

Univerzita Pardubice  
Dopravní fakulta Jana Pernera

Výukový přípravek s procesorem typu 8051

Michal Bubeník

Bakalářská práce

2014

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal Bubeník**  
Osobní číslo: **D11338**  
Studijní program: **B3709 Dopravní technologie a spoje**  
Studijní obor: **Dopravní infrastruktura: Elektrotechnická zařízení v dopravě**  
Název tématu: **Výukový přípravek s procesorem typu 8051**  
Zadávající katedra: **Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě**

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

- seznámení se se současným stavem výukových přípravků s procesorem řady 8051
- návrh řešení nových výukových přípravků s procesorem řady 8051
- návrh, výroba a oživení HW prototypu přípravku
- návrh SW a jeho realizace
- ověření vlastností HW i SW
- vypracování manuálu pro práci s finálním přípravkem

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

- Skalický, Petr, Mikroprocesory řady 8051, BEN-Technická literatura, ISBN: 80-86056-39-2
- Matoušek, David, Práce s mikrokontroléry ATMEL AT89C2051, BEN-Technická literatura, ISBN: 80-7300-094-6
- Matoušek, David, Práce s mikrokontroléry Atmel AT89LP2052, AT89LP4052, BEN-Technická literatura, ISBN: 80-7300-205-1
- Matoušek, David, Vývojový kit USB51KIT pro AT89S51 a AT89S52, BEN-Technická literatura, ISBN: 80-7300-162-4
- Matoušek, David, Práce s mikrokontroléry ATMEL AT89S8252, BEN-Technická literatura, ISBN: 80-7300-066-0
- Matoušek, David, USB prakticky s obvody FTDI, BEN-Technická literatura, ISBN: 80-7300-103-9

Vedoucí bakalářské práce:

**Ing. Ondřej Černý, Ph.D.**

Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě

Datum zadání bakalářské práce: **25. listopadu 2013**

Termín odevzdání bakalářské práce: **30. května 2014**



prof. Ing. Bohumil Culek, CSc.  
děkan

L.S.



doc. Ing. Radovan Doleček, Ph.D.  
vedoucí katedry

V Pardubicích dne 21. února 2014

## Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 28. 5. 2014



Michal Bubeník

## **Anotace**

Práce se zabývá návrhem a realizací výukového přípravku s procesorem typu 8051, který umožňuje programovat tento mikrokontrolér přes rozhraní USB a též komunikaci přes rozhraní standardu RS-232. První část obsahuje základní informace o klíčové součásti – mikrokontroléru a jeho běžných perifériích. Další část popisuje předchozí výukový přípravek pro porovnání, možná řešení nového zařízení a zvolený nový model, použitý v prototypu. Obsahuje také návrh obslužného softwaru a základní informace o programech, založených na tomto návrhu. V poslední části je ověření parametrů hardwaru (zejména parametrů komunikace), zkouška funkcí softwaru a uživatelský návod pro práci se zařízením a zmíněným softwarem pro budoucí uživatele.

## **Klíčová slova**

mikrokontrolér 8051, USB, SPI, RS-232, výukový přípravek

## **Title**

Educational device with a processor type 8051

## **Annotation**

The work deals with a design and realization of an educational device with a processor type 8051 which allows to program this microcontroller over an USB interface and also allows a communication over an interface of standard RS-232. The first part contains basic information about the key component – the microcontroller and its common peripherals. The next part describes a previous educational device for comparison, possible solutions of a new device and a chosen new model used in a prototype. It also contains a design of an utility software and basic information about programs based in this design. In the final part there is a verification of hardware parameters (especially a communication parameters), a test of software functions and an user guide for work with the device and the mentioned software for future users.

## **Keywords**

microcontroller 8051, USB, SPI, RS-232, educational device

# Obsah

|   |           |
|---|-----------|
| <b>Úvod</b>   | <b>10</b> |
| <b>1 Jádru přípravku – mikrokontrolér řady 8051</b> | <b>11</b> |
| 1.1 Jádru 8051                                      | 11        |
| 1.1.1 Paměť a její organizace                       | 12        |
| 1.1.2 Čítače a časovače                             | 12        |
| 1.1.3 Přerušeni                                     | 13        |
| 1.1.4 Sériový kanál                                 | 14        |
| 1.2 Rozšiřující periferie                           | 14        |
| 1.2.1 Paměť EEPROM                                  | 15        |
| 1.2.2 SPI rozhraní                                  | 15        |
| <b>2 Současný stav přípravků</b>                    | <b>16</b> |
| 2.1 Základní vlastnosti                             | 16        |
| 2.1.1 Nutná výbava pro provoz                       | 16        |
| 2.1.2 Ovládací prvky a komunikace                   | 17        |
| 2.2 Reálný vzhled                                   | 17        |
| <b>3 Řešení nového přípravku</b>                    | <b>19</b> |
| 3.1 Možnosti provedení                              | 19        |
| 3.1.1 Provedení s mezičlenem                        | 19        |
| 3.1.2 Provedení s externí paměti                    | 20        |
| 3.1.3 Provedení s obvodem od FTDI Chip              | 20        |
| <b>4 Návrh a výroba nového přípravku</b>            | <b>21</b> |
| 4.1 Základní blokové schéma                         | 21        |
| 4.2 Výběr součástek                                 | 21        |
| 4.2.1 Převodník od FTDI Chip                        | 21        |
| 4.2.2 Mikrokontrolér                                | 22        |
| 4.2.3 Převodník napěťových úrovní TTL na RS-232     | 22        |
| 4.2.4 Ostatní významné prvky přípravku              | 22        |
| 4.3 Provedení přípravku                             | 23        |
| 4.3.1 Schéma zapojení                               | 23        |
| 4.3.2 Uspořádání na desce plošných spojů            | 27        |
| 4.4 Výroba prototypu                                | 29        |

|          |   |           |
|----------|---|-----------|
| <b>5</b> | <b>Návrh a vytvoření obslužného softwaru .....</b>              | <b>30</b> |
| 5.1      | Návrh konfigurační aplikace .....                               | 30        |
| 5.1.1    | Detekce připojení přípravku .....                               | 30        |
| 5.1.2    | Zápis a čtení nastavení .....                                   | 31        |
| 5.2      | Realizace konfigurační aplikace .....                           | 32        |
| 5.3      | Návrh ovládacího programu .....                                 | 33        |
| 5.3.1    | Detekce připojení přípravku .....                               | 33        |
| 5.3.2    | Načtení souboru .....   | 33        |
| 5.3.3    | Vytvoření komunikace s mikrokontrolérem .....                   | 34        |
| 5.3.4    | Zápis a ověření .....   | 35        |
| 5.3.5    | Zápis a čtení zámků .....                                       | 36        |
| 5.3.6    | Zpracování jednoho bajtu ze souboru až do mikrokontroléru ..... | 36        |
| 5.4      | Realizace ovládacího programu .....                             | 37        |
| <b>6</b> | <b>Ověření vlastností přípravku .....</b>                       | <b>40</b> |
| 6.1      | Měření statických parametrů .....                               | 40        |
| 6.1.1    | Napájení obvodu FT232RL a mikrokontroléru .....                 | 40        |
| 6.1.2    | Převodník napěťových úrovní MAX232 .....                        | 40        |
| 6.2      | Měření dynamických parametrů .....                              | 40        |
| 6.2.1    | Připojení přípravku do USB a regulace spotřeby .....            | 41        |
| 6.2.2    | Nulování mikrokontroléru po startu .....                        | 41        |
| 6.2.3    | Nulování mikrokontroléru obslužným programem .....              | 42        |
| 6.2.4    | Průběh odesílání jednoho bajtu dat při programování .....       | 43        |
| <b>7</b> | <b>Ověření funkcí softwaru .....</b>                            | <b>45</b> |
| 7.1      | Zkouška funkcí konfiguračního programu .....                    | 45        |
| 7.2      | Zkouška funkcí ovládacího programu .....                        | 45        |
| <b>8</b> | <b>Manuál pro práci s přípravkem .....</b>                      | <b>48</b> |
| 8.1      | Popis přípravku .....   | 48        |
| 8.2      | Připojení k počítači a konfigurace přípravku .....              | 50        |
| 8.3      | Obslužný program .....  | 51        |
| 8.3.1    | Spuštění Programátoru .....                                     | 52        |
| 8.3.2    | Načtení souboru .....   | 53        |
| 8.3.3    | Zápis, ověření a čtení paměti mikrokontroléru .....             | 53        |
| 8.3.4    | Nastavení zámků a pojistek .....                                | 54        |
| 8.3.5    | Terminál sériové linky .....                                    | 55        |



|                                  |   |           |
|----------------------------------|---|-----------|
| 8.3.6                            | Testovací program a podpisy mikrokontroléru ..... | 56        |
| 8.3.7                            | Chybové hlášky .....                              | 57        |
| <b>Závěr</b>                     | .....   | <b>59</b> |
| <b>Seznam použité literatury</b> | .....   | <b>60</b> |
| <b>Seznam obrázků</b>            | .....   | <b>62</b> |
| <b>Seznam zkratk</b>             | .....   | <b>64</b> |
| <b>Seznam příloh na CD</b>       | .....   | <b>65</b> |

## Úvod

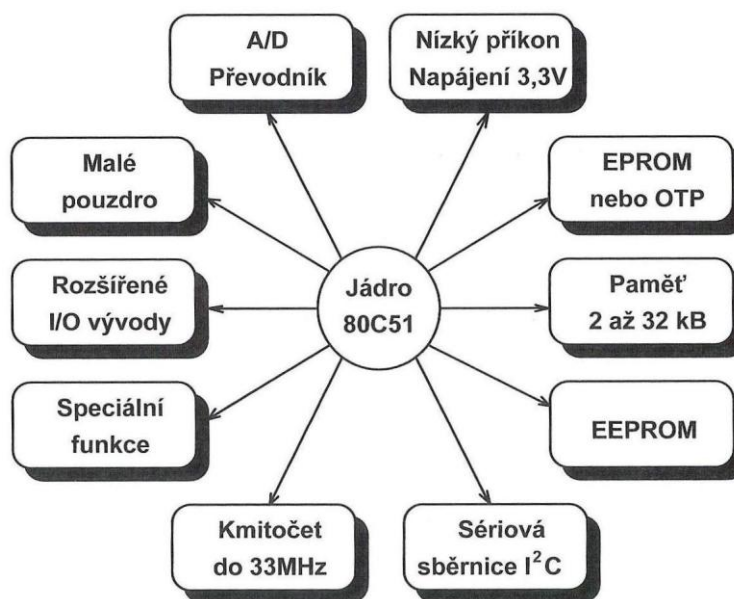
V současné době je hojně využívána mikroprocesorová technika pro řídicí účely v mnoha aplikacích, ať už jednoduchých či náročnějších. Pro náročnější aplikace je logickou volbou vytvořit celý nový řídicí systém, navržený konkrétně pro danou aplikaci. Takový systém je sice nákladnější, ale bude plnit požadovanou funkci efektivněji než systém poskládaný z existujících, univerzálních prvků. Pro jednoduché aplikace, u kterých je zároveň kladen větší důraz na cenu, je výhodnější využít právě univerzálních prvků. Do této skupiny patří mimo jiné i mikrokontroléry řady 8051, které nejsou zrovna nejmladší, ale stále pro ně lze najít široké uplatnění.

Výrobci mikrokontrolérů nabízejí ke svým produktům jak přípravky na propojení s počítačem, tak i programové prostředí. Tyto navazující produkty však představují další náklady. Uživatelé navrhující řídicí aplikace jsou často schopni navrhnout a vytvořit si vlastní zařízení a program pro konfiguraci těchto mikrokontrolérů za výrazně nižší náklady. Navíc nejsou vázáni řešením od výrobce a mohou uplatnit i vlastní požadavky či zahrnout tuto funkci přímo do finální aplikace.

Cílem této práce je návrh a vytvoření přípravku a programu pro výuku mikroprocesorové řídicí techniky, který bude poskytovat nejen základní funkci naprogramování, ale i další funkce, využitelné pro výuku a ověření činnosti zapsaného programu do mikrokontroléru. Ovládání bude jednoduché a k použití bude nutný pouze počítač s rozhraním USB a příslušnými ovladači. Tento výrobek bude nahrazovat stávající přípravek v laboratořích Katedry elektrotechniky, elektroniky a zabezpečovací techniky v dopravě, který již z technických důvodů nevyhovuje. První část práce je věnována základnímu jádru zmiňovaných mikrokontrolérů a jednotlivým základním periferiím. Dále je lehce rozebrán aktuální přípravek pro porovnání a také jsou nastíněny různé možnosti řešení nového přípravku. Následuje zvolené nové řešení a volba hlavních prvků. Nedílnou součástí je i řešení obslužného softwaru, konkrétně programu pro prvotní konfiguraci po vytvoření a programu pro obsluhu.

# 1 Jádru přípravy – mikrokontrolér řady 8051

Mikrokontrolér této řady existuje už od roku 1980 a i když jsou na trhu novější a výkonnější výrobky, je stále využíván. Z příkladu výrobců lze uvést Atmel, Philips nebo Siemens. Dodávají čipy hlavně v pouzdrech PDIP (40 vývodů), PLCC (44 vývodů), TQFP (44 vývodů), které obsahují základní jádro 80C51 s rozšiřujícími periferiemi, například paměť programu (až desítky KB, vícekrát programovatelná) a rozšířená vnitřní paměť RAM na 256 B. Některé obsahují také interní EEPROM paměť pro záznam dat, sériovou sběrnici I<sup>2</sup>C a 10bitový A/D převodník s osmikanálovým analogovým multiplexováním. Pracovní kmitočet se pohybuje většinou v rozmezí 12 až 33 MHz a vyžadováno je pouze jedno napájení, běžně kolem 5 V nebo snižené, 3,3 V. Podstatou názvu „mikrokontrolér“ je to, že je schopen samostatné činnosti bez nutnosti připojení dalších obvodů, například paměti programu a dat. (1)



Obrázek 1 Základní jádro 8051 a rozšiřující periferie (1)

## 1.1 Jádru 8051

Jádru je charakteristické harwardskou architekturou, to znamená, že paměť programu a paměť dat jsou oddělené. Základní jádro tohoto 8bitového mikrokontroléru obsahuje centrální procesorovou jednotku (CPU), vnitřní paměť RAM (kapacita 128 B), vnitřní paměť programu (kapacita 4 KB), čtyři 8bitové vstupně výstupní brány a vnitřní generátor hodin. Pro snadnější

zpracování událostí je přítomen řadič přerušení, který rozlišuje minimálně pět zdrojů: dva externí, dva od časovačů a další od sériového kanálu. K časování nebo zachytávání vnějších událostí slouží čítače (standardně jsou dva, T0 a T1), ke kterým jsou přivedeny dva vnější vstupy a hodinový signál z interního generátoru hodin. (1)

Dále jsou z čipu vyvedeny řídicí signály pro komunikaci s vnější programovou a datovou pamětí (až 64 KB). Jednotlivé části jsou propojeny společnou vnitřní sběrnici. Pro toto jádro je typické trvání jednoho strojového cyklu 12 period hodinového signálu, to znamená, že při použití 12 MHz krystalu trvá jedna jednocyklová instrukce 1  $\mu$ s. (1)

### **1.1.1 Paměť a její organizace**

Jak už bylo zmíněno, paměť je rozdělena na programovou a datovou. Tyto dvě paměti lze dále rozdělit na vnitřní a vnější (vnější paměť je fyzicky oddělená, připojená přes vstupně výstupní brány a řídicí signály a její kapacita může být až 64 KB). Mezi vnitřní a vnější programovou pamětí dochází k překrývání. Logickou úrovní na jednom ze vstupů lze určit, zda má být použita vnitřní paměť a nad její rozsah přistupovat k paměti vnější, nebo zda se používá celá vnější paměť jako paměť programu. (1)

Vnitřní datová paměť RAM o kapacitě 128 B obsahuje čtyři tzv. banky, každá po osmi registrech. Za tímto prostorem je 16 bitově přístupných registrů a zbývající oblast (48 registrů) je volně přístupná. Následujících 128 adres do 256 B je využito pro speciální registry a v tomto rozsahu adres se nachází i rozšířená paměť RAM. Přístup do rozšířené paměti je rozlišen přímým a nepřímým adresováním, nepřímým je přistupováno do rozšířené paměti. Přístup do vnější datové paměti je uskutečněn jinou instrukcí a s pomocným 16bitovým ukazatelem. Tento ukazatel omezuje tuto paměť na 64 KB. (1)

### **1.1.2 Čítače a časovače**

Základní výbavou jádra jsou dva 16bitové čítače. Nastavení a vyčítání stavu těchto čítačů je v osmibitové struktuře řešeno oddělenými registry na nižších a vyšších 8 bitů. Do těchto registrů mohou být zachytávány vnější události na dvou určených vstupech nebo vnitřní hodinový signál (funkce časovače). Přetečení registrů je následně obsluženo přerušovacím systémem. (1)

Funkce čítačů a časovačů je možná ve čtyřech různých režimech. První představuje 13bitový čítač/časovač s 5bitovým předděličem ze spodních bitů nižšího registru. Druhý mód pracuje jako 16bitový čítač/časovač, složený z obou registrů. Třetí možnost je využití pouze jednoho registru a druhý slouží pro nastavení hodnoty po přetečení. Poslední režim je možný pouze pro jeden čítač a spočívá v činnosti dvou samostatných osmibitových čítačů, každý s vlastním přerušením. K tomuto režimu je použito přerušení od druhého čítače a tak jeho současné použití je omezené. (1)

### 1.1.3 Přerušení

Pro zpracování podnětů od časovačů nebo vnějších zařízení slouží přerušovací systém, rozlišující v základu pět příčin přerušení. Dvě nezávislá přerušení od vnějších zdrojů mohou být detekována jako sestupná hrana nebo logická úroveň na předem určených vstupech. Další dvě nezávislé žádosti, od čítačů, jsou generovány při jejich přetečení. Požadavek od sériového kanálu je ve formě pouze jedno přerušení. (1)

Každý zdroj je identifikován podle vlastního příznaku. Tyto příznaky jsou při zpracování automaticky nulovány, kromě příznaků sériové linky. Z důvodu existence pouze jednoho přerušení pro tento zdroj, jsou příjem a odeslání rozlišeny dvěma příznaky. Ty nejsou automaticky nulovány a tak tento úkol zůstává na samotném obslužném programu, ve kterém může být zvoleno i pořadí zpracování. (1)

Pro správné použití přerušovacího systému je nutno znát pořadí zpracování jednotlivých zdrojů, neboli priority. Základní pořadí je pevně určeno takto: vnější přerušení 0, čítač 0, vnější přerušení 1, čítač 1 a sériový kanál. Pomocí registru priorit přerušení lze toto dále rozdělit do dvou úrovní. V rámci úrovně jsou události vyřizovány postupně a musejí počkat na vyřízení předchozích. Vyšší úroveň však může přerušit činnost z nižší úrovně přerušení. To umožňuje propracovanější strukturu ovládání programu. Pouze dvě úrovně však nemusí vždy stačit a tak novější verze mikrokontrolérů mají těchto úrovní více. (1)

Po detekci příznaku, vytvoření přerušení a případném nulování, je činnost programu pozastavena a začínou se vykonávat instrukce od tzv. vektoru přerušení. To je pevně určená adresa v paměti programu. U řady 8051 bývají tato místa od sebe vzdálena pouze 8 adres a tak se běžně využívá nepodmíněného skoku na podprogram v jiné části paměti. (1)

### 1.1.4 Sériový kanál

Sériová obousměrná komunikace je velice využívaná a tak její integrace přímo do čipu je velkou výhodou. Přítomná sériová linka umožňuje komunikaci v 8 či 9bitovém asynchronním nebo v 8bitovém synchronním režimu a současně přijímat i vysílat data s využitím minimálně třech vodičů (RxD, TxD a zem). (1)

První režim je synchronní 8bitový s pevnou přenosovou rychlostí. Jeden vodič (RxD) se využívá k příjmu a odesílání dat a druhý (TxD) pro hodinový signál. Přenosová rychlost je pevně dána jako 1/12 frekvence vnitřního oscilátoru. V druhém režimu je odesílání prováděno přes vodič TxD, příjem přes RxD a osm přenášených datových bitů je ohraničeno počátečním a koncovým bitem. Rychlost přenášení jednotlivých bitů je volitelná a k jejímu generování lze využít frekvenci přetečení čítače 1. Další režim přenáší celkem 11 bitů (1 počáteční, 9 datových a 1 koncový). Poslední datový bit je ukládán nebo vyčítán z registru řízení sériové linky. Přenosová rychlost je pevně stanovena na 1/32 nebo 1/64 frekvence oscilátoru, podle nastavení řídicích registrů. Čtvrtý režim je podobný jako předchozí, přenáší tedy 9 datových bitů, 1 počáteční a 1 koncový, celkem 11. Rychlost komunikace určuje frekvence přetečení čítače 1. (1)

Bez ohledu na nastavení funkce je odesílání spuštěno zápisem do vyrovnávacího registru, kterým je sériový kanál vybaven. Do tohoto registru je také zapisována aktuálně přijatá hodnota a tak mohou být okamžitě přijímána další data. Nová hodnota samozřejmě přepíše předchozí a tak musí být vyčtena co nejdříve. Příjem je aktivován pomocí povolujících bitů v registru řízení linky. (1)

## 1.2 Rozšiřující periferie

Kromě zmíněných rozšiřujících periférií (EEPROM, paměť programu, rozšířená RAM, I<sup>2</sup>C, A/D převodník) bývá přidáno ještě SPI rozhraní a hodně rozšířen je také komparační a záchytný systém. Ten umožňuje generování pulzů s různou šířkou (pulzně šířkovou modulaci), měření různých impulzů a obecně výkonné vytváření řídicích signálů. Díky propracovanosti je pro běžnou činnost zapotřebí minimální obsluhy ze strany programu. Vybavenost mikrokontroléru (počet periférií) se promítne do jeho rozměrů, protože při větším počtu periférií je v provedení s více vývody. (1)

### **1.2.1 Paměť EEPROM**

Z důvodu malé interní paměti RAM je pro větší množství uchovávaných dat programem nutno použít externí EEPROM paměť. Některé mikrokontroléry ji mají integrovanou přímo v pouzdru, s kapacitou jednotky KB. Její řízení a nastavení je realizováno pomocí přidaných speciálních registrů. Výhodou je pak uvolnění portů, které by byly obsazeny pro komunikaci s danou pamětí. Externí paměť lze však nadále přidat a překrývání adres o rozsahu integrované části je řešeno obdobně, jako tomu je u paměti programu. Pomocí jednoho bitu ve speciálním registru lze určit, zda se bude v překrývaném rozsahu přistupovat na interní či externí paměť. (2)

### **1.2.2 SPI rozhraní**

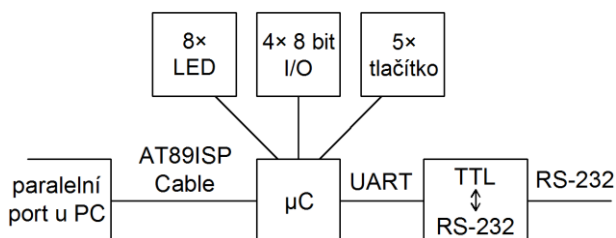
Serial Peripheral Interface, neboli sériové periferní rozhraní, je sériová synchronní komunikace pomocí dvou datových (MISO a MOSI) a jednoho hodinového vodiče (SCK). Určování řídicí a podřízené jednotky je realizováno pomocí čtvrtého vývodu (SS). V porovnání s běžným sériovým kanálem nabízí tato komunikace rychlejší přenos dat. (2)

Nejdůležitější výhodou tohoto rozhraní je, že poskytuje komunikaci vhodnou pro programování (zápis a čtení paměti programu a zámků, případně interní paměti EEPROM a pojistek) a toho je také využito. (2)

Kód programu, případně další nastavení, lze do mikrokontroléru zapsat paralelním způsobem, který vyžaduje všechny porty a proto je výhodnější vyjmutí, naprogramování v odděleném zařízení a vložení zpět, než aby se komplikovalo hardwarové řešení aplikace. S použitím SPI rozhraní není nutná žádná hardwarová úprava a programování je možné přímo v cílové aplikaci, bez zbytečného obsazení vstupně výstupních portů. Před komunikací je nutno uvést mikrokontrolér do stavu nulování a držet tento stav. Následně je použit stejný protokol jako při paralelním způsobu, jenom jsou instrukce a data odesílány sériově. (2)

## 2 Současný stav přípravků

Při vytváření práce byl dostupný starší přípravek pro programování a zkoušení mikrokontrolérů, který měl být nahrazen novým. Tento přípravek plnil svou funkci a byl využíván k výuce, avšak měl jisté nevýhody. Hlavní z nevýhod bylo připojení k počítači přes paralelní port. U nových počítačů se tento port přestává vyskytovat a u menších provedení, jako jsou notebooky, ho nenalezneme vůbec. Použití se tedy stává problematické. Další nevýhodou bylo externí napájení oddělené od komunikace a realizované přes univerzální konektor. To přinášelo potřebu mít oddělený zdroj napětí. Nakonec, jako vstup bylo použito pouze 5 tlačítek. Vložený mikrokontrolér byl 8bitový a tak pro vybavení celého portu je nutné použít 8 tlačítek.



Obrázek 2 Základní blokové schéma předchozího přípravku

### 2.1 Základní vlastnosti

Přípravek byl vybaven mikrokontrolérem AT89S8253 od firmy Atmel. Ten obsahuje kromě základního jádra 8051 další rozšíření. Mezi ně patří například interní paměť EEPROM, rozhraní SPI s možností programování a další 16bitový čítač/časovač T2 (2). Mikrokontrolér byl opatřen 12 MHz krystalem, díky kterému trval jeden strojový cyklus 1  $\mu$ s.

Dále byl vybaven integrovaným obvodem MAX232 od firmy Maxim pro převod úrovní sériové linky z TTL (5 V) na standard RS-232 ( $\pm 12$  V).

#### 2.1.1 Nutná výbava pro provoz

K použití přípravku nestačilo pouze samotné fyzické zařízení a počítač. Jak bylo zmíněno, k napájení bylo nutné mít oddělený zdroj napětí 4,5 V až 5,5 V, vyvedený na univerzální zámkový konektor. Komunikace pro přenos dat do mikrokontroléru probíhala přes kabel AT89ISP. Ten byl definován v aplikační poznámce, vydané výrobcem mikrokontroléru



Atmel (3). Na jedné straně kabelu byl konektor pro paralelní port k počítači. Na druhém konci byl konektor do přípravku, opatřený deseti kontakty ve dvou řadách.

Pro obsluhu byla kromě fyzického příslušenství nutná také určitá softwarová výbava. Ke zmíněnému kabelu AT89ISP byl vydán „Atmel Microcontroller ISP Software“ – program pro komunikaci s přípravkem od firmy Atmel (3), využívající paralelní port u počítače. Tato aplikace umožňovala zápis a ověření dat a nastavení zámků mikrokontroléru. Výběr zařízení k programování byl možný mezi pěti provedeními z řady AT89 (3).

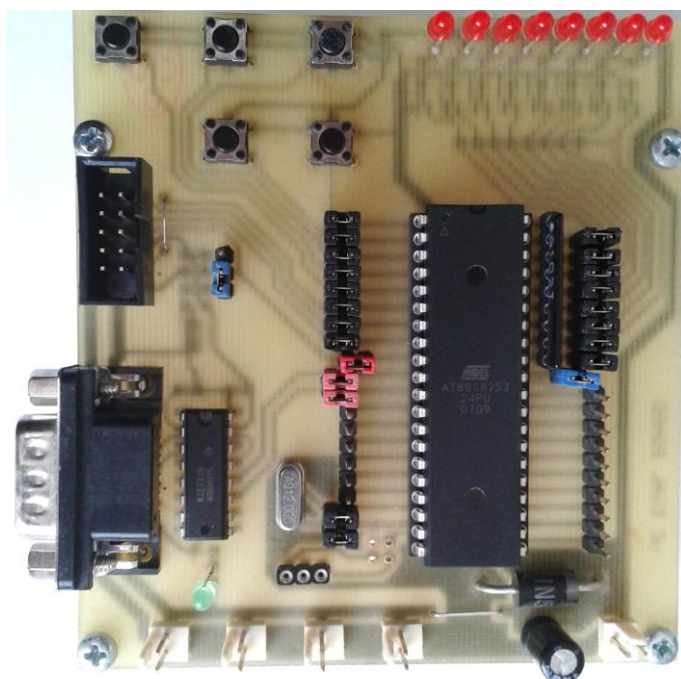
### **2.1.2 Ovládací prvky a komunikace**

Zapsaný program do mikrokontroléru bylo možno obsluhovat pomocí pěti tlačítek na jednom z 8bitových portů. Další vstupně výstupní brána byla osazena osmi LED jako výstup. Všechny vývody byly také přivedeny na oboustranné kolíky pro připojení libovolného dalšího zařízení k přípravku. Tato vlastnost však nebyla využívána. Naopak bylo používané rozhraní standardu RS-232, přivedené na sériovou linku mikrokontroléru přes převodník úrovní TTL na tento standard.

Dalším používaným rozhraním bylo samozřejmě připojení kabelu AT89ISP, přes které probíhalo programování. Tato komunikace obsahovala základní rozhraní SPI, rozšířené o pomocné napětí a ovládání nulování mikrokontroléru.

## **2.2 Reálný vzhled**

Následující obrázek obsahuje pohled shora na předchozí přípravek. U jedné strany desky je 5 tlačítek pro ovládání vstupu mikrokontroléru a 8 LED pro zobrazení výstupu. Na vedlejším kraji se nachází konektor pro AT89ISP kabel a sériové rozhraní s integrovaným obvodem MAX232. V dolní části obrázku jsou vidět konektory pro připojení napájení přípravku. Kolem mikrokontroléru jsou rozmístěny oboustranné kolíky, které je možno pomocí propojek spojit se zbytkem desky nebo použít pro připojení dalších zařízení.



**Obrázek 3** Pohled shora na předchozí přípravek

### 3 Řešení nového přípravku

Do nového přípravku bude použit opět mikrokontrolér od firmy Atmel. Dva z jeho portů budou vybaveny celkem osmi tlačítky a osmi LED. Pokud bude počet zbylých volných vývodů minimálně osm, budou vyvedeny pro připojení dalších 8bitových periférií. Dále bude opatřen sériovým rozhraním standardu RS-232, přivedeným na sériovou linku mikrokontroléru. Důležitou vlastností bude programování z počítače přes rozhraní USB bez nutnosti vyjmutí mikrokontroléru nebo hardwarové změny konfigurace přípravku. Toto vyřeší problém s paralelním portem ze staršího přípravku. Nevýhodu externího napájení vyřeší použití přívodního napětí 5 V z USB.

Součástí výbavy přípravku bude také obslužný software, sloužící hlavně k naprogramování mikrokontroléru. Dále bude umožňovat ověření zapsaných dat a nastavení zámků a pojistek. Data budou do softwaru načítána ze souboru ve formátu Intel HEX (4).

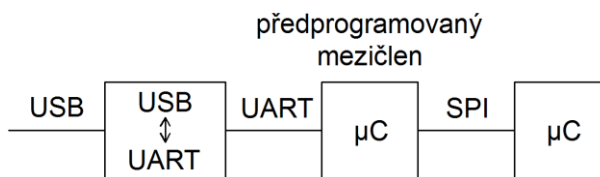
#### 3.1 Možnosti provedení

Volba provedení nového přípravku je nejvíce omezená dostupnými možnostmi propojení jednotlivých komponent. Mikrokontroléry jsou vybaveny sériovou linkou a programování je možné paralelním způsobem s využitím všech portů nebo s využitím přítomného SPI rozhraní, například u AT89S51 (5). Dále je možné, aby mikrokontrolér pracoval podle programu uloženého v externí paměti (5). Pro obsluhu USB jsou k dispozici integrované obvody od FTDI Chip, ze kterých nejjednodušší typy slouží jako převodník z USB na UART (na sériovou linku, například FT232) (6).

##### 3.1.1 Provedení s mezičlenem

Vyjmenované možnosti nabízejí z prvního pohledu přímé propojení mikrokontroléru s USB přes sériovou linku. Tato komunikace je ale dostupná pouze mimo programovací režim (5). Dále je nevhodné využívat paralelní způsob programování při dostupnosti sériového, hlavně z hlediska obsazení portů. V jedné publikaci (7) je propojení vyřešeno využitím dalšího mikrokontroléru, který pracuje jako převodník ze sériové linky na rozhraní SPI, které je vytvořeno programově na jednom z portů. Toto řešení však vyžaduje předem

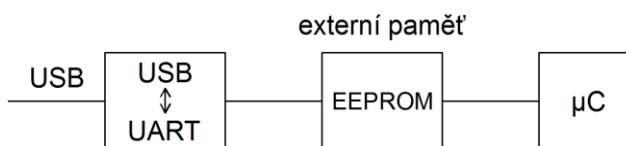
naprogramovaný mikrokontrolér, což je velký problém, když přípravek pro něj teprve vytváříme.



Obrázek 4 Řešení přípravku s mezičlenem

### 3.1.2 Provedení s externí pamětí

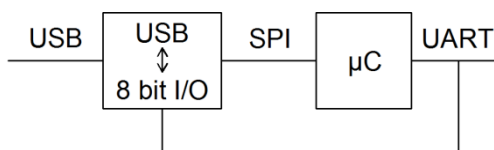
Další možností je uchovávat program v externí paměti. Do ní lze program zapsat mimo mikrokontrolér a následně ho začít používat s touto pamětí jako celou paměť programu, bez použití vnitřní části. Nevýhodou je obsazení celých dvou portů, a tak nezbyvá místo na nezbytné vstupy a výstupy pro tlačítka a LED. Tato možnost tedy není použitelná.



Obrázek 5 Řešení přípravku s externí pamětí

### 3.1.3 Provedení s obvodem od FTDI Chip

Funkce převodníků od FTDI Chip nejsou omezené pouze na jednu základní. Nabízejí také řadu dalších režimů činnosti a jedním z nich je konkrétně „Synchronous Bit Bang Mode“ – režim 8bitového univerzálního portu, obsluhovaného synchronně (8). Tento režim je dostupný například u převodníku FT232R (8) a lze ho využít k programovému vytvoření SPI rozhraní jako u provedení s mezičlenem, avšak bez nutnosti dalšího, předem naprogramovaného mikrokontroléru. Toto řešení je tedy použitelné a výhodné z hlediska počtu integrovaných obvodů.

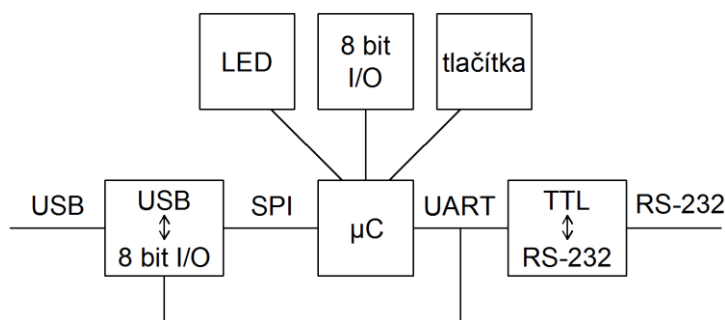


Obrázek 6 Řešení přípravku s FTDI převodníkem

## 4 Návrh a výroba nového přípravku

### 4.1 Základní blokové schéma

Pro návrh přípravku je použito řešení s obvodem od FTDI Chip, uvedené v předcházející kapitole. Po přidání dalších prvků pro vstup, výstup a ovládání bude základní blokové schéma vypadat dle následujícího obrázku.



Obrázek 7 Základní blokové schéma návrhu přípravku

### 4.2 Výběr součástek

#### 4.2.1 Převodník od FTDI Chip

FTDI Chip nabízí širokou řadu integrovaných obvodů (6) a pro tuto aplikaci jsou vhodná provedení se sériovou linkou UART s možností režimu univerzálního portu. Ve velkoobchodech s elektronickými součástkami jsou k nalezení například obvody FT232R, FT232B, FT245R, FT245B a FT2232D (9). Nejvhodnějším je potom FT232RL, který splňuje požadavky a zároveň je nejúspornější volbou. Existují také převodníky, které jsou vybavené přímo potřebným SPI rozhraním, ale za vyšší cenu. Pokud můžeme realizovat SPI komunikaci softwarově, tak je zbytečné pořizovat dražší komponenty.

FT232RL je opatřen pouzdem SSOP28 s 28 vývody ve dvou řadách. Rozměry pouzdra jsou přibližně  $5,3 \times 10,2$  mm a vzdálenost mezi vývody je 0,65 mm. (8)

## 4.2.2 Mikrokontrolér

Vhodný mikrokontrolér by měl být vybaven sériovou linkou, SPI rozhraním s možností programování a čtyřmi 8bitovými porty, aby bylo možno umístit tlačítka a LED k jedné straně. Dále by měl obsahovat vnitřní paměť EEPROM, protože připojení externí paměti dat není možné z důvodu obsazení portů. Těmto požadavkům vyhovuje mikrokontrolér AT89S8253 od firmy Atmel (2). Ten byl použit i v předchozím přípravku. Další dva typy od této firmy, AT89S51 (5) a AT89S52 (10), nejsou vybaveny interní pamětí EEPROM, ale mají provedení se stejným pouzdem a jejich programování je téměř shodné s prvním mikrokontrolérem. Přípravek by tedy měl podporovat i tyto další dva typy.

AT89S8253, stejně jako AT89S51 a AT89S52, je k dispozici v pouzdru PDIP40 se 40 vývody ve dvou řadách. Rozměry pouzdra jsou přibližně  $13,5 \times 52$  mm a vzdálenost mezi vývody je 2,54 mm. (2)

## 4.2.3 Převodník napět'ových úrovní TTL na RS-232

Přípravek má být napájen z USB napětím 5 V a pro sériovou linku standardu RS-232 je zapotřebí napětí 12 V. K vytvoření vyššího napětí lze využít integrované obvody od firmy Maxim. Tyto obvody jsou schopny, s pomocí externích kapacit 1  $\mu$ F, vytvořit potřebné napětí  $\pm 12$  V. Nejmenší dostupné provedení je dvoukanálové a tak vhodným obvodem je MAX232CPE. (11)

MAX232CPE je v pouzdru PDIP16 s 16 vývody ve dvou řadách. Rozměry pouzdra jsou přibližně  $6 \times 18,5$  mm a vzdálenost mezi vývody je 2,54 mm. (11)

## 4.2.4 Ostatní významné prvky přípravku

Interní oscilátor mikrokontroléru bude opatřen externím krystalem pro frekvenci 12 MHz. Při této frekvenci potrvá jeden strojový cyklus (a tím i krok čítačů/časovačů) 1  $\mu$ s (2). S tímto krokem bude možné provádět přesné časování například jedné sekundy.

Při využití celého portu mikrokontroléru bude proud jedním vývodem omezen přibližně na 2 mA (2). Proto jsou vhodné LED, které pracují s tímto minimálním proudem. Zároveň bude tento proud omezen rezistory, umístěnými v sérii s každou diodou. Testem byla vybrána hodnota rezistoru 1,5 k $\Omega$ , při které procházel diodou proud přibližně 2 mA.

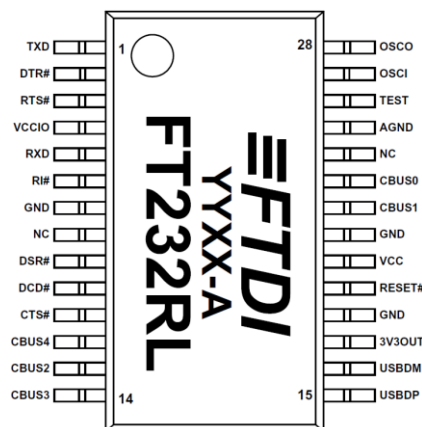
### 4.3 Provedení přípravku

Bylo navrženo základní blokové schéma a zvoleny hlavní součástky, od kterých se odvíjí použití dalších, drobných součástek. Na základě tohoto návrhu bylo vytvořeno detailní schéma zapojení a výkres navržené jednostranné desky plošných spojů.

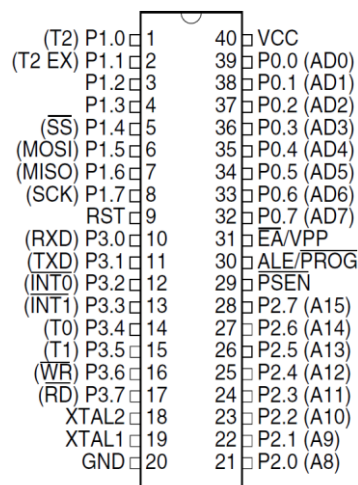
#### 4.3.1 Schéma zapojení

Schéma zapojení bylo vytvářeno v programu AutoCAD 2013 a celé je obsaženo v příloze A. K jednotlivým částem bude vždy uveden výřez uvažované oblasti.

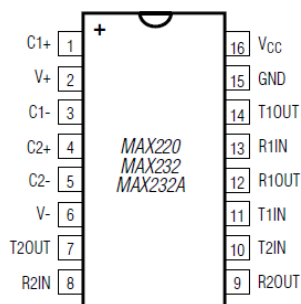
Při vytváření zapojení desky bylo potřeba znát funkci a uspořádání jednotlivých vývodů u použitých integrovaných obvodů. Následující obrázky jsou převzaty z dokumentace FTDI převodníku FT232RL (8), mikrokontroléru AT89S8253 (2) a obvodu MAX232 (11).



Obrázek 8 Popis vývodů FTDI převodníku FT232RL (8)



Obrázek 9 Popis vývodů mikrokontroléru AT89S8253 (2)



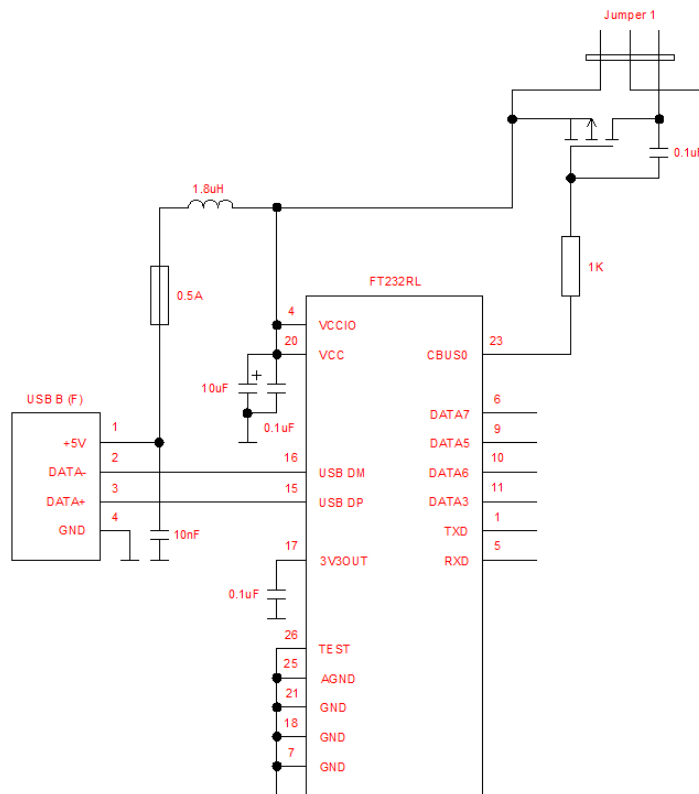
**Obrázek 10** Popis vývodů obvodu MAX232 (11)

Nejprve byla vyřešena část kolem obvodu FT232RL. K vývodům VCC a VCCIO bylo přivedeno napětí od konektoru USB přes vratnou pojistku na stálý proud 0,5 A a tlumivku o indukčnosti řádově jednotky  $\mu\text{H}$  (1,8  $\mu\text{H}$ ). Toto napájení bylo v těsné blízkosti u konektoru opatřeno filtrační kapacitou 10 nF. Datové vodiče z USB byly propojeny s odpovídajícími vývody USB DM a USB DP. K napájení integrovaného obvodu byly přidány kapacity 10  $\mu\text{F}$  a 0,1  $\mu\text{F}$  pro stabilizaci a filtraci. Vývod 3V3OUT byl taktéž vybaven kapacitou 0,1  $\mu\text{F}$ .

Standardizované USB zařízení by mělo splňovat omezení odebíraného proudu, které je 100 mA, popřípadě 500 mA, pokud tuto informaci poskytne počítači. Dále by mělo podporovat režim snížené spotřeby, kdy by proud neměl přesáhnout hodnotu 2,5 mA (8).

Obvod od FTDI obsahuje jednotku CBUS, která nabízí funkci regulace spotřeby na kterémkoliv z jejích pěti výstupů. Využit byl CBUS0, protože se nachází nejdále od ostatních použitých vývodů. Tento výstup byl přiveden k unipolárnímu tranzistoru přes RC článek s rezistorem 1 k $\Omega$  a kapacitou 0,1  $\mu\text{F}$  pro zpomalení spínání. Přes tento tranzistor vede napájení do dalších částí přípravku a pomocí propojky je umožněno jeho přemostění. Obejití tranzistoru slouží pro situaci, kdy napájení USB ignoruje datové vodiče a v tom případě by obvod FT232RL neotevřel tranzistor (například při napájení z externího zdroje k USB). Obdobné schéma zapojení FT232RL se nachází v jeho dokumentaci (8). Následující obrázek obsahuje výřez s výše popsanou částí.

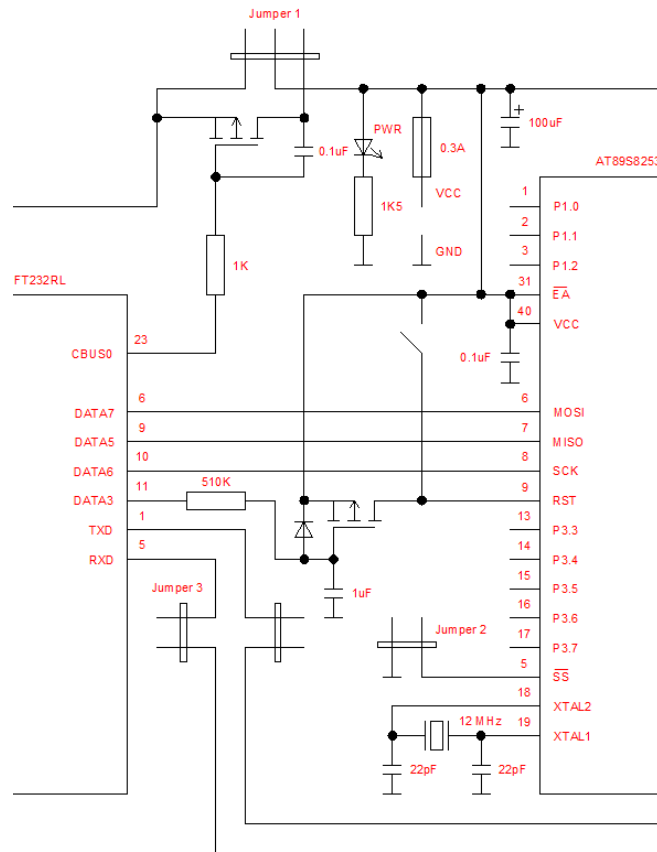




**Obrázek 11** Schéma zapojení obvodu FTDI

Další částí je propojení mikrokontroléru s FT232RL. Vodiče SPI komunikace MOSI, MISO a SCK byly propojeny s vývody DATA7, DATA5 a DATA6 (12). Pro čtvrtý vodič, SS, byla přidána možnost spojení se zemí napájení pomocí propojky. Vývod DATA3 (12) byl použit pro ovládání stavu nulování mikrokontroléru. Logická hodnota na tomto výstupu byla invertována pomocí dalšího tranzistoru a krátký reset po připojení napájení byl realizován RC článkem s vybíjecí diodou. Do článku byl použit rezistor 510 k $\Omega$ . Použití velkého odporu bylo nutné pro omezení proudu, tekoucího přes vybíjecí diodu do zbytku přípravku. S touto hodnotou odporu byla do článku postačující kapacita 1  $\mu$ F. Reset mikrokontroléru je dále umožněn spínacím tlačítkem propojeným s napájením.

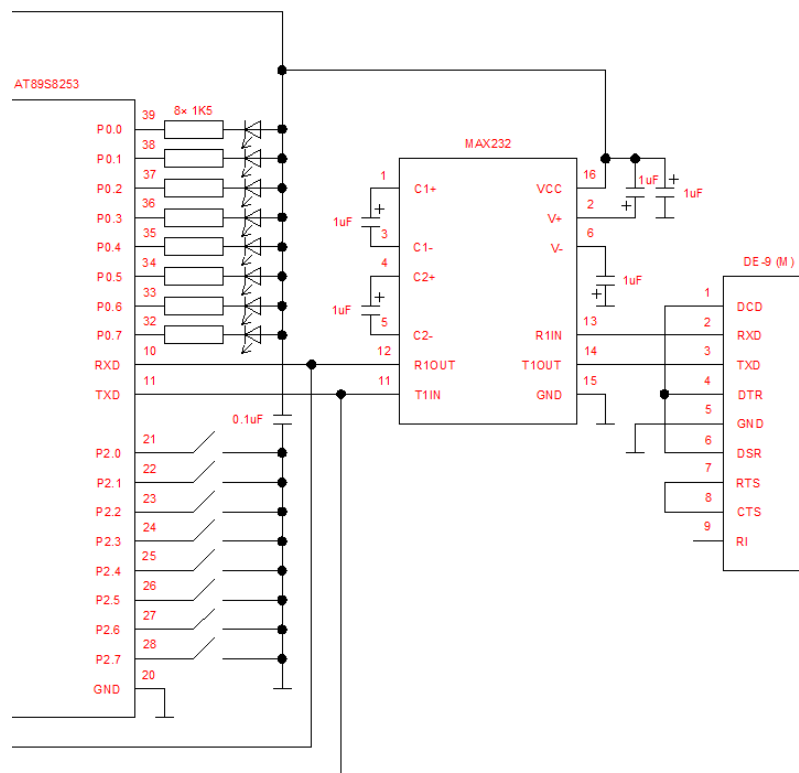
Napájení je také vybaveno indikační LED s rezistorem 1,5 k $\Omega$ , stabilizační kapacitou 100  $\mu$ F a přímo u mikrokontroléru je filtrační kapacita 0,1  $\mu$ F. Přívod pro externí zařízení je přes vratnou pojistku pro stálý proud 0,3 A. Na vstupy XTAL1 a XTAL2 generátoru hodin mikrokontroléru byl umístěn krystal pro frekvenci 12 MHz. Dále byly připojeny rozběhové kondenzátory 22 pF (dle výrobce). Následující obrázek obsahuje výše popsanou část.



**Obrázek 12** Schéma zapojení mezi obvodem FTDI a mikrokontrolérem

Na port P0 mikrokontroléru bylo připojeno 8 LED, každá s jedním rezistorem 1,5 kΩ v sérii, pro omezení proudu. K portu P2 bylo umístěno 8 spínacích tlačítek. Pro definování logické jedničky na vstupu portu P2 nebylo nutné použít rezistory vedené k napájení, protože jsou již obsaženy v mikrokontroléru (2). Mezi tyto dvě skupiny prvků byla umístěna kapacita 0,1 µF, aby se zamezilo možnému blikání diod.

Sériová linka byla propojena s obvodem MAX232 pro převod napěťových úrovní do konektoru DE-9 a také s obvodem FT232, kde je umístěna propojka pro možnost režimu „opakovače“, viditelná na předchozím obrázku. Integrovaný obvod od firmy Maxim bylo nutné opatřit pěti kapacitami 1 µF pro správnou činnost. Výstupní konektor je potom připojen pouze datovými vodiči RXD a TXD. Zbylé vodiče, sloužící pro řídicí signály, nebylo potřeba využít a tak byly spojeny přímo u konektoru pro opakování přijatých signálů. Tuto část, která dotváří celé schéma, je možné vidět na následujícím obrázku.



Obrázek 13 Schéma zapojení vstupních a výstupních prvků a zapojení sériové linky

### 4.3.2 Uspořádání na desce plošných spojů

Výkres byl vytvářen v programu EAGLE 6.5.0 Light a výsledný soubor je obsažen v příloze B.

Prvním krokem bylo umístění konektoru USB do jednoho rohu desky. K tomuto konektoru patří filtrační kapacita, kterou bylo vhodné umístit co nejbližše jeho vývodům. Dále navazuje tlumivka a vratná pojistka. Tlumivka byla zvolena ve vývodovém provedení a vzniklé místo pod ní bylo využito pro datové cesty USB. Zem napájení byla vyvedena od konektoru ke kraji desky, pro snadný rozvod na další místa. Napájení bylo vyvedeno k volné straně konektoru, kde bylo místo pro tranzistor na ovládání spotřeby přípravku, přilehlý RC článek, propojku pro přemostění a indikační diodu. Řídicí signál k tranzistoru byl přiveden přes ostatní cesty s pomocí vývodového provedení rezistoru. Napájení za tranzistorem pokračuje souběžně s krajem desky.

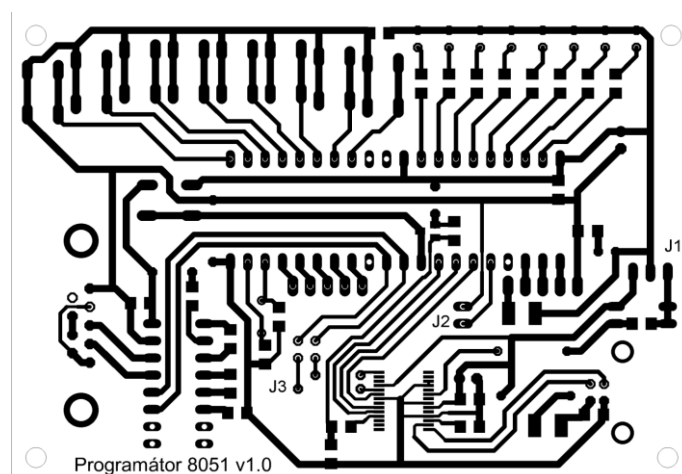
Pokračováním od USB je obvod od FTDI, který byl umístěn ke kraji desky a otočen tak, aby přilehlé kapacity a ostatní vývody využívaly místo ve stejném směru od USB rozhraní. Z důvodu vyššího počtu vývodů země napájení se tyto cesty sbíhají pod integrovaným

obvodem a vedou ke společné zemi napájení u kraje desky. Sériová linka a datové vývody z FT232 byly vyvedeny směrem do středu desky a mikrokontrolér byl umístěn podélně, pro jejich snadné napojení.

V návaznosti na mikrokontrolér byly přibližně do jedné řady seřazeny LED s rezistory v sérii a spínací tlačítka. Nejbližší možné umístění této skupiny k mikrokontroléru určuje rozměr kratší strany desky. Dvě krajní tlačítka z každé strany řady byla posunuta směrem od kraje, což poskytlo nejen místo pro rohový otvor na šroub, ale také snadnou identifikaci prostředních tlačítek při prvním pohledu (tento jev je znatelný při vyšším počtu tlačítek, v malém provedení není tak významný).

Dále bylo vedle mikrokontroléru umístěno tlačítko pro jeho nulování. Potřebné napájení, zem a cesty sériové linky byly vedeny prostorem pod mikrokontrolérem. Kraj 40vývodového integrovaného obvodu určuje umístění obvodu MAX232 kvůli propojení zmíněné sériové komunikace. Převodník napět'ových úrovní byl dále opatřen kapacitami v SMD provedení a připojen na napájení. V těsné blízkosti byl umístěn konektor DE-9 a jeho pozice určuje druhý, delší rozměr desky. Konektor byl propojen s obvodem MAX232 dvěma datovými cestami a řídicí signály byly spojeny přímo mezi vývody sériového rozhraní.

Zbylé, menší součástky byly umístěny na volná místa a dále byla snaha drobnými úpravami docílit co nejlepšího využití desky. Na závěr došlo k umístění textové identifikace propojek a názvu přípravku. Celý přípravek byl navržen na jednostrannou desku plošných spojů o rozměrech  $10 \times 7$  cm. Následující obrázek obsahuje zmenšený pohled na spodní stranu desky a není určen pro výrobu negativu.



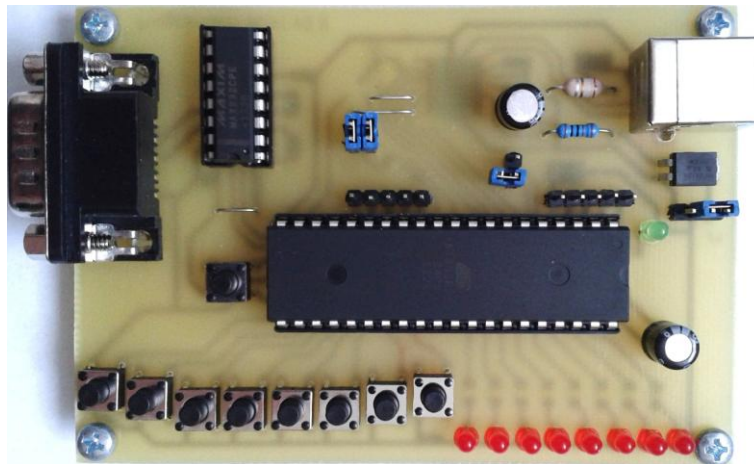
**Obrázek 14** Náhled výkresu desky pošlých spojů při pohledu na spodní stranu přípravku

## 4.4 Výroba prototypu

V průběhu vytváření detailního zapojení a uspořádání na desce plošných spojů vznikl prototyp přípravku pro ověření úprav. V rámci nich bylo upuštěno od propojení výstupního signálu o frekvenci 12 MHz z obvodu FTDI na vstup mikrokontroléru. Důvodem bylo, že výstupní signál dosahoval napětí pouze 3,3 V a jeho frekvence byla velice nestabilní.

Dále byl přidán tranzistor pro změnu logické úrovně, nutné pro nulování mikrokontroléru. Po připojení přípravku byl mikrokontrolér uveden do stavu resetu a ten bylo nutno ukončit z obslužného programu. Proto bylo nezbytné invertovat logickou úroveň výstupu z obvodu FTDI. Přestože použitý mikrokontrolér je vybaven funkcí resetování po startu (2), docházelo k problémům a byl nutný dodatečný reset. Z tohoto důvodu byl tranzistor pro inverzi logické úrovně resetu vybaven RC článkem na straně obvodu FTDI. Na následujícím obrázku je fotografie prototypu v pohledu shora.

Seznam součástek, které byly použity v prototypu, je v příloze C.



Obrázek 15 Pohled shora na prototyp přípravku

## **5 Návrh a vytvoření obslužného softwaru**

Obslužný software bude muset, kromě komunikace s mikrokontrolérem, obsluhovat také obvod od FTDI. Do vnitřní paměti obvodu je nutné zapsat nastavení, aby plnil požadované funkce. Z tohoto důvodu bude software rozdělen na dvě samostatné části – konfigurační (viz kapitola 5.1 a 5.2) a ovládací aplikaci (viz kapitola 5.3 a 5.4).

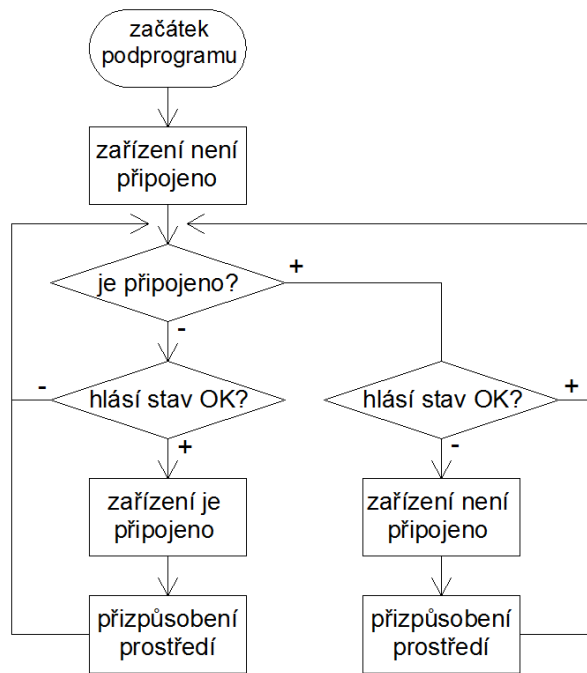
První část bude sloužit pouze k nastavení vnitřní paměti obvodu FT232RL po vytvoření přípravku. Druhá část bude poskytovat všechny ostatní funkce.

### **5.1 Návrh konfigurační aplikace**

Tato část obslužného softwaru bude vybavena grafickým prostředím, kde základními objekty budou ovládací prvky, umístěné k levému okraji okna a tabulka, představující obsah interní paměti. Program bude zjišťovat stav přípravku (připojen a nepřipojen), umožňovat zadání volitelných položek (popis, sériové číslo a odebíraný proud, dle dokumentace FT232RL (8)) a samozřejmě zápis a čtení celého nastavení paměti.

#### **5.1.1 Detekce připojení přípravku**

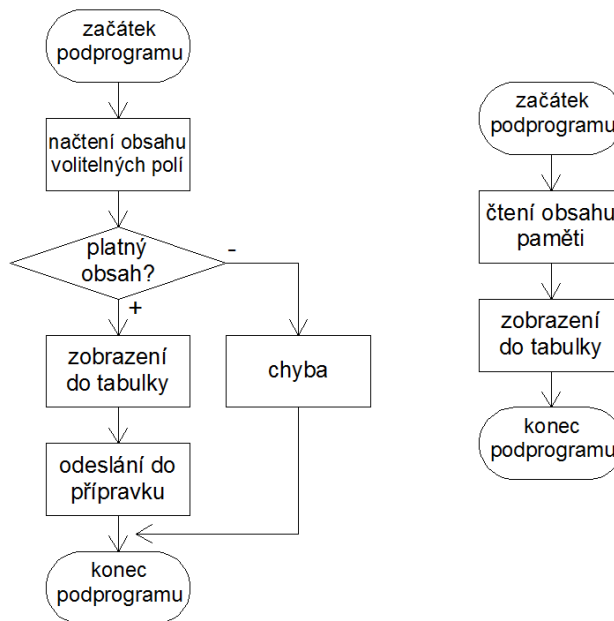
Po spuštění aplikace se začne vykonávat podprogram v intervalu do jedné sekundy. Na začátku bude vždy stav při nepřipojeném zařízení. Následně se bude v daném intervalu zjišťovat rozdíl stavu v programu a stavu, který hlásí přípravek. Při rozdílném stavu se program přizpůsobí. Základní princip činnosti je na následujícím obrázku.



Obrázek 16 Návrh základního principu činnosti při detekci přípravku

### 5.1.2 Zápis a čtení nastavení

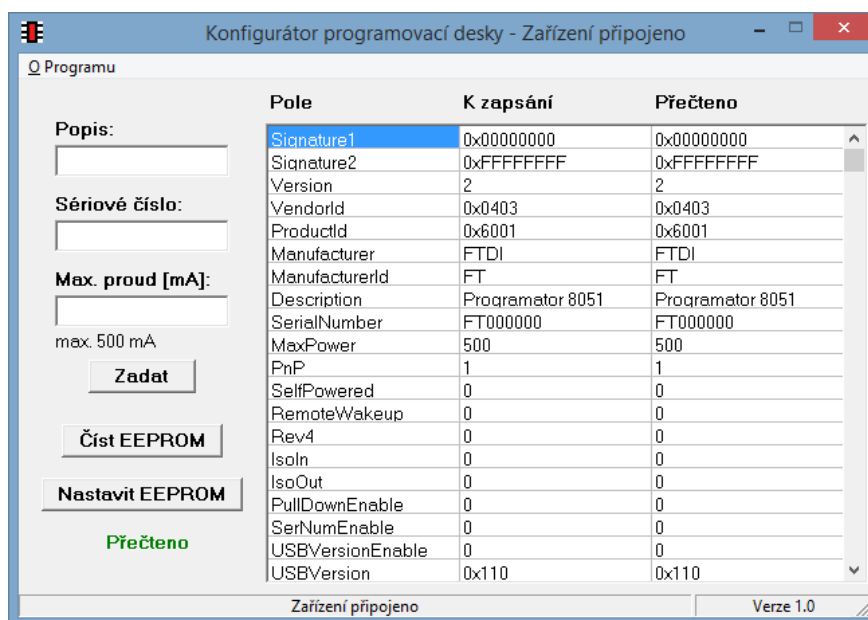
Zapisování a vyčítání paměti bude probíhat podle následujícího obrázku. Vstup bude zajištěn pomocí třech polí, každé pro jednu volitelnou položku. Následně se ověří platnost zadaných textů/čísel a zobrazí se do souhrnné tabulky, reprezentující celou paměť. Prostřednictvím ovládacích prvků bude možno zahájit zápis/čtení z/do tabulky.



Obrázek 17 Návrh postupu při zápisu a čtení nastavení přípravku

## 5.2 Realizace konfigurační aplikace

Předchozí návrh byl použit pro základní strukturu programu s názvem „Konfigurator programovací desky“. Vytvoření proběhlo v prostředí Delphi 7 a grafické provedení hlavního okna aplikace je vyobrazeno na následujícím obrázku.



Obrázek 18 Náhled grafického prostředí Konfiguratoru

Po spuštění se provádí detekce připojení zařízení každých 500 ms. Stav je zobrazován na liště u dolního kraje a v názvu okna. Ovládací prvky jsou nejdříve nedostupné a aktivují se až po připojení zařízení, protože by docházelo k chybě USB ovladačů.

Program obsahuje v levé části tři pole pro zadání volitelných položek, tlačítko „Zadat“ pro potvrzení těchto polí, „Číst EEPROM“ pro načtení hodnot z interní paměti FT232RL do posledního sloupce v tabulce a „Nastavit EEPROM“ pro zápis hodnot z druhého sloupce. Sloupec „Pole“ obsahuje identifikaci jednotlivých polí dle dokumentace od výrobce integrovaného obvodu (8). Informace o poslední provedené činnosti se zobrazuje pod tlačítky.

K realizaci softwaru je využita knihovna od FTDI (13), usnadňující použití nutných USB ovladačů (14) k obvodu FT232RL a připravená do prostředí Delphi. Celý zdrojový kód je uveden v příloze D a výsledný program je v příloze E.



### 5.3 Návrh ovládacího programu

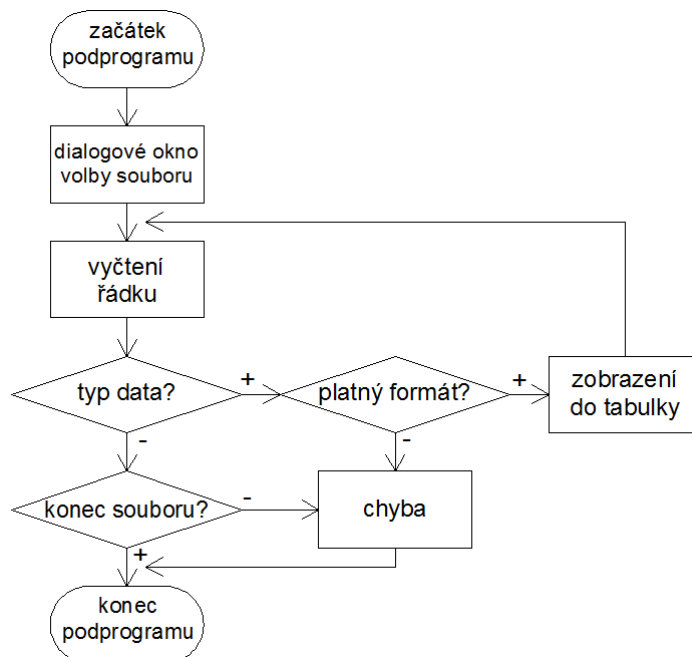
Druhá aplikace bude také vybavena grafickým prostředím s ovládacími prvky při levém kraji okna. V pravé části bude umístěna tabulka, která bude reprezentovat paměť mikrokontroléru. Opět bude probíhat zjišťování stavu přípravku (připojen a nepřipojen). Program bude umožňovat načtení souboru ve formátu Intel HEX (4), vymazání paměti mikrokontroléru, zápis obsahu z načteného souboru a ověření tohoto zápisu. Dále bude k základním funkcím patřit nastavení zámků.

#### 5.3.1 Detekce připojení přípravku

Zjišťování stavu přípravku bude probíhat stejně jako v konfigurační aplikaci. Na tuto funkci může být navázáno ověřování konfigurace hardwaru, například čtení typu mikrokontroléru.

#### 5.3.2 Načtení souboru

Import souboru do paměti aplikace bude umožněn přes dialogové okno, kde si uživatel zvolí cestu. Následně bude obsah postupně vyčítán po řádcích s platnými záznamy, které se importují do tabulky. Návrh postupu je na následujícím obrázku.

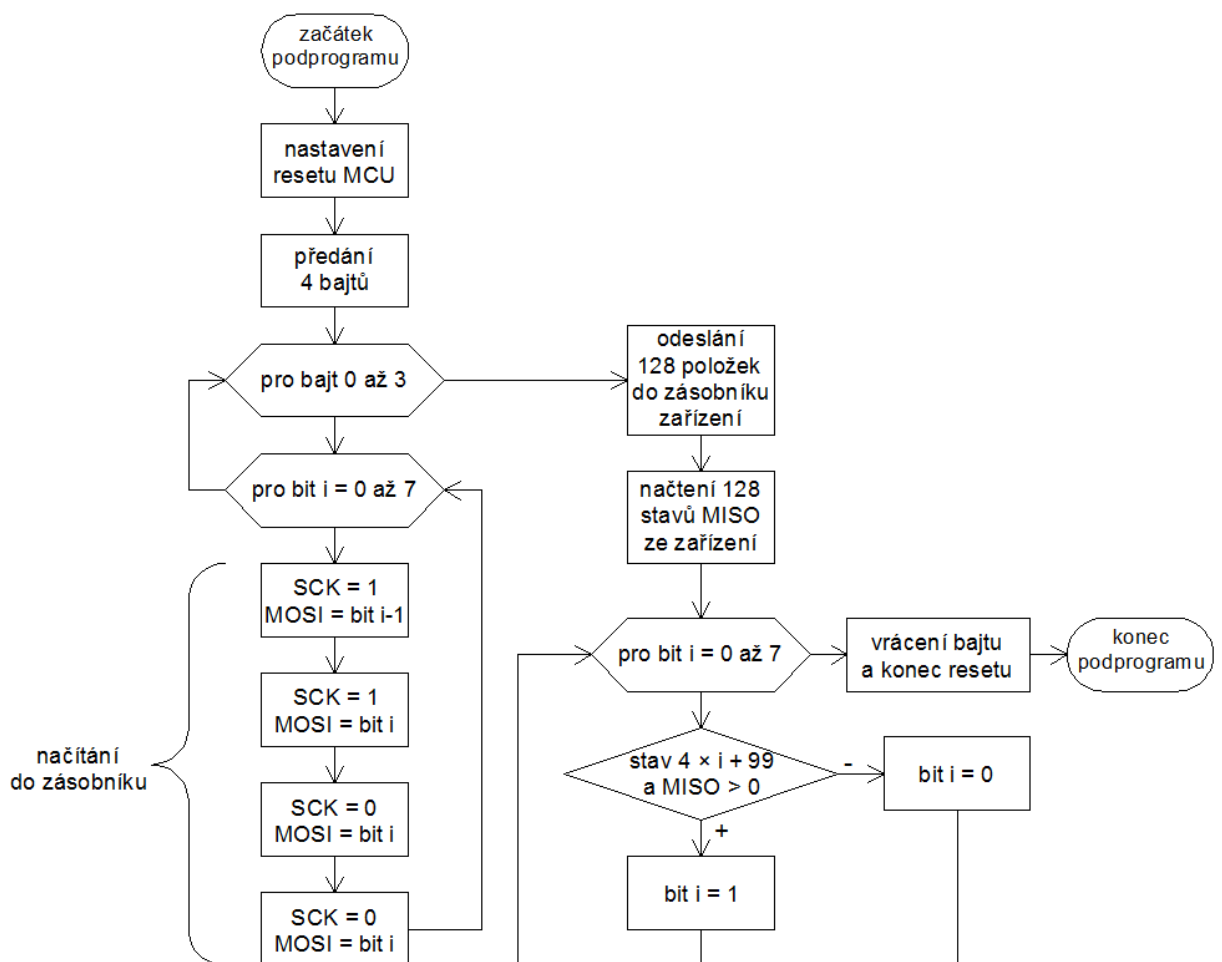


Obrázek 19 Návrh postupu při načítání souboru do softwaru

### 5.3.3 Vytvoření komunikace s mikrokontrolérem

Základní problém je vytvořit SPI komunikaci na úrovni softwaru. K dispozici jsou 4 vývody, které jsou propojené s mikrokontrolérem na MISO, MOSI, SCK a RST. Nastavování a vyčítání stavů na obvodu FT232RL probíhá přes zásobník a při nastavení se zároveň zaznamená stav na vstupu.

Podprogram dostane 4 bajty k odeslání a každý bit postupně navzorkuje do čtyř stavů se signálem SCK. Vznikne tak celkem 128 stavů do zásobníku, které odešle do zařízení. Zároveň přijme 128 stavů ze vstupu MISO, ze kterých vyjme poslední že čtyř bajtů, rozložených stejným principem. Tento přijatý bajt bude návratová hodnota. Během komunikace se drží stav nulování. Postup vychází z použití 4 bajtů v komunikačním protokolu, uvedeném v dokumentaci mikrokontroléru AT89S8253 (2).



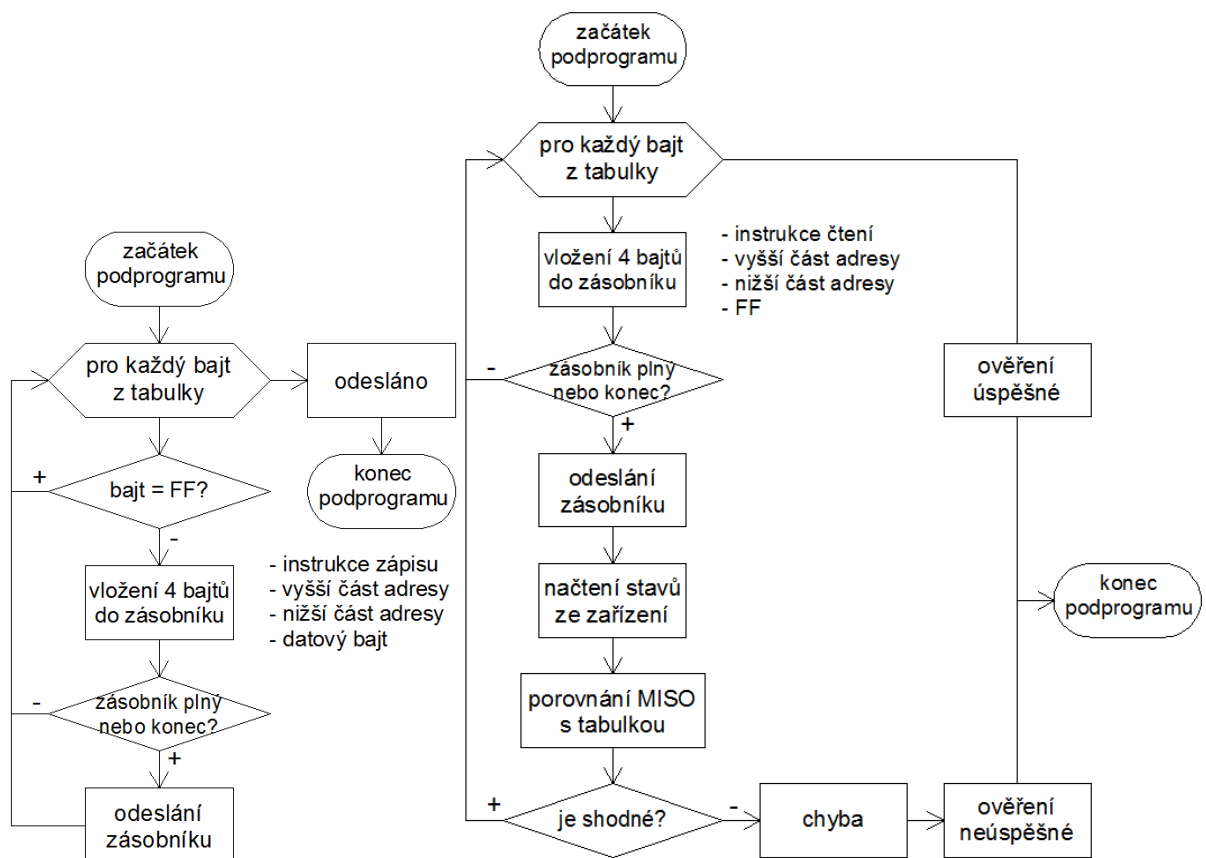
Obrázek 20 Návrh komunikace s mikrokontrolérem

### 5.3.4 Zápis a ověření

Návrh zápisu a čtení používá upravenou verzi předchozího návrhu komunikace. Odesíláno a přijímáno bude větší množství dat, a proto je výhodnější nejdříve naplnit celý zásobník než odesílat každé 4 bajty odděleně.

V průběhu zápisu se postupně načítají bloky paměti z tabulky (v tomto případě bajty) a vyfiltrují se prázdné („FF“). Ostatní se hromadí do zásobníku, opatřené jejich adresou a instrukcí zápisu. Při naplnění zásobníku se jeho obsah odešle do zařízení.

Ověřování je rozšířeno o čtení vstupu a jako odchozí bajty vkládá prázdné (FF). Po odeslání zásobníku podprogram přijme stejně velký zásobník příchozích stavů ze vstupu MISO. Tuto novou sadu stavů rozebere dle návrhu komunikace s mikrokontrolérem. Následující obrázky obsahují výše popsaný průběh zápisu (vlevo) a průběh čtení (vpravo).

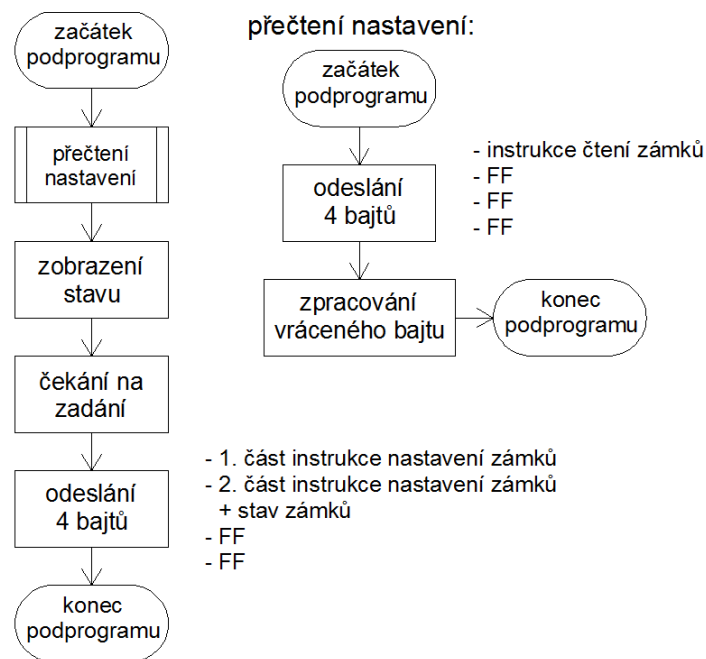


Obrázek 21 Návrh zápisu a ověření v obslužné aplikaci

### 5.3.5 Zápis a čtení zámků

Tato část by měla být v grafickém prostředí oddělena. Bude obsahovat prvky pro zobrazení a nastavení stavu zámků. K odeslání/přijímání bude použit již uvedený návrh komunikace s mikrokontrolérem, vhodný pro 4bajtový protokol.

Prostředí aplikace by také mělo zajistit, aby se po vstupu do této části provedlo čtení aktuálního nastavení (uživatel se bude rozhodovat na základě aktuálního stavu zámků). Postup při zápisu a čtení je na následujícím obrázku.



Obrázek 22 Návrh zápisu a čtení zámků

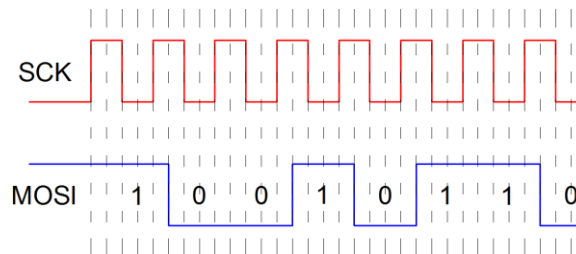
### 5.3.6 Zpracování jednoho bajtu ze souboru až do mikrokontroléru

Pro ilustraci průběhu zpracování jednoho bajtu ze souboru až na výstup jsou zobrazeny jednotlivé fáze. V načítaném souboru se na jednom řádku nachází záznam dle následujícího obrázku. Čísla jsou v hexadecimálním tvaru. Formát je ve tvaru: „:“ – začátek řádku, „01“ – počet bajtů, „00F0“ – počáteční adresa, „96“ – samotný bajt, „79“ – kontrolní součet. Tento záznam je následně vyčten a zobrazen do tabulky k příslušné adrese, jak je ukázáno na obrázku.

:0100F0009679 00F0 96

Obrázek 23 Umístění bajtu v souboru a v softwaru

Dále je bajt opatřen instrukcí zápisu a společně s jeho adresou vložen do 4bajtového protokolu ve tvaru: „XX 00 F0 96“, kde „XX“ je instrukce zápisu. Tyto čtyři bajty jsou následně rozebrány na jednotlivé bity. Každý bit je navzorkován na čtyři stavy výstupu MOSI a hodinového signálu SCK. Následující obrázek obsahuje pouze odeslání načteného bajtu (96h = 1001 0110b).

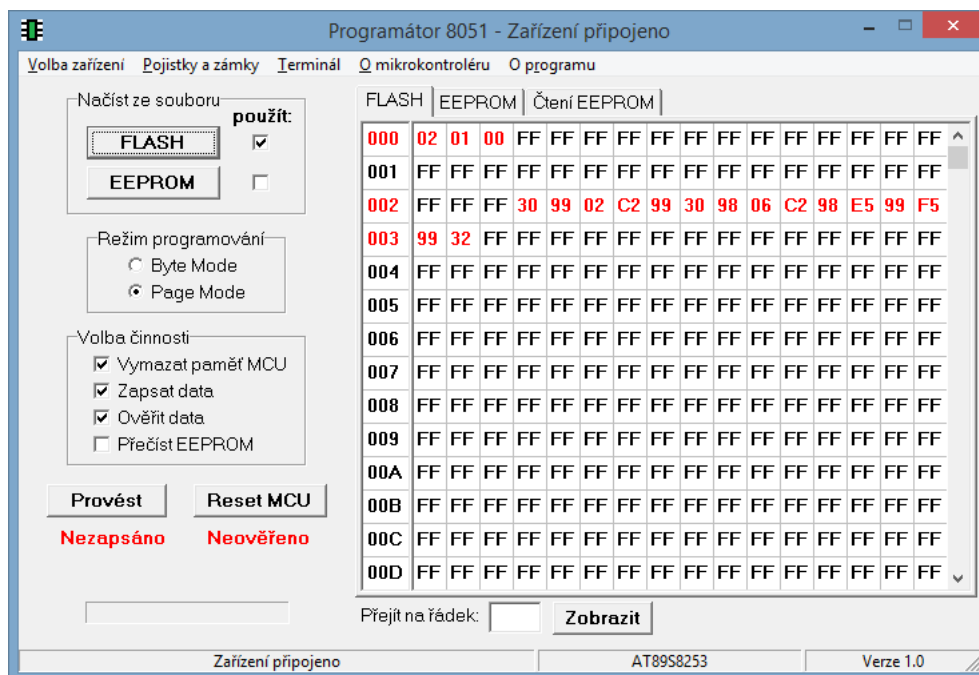


**Obrázek 24** Vzorkování komunikace s mikrokontrolérem

Svislé čáry na obrázku značí jeden stav výstupu. Celkem je tak na přenos jednoho bajtu potřeba 32 stavů, pro celý protokol pak 128.

## 5.4 Realizace ovládacího programu

Předchozí návrh byl použit jako základ programu s názvem „Programátor 8051“. Tento program byl vytvořen, stejně jako první část, v prostředí Delphi 7 a grafické provedení je vyobrazeno na následujícím obrázku.



**Obrázek 25** Náhled grafického prostředí Programátoru

Stejně jako u konfiguračního programu se po spuštění provádí detekce připojení zařízení každých 500 ms. Stav je také zobrazován na liště u dolního kraje a v názvu okna. Ovládací prvky jsou opět nejdříve nedostupné a po připojení zařízení se aktivují. Navíc se zobrazí/skryjí části, které nejsou dostupné u detekovaného mikrokontroléru. Automatickou volbu lze následně ručně změnit na horním panelu.

V levé části je umístěno načítání souboru, rozdělené podle cílové paměti. Ovládací prvky pro nastavení režimu programování a pro volbu úkonů jsou umístěny níže. Pod těmito objekty se nachází tlačítko pro zahájení činnosti a tlačítko pro nulování mikrokontroléru. Aktuální stav se pak zobrazí pod těmito tlačítky jak textově, tak i graficky naplňováním lišty.

Převážnou část okna vyplňuje tabulka, reprezentující obsah paměti mikrokontroléru. Jednotlivá umístění lze prohlížet pomocí záložek. Hodnoty načtené do tabulek jsou zvýrazněny červeně. Zároveň se označí i použití 8bitového bloku adres při levém okraji tabulky. Pro snadné nalezení konkrétní adresy je přítomno pole pro přímé zadání požadovaného řádku ve tvaru dle tabulky.

Další funkce programu jsou přístupné pomocí hlavního panelu u horního kraje okna. Dostupná je volba mikrokontroléru, správa pojistek a zámků, terminál sériové linky, informace o mikrokontroléru a o samotné aplikaci. K dodržení návrhu je důležitá správa pojistek a zámků. Grafické provedení je na následujícím obrázku.



**Obrázek 26** Okno správy pojistek a zámků

Zobrazené okno je viditelně rozděleno do dvou částí s vlastními tlačítky pro čtení a zápis, pod kterými je informace o poslední provedené činnosti. Levá část je určena pro pojistky a pravá pro zámky. Aktuální stav se po přečtení zobrazí barevně a zaškrtnutím příslušného pole. Nastavení zaškrtnutých položek se provede tlačítky „Zapsat“. Význam jednotlivých položek je uveden v dokumentaci příslušného mikrokontroléru (2).

Podrobnější popis použití programů je v Manuálu pro práci s přípravkem. K realizaci softwaru je opět využita knihovna od FTDI (13), usnadňující použití nutných USB ovladačů (14) k obvodu FT232RL a připravená do prostředí Delphi. Celý zdrojový kód je opět uveden v příloze D a výsledný program je v příloze E.

## **6 Ověření vlastností přípravku**

Výsledný přípravek musí pro správnou funkci splňovat parametry stanovené v dokumentaci jednotlivých integrovaných obvodů a také navržené hodnoty pro jejich spolupráci. Tyto parametry lze rozdělit na ustálené (statické) a na měnící se s časem (dynamické). Při měření byly otestovány důležité vlastnosti pro základní činnost přípravku.

### **6.1 Měření statických parametrů**

#### **6.1.1 Napájení obvodu FT232RL a mikrokontroléru**

Prototyp byl připojen do USB s poskytovaným napětím 5,06 V. U obvodu FT232RL bylo naměřeno napětí 5,03 V a u mikrokontroléru 5,01 V, což jsou zanedbatelné rozdíly a hodnoty splňují požadovaný rozsah v dokumentaci jak k obvodu FTDI (8), tak k mikrokontroléru (2).

Otestována byla také ochrana vratnými pojistkami. Na místě připojení externího zařízení byly na okamžik propojeny vývody pro napájení. Při tomto spojení došlo k poklesu napětí funkcí vratných pojistek a reakcí na tento pokles bylo zastavení činnosti obvodu FT232RL a uzavření tranzistoru pro řízení spotřeby. V průběhu testu nedošlo k žádnému poškození a přípravek je tedy dostatečně chráněn.

#### **6.1.2 Převodník napěťových úrovní MAX232**

V dokumentaci u obvodu MAX232 (11) je uvedeno napětí na výstupu 7,5 V až 13,2 V. V klidovém stavu bylo na výstupu naměřeno 9,51 V a tato hodnota se nachází v uvedeném intervalu.

### **6.2 Měření dynamických parametrů**

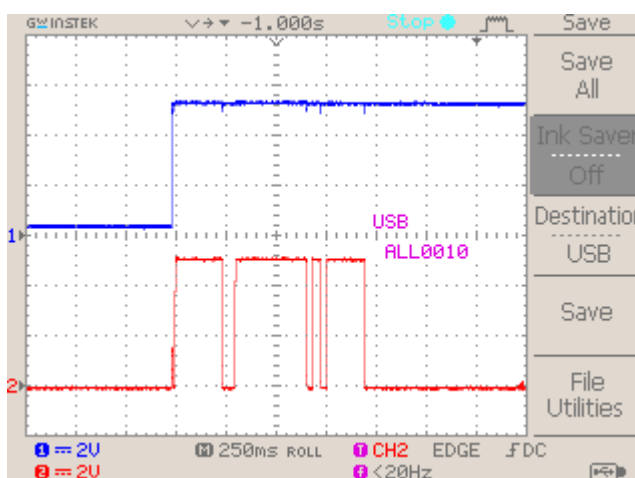
Z dynamických parametrů je významný průběh ovládní spotřeby přípravku a na to navazující napětí ve zbylé části desky. Dále je důležité nulování mikrokontroléru jak pomocí obslužného softwaru, tak při nabíjení kondenzátoru po aktivaci napájení. Nejdůležitější je průběh klíčové funkce – programování mikrokontroléru. Jednotlivá měření byla provedena na osciloskopu s možností uložení aktuální obrazovky.



## 6.2.1 Připojení přípravku do USB a regulace spotřeby

Obvod FT232RL provádí po připojení do USB inicializaci své činnosti a jeho výstupy nemusejí ihned plnit nastavenou funkci. Příkladem je vývod s nastavením pro regulaci spotřeby, na kterém se má objevit logická nula (napětí 0 V).

Následující obrázek obsahuje průběh napětí na konektoru USB (modrá barva) a výstup pro tranzistor na řízení spotřeby (červená barva). Z průběhu je patrné, že požadovaná funkce je dostupná až po přibližně jedné sekundě. Tento jev však nemá vliv na další funkce přípravku a tak připojení proběhne v pořádku.



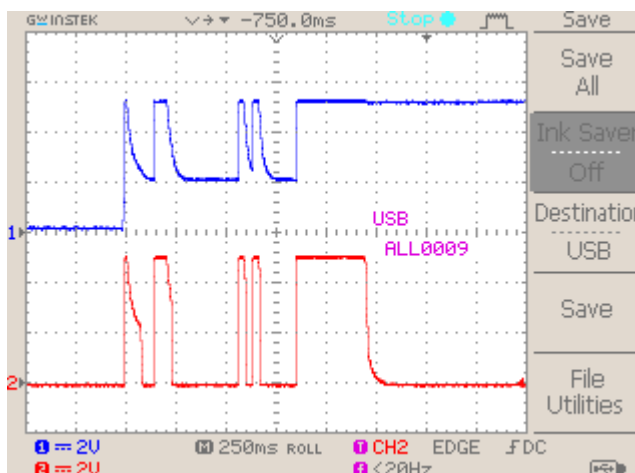
Obrázek 27 Průběh napětí po připojení přípravku

## 6.2.2 Nulování mikrokontroléru po startu

Pro správnou činnost mikrokontroléru je nutné provést nulování po aktivaci napájení, aby se zapsaný program začal vykonávat od začátku. Toho se dosáhne logickou jedničkou na příslušném vstupu po dobu alespoň dvou strojových cyklů (2). Tranzistor v nulovacím obvodu byl opatřen RC článkem, který by měl zajistit držení logické úrovně po dobu větší jak 100 ms. Tento čas je o několik řádů vyšší, než je požadováno, a tak bezpečně provede požadovaný úkon.

Následující obrázek obsahuje průběh napětí pro mikrokontrolér (modrá barva) a průběh napětí na vstupu pro nulovací signál (červená barva). V předchozím měření bylo zjištěno, že napětí není ustáleno ihned po připojení, a proto se doba určuje až od okamžiku stabilního

napětí. Na obrázku je vidět držení logické jedničky po dobu přibližně 350 ms, což splňuje požadovaný čas.



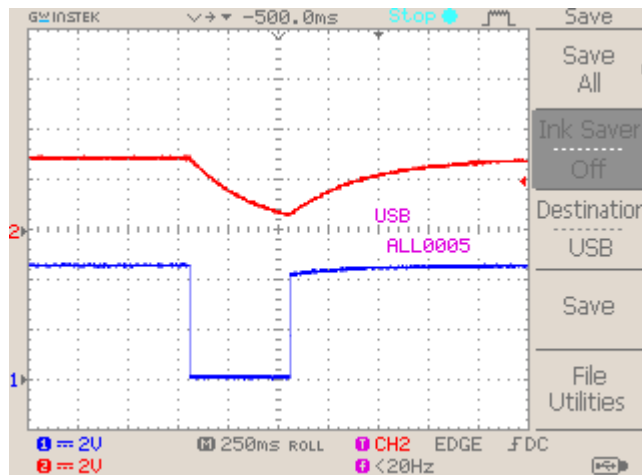
Obrázek 28 Průběh nulování mikrokontroléru po připojení přípravku

### 6.2.3 Nulování mikrokontroléru obslužným programem

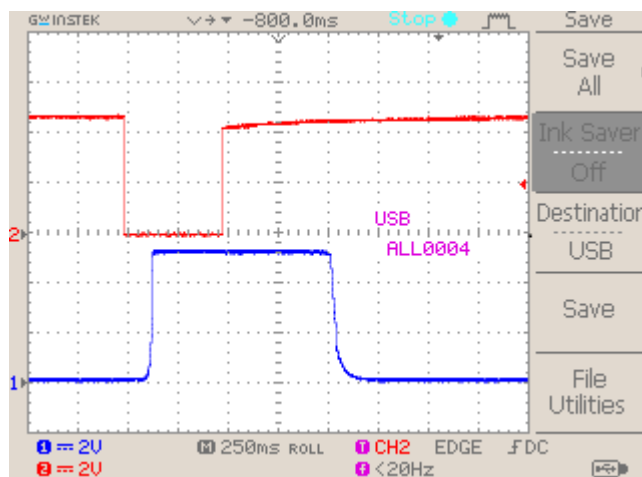
Kromě spínacího tlačítka, které přímo přivádí napětí na vstup mikrokontroléru, lze provést nulování také v obslužném softwaru. Výstupu z obvodu FT232RL, který slouží pro držení mikrokontroléru ve stavu resetu po dobu programování, je možno použít pro krátký reset bez zápisu či čtení paměti. Při změně logické hodnoty na výstupu je však třeba počítat s prodlením při vybíjení/nabíjení kapacity pro nulování po připojení. Dobu změny logické úrovně je tedy nutné navrhnout tak, aby se použitá kapacita dostatečně vybila pro otevření tranzistoru, a zároveň aby nabíjení tuto dobu zbytečně neprodloužilo. V obslužném softwaru byla nastavena hodnota 500 ms.

Obrázek 29 obsahuje průběh napětí na RC članku (červeně) a logickou úroveň na výstupu obvodu FT232RL (modře). Za dobu 500 ms můžeme pozorovat téměř úplné vybití kapacity. Můžeme tedy říct, že vždy dojde k otevření tranzistoru.

Na obrázku 30 je pro porovnání opět logická úroveň na výstupu z obvodu od FTDI (červeně) a také výsledný průběh na vstupu mikrokontroléru (modře). Při porovnání s prvním obrázkem je vidět prodleva mezi otevřením tranzistoru a začátkem vybíjení přibližně 130 ms. Opětovné zavření je pak prodlouženo o přibližně 500 ms. Výsledná doba resetu je tedy 870 ms, což je sice znatelné prodloužení, ale nulování mikrokontroléru je zaručeno a delší čas nijak nekomplikuje používání přípravku.



**Obrázek 29** Průběh nulování mikrokontroléru obslužným softwarem – ovládání tranzistoru



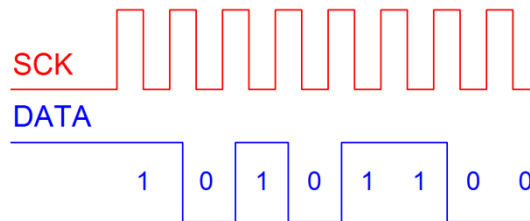
**Obrázek 30** Průběh nulování mikrokontroléru obslužným softwarem – výsledný průběh

#### 6.2.4 Průběh odesílání jednoho bajtu dat při programování

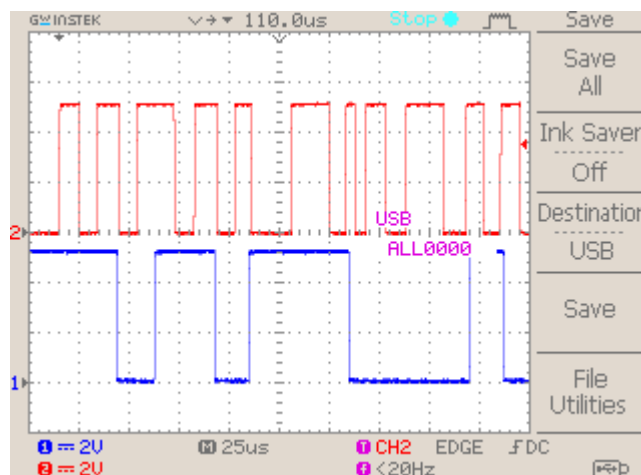
Komunikace s mikrokontrolérem musí probíhat podle parametrů určených v jeho dokumentaci (2). Jednou podmínkou je, že délka pulzu hodinového signálu musí být větší než 8 period oscilátoru. Tento čas při použití 12 MHz krystalu je přibližně 0,67  $\mu\text{s}$  a perioda změny úrovně signálu SCK je nastavena na více než čtyřnásobnou hodnotu, přibližně 2,86  $\mu\text{s}$ . Ostatní parametry jsou při tomto času mezi každou změnou zanedbatelné.

Na obrázku 31 je nakreslen předpokládaný souběh logických hodnot signálu SCK a jednoho datového vodiče. Odesílán je bajt „1010 1100“, což je část instrukce pro zahájení programování.

Obrázek 32 obsahuje zachycený průběh přenosu odesílaného bajtu. Jak je patrné, přenosová rychlost není stálá. Tento problém ale vzniká přímo v obvodu FT232RL a tak jeho odstranění není možné. Komunikace však i s touto vlastností probíhá v pořádku, protože přenos jednotlivých bitů je časován signálem SCK a přenášené bity souhlasí. Problém by nastal, pokud by nejmenší doba jedné úrovně klesla pod hodnotu  $1,33 \mu\text{s}$ . V průběhu měření byl zaznamenán rozsah  $2,9 \mu\text{s}$  až  $18,5 \mu\text{s}$ , což je přijatelné. Komunikace tedy probíhá v pořádku.



Obrázek 31 Předpokládaný průběh přenosu jednoho bajtu při programování



Obrázek 32 Skutečný průběh přenosu jednoho bajtu při programování

## 7 Ověření funkcí softwaru

Testování obslužného softwaru se skládá z ověření činnosti konfigurační a ovládací aplikace. Postupně byly vyzkoušeny všechny funkce, které je nutno projít od vytvoření přípravku po naprogramování a případné uzamčení mikrokontroléru.

### 7.1 Zkouška funkcí konfiguračního programu

Po spuštění Konfiguratoru bylo grafické prostředí neaktivní. Následně program detekoval připojené zařízení a ovládací prvky se zpřístupnily. Do polí byl zadán popis „Programator 8051“, sériové číslo „FT000000“ a maximální proud 500 mA. Kliknutím na tlačítko „Zadat“ se obsah polí úspěšně načel do tabulky. Po použití tlačítka „Nastavit EEPROM“ program krátce pracoval a jako výsledek úspěšného zápisu zobrazil pod tlačítka text „Nastaveno“.



Obrázek 33 Zkouška Konfiguratoru – úspěšné nastavení

Ověření zapsaných hodnot proběhlo následným přečtením paměti pomocí tlačítka „Číst EEPROM“ a údaje ve sloupcích „Přečteno“ a „K zapsání“ se shodovaly. Čtení a zápis probíhá nezávisle a tak lze prohlásit, že nastavení proběhlo úspěšně.

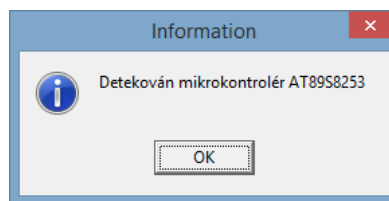


| Parameter    | Value            | Value            |
|--------------|------------------|------------------|
| Description  | Programator 8051 | Programator 8051 |
| SerialNumber | FT000000         | FT000000         |
| MaxPower     | 500              | 500              |

Obrázek 34 Zkouška Konfiguratoru – ověření zápisu

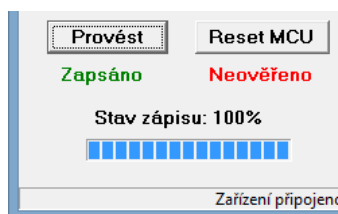
### 7.2 Zkouška funkcí ovládacího programu

Před použitím aplikace Programátor bylo nutné ukončit konfigurační aplikaci (přípravek pracuje správně pouze s jednou spuštěnou aplikací). Po spuštění byly ovládací prvky nepřístupné a po detekci připojeného přípravku se aktivovaly. Následně proběhlo zjištění vloženého mikrokontroléru jako AT89S8253 (shodné se skutečně použitým).



**Obrázek 35** Zkouška Programátoru – úspěšná detekce mikrokontroléru

Pro testování byl načten testovací program pro mikrokontrolér z původního souboru s příponou \*.hex, přístupný také pod tlačítkem v okně informací „O mikrokontroléru“. Obsah tabulky se shodoval s navrženým programem. Nastavení bylo ponecháno výchozí – Page Mode, vymazání paměti, zapsání a ověření. Následně bylo použito tlačítko „Provést“ a Programátor začal vykonávat zvolené činnosti. Nejprve se provedlo vymazání paměti a zápis. Stav se průběžně zobrazoval v dolní části.



**Obrázek 36** Zkouška Programátoru – stav zápisu

Zápis podle programu proběhl úspěšně. Poté následovalo ověření zapsaných hodnot a průběh bylo opět možno vidět v procentech a na naplňující se liště.

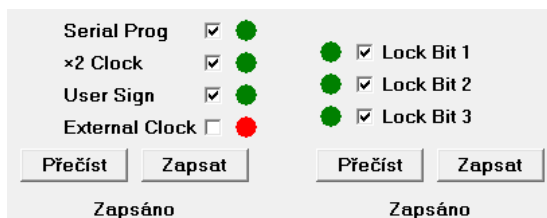


**Obrázek 37** Zkouška Programátoru – stav ověření

Se zeleným textem „Ověřeno“ byl úspěšně zapsán a ověřen nový program do mikrokontroléru. Dalším ověřením úspěchu bylo viditelné zobrazení „01010101“ na LED (hodnota 55h).

Do základní obsluhy patří také nastavení zámek v mikrokontroléru. K tomu slouží podokno „Pojistky a zámky“, dostupné na hlavním panelu při horním kraji okna Programátoru.

Po otevření okna došlo k přečtení aktuálního stavu zámků a pojistek (vložený mikrokontrolér obsahoval obě části) a zobrazení k příslušným položkám. Ověření funkce bylo provedeno zápisem všech zámků (lock bitů) a jedné pojistky („×2 Clock“). Zapsání proběhlo použitím tlačítka „Zapsat“ u obou částí.



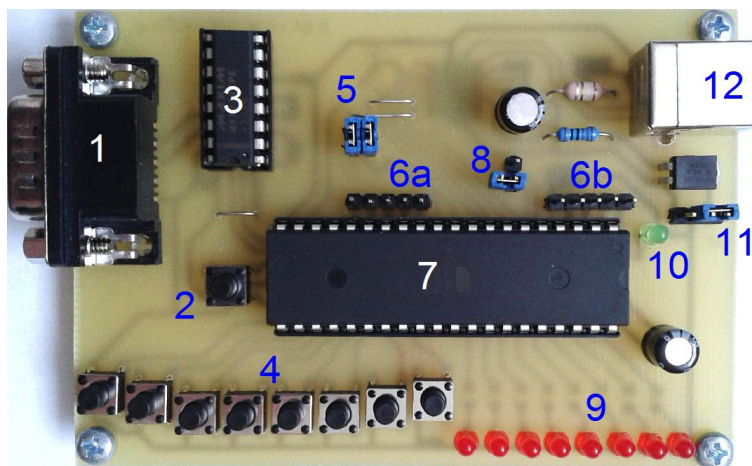
**Obrázek 38** Zkouška Programátoru – nastavení zámků

Po opětovném přečtení všech pojistek a zámků byly žádané položky vráceny jako nastavené. Čtení a zápis probíhá rozdílnými instrukcemi a prostřednictvím nezávislých částí kódu programu, takže lze prohlásit, že nastavení proběhlo úspěšně.

## 8 Manuál pro práci s přípravkem

Pokud není řečeno jinak, slovo „program“ v tomto návodu znamená soubor hodnot, určených k zápisu nebo k ověření hodnot již zapsaných do paměti mikrokontroléru.

Části přípravku:



Obrázek 39 Očíslované části přípravku

### 8.1 Popis přípravku

**1: konektor DE-9 (samec) sériové linky standardu RS-232.** Ke komunikaci jsou využity vodiče RxD, TxD a Gnd. Řízení přenosu dalšími vodiči v rámci konektoru je volitelné, pouze opakují příchozí signály.

**2: tlačítko pro reset mikrokontroléru.** Stiskem tohoto tlačítka se provede nulování mikrokontroléru.

**3: MAX232, převodník napět'ových úrovní TTL na RS-232.** Tento integrovaný obvod slouží k převodu logických úrovní 5 V na  $\pm 12$  V a je využíván pro sériovou linku. Umístění do patice umožňuje snadnou výměnu při poruše.

**4: spínací tlačítka, připojená k portu P2 na mikrokontroléru.** Tato tlačítka obsluhují celý port P2 a představují bity zleva od P2.0 do P2.7.

**5: propojka J3.** Tímto místem prochází sériová linka od terminálu v obslužném softwaru k mikrokontroléru. Základní konfigurace je dle prvního obrázku, obě propojky svisle. Možné



konfigurace jsou: propojení terminálu obslužného softwaru s mikrokontrolérem, opakování dat zpět do terminálu, opakování dat zpět do mikrokontroléru. Poslední dvě možnosti jsou kombinovatelné.



**Obrázek 40** Propojka J3 – propojení terminálu s mikrokontrolérem



**Obrázek 41** Propojka J3 – opakovač do terminálu



**Obrázek 42** Propojka J3 – opakovač do mikrokontroléru

**6: 8× datový vývod, 1× Vcc 5 V, 1× Gnd.** Tyto vývody slouží pro připojení externího zařízení k přípravku. Uspořádání je dle následujícího obrázku a odběr proudu je omezen na 300 mA.



**Obrázek 43** Vývody přípravku pro externí zařízení

**7: mikrokontrolér AT89S8253.** Zde je do patice umístěn programovaný a testovaný mikrokontrolér. Kompatibilní mikrokontroléry jsou: AT89S51, AT89S52 a AT89S8253.

**8: propojka J2.** Základní poloha propojky je rozpojeno. Toto propojení slouží k nastavení mikrokontroléru do „slave“ módu při využívání SPI komunikace nebo pro obecné nastavení logické nuly na vstupu P1.4 (2).

**9: LED, připojené k portu P0 na mikrokontroléru.** Tyto diody zobrazují stav na celém portu P0 a reprezentují bity zleva od P0.7 do P0.0.

**10: LED, indikující napájení mikrokontroléru.** Svícení této diody znamená aktivní napájení mikrokontroléru.

**11: propojka J1.** Tato propojka slouží k vynucení napájení mikrokontroléru v režimu snížené spotřeby, kdy je sice přípravek připojen, ale je neaktivní. Funkce má smysl při použití napájení USB z externího zdroje, který nevyužívá datové vodiče. Přípravek pak lze provozovat pro testování mikrokontroléru bez použití počítače.



**Obrázek 44** Propojka J1 – základní poloha



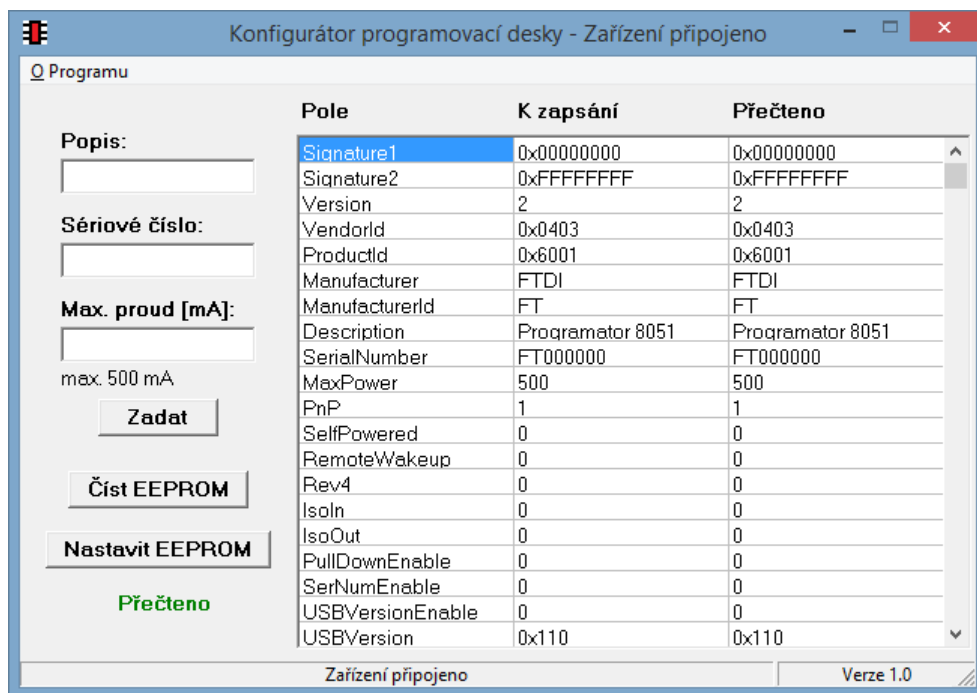
**Obrázek 45** Propojka J1 – vynucení napájení mikrokontroléru

**12: konektor USB typu B (samice).** Tento konektor slouží k propojení přípravku s počítačem pomocí USB kabelu typu A-B (standardní USB kabel, rozšířený zejména pro připojení tiskáren).

## 8.2 Připojení k počítači a konfigurace přípravku

Přípravek je otestován na operačních systémech Windows XP/Vista/7/8. K použití je nutné mít nainstalované D2XX ovladače od FTDI Chip (14), dostupné na oficiálních stránkách. Kopie těchto ovladačů je také v příloze F.

Následně připojíme přípravek pomocí USB A-B kabelu k počítači. Propojovací kabel by měl být součástí přípravku. Pokud se jedná o první připojení po vytvoření, nebo přípravek ještě nebyl nakonfigurován, je nutné nejprve spustit aplikaci Konfigurator. Po zobrazení grafického prostředí lze zadat vlastní popis přípravku, sériové číslo a maximální proud, který bude odebírán z USB (toto číslo by mělo být v rozsahu 20 až 500 mA a závisí zejména na odběru dalšího zařízení, připojeného k přípravku). Popis se doporučuje psát bez diakritiky. Po zadání se provede zápis hodnot do přípravku tlačítkem „Nastavit EEPROM“.

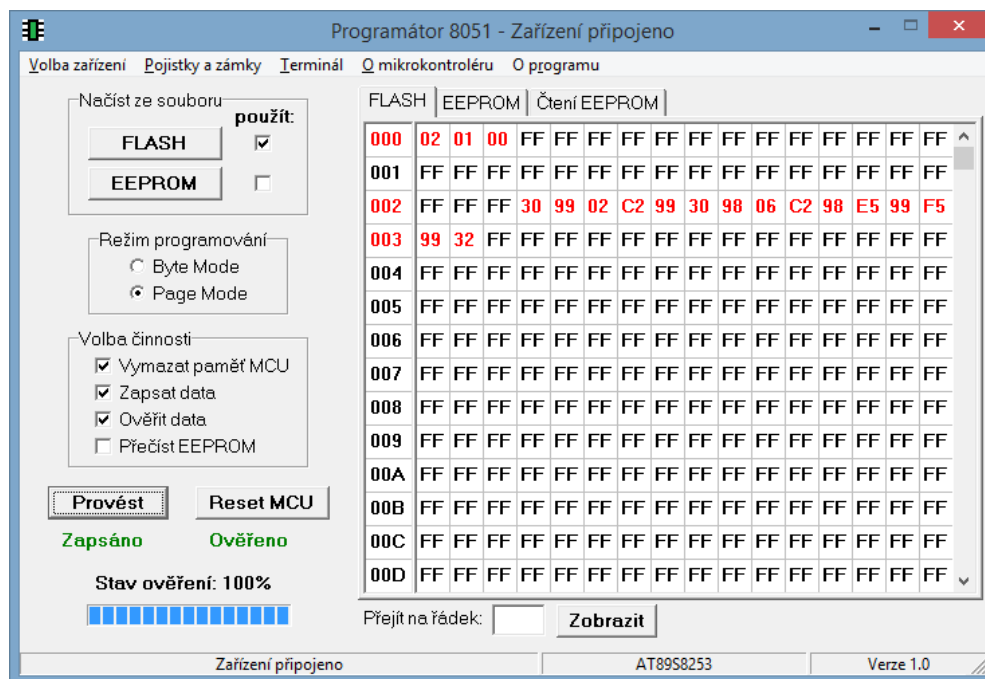


**Obrázek 46** Grafické prostředí programu Konfigurátor

Program Konfigurátor umožňuje zápis a čtení obsahu interní paměti EEPROM, umístěné v integrovaném obvodu pro obsluhu USB. V levé části grafického rozhraní se nachází pole pro zadávání volitelných hodnot, které se po stisknutí tlačítka „Zadat“ zapíší do tabulky v pravé části. Tato tabulka obsahuje tři sloupce – jeden s názvy polí a dva s hodnotami. Sloupec nazvaný „K zapsání“ obsahuje hodnoty, které jsou pouze v paměti Konfigurátoru. Do sloupce „Přečteno“ se pak načítají hodnoty z přípravku. V levé dolní části okna jsou dále tlačítka „Nastavit EEPROM“ pro zápis a „Číst EEPROM“ pro načtení hodnot do zmíněného sloupce „Přečteno“ z paměti. Text pod těmito tlačítky informuje o poslední provedené činnosti Konfigurátoru.

### 8.3 Obslužný program

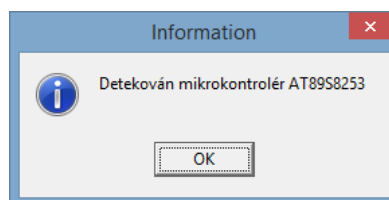
K obsluze přípravku slouží aplikace Programátor. Náhled grafického prostředí je na následujícím obrázku.



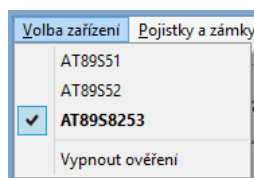
Obrázek 47 Grafické prostředí aplikace Programátor

### 8.3.1 Spuštění Programátoru

Po spuštění dojde k pokusu o zjištění vloženého mikrokontroléru. Kompatibilní jsou AT89S51, AT89S52 a AT89S8253 od firmy Atmel. Při úspěšné detekci se zobrazí dialogové okno se zjištěným typem. Prvky grafického prostředí se automaticky zobrazí/skryjí dle vybavenosti konkrétního mikrokontroléru. Volbu lze ručně změnit, ale při další činnosti se znovu ověřuje vybraný typ. Toto ověření lze vypnout (umístění této položky v hlavní nabídce je vidět na druhém obrázku níže), ale potom při špatné volbě může docházet k chybám a je na uživateli, aby toto zkontroloval.



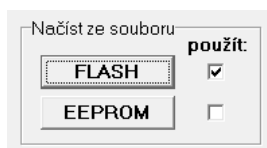
Obrázek 48 Programátor 8051 – dialogové okno detekce



Obrázek 49 Programátor 8051 – volba mikrokontroléru

### 8.3.2 Načtení souboru

Program v jazyku assembler nebo data, určená do paměti EEPROM mikrokontroléru, lze načíst ze souboru ve formátu Intel HEX (\*.hex) (4). Načítání je v levé části prostředí rozděleno do dvou tlačítek, jedno pro paměť FLASH (program) a druhé pro paměť EEPROM (data).



Obrázek 50 Programátor 8051 – načtení souboru

Záznamy ze po otevření souboru načtou do tabulky v pravé části grafického prostředí, kde má každá paměť vlastní záložku. Načtené hodnoty se zvýrazní červeně společně s adresou řádku, která představuje reálnou hexadecimální adresu bez nejnižšího bajtu (řádek obsahuje celý jeden bajt adres). Jejich použití pro další činnost lze určit zaškrtnutím příslušného pole vedle tlačítek (například stačí postupně načíst dva soubory po spuštění Programátoru a následně je střídavě používat bez nutnosti dalšího načítání). Před načtením nového souboru se všechny hodnoty z příslušné záložky nastaví na „FF“, aby nedocházelo k nechtěnému použití záznamů z více souborů.

Prohlížení každé záložky je usnadněno polem pro přímé zadání požadované adresy, umístěným pod tabulkou. Následně lze tuto adresu potvrdit klávesou „Enter“ nebo kliknutím na přilehlé tlačítko. Další možností pohybu v tabulce je použití kláves „Page Down“ a „Page Up“ či pomocí kolečka myši.

### 8.3.3 Zápis, ověření a čtení paměti mikrokontroléru

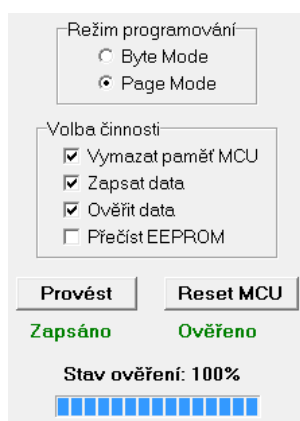
Režim komunikace a činnost Programátoru se nastaví v levé části grafického prostředí. Z režimů programování jsou dostupné „Byte Mode“ – odesílání a přijímání po bajtech a

„Page Mode“ – komunikace po stránkách. Při rozsáhlejšímu programu je rychlejší přenos po stránkách.

Z činností je možno jednotlivě vybrat vymazání paměti mikrokontroléru (FLASH i EEPROM, toto je nutno zvolit vždy před zápisem do již naprogramované paměti FLASH) a zapsání dat (zapiše obsah zobrazený v tabulkách v pravé části a dle polí „použit“ vedle tlačítek pro načtení souboru). Dále je možno ověřit shodu načtených hodnot ze souboru s hodnotami v mikrokontroléru (funkce je obdobná jako u zápisu) a přečíst obsah vnitřní paměti EEPROM (pro toto čtení slouží oddělená záložka v tabulkách). Více zvolených možností se provede v pořadí, v jakém jsou zobrazeny. Po spuštění je nastavena nejčastěji používaná konfigurace – komunikace po stránkách, vymazání paměti, zapsání dat a jejich ověření.

Zvolené úkony se provedou po kliknutí na tlačítko „Provést“. Stav je průběžně zobrazován ve spodní části jak graficky, tak i textově. Výsledek poslední provedené činnosti se objeví pod tlačítky (Zapsáno, Ověřeno, Přečteno, ...).

V této části se také nachází tlačítko pro nulování mikrokontroléru pomocí softwaru. Funkce je totožná s tlačítkem na přípravku.

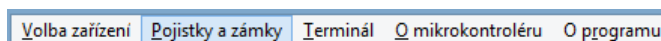


Obrázek 51 Programátor 8051 – nastavení a průběh činnosti

### 8.3.4 Nastavení zámků a pojistek

Nastavení zámků a vnitřní konfigurace mikrokontroléru lze provést v okně pojistek a zámků. Po otevření okna dojde k automatickému přečtení všech nastavení. Jejich aktuální stav je zobrazen barevně a zaškrtnutím příslušných polí. Zápis a další čtení lze provést pomocí tlačítek níže. Výsledek poslední činnosti je zobrazen textem pod tlačítky. Levá část nastavení,

viditelná na obrázku níže, je dostupná pouze u AT89S8253. Funkce jednotlivých možností jsou dány použitým mikrokontrolérem a úplný význam lze nalézt v jeho dokumentaci (2).



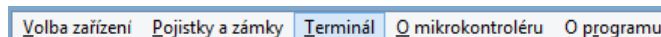
Obrázek 52 Programátor 8051 – otevření okna pojistek a zámků



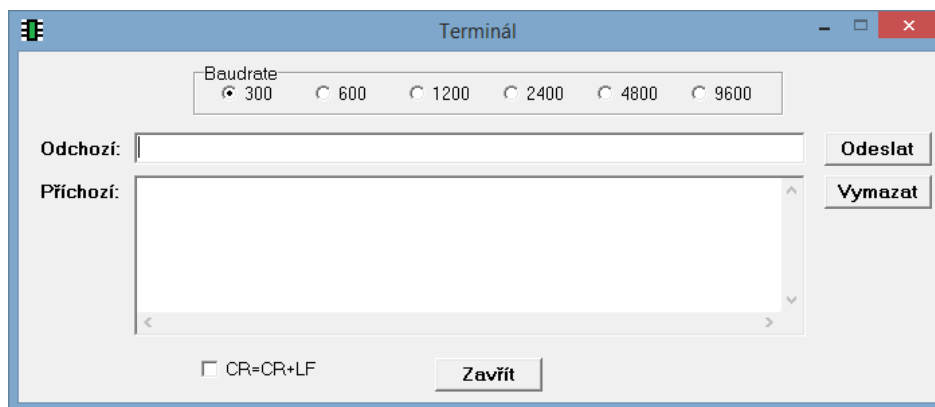
Obrázek 53 Programátor 8051 – nastavení pojistek a zámků

### 8.3.5 Terminál sériové linky

Aplikace je vybavena terminálem sériové linky, který slouží ke komunikaci s mikrokontrolérem. K činnosti je zapotřebí, aby byla propojka J3 v základní konfiguraci (dle popisu přípravku). Komunikace terminálu je možná v šesti standardních rychlostech: 300, 600, 1 200, 2 400, 4 800 a 9 600 baudů za sekundu. Odeslání textového řetězce z pole „Odchozí“ je možné po kliknutí na tlačítko „Odeslat“ nebo po stisku klávesy „Enter“. Příchozí znaky se zobrazují do pole „Příchozí“. V přijímaných zprávách lze rozlišit, zda je odřádkování tvořeno znaky CR nebo CR+LF (znak LF má v ASCII tabulce hodnotu 10 a znak CR má hodnotu 13 (15)). Pro odřádkování v poli „Příchozí“ je nutno, aby byly přijaty oba znaky. Zaškrtnutí možnosti „CR=CR+LF“ v dolní části okna znamená, že v přijímaných zprávách se vyskytuje pouze znak CR pro odřádkování.



Obrázek 54 Programátor 8051 – otevření okna terminálu

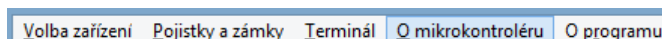


Obrázek 55 Programátor 8051 – terminál sériové linky

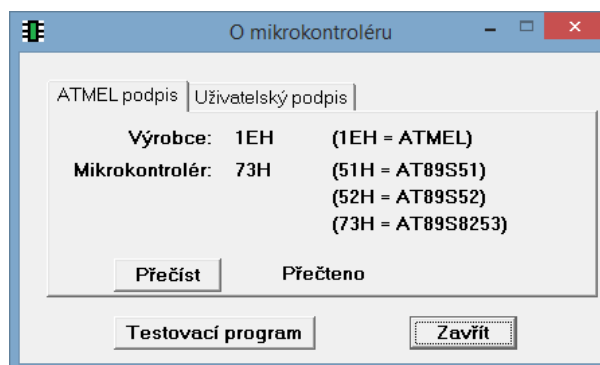
### 8.3.6 Testovací program a podpisy mikrokontroléru

Pod volbou „O mikrokontroléru“ se nachází čtení podpisu výrobce a čtení a zápis uživatelského podpisu (pouze u AT89S8253). Používané hodnoty podpisu výrobce jsou uvedeny v závorkách. Zapsání uživatelského podpisu lze, stejně jako u paměti programu, pouze jednou po vymazání celé paměti. Změna jednotlivých polí uživatelského podpisu je možná přímo v tabulce. Pro rychlé nastavení všech hodnot na „00“ nebo „FF“ jsou dostupná stejně pojmenovaná tlačítka.

V tomto okně se také nachází tlačítko pro načtení testovacího programu mikrokontroléru. Při použití tohoto tlačítka dojde k načtení programu do tabulky v hlavní části Programátoru, stejně jako kdyby byl načten ze souboru. Testovací program obsahuje opakovač sériové linky a na portu P0 zobrazí hodnotu 55h (0101 0101b).

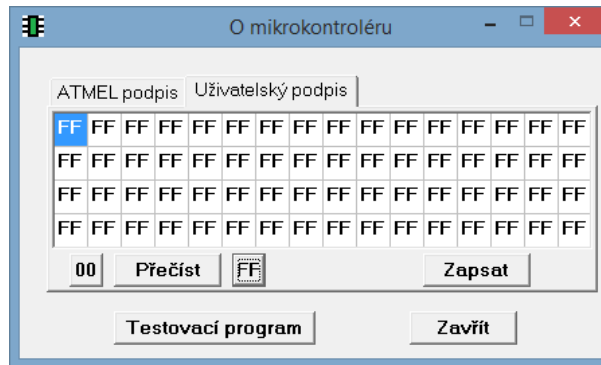


Obrázek 56 Programátor 8051 – otevření okna podpisů



Obrázek 57 Programátor 8051 – podpisy mikrokontroléru 1





Obrázek 58 Programátor 8051 – podpisy mikrokontroléru 2

### 8.3.7 Chybové hlášky

V aplikaci Programátor jsou možné tyto chybové hlášky:

**Neplatný formát souboru!** – soubor neobsahuje na jednom z řádků počáteční znak „:“, určující začátek záznamu.

**Neplatný typ záznamu na řádku X!** – soubor musí obsahovat pouze obecné záznamy (typ „00“) a konec souboru (typ „01“) dle formátu Intel HEX (4).

**Neplatný kontrolní součet na řádku X!** – každý záznam je chráněn kontrolním součtem na konci řádku a ten musí souhlasit. V opačném případě se může jednat o poškozený soubor.

**Příliš mnoho bajtů (200+) na řádku X!** – počet datových bajtů na jednom řádku je omezen na 200 hodnot, aby se nemusely importovat zbytečně dlouhé řádky. Vhodnější je záznamy rozdělit do více řádků.

**Překročení povolené FLASH adresy na řádku X!** – při importování programu je kontrolováno, zda se v souboru nenachází hodnoty na adrese, která je mimo dostupnou paměť FLASH.

**Překročení povolené EEPROM adresy na řádku X!** – při importování dat je kontrolováno, zda se v souboru nenachází hodnoty na adrese, která je mimo dostupnou paměť EEPROM.

**Nepodařilo se nastavit připojení!** – nepodařilo se nastavit výchozí logické hodnoty na USB převodníku z důvodu neznámé chyby.

**Chyba inicializace programování!** – nepodařilo se odeslat úvodní instrukci pro začátek programování z důvodu neznámé chyby.

**Zvolený mikrokontrolér nesouhlasí s podpisem!** – při kontrole vloženého mikrokontroléru nesouhlasí podpis zvoleného typu s přečtenou hodnotou. Důvodem může být špatně zvolený typ.

**Chyba mazání!** – nepodařilo se odeslat instrukci pro vymazání paměti mikrokontroléru z důvodu neznámé chyby.

**Chyba zápisu!** – nepodařilo se úspěšně odeslat všechny hodnoty určené k zapsání.

**Vypršel časový limit zápisu bloku!** – komunikace s mikrokontrolérem je rychlejší než ukládání hodnot do jeho paměti a tak musí být kontrolováno, zda je možné pokračovat v odesílání. V Programátoru je toto kontrolování časově omezeno a je počítáno s určitou rezervou, ve které se doba ukládání může pohybovat. Důvodem této chyby je překročení časové rezervy a vložený mikrokontrolér může být pomalý nebo vadný.

**Chyba ověření!** – hodnoty v tabulce grafického rozhraní nesouhlasí s obsahem mikrokontroléru.

**Chyba čtení!** – nepodařilo se správně přečíst všechny hodnoty z paměti EEPROM.

**Chyba resetu!** – nepodařilo se nastavit logické hodnoty pro reset mikrokontroléru z důvodu neznámé chyby.

## Závěr

Cílem této práce bylo navrhnout a vytvořit prototyp nového výukového přípravku, který bude sloužit k výuce Mikroprocesorové řídicí techniky v laboratoři Dopravní fakulty Jana Pernera na Univerzitě Pardubice. Nové řešení přípravku mělo odstranit nevýhody předchozího provedení, zejména problém s připojením k počítači a napájením. Dále mělo splňovat určitou vybavenost vstupními a výstupními prvky a komunikačním rozhraním. Součástí práce měl být i obslužný software.

Z důvodu návaznosti na předchozí přípravek byl nejprve proveden rozbor předchozího vyhotovení. Ten je uveden v druhé kapitole. Některé poznatky byly užitečné při vytváření nového návrhu. Jádrem zařízení zůstal stejný mikrokontrolér jako v minulém provedení. Vznikla ale možnost použití dalších dvou typů od stejného výrobce.

Prvním krokem při návrhu bylo odstranění stávajících problémů s napájením a připojením k počítači. To bylo vyřešeno použitím USB rozhraní jak ke komunikaci, tak k napájení. Dalším rozdílem je vybavení celého portu mikrokontroléru osmi tlačítky namísto pěti, protože přípravek pracuje v 8bitové architektuře. Shodně zůstalo vybavení dalšího portu osmi LED a vyvedení rozhraní standardu RS-232. Vytvoření obslužného softwaru bylo usnadněno podporou od výrobce klíčového integrovaného obvodu. Dostupné byly ovladače pro rozhraní USB a knihovna pouze nezákladnějších funkcí pro jejich použití. Celý navazující obslužný software byl pak vytvořen autorem práce. V průběhu navrhování byl vytvořen také prototyp přípravku, na kterém byla testována činnost jak hardwaru, tak softwaru.

Výsledkem práce je tedy navržení nového přípravku, který splňuje zadané požadavky a navíc poskytuje další funkce, usnadňující použití. Budoucí uživatelé mají k dispozici manuál, s jehož pomocí se mohou seznámit s přípravkem a následně si otestovat jejich vytvořený program pro mikrokontrolér. Materiály potřebné k vytvoření přípravku lze nalézt v přílohách této práce.

## Seznam použité literatury

1. **Skalický, Petr.** *Mikroprocesory řady 8051*. Praha : BEN – technická literatura, 2012. ISBN 80-86056-39-2.
2. AT89S8253 Datasheet. *Atmel*. [Online] 2010. [Citace: 20. 3. 2014.] Dostupné z: <http://www.atmel.com/Images/doc3286.pdf>.
3. AT89ISP Programmer Cable Application Note. *Atmel*. [Online] 2006. [Citace: 22. 3. 2014.] Dostupné z: <http://www.atmel.com/images/doc3310.pdf>.
4. Intel Hexadecimal Object File Format Specification. [Online] 1. 6. 1988. [Citace: 2. 4. 2014.] Dostupné z: <http://www.interlog.com/~speff/usefulinfo/Hexfrmt.pdf>.
5. AT89S51 Datasheet. *Atmel*. [Online] 2008. [Citace: 20. 3. 2014.] Dostupné z: <http://www.atmel.com/Images/doc2487.pdf>.
6. FTDI Products. *FTDI Chip*. [Online] Future Technology Devices International Ltd., 2014. [Citace: 14. 5. 2014.] Dostupné z: <http://www.ftdichip.com/FTProducts.htm>.
7. **Matoušek, David.** *Vývojový kit USB51KIT*. Praha : BEN – Technická literatura, 2005. ISBN 80-7300-162-4.
8. FT232R USB UART I.C. Datasheet. *FTDI Chip*. [Online] 2010. [Citace: 25. 3. 2014.] Dostupné z: [http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf).
9. ft2 | GM electronic. *GM electronic*. [Online] GM electronic, spol. s r. o., 2014. [Citace: 15. 4. 2014.] Dostupné z: <http://www.gme.cz/products/search?term=ft2>.
10. AT89S52 Datasheet. *Atmel*. [Online] 2008. [Citace: 20. 3. 2014.] Dostupné z: <http://www.atmel.com/Images/doc1919.pdf>.
11. +5V-Powered, Multichannel RS-232 Drivers/Receivers. *maxim integrated*. [Online] 2014. [Citace: 10. 5. 2014.] Dostupné z: <http://datasheets.maximintegrated.com/en/ds/MAX220-MAX249.pdf>.

12. Bit Bang Modes For The FT232R and FT245R. *FTDI Chip*. [Online] 2010. [Citace: 18. 4. 2014.] Dostupné z: [http://www.ftdichip.com/Support/Documents/AppNotes/AN\\_232R-01\\_Bit\\_Bang\\_Mode\\_Available\\_For\\_FT232R\\_and\\_Ft245R.pdf](http://www.ftdichip.com/Support/Documents/AppNotes/AN_232R-01_Bit_Bang_Mode_Available_For_FT232R_and_Ft245R.pdf).
13. D2XXUnit. *FTDI Chip*. [Online] 2012. [Citace: 25. 4. 2014.] Dostupné z: <http://www.ftdichip.com/Support/SoftwareExamples/CodeExamples/Delphi.htm>.
14. D2XX Direct Drivers. *FTDI Chip*. [Online] Future Technology Devices International Ltd., 2014. [Citace: 2. 5. 2014.] Dostupné z: <http://www.ftdichip.com/Drivers/D2XX.htm>.
15. Ascii Table. *www.asciitable.com*. [Online] 2010. [Citace: 5. 5. 2014.] Dostupné z: <http://www.asciitable.com/>.
16. **Matoušek, David**. *Práce s mikrokontroléry ATMEL AT89S8252*. Praha : BEN – technická literatura, 2002. ISBN 80-7300-066-0.
17. —. *Práce s mikrokontroléry ATMEL AT89LP2052, AT89LP4052*. Praha : BEN – technická literatura, 2006. ISBN 80-7300-205-1.
18. —. *Práce s mikrokontroléry ATMEL AT89C2051*. Praha : BEN – technická literatura, 2002. ISBN 80-7300-094-6.
19. —. *USB prakticky s obvody FTDI*. Praha : BEN – technická literatura, 2003. ISBN 80-7300-103-9.

## Seznam obrázků

|   |    |
|---|----|
| Obrázek 1 Základní jádro 8051 a rozšiřující periferie (1) .....                             | 11 |
| Obrázek 2 Základní blokové schéma předchozího přípravku .....                               | 16 |
| Obrázek 3 Pohled shora na předchozí přípravek .....   | 18 |
| Obrázek 4 Řešení přípravku s mezičlenem .....   | 20 |
| Obrázek 5 Řešení přípravku s externí pamětí .....   | 20 |
| Obrázek 6 Řešení přípravku s FTDI převodníkem .....   | 20 |
| Obrázek 7 Základní blokové schéma návrhu přípravku .....                                    | 21 |
| Obrázek 8 Popis vývodů FTDI převodníku FT232RL (8) .....                                    | 23 |
| Obrázek 9 Popis vývodů mikrokontroléru AT89S8253 (2) .....                                  | 23 |
| Obrázek 10 Popis vývodů obvodu MAX232 (11) .....  | 24 |
| Obrázek 11 Schéma zapojení obvodu FTDI .....  | 25 |
| Obrázek 12 Schéma zapojení mezi obvodem FTDI a mikrokontrolérem .....                       | 26 |
| Obrázek 13 Schéma zapojení vstupních a výstupních prvků a zapojení sériové linky .....      | 27 |
| Obrázek 14 Náhled výkresu desky posílých spojů při pohledu na spodní stranu přípravku ..... | 28 |
| Obrázek 15 Pohled shora na prototyp přípravku .....   | 29 |
| Obrázek 16 Návrh základního principu činnosti při detekci přípravku .....                   | 31 |
| Obrázek 17 Návrh postupu při zápisu a čtení nastavení přípravku .....                       | 31 |
| Obrázek 18 Náhled grafického prostředí Konfiguratoru .....                                  | 32 |
| Obrázek 19 Návrh postupu při načítání souboru do softwaru .....                             | 33 |
| Obrázek 20 Návrh komunikace s mikrokontrolérem .....  | 34 |
| Obrázek 21 Návrh zápisu a ověření v obslužné aplikaci .....                                 | 35 |
| Obrázek 22 Návrh zápisu a čtení zámků .....   | 36 |
| Obrázek 23 Umístění bajtu v souboru a v softwaru .....                                      | 36 |
| Obrázek 24 Vzorkování komunikace s mikrokontrolérem .....                                   | 37 |
| Obrázek 25 Náhled grafického prostředí Programátoru .....                                   | 38 |
| Obrázek 26 Okno správy pojistek a zámků .....   | 39 |
| Obrázek 27 Průběh napětí po připojení přípravku .....                                       | 41 |
| Obrázek 28 Průběh nulování mikrokontroléru po připojení přípravku .....                     | 42 |
| Obrázek 29 Průběh nulování mikrokontroléru obslužným softwarem – ovládání tranzistoru ..... | 43 |
| Obrázek 30 Průběh nulování mikrokontroléru obslužným softwarem – výsledný průběh .....      | 43 |
| Obrázek 31 Předpokládaný průběh přenosu jednoho bajtu při programování .....                | 44 |

|   |    |
|---|----|
| Obrázek 32 Skutečný průběh přenosu jednoho bajtu při programování ..... | 44 |
| Obrázek 33 Zkouška Konfiguratoru – úspěšné nastavení .....              | 45 |
| Obrázek 34 Zkouška Konfiguratoru – ověření zápisu.....                  | 45 |
| Obrázek 35 Zkouška Programátoru – úspěšná detekce mikrokontroléru.....  | 46 |
| Obrázek 36 Zkouška Programátoru – stav zápisu.....                      | 46 |
| Obrázek 37 Zkouška Programátoru – stav ověření .....                    | 46 |
| Obrázek 38 Zkouška Programátoru – nastavení zámků .....                 | 47 |
| Obrázek 39 Očíslované části přípravku.....                              | 48 |
| Obrázek 40 Propojka J3 – propojení terminálu s mikrokontrolérem .....   | 49 |
| Obrázek 41 Propojka J3 – opakovač do terminálu.....                     | 49 |
| Obrázek 42 Propojka J3 – opakovač do mikrokontroléru .....              | 49 |
| Obrázek 43 Vývody přípravku pro externí zařízení.....                   | 49 |
| Obrázek 44 Propojka J1 – základní poloha .....                          | 50 |
| Obrázek 45 Propojka J1 – vynucení napájení mikrokontroléru.....         | 50 |
| Obrázek 46 Grafické prostředí programu Konfigurator .....               | 51 |
| Obrázek 47 Grafické prostředí aplikace Programátor .....                | 52 |
| Obrázek 48 Programátor 8051 – dialogové okno detekce.....               | 52 |
| Obrázek 49 Programátor 8051 – volba mikrokontroléru .....               | 53 |
| Obrázek 50 Programátor 8051 – načtení souboru.....                      | 53 |
| Obrázek 51 Programátor 8051 – nastavení a průběh činnosti .....         | 54 |
| Obrázek 52 Programátor 8051 – otevření okna pojistek a zámků .....      | 55 |
| Obrázek 53 Programátor 8051 – nastavení pojistek a zámků.....           | 55 |
| Obrázek 54 Programátor 8051 – otevření okna terminálu .....             | 55 |
| Obrázek 55 Programátor 8051 – terminál sériové linky .....              | 56 |
| Obrázek 56 Programátor 8051 – otevření okna podpisů .....               | 56 |
| Obrázek 57 Programátor 8051 – podpisy mikrokontroléru 1 .....           | 56 |
| Obrázek 58 Programátor 8051 – podpisy mikrokontroléru 2 .....           | 57 |

## Seznam zkratek

USB – univerzální sériová sběrnice, běžně dostupná u počítačů

RS-232 – standard sériového komunikačního rozhraní, využívající napětí  $\pm 12$  V

SPI – sériové periferní rozhraní, využitelné pro komunikaci mezi integrovanými obvody

RAM – paměť s přímým přístupem, určená pro dočasné uložení dat

EEPROM – elektricky mazatelná programovatelná paměť, určená pro uložení dat

A/D převodník – obvod pro převod analogového (spojitého) signálu na digitální (diskrétní)

TTL – tranzistor-tranzistorová logika, využívající napětí 5 V

LED – světlo emitující dioda



## **Seznam příloh na CD**

**Příloha A** Detailní schéma zapojení přípravku

**Příloha B** Výkres desky plošných spojů z programu EAGLE

**Příloha C** Seznam součástek prototypu

**Příloha D** Zdrojový kód obslužného softwaru

**Příloha E** Obslužný software

**Příloha F** Kopie ovladačů od FTDI Chip

**Příloha G** Samostatný Manuál pro práci s přípravkem