

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ**

# **BAKALÁŘSKÁ PRÁCE**

**Univerzální USB programátor pro mikroprocesory**

Autor práce: Lukáš Ferkl  
Vedoucí práce: Ing. Karel Čermák

2012

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2011/2012

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš FERKL**  
Osobní číslo: **E09B0262P**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Elektronika a telekomunikace**  
Název tématu: **Univerzální USB programátor pro mikroprocesory**  
Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

### Z á s a d y p r o v y p r a c o v á n í :

1. Srovnajte dostupné univerzální programátory pro programování mikroprocesorů.
2. Navrhněte a realizujte univerzální USB programátor pro 8-bitové mikroprocesory Atmel AVR a Microchip PIC, popř. pro další platformy.
3. Napište a odlaďte jednoduchý ovládací SW pro PC.
4. Funkci sestrojeného programátoru ověřte na několika dostupných mikroprocesorech.
5. Vytvořené technické a programové prostředky podrobně popište.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **20 - 30 stran**

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

Seznam odborné literatury:

**Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.**

Vedoucí bakalářské práce:

**Ing. Karel Čermák**

Katedra aplikované elektroniky a telekomunikací

Konzultant bakalářské práce:

**Ing. Karel Čermák**


Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: **17. října 2011**

Termín odevzdání bakalářské práce: **3. června 2012**

  
Doc. Ing. Jiří Hammerbauer, Ph.D.  
děkan



  
Doc. Dr. Ing. Vjačeslav Georgiev  
vedoucí katedry

V Plzni dne 17. října 2011

## **Abstrakt**

Předkládaná bakalářská práce je zaměřena na návrh a praktickou realizaci univerzálního USB programátoru pro 8-bitové mikroprocesory Atmel AVR a Microchip PIC. Poukazuje na možnosti využití mikroprocesor s integrovaným USB modulem v kombinaci s virtuálním sériovým portem.

## **Klíčová slova**

Mikroprocesor, Univerzální programátor, Virtuální sériový port, Microchip, PIC, Atmel, AVR, USB

## **Abstract**

Ferkl, Lukáš. Universal USB programmer for microprocessors. University of West Bohemia. Faculty of Electrical Engineering. Department of Applied Electronics and Telecommunications.

Supervisor: Ing. Karel Čermák

The bachelor thesis presents the design and practical realization of the universal USB programmer for 8-bit microcontrollers Atmel AVR and Microchip PIC. It refers to possibilities of use of microcontroller with integrated USB module in combination with a virtual serial port.

## **Key words**

Microcontrollers, Universal programmer, Virtual serial port, Microchip, PIC, Atmel, AVR, USB

## Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 8.6.2012

.....  
Podpis

# Obsah

<b>OBSAH</b> .....	<b>6</b>
<b>SEZNAM SYMBOLŮ A ZKRATEK</b> .....	<b>7</b>
<b>ÚVOD</b> .....	<b>8</b>
<b>1 UNIVERZÁLNÍ PROGRAMÁTORY MIKROPROCESORŮ</b> .....	<b>9</b>
1.1 PROGRAMÁTOR PRESTO .....	9
1.2 PROGRAMÁTOR PICKIT .....	9
1.3 PROGRAMÁTOR BEEPROG .....	10
<b>2 POPIS POUŽITÝCH OBVODŮ</b> .....	<b>11</b>
2.1 MIKROPROCESOR PIC 18F4550 .....	11
2.2 OBVOD MC34063A .....	12
<b>3 VLASTNÍ KONSTRUKCE</b> .....	<b>13</b>
3.1 MOŽNÉ ZPŮSOBY ŘEŠENÍ KONSTRUKCE .....	13
3.1.1 <i>Převodník USB-RS232</i> .....	13
3.1.2 <i>Použití obvodu MCP2200</i> .....	14
3.1.3 <i>Použití mikroprocesoru s USB modulem</i> .....	14
3.2 ZVOLENÝ ZPŮSOB ŘEŠENÍ KONSTRUKCE .....	15
3.2.1 <i>Programování PIC</i> .....	15
3.2.2 <i>Programování AVR</i> .....	16
3.2.3 <i>Soubor HEX</i> .....	18
3.2.4 <i>Navržený programátor</i> .....	19
3.2.5 <i>Program pro mikroprocesor programátoru</i> .....	19
3.2.6 <i>Ovládací software</i> .....	21
3.2.7 <i>Diagramy průběhů ovládacího software</i> .....	22
3.2.8 <i>Možný další vývoj</i> .....	23
<b>ZÁVĚR</b> .....	<b>25</b>
<b>SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ</b> .....	<b>26</b>
<b>PŘÍLOHY</b> .....	<b>27</b>

## Seznam symbolů a zkratek

USB.....	Universal serial bus <i>Univerzální sériová sběrnice, slouží k připojení zařízení.</i>
VID .....	Vendor ID <i>Obsahují jej USB zařízení, identifikuje výrobce.</i>
PID .....	Product ID <i>Obsahují jej USB zařízení, identifikuje produkt výrobce.</i>
MCU .....	Micro Controller Unit <i>Integrovaný programovatelný obvod – jednočipové počítače.</i>
EEPROM .....	Electrically Erasable Programmable Read-Only Memory <i>Paměť používaná k uložení některých informací v MCU.</i>
CPLD .....	Complex Programmable Logic Devices <i>Programovatelný logický obvod.</i>
FPGA .....	Field Programmable Gate Array <i>Programovatelné hradlové pole.</i>
ARM .....	Acorn RISC Machine (Advanced RISC Machine) <i>Architektura procesorů.</i>
DIL/DIP .....	Dual In-Line Package <i>THT pouzdro integrovaných obvodů.</i>
RS232 .....	Serial standard <i>Sériové rozhraní pro přenos dat.</i>
UART .....	Universal asynchronous receiver/transmitter <i>Univerzální asynchronní přijímač/vysílač.</i>
ICSP .....	In-Circuit serial programming <i>Sériové programování osazených procesorů.</i>
ISP .....	In system programming <i>Sériové programování osazených procesorů.</i>
SPI .....	Serial peripheral interface <i>Sériové rozhraní pro přenos dat.</i>
I <sup>2</sup> C .....	Inter-Integrated Circuit <i>Multi-masterová sériová sběrnice.</i>
MISO .....	Master In Slave Out <i>Označení vodiče pro přenos dat do slave zařízení (SPI).</i>
MOSI .....	Master Out Slave In <i>Označení vodiče pro přenos do master zařízení (SPI).</i>
SCK .....	Serial clock <i>Synchronizační hodinové pulzy.</i>
A/D .....	Analog to Digital <i>Převodník analogové veličiny do číslicové hodnoty.</i>
PWM .....	Pulse Width Modulation <i>Pulzně šířková modulace.</i>
LPT .....	Line Printer Terminal <i>Paralelní rozhraní.</i>



## Úvod

Bakalářská práce je zaměřena na návrh univerzálního USB programátoru pro 8-bitové mikroprocesory Atmel AVR a Microchip PIC. Popsány jsou různé způsoby řešení při návrhu, jako hojně využívané převodníky USB - RS232, převodníky USB - I/O a využití mikroprocesoru s integrovaným USB modulem.

Využití mikroprocesoru s integrovaným USB modulem se jeví jako zajímavé řešení a zároveň ukazuje některé vlastnosti především USB modulu. Mezi takto vytvořeným zařízením a počítačem je vytvořen virtuální sériový port, který však přenáší data mnohem rychleji než standardní sériový port. Díky tomu je možný jednoduchý vývoj ovládacího software pro počítač, který zvládnou i méně pokročilí programátoři.

Cílem je vytvoření funkčního a jednoduchého univerzálního USB programátoru z dostupných součástek, se schopností naprogramovat více různých mikroprocesorů. V práci je popsán postup řešení od vytvoření fyzického zařízení schopného komunikovat s PC pomocí USB, přes popis sériového programování různých mikroprocesorů až do fáze, kdy je celý celek funkčním programátorem.

# 1 Univerzální programátory mikroprocesorů

S rozvojem integrované techniky nás různí výrobci zahrnují velkým množstvím polovodičových programovatelných obvodů, jako pamětí nebo mikroprocesorů. Mezi oblíbené, patří například mikroprocesory AVR od společnosti Atmel nebo mikroprocesory PIC společnosti Microchip. Obě společnosti a mnohé další nabízí celou řadu mikroprocesorů s různými periferiemi, velikostí pamětí nebo spotřebou. Kupovat nebo vyrábět jednoúčelové programátory na konkrétní mikroprocesor by bylo nevýhodné. Proto většina mikroprocesorů od dané společnosti má stejný způsob programování a lze tedy použít jeden programátor na velký počet různých mikroprocesorů. Na trh takovéto programátory dodává několik společností, které se snaží svoje produkty prezentovat jako rychlé, spolehlivé a levné programátory s širokou podporou programovaných obvodů.

## 1.1 Programátor PRESTO

Programátor PRESTO je univerzální USB programátor od společnosti Asix s.r.o. Disponuje podporou široké škály obvodů od mikroprocesorů společností Microchip, Atmel, Texas Instrument, Cypress, některé mikroprocesory s jádrem ARM, přes programovatelná hradlová pole CPLD a FPGA, až po paměti EEPROM a FLASH a mnoho dalšího. Celkem podporuje přes 1200 obvodů. Programátor se připojuje pomocí portu USB, lze jej tedy použít i na novějších počítačích bez dříve často používaného sériového portu. Při dodržení některých podmínek lze programovat obvody již osazené na desce plošného spoje nebo v patici. Obvody však nelze programovat přímo v programátoru, k tomu je zapotřebí dokoupit doplněk s paticí, který umožňuje přivést libovolný signál na libovolný vývod programovaného obvodu. Vnitřní obvody programátoru jsou pečlivě navrženy, s ohledem na to, aby nedošlo k poškození programátoru, nebo programovaného mikroprocesoru při nevhodném použití.

## 1.2 Programátor PICKIT

Programátor PICKit je vyvíjen pro obvody společnosti Microchip a nyní již existuje ve své třetí verzi. Podporuje programování pamětí EEPROM a velkého množství mikroprocesorů od řady PIC10F, po PIC32 až dsPIC. Bohužel programovat lze jen výhradně obvody společnosti Microchip. Naproti tomu velká výhoda je v přímé podpoře ve vývojevém

prostředí MPLAB IDE. Uživatel po napsání kódu programu může pouze jedním stiskem přeložit svůj projekt a zároveň je ihned vytvořený program naprogramován. Programovaný obvod se může programovat již osazený za předpokladu, že jsou vyvedeny vývody pro sériové programování. Další velkou výhodou je zobrazení obsahu pamětí a registrů a to přímo ve fyzickém mikroprocesoru. Díky tomu můžeme efektivně ladit program a hned jej sledovat v reálném čase, nebo lze krokovat například za účelem nalezení chyby.

### **1.3 Programátor BeeProg**

BeeProg je robustnější programátor společnosti Elnec, který podporuje přes 68000 obvodů od 291 výrobců. K počítači jej lze připojit pomocí sběrnice USB nebo některé potěší i přítomnost paralelního portu LPT. Programátor se též chlubí vysokou programovací rychlostí. Obsahuje patičky pro přímé programování obvodů v pouzdře DIL až do velikosti 48 pinů. Obvody ve větších pouzdrech lze programovat jen pomocí přídatného modulu. Samozřejmostí je možnost programovat obvody již osazené. Jelikož se jedná o profesionální programátor s podporou velkého množství obvodů s robustní konstrukcí, tak nemusí všem vyhovovat vyšší pořizovací cena.

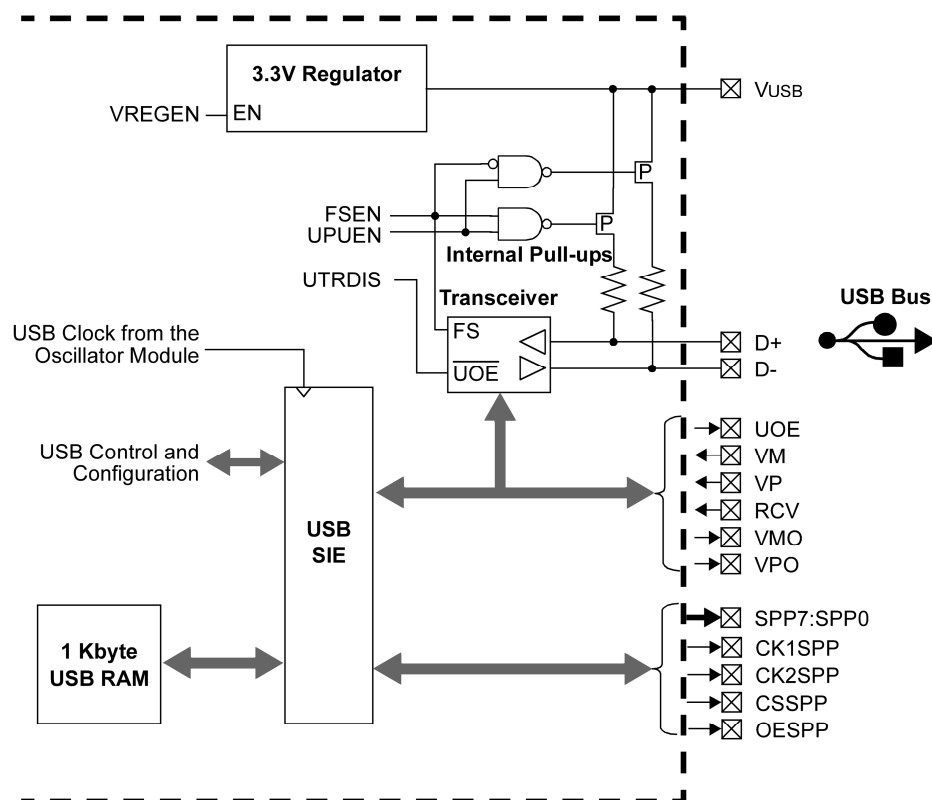
## 2 Popis použitých obvodů

Ve zvolené konstrukci jsou použity dva integrované obvody nezbytné pro správný chod celého programátoru.

### 2.1 Mikroprocesor PIC 18F4550

Ve zvoleném řešení je použit mikroprocesor PIC 18F4550 společnosti Microchip, jako hlavní obvod, který obsluhuje veškeré komunikace jak po USB, tak se samotnými programovanými mikroprocesory.

Jedná se o osmibitový mikroprocesor s harvardskou architekturou. Obsahuje programovou paměť typu FLASH o velikosti 32kB, a paměť RAM o velikosti 2kB. Maximální pracovní frekvence je 48MHz a některé periferní moduly mohou být taktovány odlišnou frekvencí než samotné jádro. Mezi perifériemi nechybí osmibitový a šestnáctibitový časovač, desetibitový A/D (analog to digital) převodník, univerzální sériový synchronní-asynchronní modul s podporou RS232, SPI a I<sup>2</sup>C, pět paralelních vstupně-výstupních bran, modul PWM a především modul USB.



Obr. 2.1 Modul USB mikroprocesoru PIC 18F4550 [4]

USB modul podporuje rychlosti Low (1,5Mb/s) a Full (12Mb/s), je tedy kompatibilní s USB 1.1 i 2.0. Dále podporuje čtyři režimy přenosu: Control (využíván zejména po připojení zařízení), Isochronus (pro přenos dat v reálném čase, bez opakování v případě chyby), Interrupt (pro občasné přenosy dat s opakováním v případě chyby) a Bulk (pro přenos relativně velkého množství dat). Modul obsahuje již vlastní zdroj referenčního napětí 3,3V. Přivedením tohoto napětí, přes pull-up rezistor na jeden z datových vodičů, je určena rychlost přenosu. Celý modul je vyroben tak, že je možné mikroprocesor připojit přímo ke sběrnici USB.

Tento mikroprocesor je poměrně rozšířený a oblíbený a to především díky příznivé ceně a obsahujícímu USB modulu. Výrobce navíc zdarma nabízí obsáhlou knihovnu odladěných ukázkových programů pro různé využití mikroprocesoru, které je možné dále modifikovat pro vlastní použití.

## 2.2 Obvod MC34063A

Pro úspěšné naprogramování mikroprocesoru PIC je potřeba přivést na vývod označený  $V_{pp}$  programovací napětí 12V až 14V. USB může napájet koncové zařízení stejnosměrným napětím 5V (specifikace stanoví minimální napětí až na 4,4V). Toto napětí je pro programování tedy nedostatečné. K tomuto účelu poslouží obvod MC34063A, určený především jako spínaný měnič z jednoho stejnosměrného napětí na jiné. Lze ho použít jak pro snížení, tak i pro zvýšení napětí. Ke své činnosti potřebuje jen několik málo externích součástek pro určení spínané frekvence, výstupního napětí a maximálního odebíraného proudu. Jistou nevýhodou může být to, že obvod potřebuje ke své činnosti indukčnost. Velikost indukčnosti závisí na požadavku maximálního proudu.

## 3 Vlastní konstrukce

Při návrhu konstrukce je třeba zvážit jakým způsobem budou data programu dopravena z počítače do programovaného mikroprocesoru, přičemž podmínkou je využití sběrnice USB. V práci jsou dále popsány některé z částí vyzkoušené pokusy, které by určitě po dokončení mohli posloužit jako plně funkční programátor. Avšak jako nejzajímavější se jeví použití mikroprocesoru s integrovaným USB modulem pro přímé připojení k počítači.

### 3.1 Možné způsoby řešení konstrukce

Popsané způsoby řešení jsou funkční a použitelné pro účely univerzálního USB programátoru mikroprocesorů. Vždy se liší způsobem komunikace s počítačem a zpracováním dat.

#### 3.1.1 Převodník USB-RS232

Na trhu je k dostání velké množství obvodů pro převod mezi sběrnici USB a RS232. Mezi nejznámější patří obvody FT232x společnosti FTDI nebo obvody CP2102 společnosti Silicon Labs. Veškeré signály jsou v úrovních 0V až 5V nebo 0V až 3,3V, což může některým zařízením nevyhovovat, zejména pokud očekávají signály v úrovních -12V až 12V.

Pro použití v programátoru by bylo nutné tyto obvody kombinovat s dalším obvodem, například s mikroprocesorem, neboť standard RS232 není podporován pro programování PIC ani AVR. Mikroprocesor by tedy sloužil buď jako prostý převodník, nebo by mohl obsahovat komplikovanější program pro efektivnější přenos dat z počítače do programovaného mikropočítače.

Výhodou je jednoduché obvodové i programové řešení. V počítači pak může programátor psát program pro jednoduchou komunikaci pomocí sériového portu a stejně tak i program (firmware) pro mikroprocesor. Nevýhodou může být nižší komunikační rychlost, což se projeví na celkové době potřebné k naprogramování a další nevýhodou je potřeba více obvodů v programátoru.

### 3.1.2 Použití obvodu MCP2200

Obvod MCP2200 je integrovaný USB – I/O převodník, který je zároveň doplněn o UART. Ve skutečnosti je obvod naprogramovaný mikroprocesorem PIC 18F14K50 s jiným značením. Tento obvod ke své plné činnosti potřebuje už jen 12MHz krystal, dva 15nF kondenzátory a dva 10uF kondenzátory.

Pro účely programátoru nás ale zajímá vstupně výstupní brána, která se dá konfigurovat a i ovládat přímo z počítače. Programátorům ovládacího software přijde vhod i sada knihoven přímo od výrobce, obsahující několik jednoduchých funkcí pro snadné programování. Vytvořený ovládací software se nemusí uživatele ptát na číslo portu nebo komplikovaně vyhledávat zařízení jako v případě použití převodníku, protože si vyhledá příslušné zařízení podle čísel PID a VID.

Jelikož lze brány jen konfigurovat (vstupní nebo výstupní), nastavovat v případě výstupní brány logické hodnoty a číst logickou hodnotu v případě vstupní brány, je veškerá komunikace s naprogramovaným mikroprocesorem řízena z počítače.

Výhodou tedy je opravdu jednoduché obvodové řešení s jedním integrovaným obvodem a několika součástek, avšak za cenu složitějšího ovládacího software v počítači.

### 3.1.3 Použití mikroprocesoru s USB modulem

Toto řešení je postavené na mikroprocesoru PIC 18F4550 (případně PIC 18F2550), kde využíváme již integrovaného USB modulu. S použitím vhodného programu v mikroprocesoru a ovladačů pro operační systém počítače, vytvoříme virtuální sériový port. K tomuto sériovému portu lze přistupovat stejně jako ke standardnímu sériovému portu, což usnadní programování ovládacího software. Mikroprocesor pak jako druhé zařízení přijímá data, která zpracuje a vykoná potřebné kroky.

Výhodou je opět jednoduché obvodové řešení a ulehčené programování ovládacího software. Nevýhodou je naopak složitý program v mikroprocesoru, který musí obsluhovat i samotnou komunikaci po sběrnici USB.

## 3.2 Zvolený způsob řešení konstrukce

Jako neoptimálnější řešení byl zvolen virtuální sériový port, protože nabízí jednoduchou komunikaci programátoru s počítačem a bude tak využita možnost integrovaného USB modulu v mikropočítači.

Celou práci je rozdělena do několika kroků. Bylo potřeba prozkoumat možnosti a úskalí zvolené metody komunikace s počítačem, zjistit způsoby programování PIC i AVR a tomu přizpůsobit výsledný programátor s ohledem na potřebné vývody a jejich napěťové úrovně. Následovalo napsání programu pro mikroprocesor i pro počítač a ozkoušení celku na několika mikroprocesorech. Hlavní prioritou bylo sestrojít programátor levný a sestavený z dostupných součástek.

### 3.2.1 Programování PIC

Programování mikroprocesoru PIC se provádí pomocí sériové komunikace s programovaným mikroprocesorem. Je možné zapisovat a číst jak z programové FLASH paměti, tak i z datové paměti EEPROM a číst nebo upravovat konfigurační slova. V případě řady procesorů PIC 16F jsou slova velká 14 bitů.

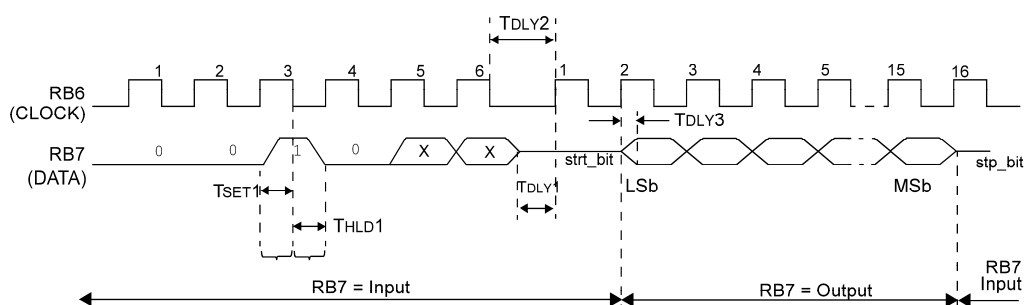
Pro komunikaci s programovaným mikroprocesorem je třeba tří vodičů. Pro uvedení mikroprocesoru do stavu, kdy jej lze programovat je třeba přivést programovací napětí (12V až 14V) na vývod označený  $V_{pp}$ . Signál *CLOCK* slouží k synchronizaci komunikace a programovaný mikroprocesor je vždy slave, tedy *CLOCK* negeneruje. Vodič *DATA* je obousměrný a jsou jím data přijímána i odesílána. Je tedy nutná přesná synchronizace s programátorem, aby nenastala situace, kdy programátor i programovaný mikroprocesor vysílají současně. Pokud by například programátor nastavil logickou 0 a programovaný mikroprocesor zároveň log1, tekli by mezi vývody velké proudy, které by mohli poškodit jak programátor, tak i programovaný mikroprocesor.

Napájecí napětí  $V_{dd}$  musí být v rozsahu 4,5V až 5,5V proti  $V_{ss}$ . Výrobce povoluje i nižší, ale pro mazání obou pamětí je potřebné napětí minimálně 4,5V. Napěťové úrovně signálů *CLOCK* a *DATA* se mohou lišit od napájecího maximálně o 0,2V.

Veškerá komunikace z pohledu programátoru začíná odesláním příkazu. V závislosti na příkazu mohou následovat data. Přenos dat začíná start bitem, za ním následuje 14bitů a je ukončen jedním stop bitem. Před započítím přenosu dat je nutné správně nastavit datový vývod programátoru (u programovaného mikroprocesoru je toto nastavení prováděno



automaticky). V případě vysílání dat je nastaven jako výstupní a v případě příjmu dat je nastaven jako vstupní, tedy má vysokou impedanci. Po odeslání příkazu je nutné počkat stanovenou dobu, po kterou programovaný mikroprocesor vykonává odpovídající úkony. Délka této doby se opět liší podle odeslaného příkazu.



Obr. 3.1 Diagram komunikace při čtení dat z programové paměti [1]

Programovaný mikroprocesor je taktován signálem *CLOCK*, jehož frekvence nesmí být větší než stanovená výrobcem. Například u mikroprocesoru PIC 16F628A je možná maximální frekvence 5MHz.

Seznam příkazů, jejich tvar i časování, vývojové diagramy a podrobné informace ke konkrétnímu mikroprocesoru jsou k dispozici na stránkách výrobce.

### 3.2.2 Programování AVR

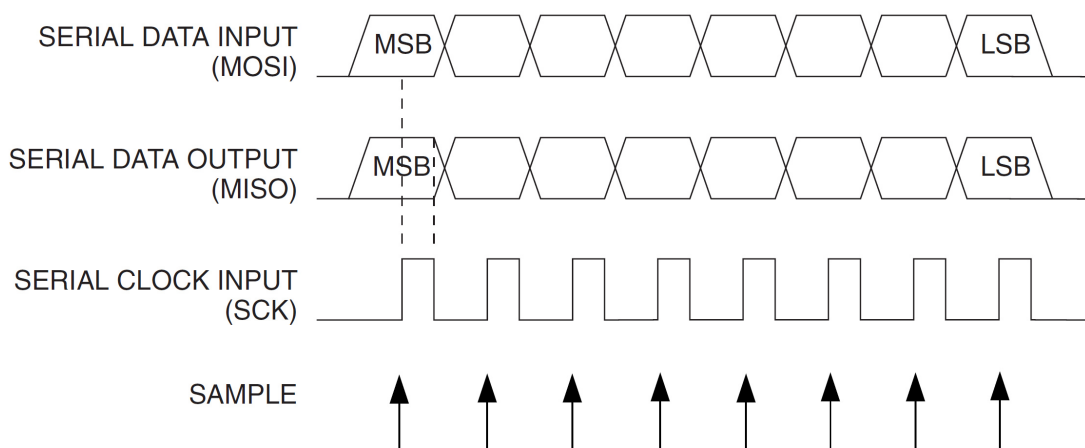
Mikroprocesory Atmel AVR umožňují programovat sériově i paralelně. Paralelní programování vyžaduje sedmnáct signálů, a proto se hodí pro použití ještě před samotným osazením mikroprocesoru na desku plošného spoje nebo do patice. Paralelní programování je vždy použitelné, protože sériové programování lze zakázat (základní nastavení však sériové programování umožňuje). Navržený programátor podporuje pouze sériové programování z důvodů nižšího počtu signálových vodičů.

Mikroprocesor pro sériové programování (označované jako ISP – In System Programming) využívá ke komunikaci s programátorem periférii SPI. Komunikace tedy probíhá po třech vodičích. Prvním signál slouží k synchronizaci, označuje se *SCK* a pulzy jsou generovány programově. Další dva signály pro přenos dat se značí *MOSI* (přenos z programátoru do mikroprocesoru) a *MISO* (přenos z mikroprocesoru do programátoru). Aby bylo s mikroprocesorem možné komunikovat, je nutné ještě přivést log 0 na vývod označený

RESET. Napájecí napětí může být v rozsahu 2,7V až 5,5V a logické úrovně na vodičích *SCK*, *MOSI* a *MISO* se mohou od napájecího lišit maximálně o 0,2V.

Komunikace vždy probíhá po blocích, kde jeden blok je složen ze čtyř bytů. Vždy jsou odeslány a zároveň přijaty 4 byty. V první bytu je odeslán příkaz případně vyšší bity příkazu. Ve druhém bytu odesíláme zbytek, tedy nižší bity příkazu nebo pokud se příkaz vešel do prvního bytu, vyšší bity adresy. Třetí byte může být prázdný nebo obsahuje nižší bity adresy. Poslední čtvrtý byte slouží pouze pro přenos dat. Pokud k provedení příkazu nejsou data potřeba, je čtvrtý byte také prázdný.

Aby bylo možno s programovaným mikroprocesorem komunikovat, je třeba nejprve odeslat příkaz na povolení sériového programování. Pokud mikroprocesor odpoví korektně, můžeme začít číst, mazat nebo zapisovat do programové FLASH paměti, datové EEPROM paměti nebo měnit konfigurační byty (fuse bity, fuse high bity a lock bity).



Obr. 3.2 Průběh odeslání a příjmu jednoho byte [3]

Programovaný mikroprocesor je taktovaný hodinovým signálem, který je generován vlastním integrovaným nebo vnějším oscilátorem. Od frekvence tohoto oscilátoru se pak odvíjí i maximální možná frekvence *SCK*, které slouží pouze k synchronizaci přenosu dat. Signál *SCK* má vždy menší frekvenci než na jakou je taktován programovaný procesor. Obecně lze říci, že pro většinu AVR mikroprocesorů musí být *SCK* 4x menší než frekvence oscilátoru. Například pro ATmega8 výrobce udává, že pokud je mikroprocesor taktován pod 12MHz, musí být frekvence *SCK* minimálně 2x menší a pokud je taktován nad 12MHz, *SCK* musí být minimálně 3x menší. Přesný poměr frekvencí lze opět získat pro konkrétní mikroprocesor z dokumentace od výrobce.

### 3.2.3 Soubor HEX

Soubor HEX byl navržen a je používán k reprezentaci obsahu paměti, kde data jsou kódována do ASCII. Jedná se tedy o textový soubor a to z důvodu, aby bylo možné jednoduše soubor otevřít v textovém editoru. Veškerá data jsou reprezentována znaky z šestnáctkové soustavy po bytech, tedy jeden byte je reprezentován dvěma ASCII znaky. Data jsou v souboru zapsána do řádků označovaných jako záznam, kde každý záznam představuje část dat a je definováno, jak takový záznam má vypadat.

Každý záznam začíná znakem dvojtečky, který informuje o začátku záznamu. Za ním následují dva znaky mající význam počtu bytů v záznamu. Tato informace je také v šestnáctkové soustavě, takže například hodnota 10 znamená, že v záznamu je šestnáct bytů dat. Následují čtyři znaky adresy, adresa je zapsána do dvou bytů. Pro lepší čitelnost je zde adresa zapsána od významnějšího bytu obsahující MSB. Na to je třeba dát pozor, protože ve zbytku záznamu jsou byty řazeny za sebou jako v samotné paměti. Pokud slovo v paměti mikroprocesoru bude obsahovat například hodnotu 0x1A35, bude v souboru HEX zapsáno ve znacích jejichž pořadí bude 351A. Další dva znaky specifikují typ dat a nabývají nejčastěji hodnoty od 00 do 05. Nejvíce se setkáme s hodnotou 00 (v záznamu jsou samotná data) a 01 (záznam neobsahuje data, soubor zde končí). Dále již následují samotná data, kde počet byte v záznamu je informován druhým a třetím znakem. Počet znaků v souboru pro zaznamenání dat, je dvojnásobkem počtu datových bytů. Za data nalezneme ještě dva znaky jako kontrolní součet pro ověření správnosti dat. Kontrolní součet je součet všech bytů v záznamu, mimo první dvojtečky a posledního bytu s kontrolním součtem. Z tohoto součtu pomocí dvojkového doplňku získáme hodnotu kontrolního součtu. Důležité je zachovat velikost bytu, při součtu nad hodnotu 0xFF, dojde sice k přenosu do vyššího řádu, ale tuto informaci zahazujeme a dále nevyužijeme. Veškerá čísla v následujícím příkladu jsou hexadecimální.

$$04 + 00 + 06 + 00 + F8 + 94 + FF + CF = BE \text{ (binárně } 1011 \ 1110)$$

Dvojkový doplněk z BE je 42 (binárně 0100 0010).

Na úplném konci záznamu se nachází jeden nebo dva znaky pro ukončení řádku a jsou jím LF (0x0A - Line Feed) nebo CR (0x0D - Carriage Return). Na konci se může vyskytnout pouze jeden z nich, nebo zároveň oba. Záleží na software, který HEX soubor generoval. Tyto znaky jsou neviditelné a v běžném textovém editoru je nevidíme.

```
:1000000012C019C018C017C016C015C014C013C044
:1000100012C011C010C00FC00EC00DC00CC00BC06C
:100020000AC009C008C011241FBECFE5D4E0DEBF5E
:10003000CDBF09D015C0E4CF80E090E001C0000042
:100040000197E9F708951F93A59A10E1AD9AF4DF9F
:10005000085B38F7585BBF0DF15BBEEDF15BAF6CF24
:04006000F894FFCF42
:00000001FF
```

Obr. 3.3 Náhled obsahu souboru HEX

### 3.2.4 Navržený programátor

Jako základ univerzálního USB programátoru je použit mikroprocesor PIC 18F4550, který je snadno dostupný, cenově příznivý a obsahuje modul USB. Dalších okolních součástek není dále potřeba velké množství, což zjednodušuje celou konstrukci. Hodinové pulzy pro mikroprocesor jsou generovány s pomocí 20MHz krystalu a následně fázovým závěsem násobeny na frekvenci 48MHz, kterou vyžaduje modul USB pro správnou činnost. Programovaný mikroprocesor je pak připojen přímo. Pouze na vodičích označených *SCK*, *MOSI* a *MISO/Data* jsou ochranné rezistory 100Ω. Celý programátor včetně programovaného mikroprocesoru je napájen napětím 5V z USB. Programování mikroprocesorů PIC vyžaduje programovací napětí okolo 13V. Jako zdroj 13V je použit obvod MC34063A ve funkci spínaného step-up měniče stejnosměrného napětí s maximálním proudem 10mA. Toto napětí je přivedeno k programovanému mikroprocesoru přes tranzistor BC557 ve funkci spínače. Schéma jednoduchého programátoru je k dispozici v příloze A.

### 3.2.5 Program pro mikroprocesor programátoru

Výrobce mikroprocesoru PIC 18F4550 nabízí řadu funkčních ukázkových programů, které je možné dále použít jako základ vlastního vývoje. Pro účely programátoru byl vybrán ukázkový program pro převodník USB na RS232. Po připojení do počítače, takto naprogramovaného mikroprocesoru, se v systému objeví nový sériový port, ke kterému lze přistupovat jako ke standardnímu sériovému portu. Jedná se o virtuální sériový port, který fyzicky neexistuje. Data se ve skutečnosti odesílají po USB do mikroprocesoru, kde jsou následně zpracována. Jako výstup převodníku slouží integrovaný modul univerzálního sériového vysílače a přijímače. Omezením takového převodníku je nastavení komunikační rychlosti. Zatímco virtuální sériový port takovéto nastavení ignoruje a využívá maximální

možnou rychlost sběrnice USB, integrovaný modul UART má na pevně definovanou komunikační rychlost, kterou lze změnit pouze změnou programu v mikroprocesoru.

V ukázkovém programu je již připravena obsluha USB a je již předem nastaveno VID a PID, které lze změnit. To ovšem není doporučeno a společnost Microchip umožňuje využití jejího jednoho VID a PID k nekomerčním účelům (tyto ID jsou v programu již v základu). V opačném případě je nutné si vlastní VID zakoupit.

Příjem dat je řešen nezávisle na hlavní smyčce programu a podprogram pro zajištění příjmu dat je volán při přerušení s USB modulu. Přijaté byty jsou ukládány do proměnné, sloužící jako zásobník o velikosti 64 bytů (velikost lze měnit). Odesílání dat je řešeno obdobně. Do proměnné, sloužící jako výstupní zásobník o velikosti 64 bytů (také lze změnit), vložíme data, která je třeba odeslat. Poté zavoláme funkci `CDCTXService()`, která zajistí spolehlivé odeslání po virtuálním sériovém portu do počítače.

V hlavní smyčce programu je pouze kód pro předání dat mezi UART a zásobníky. Tento kód není potřeba a je nahrazen vlastním kódem pro potřeby programátoru. Přijátá data v zásobníku jsou porovnávána a jsou využita teprve ve chvíli, kdy jsou přijata veškerá potřebná. Podle přijatých dat je následně rozhodnuto, jaké úkony budou provedeny a jsou případně volány příslušné podprogramy.

První přijatý byte je příkaz, na jehož základě se programátor připraví k dané operaci a je-li třeba, čeká na data. Po provedení potřebných úkonů, odesílá programátor zpět do počítače byte s výsledkem (zda příkaz byl proveden úspěšně či nikoliv). Odpověď zasláná do počítače, může obsahovat i více než jeden byte. Taková situace může nastat pokud požadujeme, aby programátor přečetl z programovaného mikroprocesoru například jedno slovo z obsahu FLASH paměti. V takovém případě jsou do počítače odeslány dva byty (jedno slovo u PIC 14 bitů), případně čtyři, pokud programovaným mikroprocesorem je AVR. Jsou příkazy i takové, které pouze nastavují globální proměnné jako informaci jaký je programovaný mikroprocesor nebo jak dlouhá je perioda hodinového signálu *SCK*.

Hodinové pulzy *SCK*, sloužící k synchronizaci komunikace mezi programátorem a programovaným mikroprocesorem, jsou generovány programově. Po každé změně logického stavu je volána funkce `Delay_auto()`, která pozdrží běh hlavní smyčky programu na předem stanovenou dobu (tato doba je nastavena příkazem z počítače).

Seznam všech příkazů a jejich význam je uveden v příloze. Veškeré zdrojové kódy jsou napsány v jazyku C. Takto navržený a naprogramovaný programátor lze použít k programování mikroprocesorů AVR ATtiny13A, ATtiny24, ATtiny44, ATtiny84,

ATmega48, ATmega88, ATmega8, ATmega16, ATmega32, ATmega64 a většiny mikroprocesorů PIC řady 16F.

### 3.2.6 Ovládací software

Ovládací program v počítači je napsán v jazyce C a slouží spíše jako demonstrace funkčnosti programátoru. Podporuje programování procesorů PIC 16F628A, PIC 16F84A a AVR ATmega8. Uživatel může libovolně volit soubor HEX (pokud je zvolen neexistující soubor, je automaticky vytvořen a připraven pro zápis), číslo sériového portu a rychlost zápisu a čtení. Uživatel je také pomocí ovládacího programu informován o právě probíhající akci a jejím stavu.



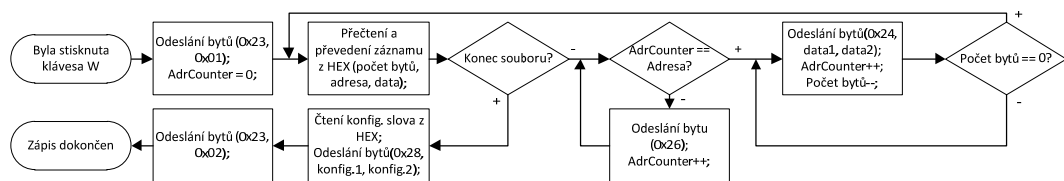
Obr. 3.4 Okno ovládacího software

Jakmile uživatel zvolí programovaný mikroprocesor a dobu periody signálu *SCK*, jsou tyto informace odeslány programátoru a program se dostává do hlavní smyčky, která pouze obsahuje kód pro snímání stisků kláves. Po stisku klávesy je buď proveden jednoduchý sled úkonů, například odeslání instrukce na smazání, nebo zavolán podprogram, jemuž je předána informace, která klávesa byla stisknuta. Pokud uživatel zvolí zápis, je otevřen soubor HEX. Ze souboru je vždy přečten celý záznam a zároveň ihned převeden ze znaků do čísel. Následně jsou načtená data odeslána programátoru. V případě programování mikroprocesoru PIC není odesílána adresa dat. Ta je inkrementována automaticky, ale není možné její hodnotu zjistit. Proto je nutné inkrementovat vlastní proměnnou po každém odeslaném příkazu pro zápis do paměti FLASH, nebo příkazu pro inkrementaci pokud na aktuální adresu

nemá být zapsáno žádné slovo. Při opačném postupu, čtení a následně ukládání do souboru HEX, probíhá čtení na všech adresách, ale informace je do souboru uložena jen tehdy, nachází-li se na aktuální adrese nějaká data. Pokud je slovo prázdné, je vrácena hodnota 0x3FFF v případě PIC nebo 0xFFFF v případě AVR.

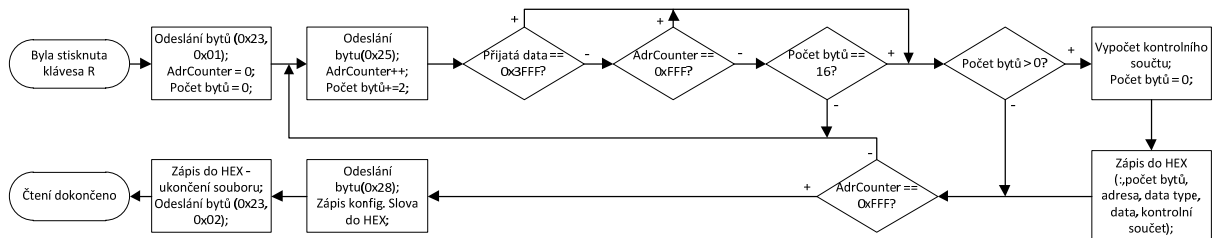
### 3.2.7 Diagramy průběhů ovládacího software

Následující diagramy znázorňují, jak probíhají jednotlivé činnosti Při programování mikroprocesoru PIC 16F628A a AVR ATmega8. Adresa ukazující na místo v paměti mikroprocesoru čítá slova, zatímco adresa uložená v souboru HEX čítá každý byte. Jedno slovo je v HEX souboru vyjádřeno dvě byty, proto adresa z HEX je dělena dvěma. Předpokladem je, že uživatel již zadal všechny potřebné informace jako název souboru, číslo portu a dobu periody signálu *SCK*.



Obr. 3.5 Zápis do paměti mikroprocesoru PIC

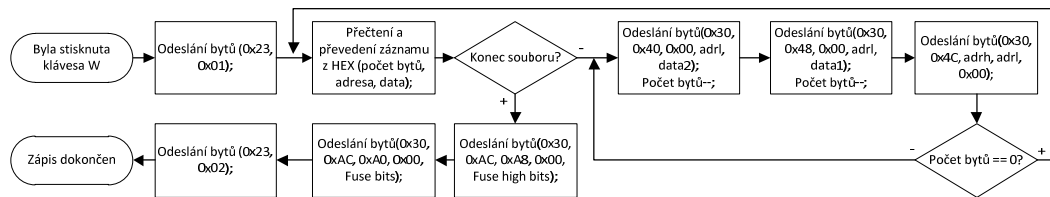
Příkaz 0x23 slouží k nastavení programovacího napětí. Hodnota 0x01 znamená, že na vývod mikroprocesoru označený jako *Vpp* bude přivedeno napětí 13V. Hodnota 0x02 naopak vývod připojí k zemi. Příkazem 0x24 zapíšeme 14bit slovo, jehož hodnota je přenesena v následujících dvou bytech s MSB napřed.



Obr. 3.6 Čtení z paměti mikroprocesoru PIC

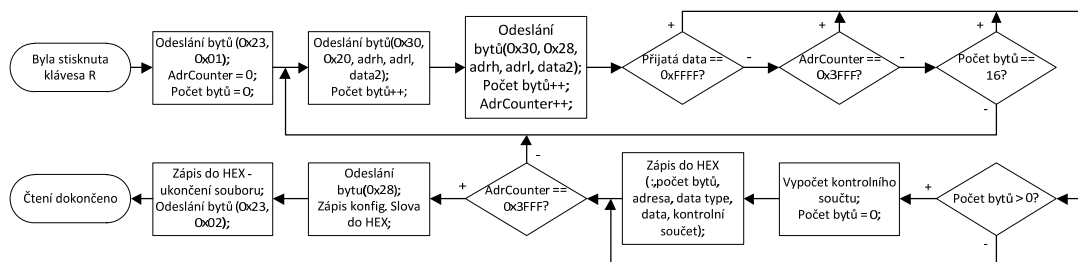
Příkazem 0x25 si vyžádáme slovo z paměti mikroprocesoru. Pokud se rovná 0x3FFF, v paměti na aktuální adrese nejsou žádná data a proto toto slovo do souboru HEX neukládáme. Téměř na konec souboru zapíšeme konfigurační slovo, které si vyžádáme

příkazem 0x28. Testováním hodnoty 0x2FFF v adrese zjišťujeme, zda nejsme na konci paměti. Následné čtení by se provádělo opět od adresy 0x0000.



Obr. 3.7 Zápis do paměti mikroprocesoru AVR.

Pro komunikaci s mikroprocesorem AVR slouží jediný příkaz 0x30. Za ním musí následovat čtyři byty a vráceny jsou taktéž čtyři byte. V případě zápisu se vrátí pouze číslo příkazu ve druhém bytu. Byte označený *adrh*, obsahuje vyšší část adresy a *adrl* naopak nižší část adresy. Zde není potřeba adresu speciálně inkrementovat, protože je součástí příkazu pro zápis.



Obr. 3.8 Čtení z paměti mikroprocesoru AVR

Čtení je velmi obdobné, opět se využívá pouze příkazu 0x30, ale tentokrát následují jiné byty. Změna je také ve čtvrtém bytu, kde data nejsou odesílána, ale přijímána.

Komunikace se mikroprocesorem AVR se může zdát trochu komplikovaná. Vždy je odesíláno pět bytů. První je příkaz pro programátor, který znamená, že hodláme předat mikroprocesoru AVR nějaká data. Další čtyři byty jsou následně odeslány do programovaného mikroprocesoru AVR.

### 3.2.8 Možný další vývoj

Jelikož vytvořený programátor je první zhotovená funkční verze, je možno přidat dodatečné funkce nebo vylepšit stávající.



Ovládací software se dá ovládat pouze klávesnicí. Proto dobrým zlepšením může být vytvoření ovládacího software s přehledným grafickým prostředím nebo přidání automatického hledání zařízení, aby uživatel nemusel vždy port zadávat ručně. Tyto úpravy by výrazně zefektivnili práci s takovýmto ovládacím software. Další zlepšení by mohlo být ve vytvoření snadno upravitelného seznamu podporovaných obvodů, i s jejich údaji o velikosti paměti, adresy konfiguračních bytů atd., pro snadné přidání nových informací. Z tohoto seznamu by ovládací software čerpal potřebné informace o mikroprocesorech.

Rychlejšího zápisu nebo čtení by mohlo být dosaženo lepším odladěním nebo předěláním stávajících částí programu pro programátor. Navíc paměť řídicího mikroprocesoru je z velké části volná, a je tedy možné přidávat další funkce. Jako například jednoduchý logický analyzátor, se schopností sledovat až 4 signály a možností grafického výstupu na počítači.

## Závěr

Cílem práce bylo navrhnout a zkonstruovat funkční univerzální USB programátor mikroprocesorů. Nejprve bylo potřeba zprovoznit a porozumět komunikaci mezi počítačem a programátorem, bez čehož by nebylo možné postavit funkční programátor. Následně vytvořit ovládací software i program pro mikroprocesor programátoru, porozumět způsobu sériového programování PIC i AVR a navrhnout obvod se všemi náležitostmi včetně zdroje programovacího napětí.

Veškeré části byly nakonec úspěšně dokončeny a vznikl funkční programátor. Jeho funkce byly ověřeny na několika dostupných mikroprocesorech. Doba potřebná k zápisu nebo čtení se pohybuje v jednotkách až desítkách sekund, v závislosti na zvolené rychlosti a velikosti souboru HEX.

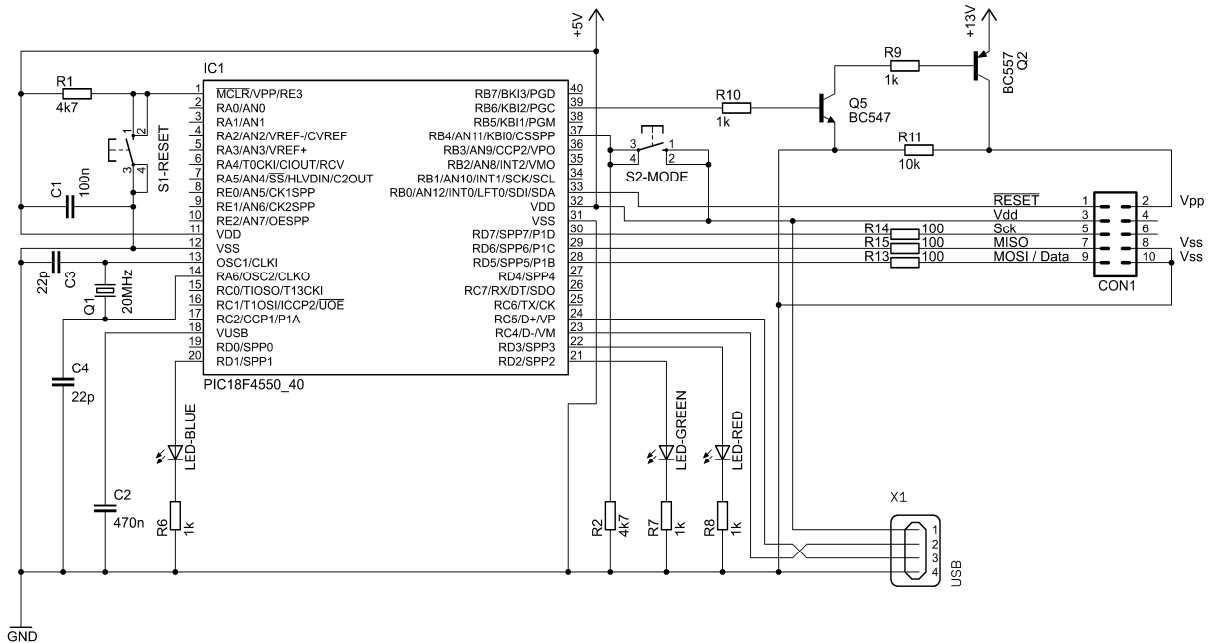
Programátor je sestaven ze dvou integrovaných obvodů a několika málo součástek. Díky tomu je jeho pořizovací cena velmi příznivá. Proto uspokojí především každého začínajícího vývojáře a díky připojení přes sběrnici USB i každého kdo používá novější počítač bez sériových nebo paralelních portů.

## Seznam literatury a informačních zdrojů

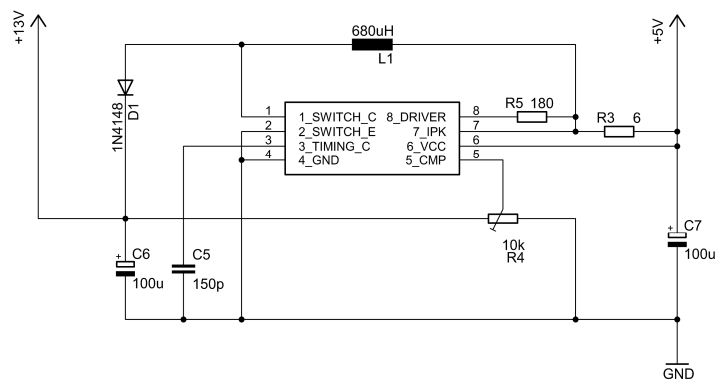
- [1] MICROCHIP. *Microchip Programming specification* [online]. [cit. 2012-06-06]. Dostupné z: <http://www.microchip.com/TechDoc.aspx?type=Programming>
- [2] PINKER, Jiří. *Mikroprocesory a mikropočítače*. 1. vyd. Praha: BEN - technická literatura, 2004, 159 s. ISBN 80-730-0110-1.
- [3] ATMEL. *Datasheet ATmega8(L) Complete* [PDF]. 2/2011 [cit. 2012-06-06]. Dostupné z: <http://www.atmel.com/Images/doc2486.pdf>
- [4] MICROCHIP. *PIC18F2455/2550/4455/4550 Data sheet* [PDF]. 27.10.2009 [cit. 2012-06-06]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>
- [5] INTEL CORPORATION. *Hexadecimal Object File Format Specification* [PDF]. 6.1.1988 [cit. 2012-06-06]. Dostupné z: <http://www.microsym.com/editor/assets/intelhex.pdf>
- [6] MICROCHIP. *MCP2200 Data Sheet* [PDF]. 10.3.2011 [cit. 2012-06-06]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/22228B.pdf>
- [7] HEROUT, Pavel. *Učebnice jazyka C*. 6. vyd. České Budějovice: Kopp, 2009, 271, viii s. ISBN 978-80-7232-383-8.

## Přílohy

### Příloha A – Schéma programátoru



Obr. A.1 Schéma programátoru.



Obr. A.2 Schéma zdroje napětí 13V s obvodem MC34063A

Obě schémata vychází z doporučeného zapojení výrobcem s minimální úpravou, aby vyhovovala potřebám programátoru. Modrá LED dioda indikuje činnost USB a množství prováděných operací. Po připojení k počítači jsou odeslány informace o zařízení a nastavení přenosu dat. Jakmile je sběrnice připravena na přenos dat, modrá LED dioda začne blikat. Rychlost blikání se snižuje s náročnějšími operacemi, které jsou aktuálně prováděny. Zelená LED dioda indikuje úspěšné spojení s ovládacím software v počítači a červená LED dioda svítí, jakmile probíhá komunikace s programovaným mikroprocesor.

**Příloha B – Seznam použitelných příkazů pro komunikaci s programátorem**

Tab. B.1 Seznam příkazů

<b>Příkaz</b>	<b>Data</b>	<b>Vrací</b>	<b>Název a popis</b>
<b>0x67</b>	-	<byte1>	<b>Global Start</b> <i>Povolí příjem příkazů. Vrací číslo verze HW .</i>
<b>0x68</b>	-	-	<b>Global End</b> <i>Zakáže příjem dalších příkazů.</i>
<b>0x69</b>	-	<ID1>	<b>HW ID 1</b> <i>Ověřovací instrukce. Vrací ID1 = 0x4A.</i>
<b>0x6A</b>	-	<ID2>	<b>HW ID 2</b> <i>Ověřovací instrukce. Vrací ID2 = 0xF0.</i>
<b>0x80</b>	-	<valid>	<b>Erase program</b> <i>Smaže programovou paměť.</i>
<b>0x20</b>	<func#>	<valid>	<b>Choice func.</b> <i>Volba funkce (nastaví i příslušné piny).</i>
<b>0x21</b>	<byte1>	<valid>	<b>Clk speed</b> <i>Nastavení rychlosti hodin (<math>T = 2 * \text{byte} * 1\text{ns}</math>).</i>
<b>0x22</b>	<byte1>	<valid>	<b>Clk slow speed</b> <i>Nastavení rychlosti hodin (<math>T = 2 * \text{byte} * 1\mu\text{s}</math>).</i>
<b>0x23</b>	<on_off>	<valid>	<b>Vpp/<math>\overline{\text{RESET}}</math></b> <i>Zapne/Vypne programovací napětí nebo nastaví log0/log1 na vývodu <math>\overline{\text{RESET}}</math>.</i>
<b>0x24</b>	<byte1><byte2> <sup>(1)</sup>	<valid>	<b>Write program<sup>(2)</sup></b> <i>Zápis do programové paměti a posun adresy.</i>
<b>0x25</b>	-	<byte1><byte2> <sup>(1)</sup>	<b>Read program<sup>(2)</sup></b> <i>Přečtení z programové paměti a posun adresy.</i>
<b>0x26</b>	-	<valid>	<b>Adress++<sup>(2)</sup></b> <i>Posune ukazatel programové paměti.</i>
<b>0x27</b>	<byte1><byte2> <sup>(1)</sup>	<valid>	<b>Write Config.<sup>(2)</sup></b> <i>Zapiše konfigurační slovo.</i>
<b>0x28</b>	-	<byte1><byte2> <sup>(1)</sup>	<b>Read Config.<sup>(2)</sup></b> <i>Přečte konfigurační slovo.</i>
<b>0x30</b>	<byte1><byte2> <byte3><byte4> <sup>(1)</sup>	<byte1><byte2> <byte3><byte4> <sup>(1)</sup>	<b>Atmel ISP<sup>(3)</sup></b> <i>Odešle a přijme 4 byty.</i>

1: První byte má vyšší význam, obsahuje MSB.

2: Příkaz lze použít jen při programování mikroprocesoru PIC.

3: Příkaz lze použít jen při programování mikroprocesoru AVR.

Tab. B.2 Hodnoty a významy bytů

<b>Byte</b>	<b>Hodnoty</b>	<b>Význam</b>
<valid>	0x01	Dokončeno úspěšně.
	0x02	Dokončeno neúspěšně.
	0x03	Nedokončeno - chyba v průběhu.
	0x04	Neznámý příkaz.
<byte>	0x??	Libovolná hodnota. <sup>(1)</sup>
<func#>	0x01	Programování PIC.
	0x02	Programování AVR.
<on_off>	0x01	Zapne, aktivuje, povolí.
	0x02	Vypne, deaktivuje, zakáže.

1: Muže nabývat hodnot od 0x00 do 0xFF (dekadicky od 0 do 255).