání a verifikaci datové i programové paměti a nastavování lock i fuse bitů procesoru. Na desce programátoru jsou 3 precizní patice pro programované obvody v pouzdrech DIP8, DIP20 a DIP40. U většiny obvodů funguje autodetekce typu pomocí signature bytes, tzn. že není nutné nastavovat typ ručně. Hlavním vylepšením je ale ISP konektor pro programování procesorů přímo v aplikaci, které výrazně urychluje odlaďování programů. Reset programovaného procesoru je možné provádět pomocí tlačítka v liště obslužného programu.



Jádem programátoru je opět procesor AT89C51, zajišťující všechny operace s programovanou součástkou, která se vkládá do příslušné precizní patice. Napájení zajišťuje externí zdroj 12V/300mA. Napájení indikuje zelená LED, probíhající operaci červená LED. K počítači je programátor připojen přes sériový port.

Tab. č 8. Seznam podporovaných procesorů a pamětí

<b>Procesory 51</b>	AT89C51, AT89LV51, AT89C52, AT89LV52, AT89C55, AT89LV55, AT89S8252, AT89LS8252, AT89S53, AT89LS53 AT89C1051, AT89C1051U, AT89C2051, AT89C4051
<b>Procesory AVR</b>	AT90S1200, AT90S2313
<b>Procesory s podporou ISP</b>	AT89S8252, AT89LS8252, AT89S53, AT89LS53, AT90S1200, AT90S2313, AT90S2323, AT90LS2323, AT90S2343, AT90LS2343, AT90S2333, AT90LS2333, AT90S4433, AT90LS4433, AT90S4434, AT90LS4434, AT90S8535, AT90LS8535
<b>Paměti EEPROM</b>	AT24C01A, AT24C02, AT24C04, AT24C08, AT24C16, AT24C32, AT24C64, AT24C128, AT24C256, AT24C512, AT93C46, AT93C56, AT93C57, AT93C66, AT93C86



Obr. č 12. Rozložení součástek na desce plošného spoje

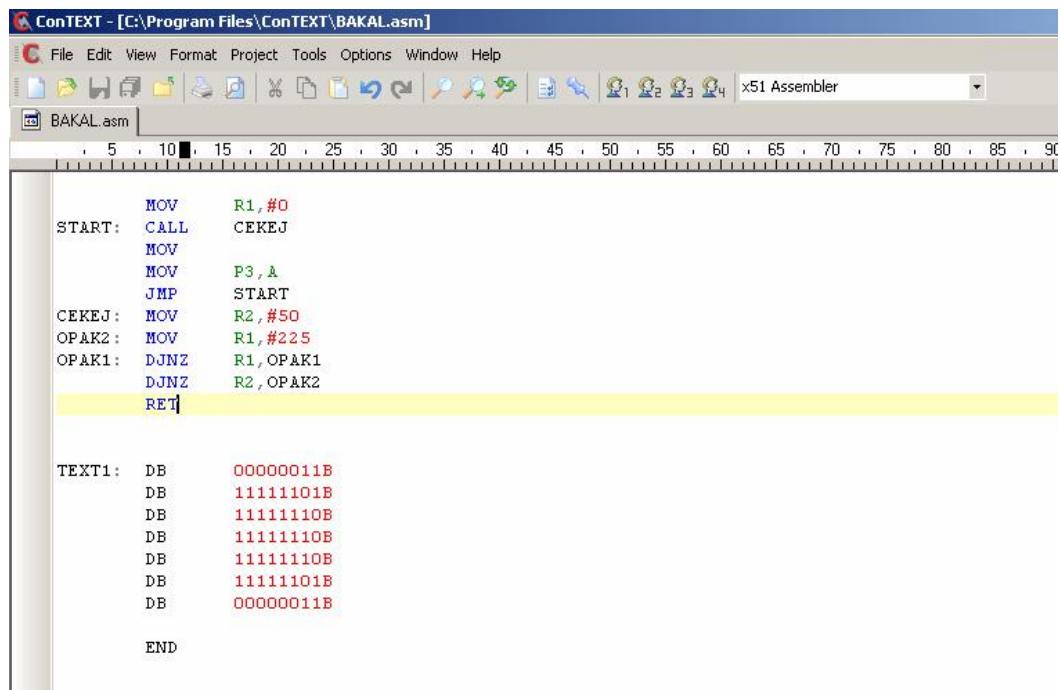
Na obrázku 12 je rozmístění nejdůležitějších součástek na desce plošného spoje. Integrovaný obvod IC3 je mikrokontrolér AT89C4051, který slouží k programování ostatních mikrokontrolérů IC1, IC4 a IC5. Tyto mikrokontroléry mohou být IC1- AT89C1051, IC4 – AT89C4051 a IC5 – AT89C2051.

## 5 PROGRAMOVÁ ČÁST

### 5.1 Programovací prostředí ConTEXT a Patmel3

#### 5.1.1 ConTEXT

Program ConTEXT je volně šiřitelný (freeware) textový editor. Používá se jako náhrada standardního textového editoru Notepad, který je součástí Windows. Jeho výhoda spočívá především v tom, že po nainstalování příslušného balíčku (highlighter) zvýrazňuje syntaxy programovacích jazyků. V mém případě jsem použil highlighter x51 Assembler.



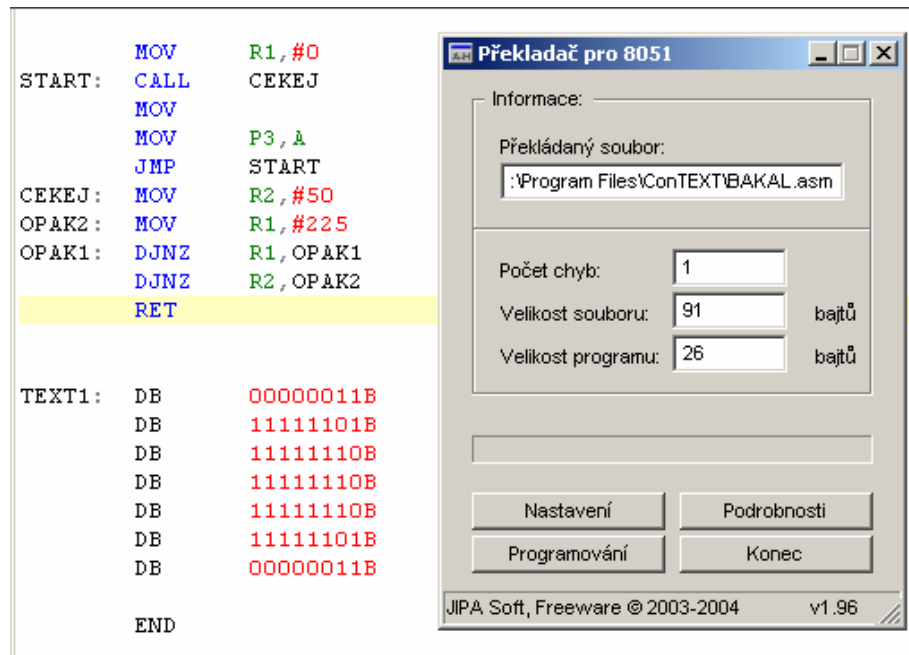
```
ConTEXT - [C:\Program Files\ConTEXT\BAKAL.asm]
File Edit View Format Project Tools Options Window Help
BAKAL.asm
x51 Assembler
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90
MOV R1, #0
START: CALL CEKEJ
MOV P3, A
JMP START
CEKEJ: MOV R2, #50
OPAK2: MOV R1, #225
OPAK1: DJNZ R1, OPAK1
DJNZ R2, OPAK2
RET

TEXT1: DB 00000011B
DB 11111101B
DB 11111110B
DB 11111110B
DB 11111110B
DB 11111101B
DB 00000011B

END
```

Obr. č 13. Highlighter x51 Assembler

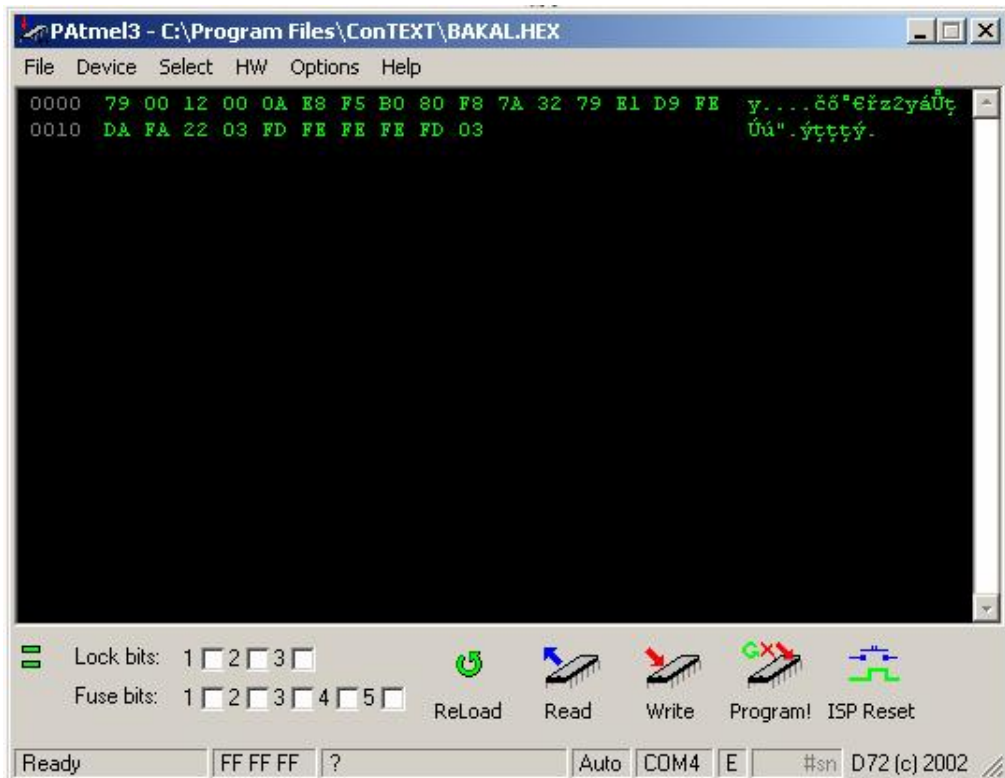
Další výhodou tohoto prostředí je možnost nainstalování softwaru pro programátor EmulEasy51, který nastaví uživatelská makra v ConTEXTu na klávesy F9 (kompilace), F10 (programování) a F12 (nastavení EmulEasy51). Jelikož jsem používal programátor PATmel3, využil jsem pouze kompilaci souboru. Během kompilace EmulEasy51 přeloží uživatelský program do strojového kódu a uloží ho do souboru s příponou HEX, který lze použít v jiném programátoru.



Obr. č 14. Kompilátor EmulEasy51

### 5.1.2 PATmel3

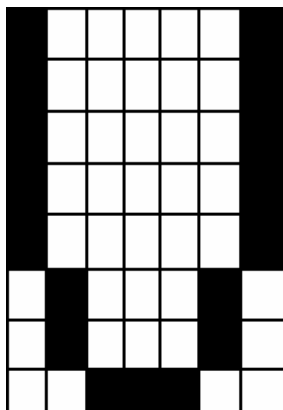
PATmel je volně šiřitelný software k programátoru PATmel3. Pracuje tak, že data z vstupního souboru (soubor strojového kódu s příponou HEX) zkopíruje do programátoru a ten naprogramuje mikrokontrolér. Komunikace mezi programátorem a softwarem probíhá přes seriovou linku počítače (v mém případě přes replikátor seriového portu do USB). Přestože by měl program zjistit automaticky jaký typ mikrokontroléru je v patci programátoru (díky přečtení signature bytes), autor uvádí na svých webových stránkách, že se setkal s problémem u typu AT89C2051, kdy se přepsaly tyto byty a typ bylo nutné nastavit ručně. Já jsem ve své práci tento typ používal a programátor vždy automaticky našel typ mikrokontroléru.



Obr. č 15. Software k programátoru PATmel3

## 5.2 Zobrazení písmen

Zobrazování jsem volil tak, že písmena jsou zobrazena v matici bodů 8x7 nebo 5x7, vytvořené pomocí LED diod, v závislosti na délce zobrazovaného textu. Dolnímu bodu sloupce odpovídá pin P1.0 a hornímu P1.7.



Obr. č 16. Zobrazení písmena U

### 5.3 Nastavení časovačů

Časovače jsem v programu použil dva. Časovač 0 pro dobu jedné otočky vrtule a časovač 1 pro dobu zobrazení jednoho sloupce textu. Časovače jsem nastavil tak, že se spouští bitem TR a pracují v režimu 1 (16-bitový čítač). Dále je potřeba časovač přednaplnit hodnotou, od které bude čítat až do přetečení a následného vyvolání přerušení. Nejprve jsem spočítal dobu trvání strojového cyklu při frekvenci krystalu 24MHz (1).

$$T = \frac{12}{f} = \frac{12}{24 \cdot 10^6} = 0,5 \text{ms} \quad (1)$$

Použitý mororek má podle výrobce 2000 otáček/min, z čehož jsem následně spočítal dobu tvání jedné otáčky (2).

$$f = \frac{f}{60} = \frac{2000}{60} \cong 33,3 \text{ot} / \text{s} \quad (2)$$

$$T = \frac{1}{f} = \frac{1}{33} = 0,03 \text{s} = 30000 \text{ms}$$

Z toho vyplývá, že při době trvání strojového cyklu 0,5μs čítač musí napočítat 60000 strojových cyklů než dojde k přerušení. Jelikož je čítač 16-bitový (65536), je potřeba ho naplnit hodnotou 5536. Toho lze docílit třemi způsoby.

1. Převedením čísla 5536 do binární soustavy a prvních 8 bitů uložením do registru TH0 a zbylých 8 do registru TH0.

```
MOV     TH0, #00010101B
MOV     TLO, #10100000B
```

Obr. č 17. Nastavení časovače 1. metodou

2. Uložením do registrů TH0 a TLO pomocí operandu LOW a HIGH.

```
MOV     TH0, #HIGH (5536)
MOV     TLO, #LOW (5536)
```

Obr. č 18. Nastavení časovače 2. metodou

3. Uložením počtu impulsů potřebných k vyvolání přerušení se záporným znaménkem do registrů TH0 a TH1 pomocí operandu LOW a HIGH.



```

MOV     TH0, #HIGH (-60000)
MOV     TLO, #LOW (-60000)

```

Obr. č 19. Nastavení časovače 3. metodou

Ve svém programu jsem používal pro oba časovače 3. metodu, protože při ní není potřeba převádět počet impulsů do dvojkové soustavy, ani odečítat od maximální hodnoty časovače. Časovač 1 používám na zobrazení jednoho sloupce textu. Jeho hodnota je tedy závislá na zobrazovaném textu. Například pro nápis „U T B“ má hodnotu 3750 nebo pro nápis „FAKULTA“ má hodnotu 1875. Mimo tyto dva časovače jsem si vytvořil ještě jeden programový časovač, který časuje dobu zobrazení jednoho řádku textu. Tento časovač jsem vytvořil z časovače 0 a z registrů R3 a R4. R4 má implicitní hodnotu 255 a R3 100. Po každém přetečení časovače 0 se sníží obsah R4 o 1. Pokud klesne R4 až na 0, sníží se hodnota R3 o 1 a do R4 se opět uloží hodnota 255. Až hodnota R3 klesne na 0, zobrazí se na displeji další řádek a registry se obnoví do původních hodnot.

#### 5.4 Přerušeni a jeho obslužné rutiny

Možnost kontroly přetečení časovačů pomocí přerušeni jsem využil u obou dvou použitých časovačů. Do registru IE jsem uložil hodnotu 10001010b. To znamená, že jsou nastaveny bity EA (povolení veškerých přerušeni), ET1 (přerušeni od časovače 1) a ET0 (přerušeni od časovače 0). Registr úrovní priorit přerušeni IP jsem ponechal bezezměny (všechny přerušeni mají úroveň priority 0), protože mně zcela vyhovuje vnitřní výběrová sekvence.

```

ORG     OH
JMP     START
ORG     OBH
JMP     TIMERO
ORG     1BH
JMP     TIMER1

```

Obr. č 20. Vektory přerušeni

Z důvodu malého místa v paměti mezi jednotlivými vektory přerušení se na prvním řádku vektoru přerušení používá skoková instrukce do jiné části programu, kde je celá obsluha přerušení. Používá se direktiva ORG, jak ukazuje obrázek 20.

Přetečení časovače 0 znamená, že vrtule se nachází ve výchozí pozici a řádek textu se musí znovu vykreslit. V jeho obsluhovací rutině je nejprve vynulován bit TR0. Tím dojde k jeho zastavení. Poté je do akumulátoru vložena hodnota z registru R1. V registru R1 je počet sloupců právě zobrazovaného textu. Následuje část programového čítače, který je popsán v kapitole 5.3. Pak už jenom následuje naplnění čítače na výchozí hodnotu (5536) a jeho spuštění a spuštění časovače 1. V obsluze přerušení od časovače 1 je jako první vynulován řídicí bit TR1. Následuje porovnání registrů R0 a R1. V registru R0 je uložena pozice právě zobrazovaného sloupce v řádku textu. Jelikož instrukce CJNE (Compare and Jump if Not Equal) nedovoluje přímo porovnávat dva registry, je potřeba registr R1 zkopírovat do akumulátoru a potom ho porovnat přímo s adresou registru R0 (jeho adresa je 0H). V případě že hodnoty nejsou stejné (ještě není zobrazován poslední sloupec textu), vybere se z tabulky další řádek. Výběr probíhá přes Data Pointer (DPTR) a akumulátor, ve kterém je obsah registru R0. Číslo v DPTR udává bázovou adresu v paměti programu (adresu začátku tabulky sloupců textu). K hodnotě DPTR se přičte hodnota akumulátoru, která vyjadřuje posunutí od bázové adresy. Součet DPTR a akumulátoru se vezme jako nepřímá adresa, která ukazuje do paměti programu a obsah se uloží do akumulátoru a pošle na port P1. Následuje inkrementace registru R0 a znovunastavení a spuštění časovače. Pokud už nastal konec řádku, tak jsou na port P1 posány samé jedničky a časovač už není znovu spuštěn. Znovu je spuštěn až při přetečení časovače 0.

## 5.5 Hlavní smyčka programu a tabulky textu

Hlavní smyčka začíná nastavením všech tří časovačů, jejich naplnění a nastavení přerušení. Dále je na pin P3.7 přivedena logická 0, která aktivuje budič diod 74LS245N. Následuje uložení do DPTR první tabulka textu. Instrukce CJNE porovnává obsah registru R3. Pokud není nula, což znamená že ještě není čas ke změně řádku, instrukce skočí na ten řádek, kde začíná. To je zajištěno symolem \$ v místě pro návěští. Pokud je R3 rovno 0, začne se

zobrazovat další řádek textu. Nejprve se do registru R1 uloží celkový počet sloupců textu, nastaví se znovu programový časovač a program čeká na instrukci CJNE, dokud nebude hodnota registru R3 nulová. Takto se zobrazí všechny řádky textu a program je zacyklený v nekonečné smyčce.

Každá tabulka musí začínat návěštím a její jednotlivé řádky instrukcí DB, která uloží 8-bitové hodnoty (čísla) do paměti programu. Řádky by měly být odděleny čárkou, nebo každý na jednom řádku. Tabulka je vždy složena z písmen a dvěma prázdnými sloupci jako mezery mezi nimi. Mezery mezi písmeny by mohly být řešeny pomocí časovače, ale oba dva už jsou používány.

## ZÁVĚR

Hlavním cílem této bakalářské práce bylo navrhnout a realizovat soustavu, která by na listech točící se vrtule zobrazovala reklamní nápisy, za použití mikrokontroléru architektury 8051.

První část se zabývá mikrokontrolérem ATMEL AT89C2051. Nejprve jsou popsány jeho hardwarové vlastnosti (rozložení pinů, zapojení krystalu,..), poté zbůsob, jakým lze tento mikrokontrolér naprogramovat a nakonec jsou popsány jedny z jeho nejdůležitějších a nejpoužívanějších funkcí: časovače/čítače a přerušovací systém.

Ve druhé části je navržena model vrtule v programu EAGLE. Dále je vysvětleno schema zapojení a rozložení součástek na desce plošného spoje a jeho smotná výroba. V další části je popsán použitý programátor PAtmel3 od ing. Davida Hofra. Výhodou tohoto programátoru je, že je možné stáhnout z autorových stránek jeho schema zapojení, desku plošného spoje a tento programátor samostatně vyrobit, popřípadě objednat hotový nebo jako stavebnici.

V programové části je popsáno nejprve využití softwarových prostředků k napsání a následnému zapsání řídicího programu do paměti mikrokontroléru. Jako nejvhodnější se ukázala volba textového editoru ConTEXT s doinstalovaným balíčkem pro zvýrazňování syntaxí v jazyce symbolických adres Assembler x51 a kompilátorem EmulEasy51. Pro zapsání programu do mikrokontroléru byl použit uživatelský software k programátoru PAtmel3. Řídicí program se skládá z páce s časovači, přerušením a tabulkami.

Výhodou tohoto systému je nízká cenová náročnost. Použitý mikrokontrolér je v prozeji za 35,- Kč. Nevýhodou je zaprvé malá aerodynamičnost vrtule, tudíž časové konstanty pro časovače je nutné ještě ručně opravit a zadruhé proudový odběr použitého modelářského motorku, který mže dostahovat hodnoty až 10A.

## ZÁVĚR V ANGLIČTINĚ

The main aim of this thesis was to design and create a system, which can display advertisement text on a rotating propeller using architecture 8051 microcontroller AT89C2051.

The first part describes microcontroller AT89C2051. First of all, hardware features are described (pin configuration, oscillator configuration, ...), then method of programming this microcontroller and finally one of its most important and most useful features: counters/timers and interrupt system.

In the second part a model of propeller is designed using software EAGLE. Then next wiring diagram is described, device positions on a printed circuit board and manufacturing. A programmer PATmel3, used to program FLASH memory, designed by Ing. David Hofr is described in next part. Advantage of this programmer is possibility of downloading all necessary documents to make own programmer or ordering a construction set.

In software part used software is described. The most useful development software was text editor ConTEXT with Assembler x51 highlighter and EmulEasy51 compiler. A Patme3 software was used to write a control program into a FLASH memory. Control program is divided to three parts. Counters/Timers, interrupt routine and working with tables.

The main advantage of this system is low cost. The cost of used microcontroller is 35 CZK. Disadvantages are low aerodynamic of the propeller and the current drain of used engine, which can be 10A.

## SEZNAM POUŽITÉ LITERATURY

- [1] HRABOVSKÝ, M.: Eagle pro začátečníky, Praha, BEN, 2002
- [2] [www.intel.com](http://www.intel.com)
- [3] [www.atmel.com](http://www.atmel.com)
- [4] Fronc Vojtěch, Klůčik Ján: Mikrokontroléry ATMEL s jádrem 8051, Praha, BEN, 2002
- [5] Katalog EZK Rožnov pod Radhoštěm, 2007
- [6] Vacek Václav: Učebnice programování ATMEL s jádrem 8051

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

AT	ATMEL
C/T	Counter/Timer – časovač/čítač
LED	Light-Emitting Diode
DPS	Deska plošného spoje
DPTR	Data Pointer
PEROM	Programmable and Eerasable Read-Only Memory

**SEZNAM OBRÁZKŮ**

<i>Obr. č 1. Rozložení pinů mikrokontroléru AT89C2051 .....</i>	<i>11</i>
<i>Obr. č 2. Blokovaná schéma mikrokontroléru AT89C2051 .....</i>	<i>12</i>
<i>Obr. č 3. Zapojení vnitřního oscilátoru .....</i>	<i>14</i>
<i>Obr. č 4. Zapojení externího hodinového signálu .....</i>	<i>14</i>
<i>Obr. č 5. Registr TMOD.....</i>	<i>22</i>
<i>Obr. č 6. Registr TCON.....</i>	<i>24</i>
<i>Obr. č 7. Registr IE.....</i>	<i>25</i>
<i>Obr. č 8. Registr IP .....</i>	<i>27</i>
<i>Obr. č 9. Schema zapojení vrtule .....</i>	<i>30</i>
<i>Obr. č 10. Osazená deska plošného spoje .....</i>	<i>31</i>
<i>Obr. č 11. Schema zapojení PAtmel3.....</i>	<i>34</i>
<i>Obr. č 12. Rozložení součástek na desce plošného spoje.....</i>	<i>35</i>
<i>Obr. č 13. Highlighter x51 Assembler .....</i>	<i>37</i>
<i>Obr. č 14. Kompilátor EmulEasy51.....</i>	<i>38</i>
<i>Obr. č 15. Software k programátoru PAtmel3.....</i>	<i>39</i>
<i>Obr. č 16. Zobrazení písmena U.....</i>	<i>39</i>
<i>Obr. č 17. Nastavení časovače 1. metodou .....</i>	<i>40</i>
<i>Obr. č 18. Nastavení časovače 2. metodou .....</i>	<i>40</i>
<i>Obr. č 19. Nastavení časovače 3. metodou .....</i>	<i>41</i>
<i>Obr. č 20. Vektory přerušení .....</i>	<i>41</i>



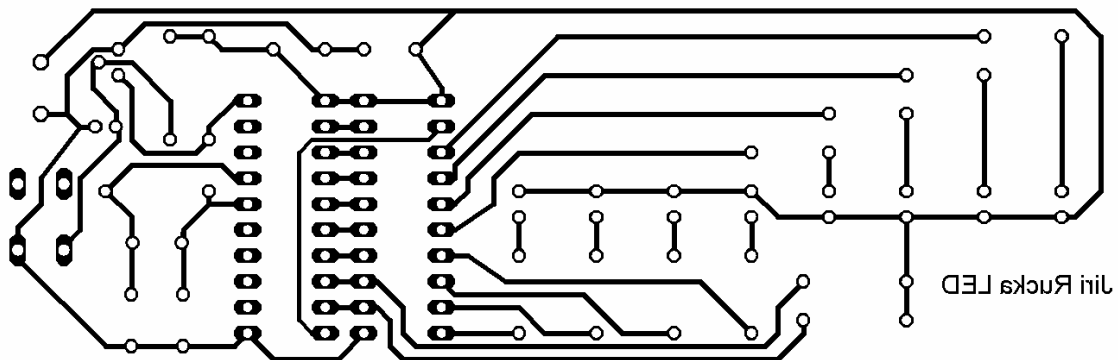
**SEZNAM TABULEK**

<i>Tab. č 1. Alternativní použití pinů portu P3.....</i>	<i>13</i>
<i>Tab. č 2. Mapa Registru speciálních funkcí a implicitní hodnoty.....</i>	<i>15</i>
<i>Tab. č 3. Módy Lock-bitových zámků.....</i>	<i>17</i>
<i>Tab. č 4. Tabulka režimů.....</i>	<i>23</i>
<i>Tab. č 5. Zdroje přerušení.....</i>	<i>25</i>
<i>Tab. č 6. Tabulka vektorů přerušení.....</i>	<i>26</i>
<i>Tab. č 7. Rozpiska součástí.....</i>	<i>31</i>
<i>Tab. č 8. Seznam podporovaných procesorů a pamětí.....</i>	<i>35</i>

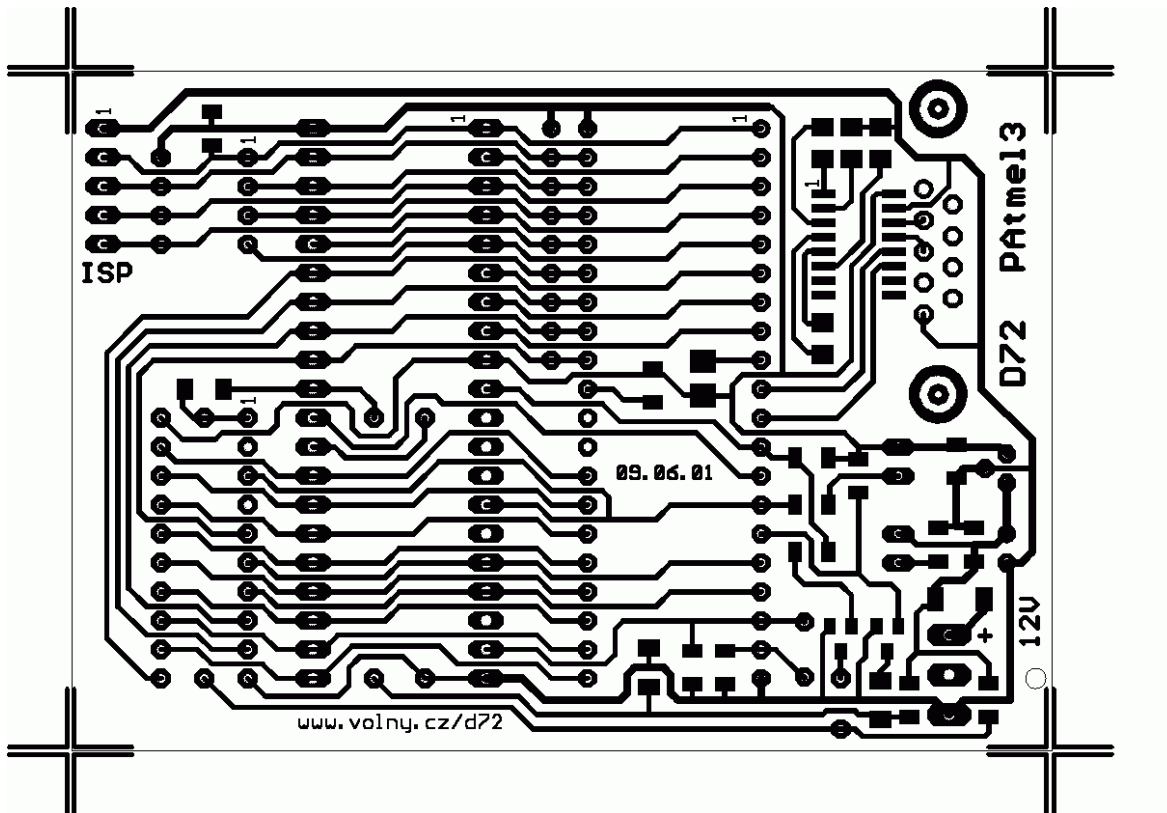
## SEZNAM PŘÍLOH

- P I Deska plošného spoje vrtule – strana spojů
- P II Deska plošného spoje programátoru - strana spojů
- P III Programátor PAtmel3
- P IV Eeltronické přílohy a verze bakalářské práce na CD

**PŘÍLOHA P I: DESKA PLOŠNÉHO SPOJE VRTULE – STRANA  
SPOJŮ**



PŘÍLOHA P II: DESKA PLOŠNÉHO SPOJE PROGRAMÁTORU –  
STRANA SPOJŮ



### **PŘÍLOHA P III: PROGRAMÁTOR PATMLEL3**

