

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA ELEKTROTECHNICKÁ

KATEDRA TECHNOLOGIÍ A MĚŘENÍ

DIPLOMOVÁ PRÁCE

**Komunikační rozhraní USB pro elektronické zdroje a
zátěže ITECH**

Anotace

Cílem práce je vytvořit obslužný program v jazyce C pro zařízení, které zřizuje komunikaci mezi PC a elektronickou zátěží. Hlavní funkcí komunikačního zařízení je překládat komfortní uživatelské příkazy do formátu, který je srozumitelný pro elektronickou zátěž IT8512C. Tyto příkazy jsou odesílány z terminálu PC přes rozhraní USB. Zařízení zprávy nejprve převede z rozhraní USB na RS232 a následně odešle řídicímu mikrokontroléru. Ke konverzi USB/RS232 je použit převodník FT232R. Mikrokontrolér na základě tohoto požadavku vygeneruje 26 bytů dlouhou zprávu a odešle ji druhým sériovým portem do elektronické zátěže. Tento přenos funguje obousměrně. Obdobně je tedy možno zjišťovat údaje z elektronické zátěže a zobrazovat je na terminálu PC.

Klíčová slova

Elektronická zátěž, komunikační zařízení, PC, sériový port, mikrokontrolér, obslužný program, převodník USB/RS232.

Abstrakt

The aim of the work is to create the utility program in language C for the device that allows communication between the computer and the electronic load. The main function of communication device is conversion comfortable custom commands to the format that is understood by the electronic load IT8512C. These commands are sent from the PC terminal through the USB interface. The device converts these commands from the USB interface to the RS232 interface and sends them the control microcontroller. The converter FT232R is used for conversion USB to RS232. The microcontroller generates 26 bytes long message depending on the requirement and sends it second serial port to the electronic load. The transfer works in both directions. It is possible to collect the data from the electronic load and display them on the PC terminal.

Key words

Electronic load, communication device, computer, serial port, microcontroller, utility program, converter USB/RS232.

Prohlášení

Předkládám tímto svou závěrečnou diplomovou práci na téma **Komunikační rozhraní USB pro elektronické zdroje a zátěže ITECH** k obhajobě a posouzení odborné komisi. Tímto prohlašuji, že práci jsem vypracoval sám, za pomoci Ing. Aleše Voborníka, odborných knih a jednotlivých technických dokumentací uvedených na konci mé práce.

V Plzni dne 9. 5. 2014

Josef Suchý

Poděkování

Moc rád bych tímto poděkoval vedoucímu práce Ing. Aleši Voborníkovi za účinnou metodickou, pedagogickou a odbornou pomoc, za vypůjčená zařízení a další cenné rady při zpracování mé diplomové práce.

Úvod	9
Seznam použitých zkratk	10
1. Elektronická zátěž IT8512C	11
1.1. Popis	11
1.1.1. Popis předního panelu	11
1.1.2. Popis zadního panelu	12
1.2. Operační módy	12
1.2.1 Konstantní proud (CC)	12
1.2.2 Konstantní napětí (CV)	13
1.2.3 Konstantní odpor (CR)	13
1.2.4 Konstantní výkon (CW)	14
1.3. Test baterie	14
1.3.1 Test kapacity	15
1.4. Transientní zkoušky	15
1.5. Ochranné prvky	16
1.5.1 Ochrana proti přepětí	16
1.5.2 Ochrana proti nadproudu	17
1.5.3 Další ochrany	17
1.6. Komunikace skrz I/O periférii.....	17
2. Komunikační zařízení	19
2.1. Popis	19
2.2. Převodník USB/RS232	21
3. Mikrokontrolér DS89C440	22
3.1. Základní popis	22
3.2. Architektura	23
3.2.1 Alu.....	23
3.2.2 I/O porty	23
3.2.3 Čítače/časovače	23
3.2.4 Sériové porty.....	24
3.2.5 Zásobník	24
3.2.6 Flash paměť	24
3.2.7 Pracovní registry	24
3.2.8 Mapa paměti	24
3.2.9 Mapa registrů	24
3.2.9 Adresová/datová sběrnice.....	25
3.2.10 Watchdog.....	25

3.2.11	Přerušení	25
3.3.	Sériové porty	26
3.3.1	Módy	27
3.3.2	Podrobný popis módu 1	27
3.3.3	Detekce rámcové chyby.....	28
3.3.4	Inicializace sériového portu.....	28
3.4.	Správa a kontrola napájení.....	29
3.4.1	Monitoring napětí	29
3.4.2	System včasného varování před výpadkem napětí.....	29
3.4.3	Snížení kmitočtu	30
3.4.4	Klidový mód	31
3.4.5	Stop mód.....	32
3.4.6	Redukce rychlosti.....	32
3.4.7	Režim řízení spotřeby	33
3.4.8	Switchback.....	34
3.5.	Přerušení.....	35
3.5.1	Zdroje přerušení.....	36
3.5.2	Priority přerušení.....	37
3.5.3	Reakční doba obsluhy přerušení	37
4.	Obslužný program	38
4.1.	Funkce programu	38
4.2.	Detailní popis programu	40
4.2.1	Výpočet baud-rate a nastavení řídicích registrů.....	40
4.2.2	Příjem zprávy od PC	41
4.2.3	Porovnání obdržené zprávy s příkazovou sadou	42
4.2.4.	Odeslání vygenerované zprávy přístroji.....	42
4.2.5.	Odpověď elektronické zátěže na požadavek	43
5.	Závěr	44
6.	Literatura	45
7.	Seznam příloh	46
	Příloha 1	47
	Příloha 2.....	48

Úvod

Předmětem této diplomové práce je zajištění elektronické komunikace mezi PC a elektronickou zátěží IT8512C. Komunikaci zajišťuje speciální komunikační zařízení. Do tohoto zařízení je implementován řídicí program, jehož napsání je hlavním cílem mé práce.

Text diplomové práce se skládá z několika bodů. V první řadě se budu věnovat elektronické zátěži IT8512C. Popíši její základní vlastnosti, parametry a měřicí zkoušky. Poté se zaměřím na komunikaci mezi elektronickou zátěží a PC. Uvedu princip komunikace, přenosové rychlosti a formát odesílané zprávy. V této kapitole se také zmíním o standardizovaném zařízení IT-E131, které je ke komunikaci použito. V další části textu představím zdokonalené komunikační zařízení a porovnáím ho se zařízením IT-E131. U zdokonaleného zařízení podrobně popíši všechny důležité části a ukáži výhody oproti standardnímu zařízení. V další kapitole se budu speciálně věnovat mikroprocesoru DS89C440, který celé zařízení řídí. V této kapitole procesor důkladně popíši a uvedu základní vlastnosti a parametry. Podrobněji se zaměřím na dva sériové kanály, kterými procesor disponuje. Poslední část textu věnuji vlastnímu řídicímu programu. Uvedu základní funkci programu a popíši jeho jednotlivé části. V této kapitole také vysvětlím nastavení jednotlivých sériových kanálů a vzorce pro výpočet přenosové rychlosti.

V závěru práce bych rád shrnul veškeré poznatky, které jsem při vypracování textu nabyl a zhodnotím zde základní cíle a výsledky projektu.

Seznam použitých zkratek

elektrické napětí	U [V]
elektrický proud	I [A]
elektrický odpor	R [Ω]
elektrický výkon	P [W]
frekvence	f [Hz]
čas	t [s]
mód konstantního proudu	CC
mód konstantního napětí	CV
mód konstantního odporu	CR
mód konstantního výkonu	CW
speciální funkční registr	SFR
obsluha rutiny přerušení	ISR
milión instrukcí za sekundu	MIPs

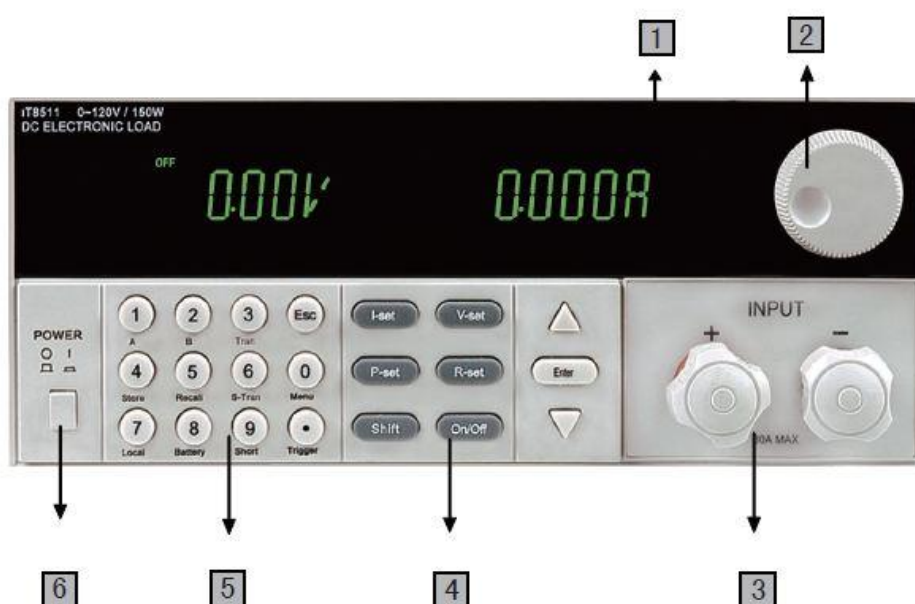
1. Elektronická zátěž IT8512C

1.1. Popis

Elektronická zátěž je používána pro návrh, výrobu a ohodnocení DC zdrojů, baterií, nabíječek, apod. Zařízení obsahuje procesor, I/O periférii RS232, ovládací prvky na předním panelu, displej, generátor pro transienční zkoušky, inteligentní chladicí systém, atd. Elektronická zátěž může pracovat v režimu konstantního proudu (CC), konstantního napětí (CV), konstantního odporu (CR) a konstantního výkonu (CW). Zátěž IT8512C se pyšní vysokou přesností a rozlišením jak u napětí (1 mV), tak u proudu (0,1 mA). [1], [2]

Zařízení je napájeno střídavým napětím o hodnotě 110 V, nebo 230 V (na zadním panelu disponuje přepínačem) a frekvenci 50/60 Hz. Maximální vstupní napětí přivedené na svorky je 120 V, proud od 1 mA do 60 A a maximální výkon je 300 W. [2]

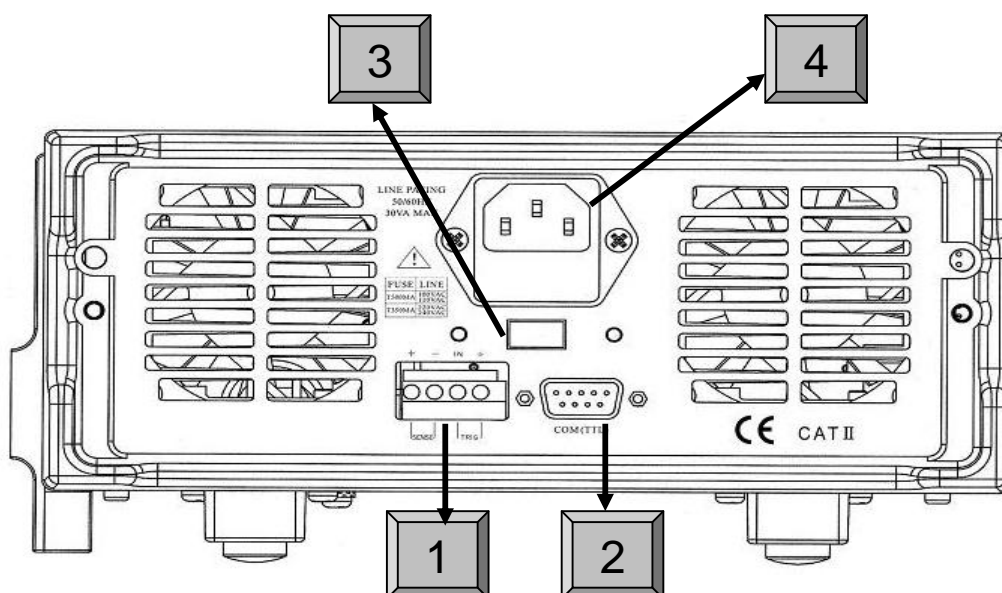
1.1.1. POPIS PŘEDNÍHO PANELU



Obr. 1.1.1. Přední panel přístroje IT8512C [1]

1. Přední panel pro zobrazení naměřených hodnot
2. Rotační ovládací tlačítko
3. Klávesnice
4. Numerická klávesnice
5. Tlačítko ON/OFF
6. Vstupní svorky

1.1.2. POPIS ZADNÍHO PANELU



Obr. 1.1.2. Zadní panel přístroje IT8512C [1]

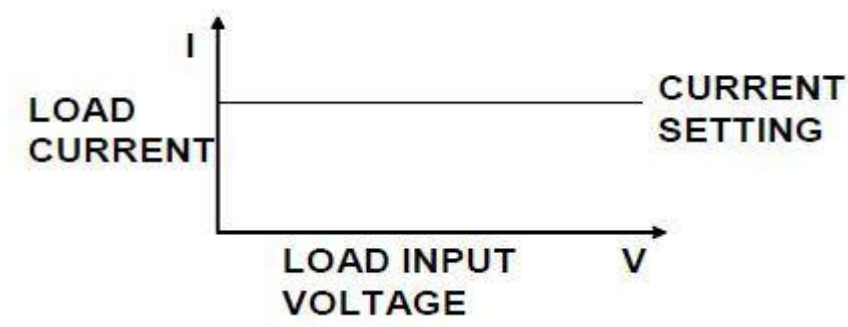
1. 4-pinový konektor pro spouštění a vzdálenou kontrolu přístroje
2. 9-pinový COM port
3. Přepínač mezi napájecím napětím 110/220 V
4. Napájecí konektor IEC320

1.2. Operační módy

1.2.1 KONSTANTNÍ PROUD (CC)

V tomto režimu se proud rychle dostane do přeprogramované hodnoty bez

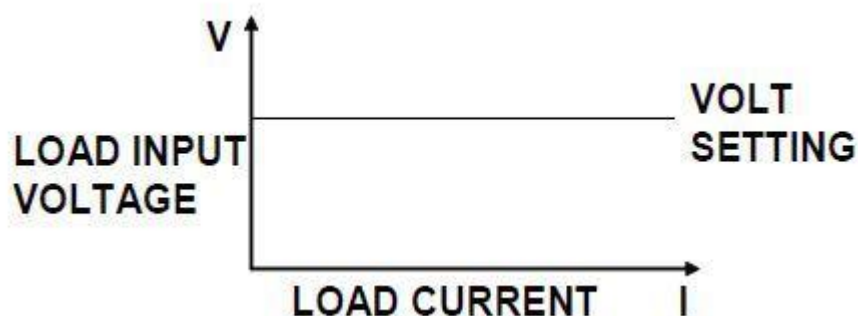
ohledu na vstupní napětí. CC mód se všemi parametry může být nastaven buď na předním panelu, nebo vzdáleně pomocí PC a komunikační sběrnice. Při rozsahu 0 - 3 A má tento mód rozlišení 0,1 mA a přesnost $\pm (0,1\%+0,1\%FS)$. V rozsahu 0 - 60 A je rozlišení 1 mA a přesnost $\pm (0,2\%+0,15\%FS)$. [1]



Obr. 1.2.1 Mód konstantního proudu [1]

1.2.2 KONSTANTNÍ NAPĚTÍ (CV)

Při tomto módu se snaží elektronická zátěž udržet nastavené napětí bez ohledu na protékající proud. Režim je opět možné nastavit jak na přístroji, tak speciální instrukcí z PC. Tento mód disponuje taktéž dvěma rozsahy. V prvním regulujeme napětí od 1,5 V do 18 V s přesností $\pm (0,05\%+0,02\%FS)$ a rozlišením 1 mV. Druhý rozsah je od 1,5 V do 120 V, přesnost je zde $\pm (0,05\%+0,025\%FS)$ a rozlišení 10 mV. [1]

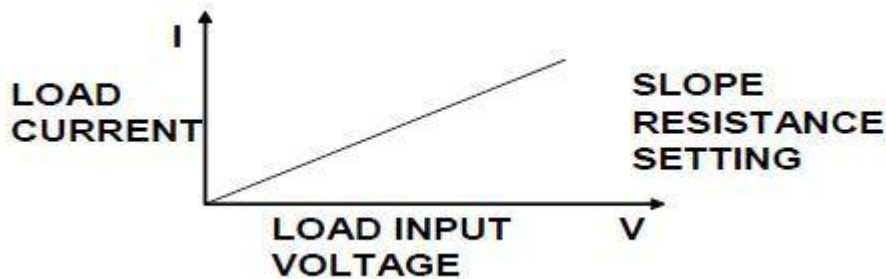


Obr. 1.2.2 Mód konstantního napětí [1]

1.2.3 KONSTANTNÍ ODPOR (CR)

V tomto režimu se zařízení snaží udržet lineární V/A charakteristiku a

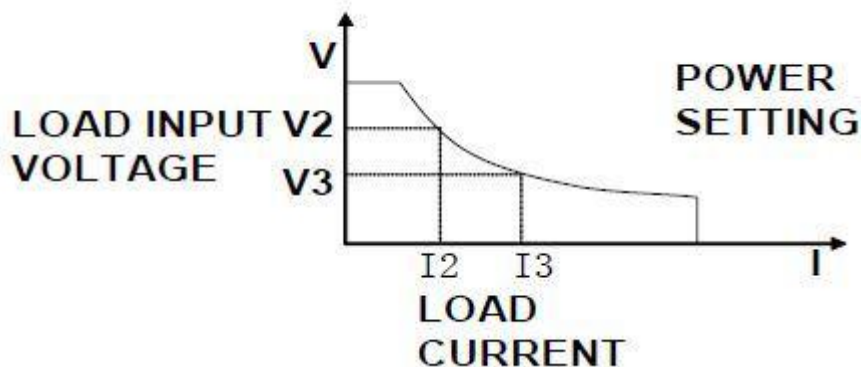
zachovat tak předem nastavený odpor. Nastavení probíhá obdobně jako v minulých případech, ale v tomto módu zařízení disponuje čtyřmi rozsahy. Nejnižší je 0,1 - 10 Ω a nejvyšší 1 - 4 k Ω . [1]



Obr. 1.2.3 Mód konstantního odporu [1]

1.2.4 KONSTANTNÍ VÝKON (CW)

Zde elektronická zátěž spotřebovává energii v souladu s naprogramovanou hodnotou bez ohledu na vstupní napětí. Režim se všemi parametry lze opět nastavit přímo na přístroji, nebo lze přístroji odeslat instrukci pomocí sériové linky. Rozsahy jsou 0 - 100 W a 100 - 300 W.[1]



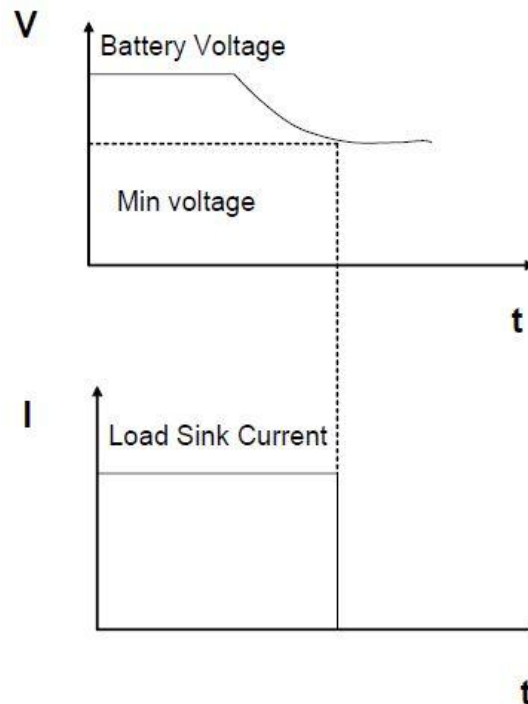
Obr. 1.2.4 Mód konstantního výkonu [1]

1.3. Test baterie

Nejlépeším způsobem testování, zdali baterie pracuje dobře, nebo ne, je testování se zátěží. Elektronická zátěž může být použita k testování jakýchkoliv typů baterií. Jelikož baterie je komponent s nízkou životností, je nutné provádět testy periodicky po určité době. [1].

1.3.1 TEST KAPACITY

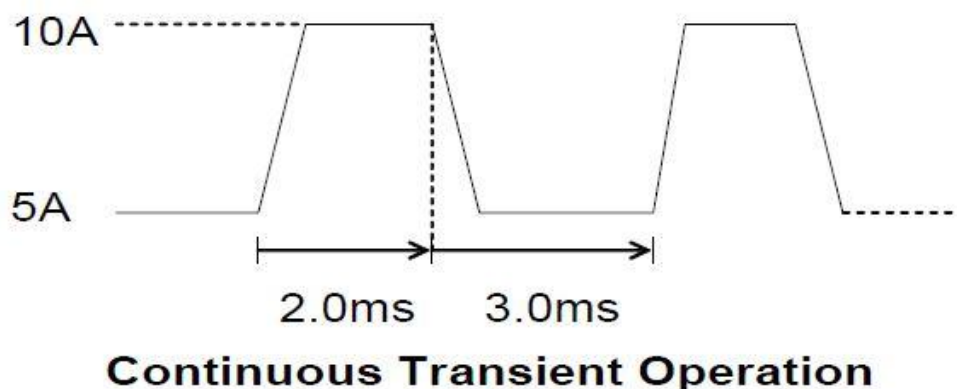
Pro test kapacity je použit mód konstantního proudu. Pro měření je na elektronické zátěži nastaven určitý vybíjecí proud a měří se čas, za který napětí na baterii klesne pod kritickou mez. Tento čas se samozřejmě liší s velikostí vybíjecího proudu, ale dá se říci, že čím je čas delší, tím má baterie větší kapacitu. Průběh můžete vidět na grafu 2.3.1. [1]



Graf 1.3.1 Průběh testování baterie [1]

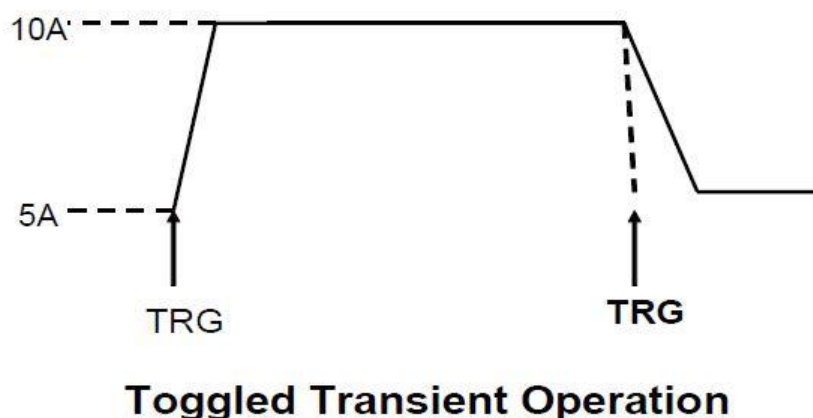
1.4. Transientní zkoušky

Transientní zkoušky dovolují elektronické zátěži periodicky přepínat mezi dvěma úrovněmi zatížení a tím zjišťovat jak reaguje elektrický zdroj na rychlé změny. Tyto zkoušky v podstatě simulují přechodové děje, které vznikají v obvodu s elektrickým zdrojem. U transientních zkoušek je důležité vědět, že mohou být spuštěny pouze v režimu konstantního proudu (CC), nebo konstantního napětí (CV). Tento test může být nastaven přímo na přístroji, nebo vzdáleně speciálním příkazem. Pro tyto zkoušky existují 3 režimy. Jsou to continuous, pulse a toggled. U režimu continuous se nastavují dvě úrovně proudu (zátěž) a doby po kterých se budou obě úrovně přepínat, viz obr. 1.4.1.



Obr. 1.4.1 Průběh signálu v režimu continuous [1]

U tohoto módu se tedy zadávají pouze parametry a po spuštění již běží zkouška samostatně. Tímto se liší od zbylých dvou režimů. Režimy pulse i toggled pracují s tzv. trigger signálem. U režimu pulse se tímto signálem přepíná zátěž do druhé úrovně, kde je stanovený čas, po kterém se zátěž vrátí zpět do původní velikosti. Nastavují se zde tedy dvě úrovně zátěže a jeden časový interval. V režimu toggled se časový interval nenastavuje vůbec, jelikož obě dvě úrovně jsou přepínány trigger signálem. Tento režim je zobrazen na obrázku 1.4.2. [1] [2]



Obr. 1.4.2 Průběh signálu v režimu toggled [1]

1.5. Ochranné prvky

1.5.1 OCHRANA PROTI PŘEPĚTÍ

Pokud vstupní napětí překročí mez nastavenou uživatelem, elektronická zátěž

automaticky odpojí vstup (input = OFF) a upozorní uživatele zvukovým signálem. Následně se na displeji objeví varovný nápis OVER VOLTAGE. Pro tento typ zátěže je limitní vstupní napětí 120 V + 5%, tedy 126 V. [1]

1.5.2 OCHRANA PROTI NADPROUDU

V módech konstantního proudu, odporu a výkonu je zatěžovací proud omezen mezní hodnotou nastavenou uživatelem. Maximální možná hodnota je pro tento model 60A. Pokud je tato hodnota překročena, zátěž vstoupí do ochranného stavu. Je-li v tomto stavu zátěž provozována v módech CR, nebo CW je automaticky přepnuta do režimu konstantního proudu CC. Pokud je zátěž provozována v CV módu a zároveň provádí transientní zkoušky, je uživatel na nadproud upozorněn zvukovým signálem a blikající hodnotou limitního proudu. [1]

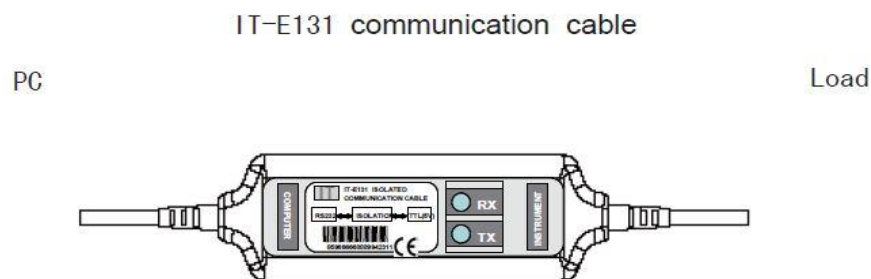
1.5.3 DALŠÍ OCHRANY

Mezi další ochrany patří například tepelná ochrana, kde přístroj pomocí senzoru kontroluje teplotu uvnitř přístroje. Jestliže tato teplota dosáhne hodnoty 80°C a vyšší, je přístroj uveden do ochranného stavu, kdy se vypne vstup a na displeji se objeví nápis OVER HEAT. [4]

Přístroj také dokáže rozpoznat, má-li DC zdroj zapojen opačně polarity. V tomto případě uživatele upozorní zvukovým znamením a zobrazí na displeji REVERSE VOLTAGE. [1]

1.6. Komunikace skrz I/O periferii

Jak již bylo řečeno, elektronická zátěž může být ovládána skrz klávesnici na předním panelu, nebo vzdáleně pomocí PC a I/O periferie. Přístroj má pro tento režim v sobě zabudované rozhraní RS232. Ke komunikaci je použito speciálního komunikačního zařízení IT-E131, které je zobrazeno na obrázku 1.6.1. [1]



Obr. 1.6.1 Zařízení IT-E131

Pro správnou komunikaci mezi zařízením a PC je nutné také nastavit přenosovou rychlost. Tato rychlost je defaultně nastavena na 4800 Bd, ale může být změněna pomocí tlačítek na předním panelu, nebo speciálního softwarového příkazu na rychlost 9600, 19200 nebo 38400 Bd. [1]

Řešení pomocí tohoto zařízení má několik nevýhod. Předně rozhraní pro připojení PC je RS232. Tento konektor je již poměrně zastaralý a na nových stolních počítačích a v laptotech se již standardně nevyskytuje. Další nevýhodou je formát zprávy, který musí být elektronické zátěži poslán. Pro každý jednotlivý příkaz musí být přístroji posláno 26-bytové slovo. Každé slovo musí začít startovním bytem o hodnotě AAH. Dále následuje adresa. Tato adresa může být v rozsahu 0 - FEH a lze ji nastavit speciálním příkazem. Třetí byte identifikuje typ příkazu. Ve 4. až 25. bytu je blíže specifikován jednotlivý příkaz. V této části je nastavena například volba režimu, maximální proud/napětí/odpor/výkon, parametry pro přechodové děje, aj. Poslední byte obsahuje kontrolní součet a slouží pro detekci potenciální chyby přenosu. V tabulce 1.6.1 je zobrazen obecný formát zprávy a v tabulce 1.6.2 formát zprávy, který na elektronické zátěži nastaví mód konstantního napětí. [1]

Posílat pro každý příkaz tento formát zprávy je dosti nepraktické, proto cílem mé práce bylo napsat program, který tento složitý řetězec vytvoří pomocí komfortních ovládacích příkazů podobných jazyku SCPI. Program včetně zařízení, na které byl implementován, bude podrobně popsán v dalších kapitolách.

Tab. 1.6.1 Všeobecný formát zprávy

VŠEOBECNÝ FORMÁT ZPRÁVY	
1. byte	START BYTE (AAH)
2. byte	ADRESA (00H - FEH)
3. byte	VÝBĚR PŘÍKAZU
4. až 25. byte	OBSAH INFORMACE
26. byte	KONTROLNÍ SOUČET

Tab. 1.6.2 Formát pro nastavení módu konstantního napětí

FORMÁT ZPRÁVY PRO NASTAVENÍ MÓDU KONSTANTNÍHO NAPĚTÍ	
1. byte	AAH
2. byte	00H
3. byte	28H
4. byte	01H
5. až 25. byte	00H
26. byte	D3H

2. Komunikační zařízení

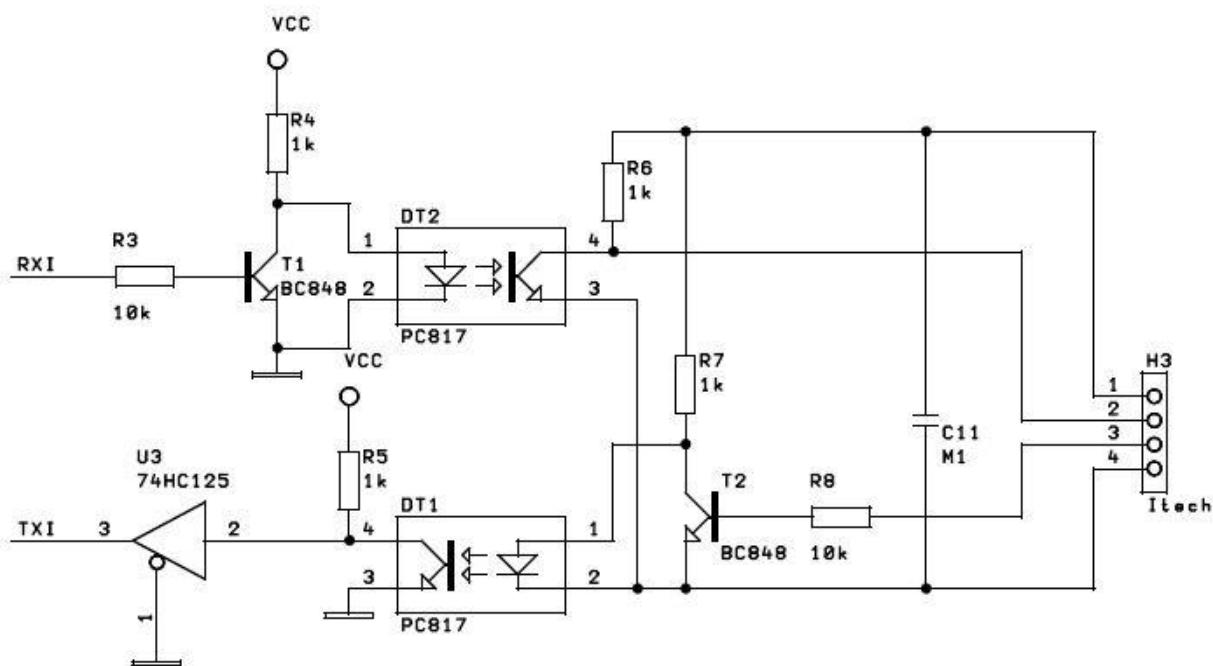
Jak bylo zmíněno výše, použití standartního zařízení ke komunikaci mezi PC a elektronickou zátěží má své nevýhody. Cílem mé práce bylo implementovat program na zdokonalené zařízení, které tyto problémy vyřeší. Primárně se jedná o transformaci dnes již méně používaného rozhraní RS232 na rozhraní USB a vytvoření komfortního ovládání pomocí příkazů podobných jazyku SCPI.

2.1. Popis

Komunikační zařízení je postaveno na procesoru DS89C440. Tento mikroprocesor vychází z architektury procesorů 8051, a má tedy i stejnou instrukční sadu. Hlavním důvodem, proč byl vybrán právě tento procesor, je vyšší výkon než standartní 8051 (při stejné taktovací frekvenci) a především to, že obsahuje dva sériové porty. Dva sériové porty jsou pro toto zařízení nezbytné, neboť prvním portem je obsluhováno PC a druhý port slouží ke komunikaci s elektronickou zátěží. Podrobný popis mikrokontroléru je uveden v další kapitole. [5]

Jak již bylo zmíněno, sériový port je v dnešní době na počítačích a notebookech spíše vzácností, proto je mezi mikrokontrolér a počítač vložen

převodník USB/RS232. Jako převodník byl použit model z řady FTDI, konkrétně FT232R, který je popsán níže. [3] Z důvodu rušení jsou piny sériového portu vedeného k elektronické zátěži galvanicky odděleny optočleny PC817. Jedná se o tranzistorové optočleny s výstupním napětím až 35 V a výstupním proudem 50 mA. Izolační napětí je 5 kV. Schéma zapojení optočlenů vidíme na obrázku 2.1.1. [4]



Obr. 2.1.1 Schéma zapojení optočlenů

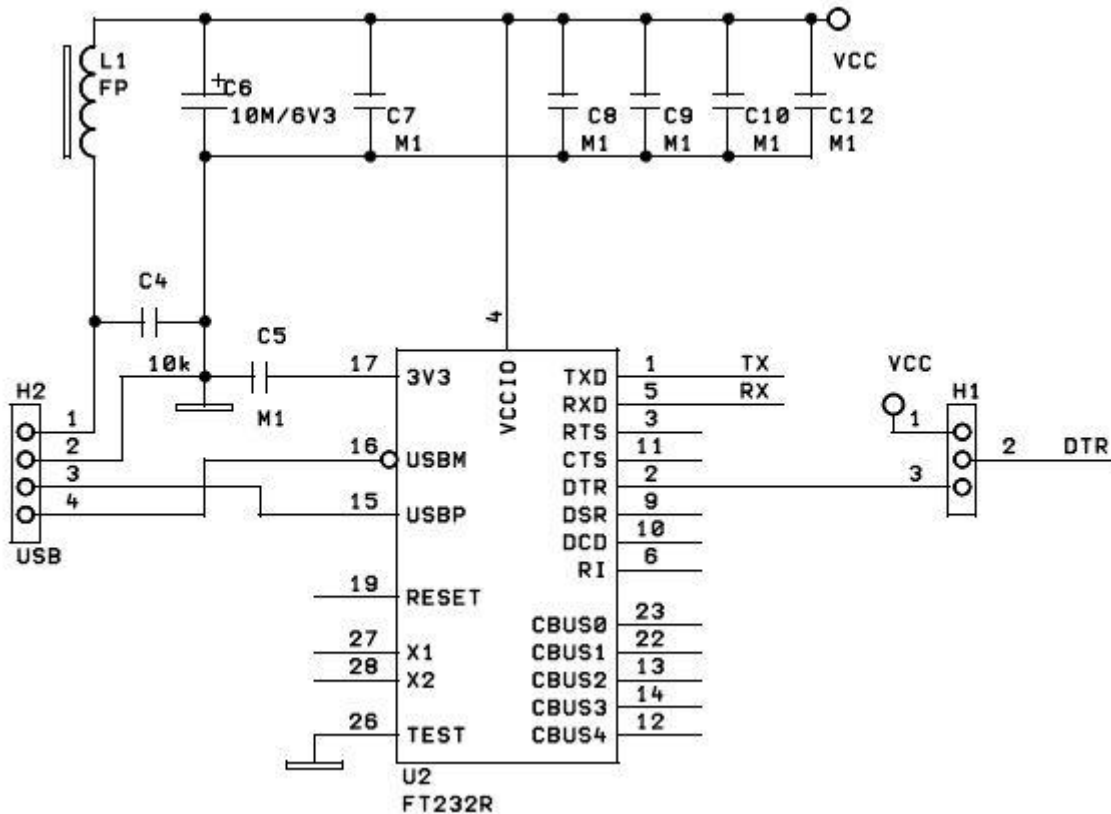
Z důvodu přesného výpočtu přenosové rychlosti, která je nezbytná pro správné fungování obou sériových portů, je zvolen rezonanční krystal s frekvencí 11,0592 MHz. Vzorec pro výpočet s bližším vysvětlením naleznete v kapitole 4.2 s názvem *Detailní popis programu*. Pro komunikaci s PC byla zvolena přenosová rychlost 115200 Bd a pro komunikaci s elektronickou zátěží rychlost 9600 Bd.

K programování procesoru je využito rozhraní USB bez nutnosti použití speciálního programátoru. Pro přenos programu do procesoru jsou využity piny $\overline{\text{PSEN}}$ a $\overline{\text{EA}}$. Přes USB je také řešeno napájení celého zařízení. Rozhraní USB pracuje s napětím 5 V, toto napětí vyhovuje jak mikroprocesoru, tak převodníku FTDI, proto nebylo nutné do zařízení umisťovat napěťové stabilizátory nebo podobné obvody pro úpravu napětí.

2.2. Převodník USB/RS232

Pro převod ze sériové linky na USB byl použit převodník FT232R. Díky příznivému poměru výkon/cena patří tento, ale i ostatní převodníky firmy FTDI (Future technology device integrations) k tomu nejlepšímu a nejpoužívanějšímu typu na trhu. Díky své jednoduchosti a kompatibilitě (podporuje jej drtivá většina operačních systémů) se stal základem velkého počtu zařízení jako např. USB telefony, USB rozhraní pro MP3 přehrávače, digitální kamery, atd. [3]

FT232R disponuje přenosovou rychlostí 300 Bd až 3 MBd. Integrovaný obvod obsahuje také interní vyrovnávací paměť o velikosti 384 B (PC \rightarrow zařízení) a 128 B (zařízení \rightarrow PC). Komunikace pomocí RS232 je vedena třemi hlavními vodiči RXD (příjem), TXD (vysílání) a společné zemi GND. Oproti svým předchůdcům od firmy FTDI má tento převodník navíc integrované hodiny (6, 12, 24 a 48 MHz), 1024 B interní EEPROM, integrované USB rezistory, aj. Napájecí napětí je v rozmezí 3,3 V - 5,25 V a může být řešeno přímo z USB portu. [3]



Obr. 2.2.1. Zapojení převodníku FT232R

Na obrázku 2.2.1. vidíme zapojení převodníku přímo v zařízení. Z obrázku je patrné, že nebylo nutné připojit všechny piny. Pro chod převodníku jsou nejdůležitější piny USBM a USBP, které slouží jako datové vodiče pro USB a piny TXD a RXD, které slouží jako datové vodiče pro rozhraní RS232. Integrovaný obvod je napájen pinem VCCIO přes USB konektor přímo z PC. Dalším pinem je 3V3. Je to výstup z integrovaného regulátoru LDO. Tento vývod jsme nepotřebovali, proto byl připojen přes kondenzátor na zem. Jeho hlavní využití je interní napájení 3,3 V pro USB transceiver, ale dá se také použít k napájení VCCIO obvodu. Zapojen je ještě pin DTR (Data Terminal Ready), který kontroluje výstupní signály pro RS232. [3]

K nevyužitým pinům patří např. CBUS1-5, kterým lze konfigurovat výstupy i vstupy, regulaci výkonu, kmitočet výstupního signálu aj. Piny X1 a X2 lze použít jak vstup/výstup oscilátoru. Pro použití interního oscilátoru musí být připojena analogová zem AGND. Na integrovaném obvodu nalezneme ještě testovací pin, který musí být v provozu uzemněn, resetovací pin a piny pro řízení a kontrolu vstupních signálů RS232. [3]

3. Mikrokontrolér DS89C440

3.1. Základní popis

Mikrokontrolér DS89C440 vychází z architektury procesorů 8051, ale ve srovnání s nimi poskytuje vyšší výkon. Největší inovací je použití pouze jednoho taktovacího cyklu pro vykonání instrukce (ve výchozím režimu) na rozdíl od 12 taktů, které potřeboval původní procesor 8051. Nový design procesoru tedy dokáže vykonávat instrukce, při stejné taktovací frekvenci, až 12 krát rychleji než originální procesor 8051. Účinnější konstrukce umožňuje analogicky použít menší taktovací rychlost a dosáhnout tak výkonu původního mikrokontroléru 8051, ale při podstatném snížení příkonu. [5]

Výše bylo uvedeno, že mikrokontrolér DS89C440 vychází z architektury procesorů 8051, má tedy i stejnou instrukční sadu. Dále obsahuje 32 kB flash paměti, 1 kB paměti RAM, čtyři 8-bit paralelní brány, tři 16-bit čítače/časovače, dva sériové kanály, watchdog, atd. Procesor poskytuje 256 bitů paměti RAM pro

proměnné a zásobní. Při maximální taktovací frekvenci 33 MHz dosahuje výkon 33 MIPS, což je 33 miliónů instrukcí za vteřinu. Tato úroveň výpočetního výkonu je srovnatelná s 16-bit procesory, ale bez zvýšení nákladů a složitosti při zavedení 16-bit rozhraní. [5]

Procesor obsahuje režim řízení spotřeby, který mu dynamicky dovolí měnit vnitřní taktovací frekvenci, vzhledem k zatížení procesoru. Taktovací frekvence se mění od 1 do 1024 taktů na jeden cyklus (jednu provedenou instrukci). Procesor se dokáže vrátit do nejvyšší provozní rychlosti v případě obdržení požadavků na přerušení, případně požadavkem na obsluhu sériového portu. [5]

3.2. Architektura

Architektura je založena na průmyslovém standartu 87C52. Procesor využívá interní registry pro ukládání a řízení periférií. Funkční schéma procesoru je na obrázku 3.2.1. [5]

3.2.1 ALU

Je využívána k provádění matematických operací a k obecnému rozhodování. Používá dva speciální funkční registry jako zdroj a cíl dat pro všechny operace. Jedná se o akumulátor a registr B. Akumulátor jako takový je zdrojem a cílem pro mnoho operací zahrnující matematiku, pohyb dat a rozhodování. Registr B je využíván při násobení a dělení. Není-li použit pro tyto účely, dá se využít jako všeobecný paměťový prostor. [5]

3.2.2 I/O PORTY

Procesor obsahuje čtyři 8-bit I/O porty. Tyto porty (brány) slouží procesoru ke komunikaci s okolím. Každý I/O port má adresu ve speciálním funkčním registru (SFR) a může z něj být čteno, nebo je možno do něj zapisovat. [5]

3.2.3 ČÍTAČE/ČASOVAČE

V procesoru jsou k dispozici tři 16-bit čítače/časovače. Každý čítač je složen ze dvou SFR (TLx a THx). Z těchto registrů je možno softwarově číst i do nich zapisovat. Funkce a mód jsou nastaveny v registrech TMOD a TCON ze SFR. [5]

3.2.4. SÉRIOVÉ PORTY

Mikrokontrolér DS89C440 obsahuje dva sériové porty. Každý port je samostatně adresovatelný a může být použit jak pro čtení, tak pro zápis dat. Každý port má své vlastní řídicí registry. Podrobnější popis je uveden v další části textu. [5]

3.2.5. ZÁSOBNÍK

Je to speciální prostor v paměti RAM, kam se v případě přerušení programu, nebo volání podprogramu, ukládají návratové adresy. Zásobník, je-li to potřeba, může být také použit k uložení akumulátoru nebo i jiných proměnných. Zásobník pracuje jako paměť typu FIFO, což znamená, že poslední uložená data jsou vyčítána jako první. Pro svou funkci potřebuje tzv. stack pointer, neboli ukazatel zásobníku, který udává adresu posledního uloženého prvku. [5]

3.2.6. FLASH PAMĚŤ

Na chipu je paměť programu realizována ve flash paměti. Tato paměť může být naprogramována pomocí interního programátoru, nebo standardním flash, případně EEPROM programátorem. [5]

3.2.7. PRACOVNÍ REGISTRY

Prvních 32 bytů z paměti RAM může být použito jako čtyři banky po osmi pracovních registrech na vysokorychlostní přesun dat. Využití jednotlivých bank je dle potřeby rychle softwarově přepínáno. Část pracovních registrů může být využito jako ukazatele (pointry) na zbylé sekce z paměti RAM. [5]

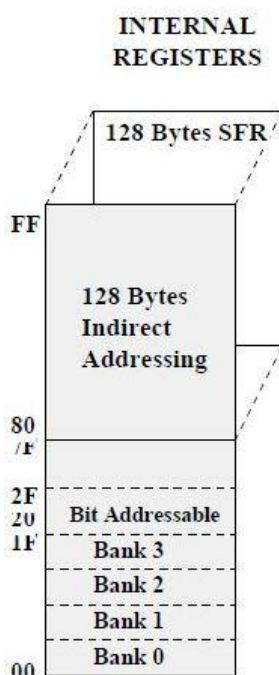
3.2.8. MAPA PAMĚTI

Mikroprocesor využívá adresování paměti, které mu dovoluje oddělit programovou a datovou paměť. Programové a datové segmenty se mohou překrývat, protože jsou dostupné různými způsoby. Program je do paměti přiveden automaticky, neexistuje žádná instrukce, která by dovolovala do této části paměti softwarově zapisovat. Datová paměť je přístupná pomocí instrukce MOVX. [5]

3.2.9. MAPA REGISTRŮ

Jak je vidět na obr. 3.2.1. mapa registrů je zcela oddělena od paměti programu i dat. Pro přístup existuje oddělená třída instrukcí. V procesoru

nalezneme 256 potenciálních registrů. V praxi můžeme k 128 registrům přistupovat přímo a k druhé půlce pouze nepřímo. [5]



Obr. 3.2.1. Schéma mapy registrů [6]

3.2.10. ADRESOVÁ/DATOVÁ SBĚRNICE

Procesor dokáže adresovat 64 kB pro paměť programu a až 64 kB pro paměť dat. Paměť dat je složena z interní paměti, která je přímo v procesoru a externí paměti, která může být připojena přes porty P0 a P2. [5]

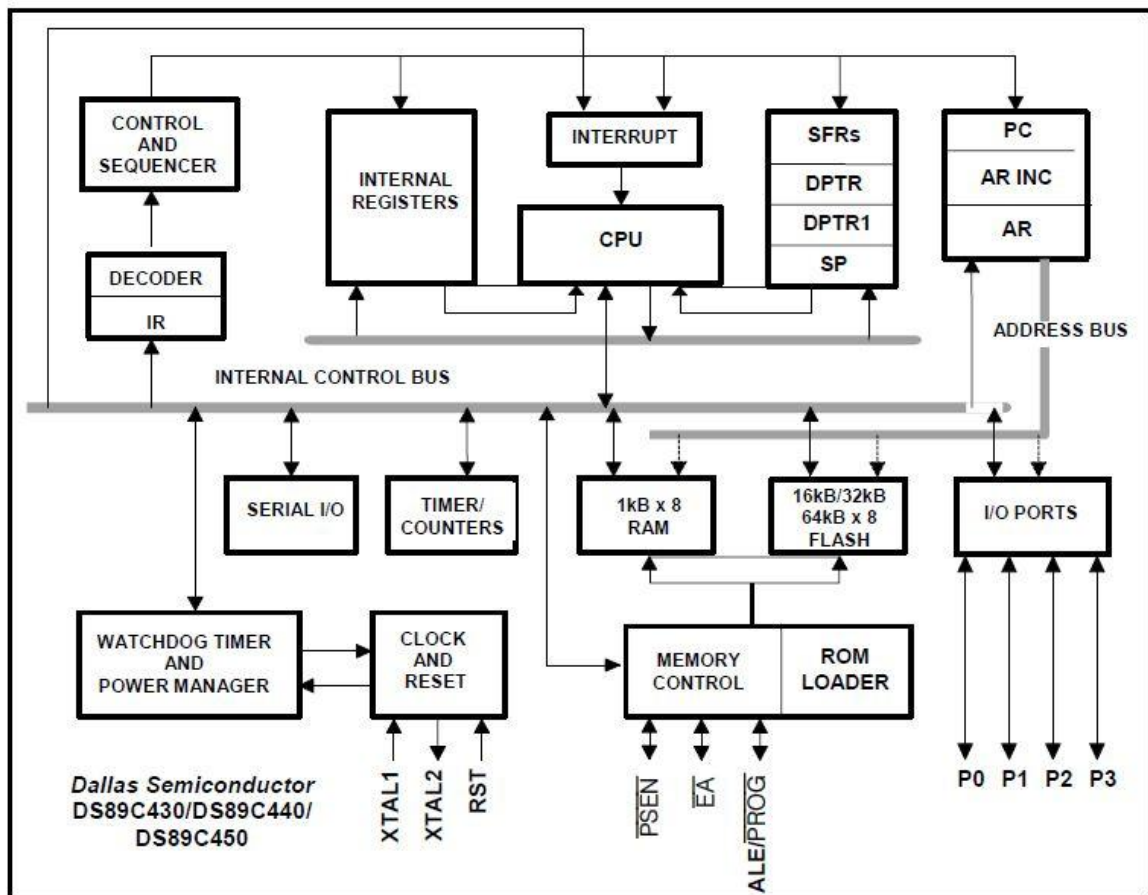
3.2.11. WATCHDOG

Je to počítačová periferie, která dokáže resetovat program v případě chyby. Tyto chyby mohou vzniknout jak softwarově, tak hardwarově. Povolím-li watchdogu chod, spustí se čítač, který je inkrementován s každým taktem krystalu. Čítač musí být periodicky resetován, jestliže nedojde k resetu, čítač přeteče a program se restartuje. Je-li watchdog správně implementován do programu a dojde k resetu, značí to, že program je v některé části zacyklen a nemůže pokračovat ve správném chodu. [5]

3.2.12. PŘERUŠENÍ

Mikrokontrolér obsahuje 13 zdrojů přerušení. Každé jednotlivé přerušení má svůj vlastní vektor, příznak přerušení a svojí vlastní prioritu. Všechna přerušení je

možno globálně povolit, případně zakázat. Podrobnější popis všech druhů přerušení je uveden v kapitole 3. 5. [5]



Obr. 3. 2. 1. Funkční schéma mikrokontroléru DS89C440

3.3. Sériové porty

Jak již bylo řečeno, mikroprocesor nabízí dva zcela nezávislé sériové porty (UART) s detekcí chyb a automatickým rozpoznáním adres. Oba porty mohou být provozovány současně ve stejných, nebo různých režimech a komunikačních rychlostech. Každý sériový port je schopen jak synchronního, tak asynchronního přenosu. V synchronním režimu mikrokontrolér generuje hodiny a provozuje UART v half-duplex módu. V asynchronním režimu je možné využít i full-duplex mód, komunikace tedy může být současně obousměrná. Přijatá data se ukládají do speciálního paměťového registru. Tyto registry umožňují sériovému portu

přijímat nová data dříve, než software přečte ta předchozí. Každý UART má přiřazen vlastní řídicí registr (SCON0, SCON1) a vlastní vysílací/přijímací registr (SBUF0, SBUF1). [5]

Každý port může být použit ve čtyřech provozních režimech. Tyto režimy nabízejí různé komunikační protokoly a různé přenosové rychlosti. Podrobný popis jednotlivých režimů je uveden v další části textu. [5]

3.3.1 MÓDY

Procesor disponuje čtyřmi módy pro sériovou komunikaci. Oba porty mohou pracovat ve stejném, nebo odlišném módu i rychlosti. [5]

Mód 0 zajišťuje synchronní komunikaci s externími zařízeními. Tento mód je běžně používán ke komunikaci se sériovými zařízeními. Přenos dat poskytuje pin RXD, zatímco pin TXD je používán pro přenos hodinového signálu. Tento mód může být použit pouze v half-duplex režimu. [5]

Mód 1 je používán pro standardní full-duplex asynchronní komunikaci. Tento mód důkladně popíši v další části textu, protože je využit v obslužném programu. [5]

Asynchronní přenos dat využívá také mód 2. Tento mód odesílá zprávy o celkové velikosti 11 bitů. Z těchto jedenácti bitů je první start bit, další devět bitů je datových a poslední bit je tzv. stop bit. Jelikož registr SBUF (SBUF1) má velikost pouze 8 bitů, poslední devátý datový bit nemůže být umístěn v tomto registru, ale nachází se v řídicím registru SCON0.1 (SCON1.3). Pokud není použit režim řízení spotřeby, přenosová rychlost je dána frekvencí oscilátoru. Tato frekvence je pro přenos dělena 64, případně 32 (nastaví se v registru SMOD). [5]

Mód 3 pracuje obdobně jako mód 2, ale přenosová rychlost je zde generována časovačem. Tento způsob dělá přenosovou rychlost libovolně volitelnou uživatelem. [5]

3.3.2 PODROBNÝ POPIS MÓDU 1

Jak již bylo napsáno výše, tento mód pracuje s asynchronním přenosem a využívá se ke komunikaci s PC, modemy a podobnými druhy zařízení. Kompletní zpráva se skládá z deseti bitů. První je start bit, dále osm datových bitů a

nakonec stop bit. Z datového balíku se nejprve odesílá bit s nejnižším významem LSB. Přenosová rychlost je generována časovačem 1 případně časovačem 2, který je v autoreload režimu. Přenos prvního bitu začíná po prvním převrácení (dopočtením do konce) čítače. Nastavení registru THx, případně TLx určuje rychlost přenosu (baud-rate). [5]

Nyní bude popsáno odeslání zprávy pomocí kanálu UART0, ale tento postup je totožný i pro kanál UART1. Odeslání zprávy začíná nakopírováním bytové informace do registru SBUF. Přenos probíhá na pinu TXD. Jak již bylo zmíněno výše, nejprve je odeslán start bit, následuje obsah registru SBUF a poslední je stop bit. Všechny bity jsou přenášeny s prodlevou, kterou stanovuje generátor baud-rate. Po odeslání posledního bitu je příznak přenosu TI hardwarově nastaven na logickou 1. [5]

Jestliže chceme sériový kanál používat i pro příjem, je nutné nastavit bit REN (SCON0.4 nebo SCON1.4) na logickou 1. Pro UART0 příjem probíhá na pinu RXD. Data jsou posouvány ve zvoleném baud-rate. Jakmile je naplněn registr SBUF, příznak příjmu RI je nastaven na logickou 1 a následně je vyvoláno přerušení. Každý bit datového paketu je 3x testován, pro odhalení potenciální chyby. Jsou-li výsledky různé, vyhrává většinová hodnota. Tímto způsobem je značně eliminována možná chyba přenosu. [5]

3.3.3 DETEKCE RÁMCOVÉ CHYBY

K rámcové chybě dochází v případě, že není detekován platný stop bit, což má za následek možné nesprávné přijetí sériového slova. UART dokáže zaznamenat rámcovou chybu a oznámit jí uživateli pomocí bitu FE_0 nebo FE_1 (SCON0.7 nebo SCON1.7). [5]

3.3.4 INICIALIZACE SÉRIOVÉHO PORTU

Aby bylo možno sériový port používat, je nezbytné ho nejprve inicializovat. Jedná se o výběr režimu, časovou základnu a v případě nutnosti inicializaci generátoru baud-rate. Pokud je vše správně nastaveno, sériová komunikace běží. V režimu 0 synchronizaci zajišťuje hodinový signál, který je generován na pinu TXD. Pro generování hodin musí být bit RI nastaven do logické 0 a bit REN do logické 1. V ostatních případech nastavení bitu REN do logické 1 umožňuje

sériový příjem, ale externí zařízení musí komunikaci zahájit odesláním start bitu.[5]

3.4. Správa a kontrola napájení

Procesor má v sobě několik funkcí, které dokáží ovlivňovat spotřebu energie a vhodně ji spravovat. Tyto funkce také dokáží kontrolovat hodnotu napájecího napětí a dokáží snížit spotřebu energie v závislosti na zatížení procesoru. Kontrola napájecího napětí má uplatnění hlavně u nespolehlivých nebo bateriových zdrojů. Pro správu napájení využívá procesor tři registry WDCON, EIE a PCON. [5]

KONTROLA NAPÁJECÍHO NAPĚTÍ

3.4.1 MONITORING NAPĚTÍ

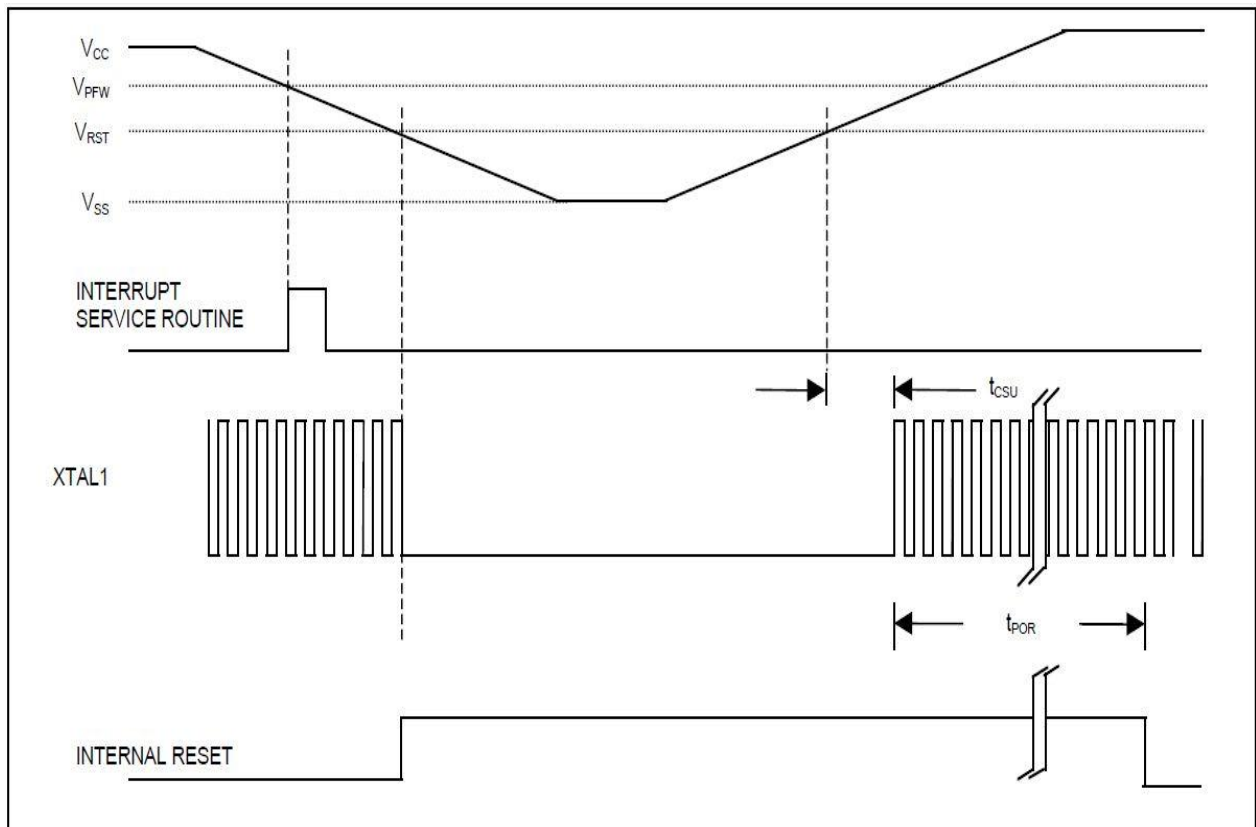
Pro přesný monitoring napětí je využita napěťová reference typu bandgap. Tato reference pracuje společně s dalšími analogovými obvody. Napájecí napětí je porovnáváno s předem stanovenými prahovými hodnotami napětí V_{PFW} a V_{RST} . Podrobné informace k těmto hodnotám a bližší specifikaci k monitoringu napětí je k nalezení v technické dokumentaci k tomuto procesoru. [5]

3.4.2 SYSTÉM VČASNÉHO VAROVÁNÍ PŘED VÝPADKEM NAPĚTÍ

Napěťová reference je v případě poklesu napětí pod kritickou mez V_{PFW} schopna generovat přerušení. K tomuto účelu je použita napěťová reference typu bandgap. Včasné varování před výpadkem napájecího napětí umožňuje systému uložit kritické parametry do energeticky nezávislé paměti a dát externí funkce do bezpečného stavu. Obsluha přerušení není generována automaticky s každým poklesem napětí pod kritickou mez, ale musí být softwarově aktivována bitem EPFI (WDCON.5). Bit PFI (WDCON.4), slouží jako příznakový bit a je nastaven na logickou 1 vždy, když je napětí menší než V_{PFW} (i v případě, že přerušení je zakázáno). Tento bit by měl být ihned po přečtení vrácen do logické 0. [5]

Jestliže napětí klesne pod mez V_{RST} , proběhne tzv. power-fail přerušení,

zařízení se automaticky zastaví a všechny výstupní porty jsou vynulovány. Tento stav trvá do doby, než napětí VCC nevzroste natolik, aby bylo možno výstupní porty napájet. Jelikož napětí V_{RST} je nižší než napětí V_{PFW} , mikrokontrolér má možnost uložit veškerá potřebná data a umístit se do tzv. bezpečného stavu ještě před zastavením činnosti. Tuto funkci nelze vypnout, je vyvolávána automaticky.[5]



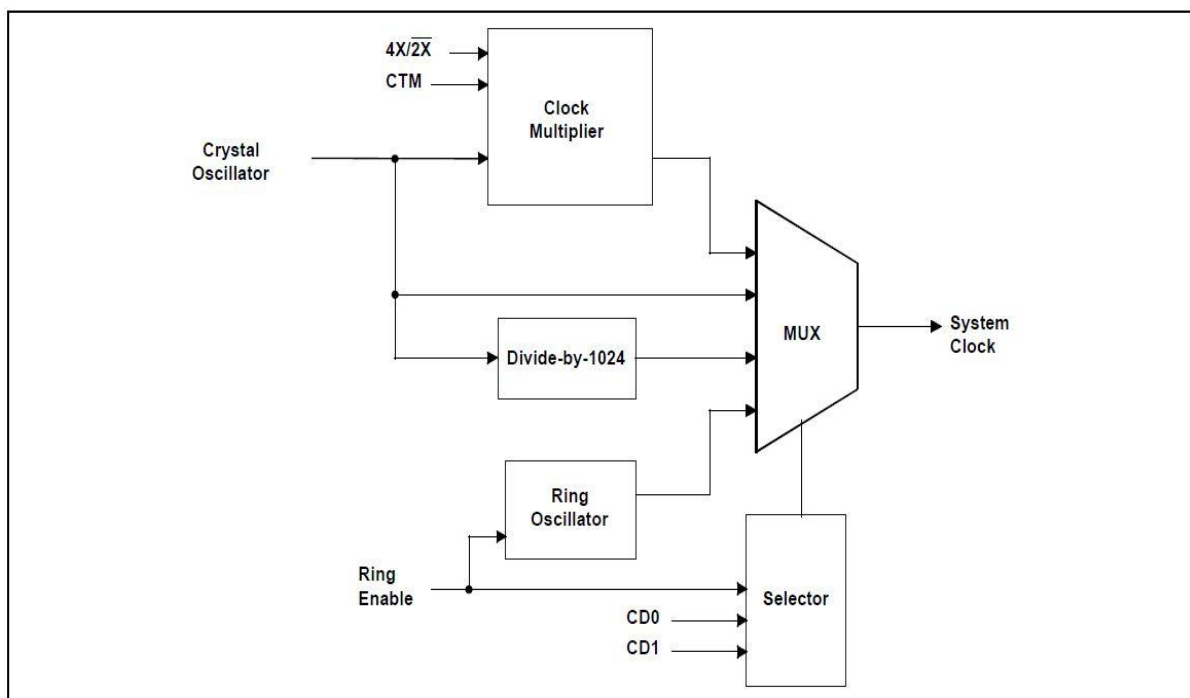
Obr. 3.4.1 Schéma kontroly napájecího napětí [5]

Pro opětovné spuštění procesoru musí napájecí napětí překročit hodnotu V_{RST} . Pokud se tak stane po dobu delší, než je 65536 taktů procesoru (z důvodu ustálení napájecího napětí), procesor automaticky obnoví svou činnost. Kompletní schéma kontroly napájení se všemi napěťovými mezemi můžeme vidět na obr. 3.4.1. [5]

SPRÁVA NAPÁJECÍHO NAPĚTÍ

3.4.3 SNÍŽENÍ KMITOČTU

Při nízkém zatížení mikrokontroléru můžeme snížit kmitočet hodinového signálu a tím i snížit příkon. V případě potřeby může být frekvence krystalu dělena 1024. Pro řízení děličky kmitočtu jsou použity bity CD0 a CD1. Při vyšším zatížení je v procesoru k dispozici i interní násobička hodinového signálu, která dovoluje procesoru vykonávat instrukce 2x nebo 4x rychleji, nežli je kmitočet krystalu. Pro příklad, mám-li krystal o jmenovitém kmitočtu 10 MHz, vykonání instrukce trvá 100 ns. V případě vynásobení hodinového signálu 4x se vykonání jedné instrukce zkrátí na 25 ns. Je-li procesor málo zatížen, mohu kmitočet vydělit 1024 a v tomto případě trvá vykonání instrukce 102,4 μ s. Celý systém řízení kmitočtu je zobrazen na obr. 3.4.2. Bližší informace k této funkci jsou k dispozici v technické dokumentaci procesoru. [5]



Obr. 3.4.2 Schéma řízení kmitočtu [5]

3.4.4 KLIDOVÝ MÓD

Klidový režim pozastaví veškeré procesy mikrokontroléru tím, že drží čítač instrukcí ve statickém stavu. Žádné hodnoty nejsou stahovány a žádné nejsou zpracovávány. Tímto stavem šetříme značné množství energie oproti plnému provozu. Výhodou tohoto režimu je, že dokáže v případě potřeby rychle reagovat na vyvolání obsluhy přerušení a vrátit procesor z klidového módu zpět do plné zátěže. Takt procesu zůstává aktivní, takže všechny časovače, sériové porty,

watchdog i systém včasného varování před výpadkem napájení pracují. Tento mód je vyvolán bitem PCON.0, který musí být nastaven do logické 1. Existují dva způsoby, jak klidový režim ukončit. Prvním z nich je jakýmkoliv druhem přerušení, kdy návratová instrukce RETI automaticky maže bit PCON.0. Klidový režim může být také ukončen vyvoláním jednoho ze tří zdrojů resetu, kdy je po obnovení činnosti procesoru bit PCON.0 taktéž vrácen do logické 0. [5]

3.4.5 STOP MÓD

Tzv. Stop mód je stav nejnižší spotřeby energie. Tohoto stavu je dosaženo vypnutím taktu procesoru. V tomto stavu není možné zpracovávat informace, spouštět časovače/čítače, ani komunikovat skrze sériové porty. Software může vyvolat tento mód nastavením stop bitu v registru PCON (PCON.1). Tento mód může být zrušen dvěma způsoby. Prvním způsobem je vyvolání přerušení, které ke svému chodu nepotřebují hodinové impulsy. Jedná se o vnější přerušení a power-fail přerušení. Aby mohlo být použito power-fail přerušení, musí být povolena reference bandgap. Po příchodu žádosti o přerušení procesor načte adresu IRS a vypne stop mód. Druhým způsobem ukončení stop módu je využitím resetu. Pro vyvolání resetu existuje pin RST. Při využití tohoto pinu je po obnovení funkce mikroprocesoru stop mód také vypnut. K vyvolání resetu nemůže být použit watchdog, protože ten ke své práci potřebuje hodinové impulsy. [5]

3.4.6 REDUKCE RYCHLOSTI

Mikrokontrolér DS89C440 je vyroben CMOS technologií. To znamená, že pro stejný výkon spotřebovává méně energie než ostatní procesory 8051. Při stejném příkonu běží software procesor v průměru 10x rychleji nežli na ostatních 8051. Chceme-li tedy snížit spotřebu energie a stačí nám výkon klasického procesoru 8051, snížíme takt 10x. Tabulka 3.4.1 ukazuje přibližné rychlosti, při kterých procesor DS89C440 dokáže plnit stejnou práci jako procesor 8051. Pro vysvětlení, v prvním sloupci je vidět rychlost krystalu, ve druhém počet zpracovaných instrukcí za sekundu pro klasickou 8051 a ve třetím potřebnou frekvenci krystalu pro zpracování stejného počtu instrukcí procesorem DS89C440. Tyto údaje musí být brány s jistou rezervou, protože skutečné

zlepšení se liší v závislosti na vykonávaných instrukcích v programu. [5]

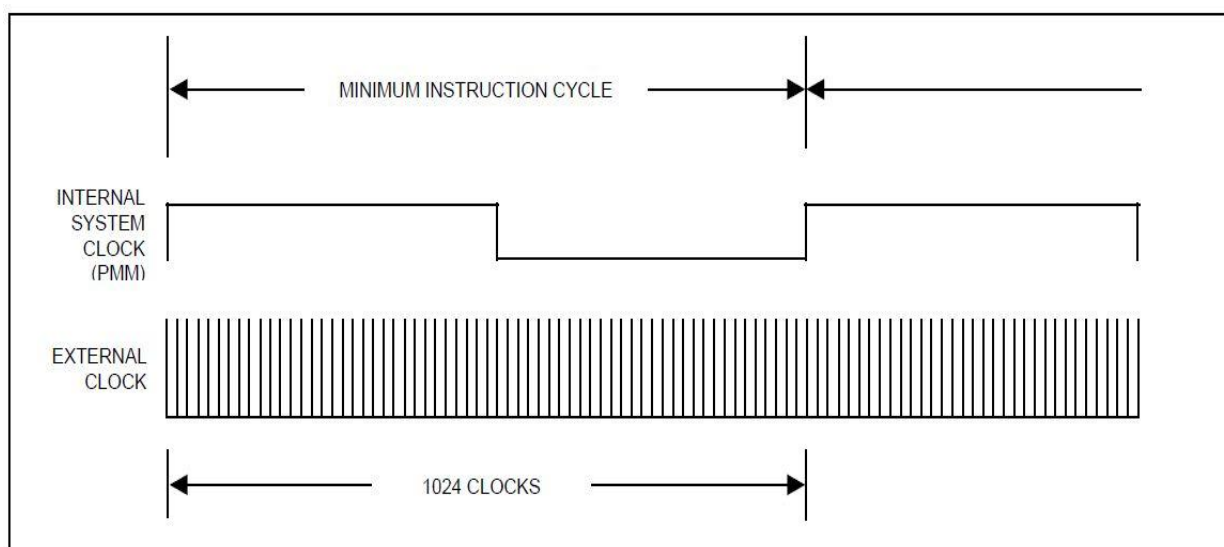
Tab. 3.4.1 Srovnání výkonu klasického procesoru 8051 s procesore DS89C440 [5]

ORIGINAL 8051 CRYSTAL SPEED (MHz)	MIPS	ULTRA-HIGH-SPEED FLASH MICROCONTROLLER CRYSTAL SPEED (MHz)
16	1.3	1.6
20	1.6	2.0
24	2.0	2.4
33	2.7	3.3
40	3.3	4.0

3.4.7 REŽIM ŘÍZENÍ SPOTŘEBY

Spotřeba energie v CMOS procesorech je funkcí provozní frekvence. Režim správy napájení (PMM) umožňuje softwaru dynamicky měnit provozní frekvenci v závislosti na vynaloženém výpočetním výkonu. Namísto standardního jednoho taktu na strojový cyklus, PMM využívá 1024 taktů na cyklus pro úsporu energie, viz obr. 3.4.3. K PMM byly přidány některé speciální algoritmy pro zlepšení běhu. Jedná se hlavně o funkci opětovného přepnutí (switchback), která dokáže takřka okamžitě vrátit procesor z režimu 1024 taktů na cyklus do režimu 1 takt na cyklus. Tato funkce je vyvolána detekcí povoleného externího přerušení, nebo přijetím start bitu od sériového portu. Switchback bude podrobně popsán v kapitole 3.4.8. [1]

Je třeba zvýšit pozornost při práci s časovači v režimu PMM. Vzhledem k tomu, že časovače odvozují svou časovou základnu z vnitřních hodin, které jsou v tu dobu ovlivněny režimem PMM, není možné generovat přenosové rychlosti pro sériové porty. V tomto případě je doporučeno využívat funkce switchback, pro rychlý návrat do standardního režimu procesoru. [5]



Obr. 3.4.3 Řízení rychlosti procesoru v režimu PMM [5]

3.4.8 SWITCHBACK

Switchback je funkce, která řeší jeden z nejdůležitějších problémů při spojení elektrické energie. Většina aplikací není schopna použít stop mód případně klidový mód, protože vyžadují nepřetržité výpočty. V obvyklém případě návrháři systému nemohou snížit provozní rychlost pod mez, kterou vyžaduje nejrychlejší událost. To znamená, že systém musí být provozován na své nejvyšší rychlosti, i když to mnohdy není nutné. Switchback umožňuje systému provoz na relativně pomalých rychlostech a rychlé přepnutí do vyšší rychlosti v případě externí události. Tato funkce je povolena nastavením bitu SWB (PMR. 5) na logickou 1. Jak již bylo napsáno výše, switchback může být vyvolán několika způsoby:

- Externím přerušením 0/1/2/3/4/5 (*Více informací o externích přerušeních můžete nalézt v technické dokumentaci k procesoru*)
- Detekováním start bitu, sériový port 0/1
- Načtením zprávy do vysílacího registru (bufferu), sériový port 0/1
- Watchdog resetem
- Spínacím (power-on) resetem
- Externím resetem

V případě nutnosti přepnutí do vyšší provozní rychlosti obsluhou sériového kanálu, není switchback aktivován přímo přidruženým přerušením. Děje se tak z toho důvodu, že v režimu PMM není schopen procesor správně detekovat

přijímací start bit. Místo toho je switchback vyvolán načtením dat do registrů SBUF a SBUF1 a s tím spojeném nastavení příznaků příchodu/odchodu zprávy (RI_0, TI_0, RI_1, TI_1). [5]

3.5. Přerušení

Přerušení umožňuje softwaru dynamicky reagovat na neplánované nebo asynchronní události. Dojde-li k přerušení, mikroprocesor očekává vyvolání rutiny přerušení (ISR). ISR je umístěn na předem stanovené adrese, jak je uvedeno v tabulce 3.5.1. Po vyvolání přerušení tedy program vykoná instrukce na předem dané adrese a vrací se zpět do hlavního programu pomocí instrukce RETI. RETI vrátí program k instrukci, která měla následovat před vykonáním přerušení. Jakmile přerušení probíhá, může být pozastaveno pouze přerušením s vyšší prioritou. Procesor má celkem 13 druhů přerušení a 5 úrovní priorit. Dojde-li k některému přerušení, je vyvolán příznak přerušení (flag). Tento příznak je pro každý druh přerušení jiný a je vyvolán bez ohledu na to, je-li přerušení povoleno, nebo ne. [5]

Jak již bylo řečeno, každé přerušení musí být individuálně povoleno, aby bylo možno vyvolání ISR. Aby přerušení proběhlo, musí být mimo nastavení tohoto individuálního povolení přerušení ještě nastaveno globální povolení přerušení EA. Je-li bit EA v logické 0, není povoleno žádné přerušení, bez ohledu na individuální nastavení jednotlivých přerušení. Jedinou výjimkou je přerušení power-fail, které vyžaduje pouze nastavení bitu EPFI. Bit EA na něj nemá žádný vliv. Všechny typy přerušení, včetně jejich příznakových bitů, maskovacích bitů a adresy můžeme vidět v tabulce 3.5.1. [5]

Tab. 3.5.1 Souhrn všech přerušení [5]

PŘERUŠENÍ	ADRESA PŘERUŠENÍ	PŘÍZNAK PŘERUŠENÍ	POVOLENÍ PŘERUŠENÍ	PŘIROZENÁ POSLOUPNOST
Power-fail přerušení	33h	PFI (WDCON.4)	EPFI (WDCON.5)	0 (nejvyšší)
Externí přerušení 0	03h	IE0 (TCON.1)	EX0 (IE.0)	1
Přetečení čítače 0	0Bh	TF0 (TCON.5)	ET0 (IE.1)	2
Externí přerušení 1	13h	IE1 (TCON.3)	EX1 (IE.2)	3
Přetečení čítače 1	1Bh	TF1 (TCON.7)	ET1 (IE.3)	4
Sériový port 0	23h	RI_0 (SCON0.0) TI_0 (SCON0.1)	ES0 (IE.4)	5
Přetečení čítače 2	2Bh	TF2 (T2CON.7) EXF2 (T2CON.6)	ET2 (IE.5)	6
Sériový port 1	3Bh	RI_1 (SCON1.0) TI_1 (SCON1.1)	ES1 (IE.6)	7
Externí přerušení 2	43h	IE2 (EXIF.4)	EX2 (EIE.0)	8
Externí přerušení 3	4Bh	IE3 (EXIF.5)	EX3 (EIE.1)	9
Externí přerušení 4	53h	IE4 (EXIF.6)	EX4 (EIE.2)	10
Externí přerušení 5	5Bh	IE5 (EXIF.7)	EX5 (EIE.3)	11
Watchdog přerušení	63h	WDIF (WDCON.3)	EWDI (EIE.4)	12

3.5.1 ZDROJE PŘERUŠENÍ

Zdroje přerušení lze rozdělit do několika kategorií: externí zdroje, zdroje na bázi časovačů, přerušení sériovou komunikací a selhání napájení. Každý typ má specifickou adresu, masku, příznak i prioritu. V našem případě pracujeme s přerušením od sériové komunikace a z důsledku selhání napájení. [5]

PŘERUŠENÍ OD SÉRIOVÉ KOMUNIKACE

Každý sériový port je schopen generovat přerušení a má vlastní povolení

přerušení, adresu, příznak i prioritu. Jednotlivé porty mají dva příznaky přerušení (RI, TI). Tyto příznaky používá ISR aby zjistil, zda přerušení přichází od příjmu, nebo od vysílání. Když UART dokončí přenos zprávy, nastaví se bit TI a je generováno přerušení (pokud je povoleno). Příznak RI je generován při obdržení zprávy. Oba příznaky nejsou generovány, dokud není zpráva kompletně přijata (RI), případně odeslána (TI). [5]

PŘERUŠENÍ Z DŮSLEDKU SELHÁNÍ NAPÁJENÍ

Mikrokontrolér má schopnost generovat přerušení pokud napájecí napětí VCC klesne pod stanovenou mez. Porovnáním aktuálního napájení s interní referencí mikrokontroléru můžeme posoudit, zdali napětí kleslo pod mez VPFW. Pokud se tak stane, je vyvolána rutina přerušení (pokud je povolena). Power-fail přerušení zůstává v platnosti tak dlouho, dokud VCC zůstává pod hladinou VPFW. Toto přerušení má nejvyšší prioritu, která nemůže být uživatelem změněná. Jak již bylo zmíněno, toto přerušení je povoleno výhradně bitem EPFI a na globálním povolovacím bitu EA je zcela nezávislé. [5]

3.5.2 PRIORITY PŘERUŠENÍ

Mikrokontrolér má pět prioritních úrovní přerušení. Nejvyšší priorita má číslo 4 nejnižší 0. Power-fail přerušení, pokud je povoleno, má přidělenou nejvyšší prioritu 4, zatímco zbývající přerušení lze individuálně nastavit mezi 3, 2, 1 nebo 0 prioritní úroveň. Obsluha rutiny přerušení může být pozastavena pouze přerušením s vyšší prioritou. Jsou-li současně vyvolány dva zdroje přerušení se stejnou prioritou, rozhoduje přirozená posloupnost, viz tab. 3.5.1. [5]

3.5.3 REAKČNÍ DOBA OBSLUHY PŘERUŠENÍ

Reakční doba obsluhy přerušení je závislá na více parametrech a pohybuje se v rozmezí 4 až 18 cyklů procesoru. S přehledem nejvyšší latence dosahuje přerušení, pokud právě pobíhá jiné přerušení se stejnou nebo vyšší prioritou. V tomto okamžiku musí být nejdříve vykonáno právě probíhající přerušení a až poté může být vyvoláno přerušení s nižší prioritou. V ostatních případech je doba odezvy závislá na právě probíhající instrukci. Nejrychlejší reakční dobou jsou čtyři cykly, kde 1 cyklus je na detekci žádosti o přerušení a 3 cykly na provádění

instrukce LCALL, která volá vlastní obsluhu přerušení. Naopak maximální prodleva vzniká při provádění instrukce JBC, která maže bity IE, IP0, EIE, nebo EIP, a pak vykoná instrukci DIV (dělení). Tato sekvence trvá celkově 18 cyklů a zahrnuje 1 cyklus pro detekci přerušení, 4 cykly pro dokončení instrukce JBC, 10 cyklů pro vykonání instrukce DIV a 3 cykly pro instrukci LCALL. Tato maximální doba odezvy nepředpokládá, že ve frontě existuje přerušení s vyšší prioritou, ani že právě nějaké přerušení s vyšší nebo stejnou prioritou probíhá. [5]

4. Obslužný program

Pro správný chod zařízení bylo nutné vytvořit obslužný program. Napsání programu pro procesor DS89C440 byl základní cíl mé diplomové práce. Program je psaný v jazyce C, jako vývojové prostředí byl použit program μ Vision 4 [7]. Pro vlastní (fyzické) nahrání programu do procesoru byl použit program Loader420 [8] a jako terminálový vstup z PC program Hercules [9].

4.1. Funkce programu

Program jako celek funguje tak, že přijme specifický požadavek od uživatele přes terminálový vstup. Po přijetí zprávy ji porovná s existujícím seznamem příkazů, a pokud se příkaz shoduje, vygeneruje specifický formát zprávy, který odešle elektronické zátěži. Pokud se jedná o příkaz, který nevyžaduje odpověď, elektronická zátěž jej provede a vyčká na další příkazy. Pokud se jedná o požadavek, na který je vyžadována odpověď, program vyčkává na zprávu od zátěže. Tato zpráva se skládá z 26 bytů. Po přijetí posledního bytu je zpráva zpracována a výsledek je zobrazen v programu Hercules. Po zobrazení výsledku program opět očekává zprávu od uživatele. Jak již bylo psáno v kapitole 1.6, tento překlad se provádí hlavně pro komfortní, uživatelsky příjemné ovládání přístroje. Nejdůležitější příkazy včetně popisu jejich funkce můžeme nalézt v tabulce 4.2.1.

Tímto způsobem lze implementovat jednodušší požadavky přístroji jako zapnutí/vypnutí vstupu, nastavení/čtení pracovního módu, apod. Elektronická zátěž ale umí vykonávat také složitější operace, které vyžadují číselný parametr. Takovým příkazem je například nastavení maximálního vstupního napětí, kde

nastavuji jednu hodnotu, případně nastavení a spuštění transienční zkoušky, kde nastavuji dokonce hodnoty čtyři (u režimu continuous). Číselné hodnoty a případně i nastavovací parametry se posílají ve 4. až 25. bytu zprávy. Každá nastavovaná/měřená veličina má jinou minimální hodnotu. Například pro elektrické napětí je minimální hodnotou 1mV. Chceme-li tedy na elektronické zátěži nastavit např. 5 V, musí tento údaj program nejprve vynásobit 1000krát, a poté převést do hexadecimálního formátu. Číselná hodnota elektrického napětí si vymezuje čtyři byty z celkové velikosti zprávy. Číslo, jako parametr u jakékoliv veličiny, je vždy reprezentováno tak, že první byte z vymezeného prostoru je branný jako nejnižší a poslední byte jako nejvyšší. [1] Pro příklad, mám-li tedy nastavit 5V, nejprve údaj převedu na milivolty, což je 5000 a poté číslo převedu do hexadecimální soustavy. Výsledné číslo je 00001388H (údaj je zpředu doplněn o nuly, aby bylo lépe patrné, že je nutné poslat 4 byty). Nejnižší byte z tohoto čísla má hodnotu 88H, je tedy odeslán jako první, poté byte 13H a následně dva nulové byty. Obdobný proces probíhá i obráceně, kdy zátěž odesílá hodnoty a program je musí převést zpět do dekadické soustavy a zobrazit na terminále PC. Jak již bylo zmíněno, každá veličina má jinou minimální hodnotu a v případě času i jiný vymezený prostor. Tyto údaje můžeme porovnat v tabulce 4.1.1.

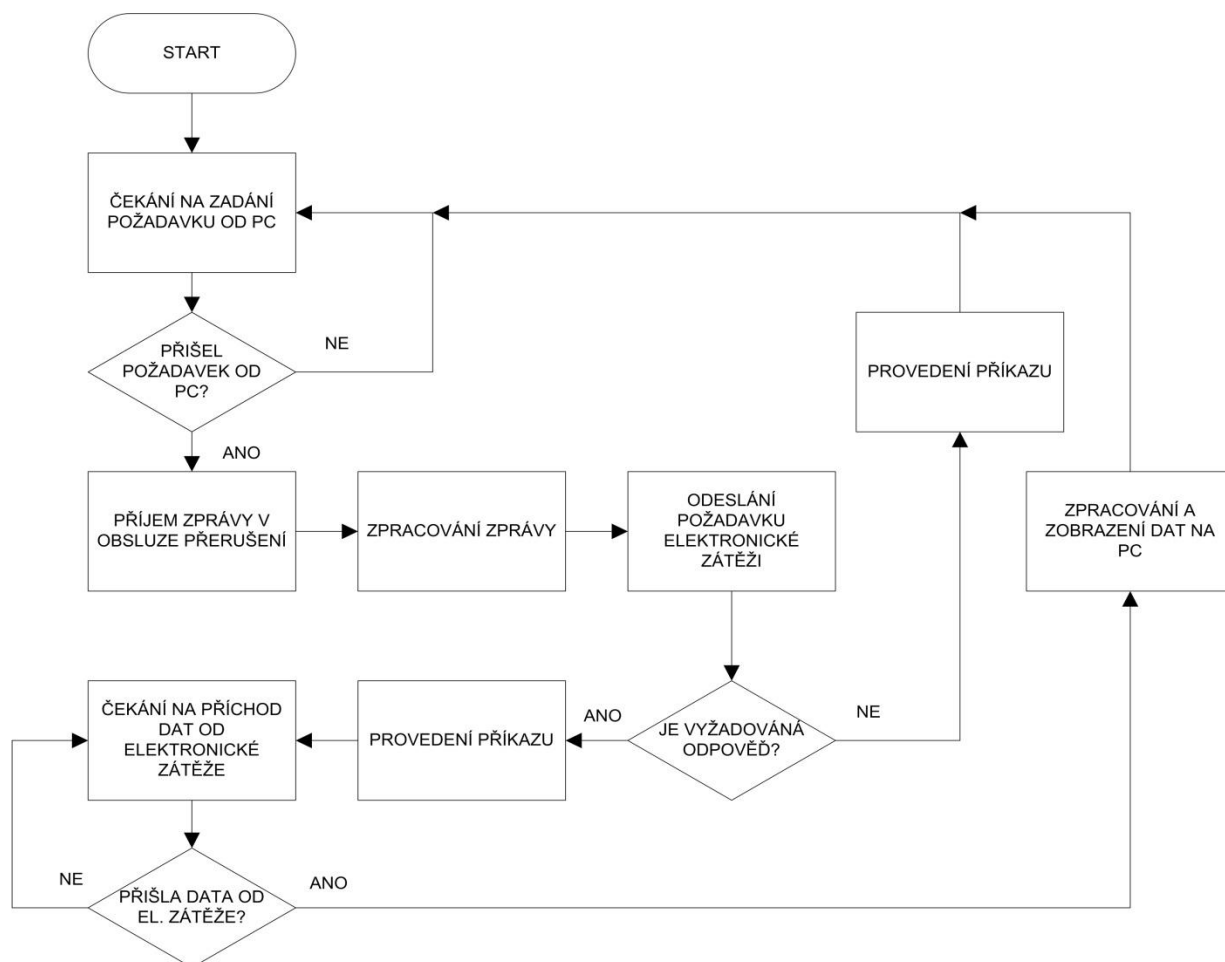
Tab. 4.1.1. Krajní parametry po jednotlivé veličiny

VELIČINA	MIN. HODNOTA	PŘIDĚLENÝ ROZSAH	POTENCIÁLNÍ MAX. VELIKOST
Elektrické napětí	1 mV	4 byty	4,29 MV
Elektrický proud	0,1 mA	4 byty	429,49 kA
Elektrický odpor	1 mΩ	4 byty	4,29 MΩ
Elektrický výkon	1 mW	4 byty	4,29 MW
Čas	0,1 ms	2 byty	6,55 s

Z tabulky je patrné, že pro každou veličinu je přidělený rozsah a z toho vyplývající i potenciální maximální velikost dostatečná s obrovskou rezervou. Pro představu maximální vstupní napětí pro náš typ zátěže je 120 V (pro typy IT8511B a IT8512B je maximální napětí 500V) a v tabulce vidíme, že potenciální možné napětí je 4,29 MV. [1]

4.2. Detailní popis programu

Na obrázku 4.2.1 vidíme vývojový diagram pro obslužný program. Vysvětlení všech bloků včetně názvů funkcí, přerušení, výpočty pro přenosové rychlosti, nastavení řídicích registrů, aj. si vysvětlíme v další části této kapitoly.



Obr. 4.2.1. Vývojový diagram

4.2.1 VÝPOČET BAUD-RATE A NASTAVENÍ ŘÍDICÍCH REGISTRŮ

V prvním bloku vývojového diagramu nejprve nastavíme řídicí registry sériového portu 0, 1 (SCON, SCON1) a poté řídicí registr časovačů TMOD. Řídicí registry SCON a SCON1 nastavíme shodně tak, aby mohli odesílat i přijímat a aby pracovali v modu 1, který je popsán v kapitole 3.3.2. Registr TMOD je nutné nastavit z důvodu generování přenosové rychlosti obou sériových portů. K těmto účelům je použit časovač 1 a časovač 2 (časovač 0 neumí pracovat jako

generátor baud-rate). Pro generování přenosové rychlosti sériového portu 0 je plněn pouze horní registr čítače TH1. Výpočet přenosové rychlosti v modu 1 pak vypadá takto:

$$TH1 = 256 - \frac{2^{SMOD} * f.krystalu}{384 * \text{přenosová rychlost}}$$

Přenosová rychlost pro sériový kanál 0 byla zvolena 115200 Bd. Frekvence krystalu je 11,0592 MHz a SMOD se rovná 0. Pro tyto hodnoty pak TH1 vychází 253. Pro generování přenosové rychlosti 115200 Bd tedy musíme přednastavit registr TH1 na hodnotu 253 (0xFD). Pro generování přenosové rychlosti u sériového portu 1 je nutné naplnit dva speciální registry čítače 2. Jedná se o registry RCAP2L a RCAP2H. Pro výpočet byl použit vzorec:

$$RCAP2H, RCAP2L = 65536 - \frac{\frac{f.krystalu}{2}}{16 * \text{přenosová rychlost}}$$

U tohoto kanálu byla použita rychlost 9600 Bd. Výsledné číslo vyšlo 65500, což znamená, že při převodu na hexadecimální tvar bylo do horního bytu RCAP2H vloženo číslo 0xFF a do dolního bytu RCAP2L číslo 0xFD.

Nakonec bylo nutné ještě povolit přerušení od sériového portu 0, 1 a nastavit centrální bit, který povolí všechna přerušení (EA).

4.2.2 PŘÍJEM ZPRÁVY OD PC

Poté co jsou nastaveny všechny nezbytné parametry, mikrokontrolér čeká na příjem požadavku od PC. Jakmile přijde zpráva, mikrokontrolér ji pozná tak, že je nastaven příznakový bit RI1 a je aktivováno přerušení od sériového kanálu 1. Po aktivaci přerušení program vykonává instrukce z adresy 0x3B, kde je umístěna rutina obsluhy přerušení, viz tabulka 3.5.1.

V obsluze přerušení nejprve program vymaže příznakový bit RI, a následně vyčte hodnotu z registru SBUF, kde je vlastní obsah zprávy a uloží ji do proměnné typu unsigned char. Poté proběhne kontrola, zdali se nejedná o malé písmeno. Pokud ano, program znak převede na velké písmeno. Následně je znak uložen do jednoho ze čtyř připravených polí. Jedná-li se o první znak, je automaticky uložen do pole typu unsigned char jménem příjem. Po dokončení se

program vrací zpět z rutiny přerušení a očekává další znak. Tento průběh pokračuje do té doby, než přijde znak dvojtečky. Po tomto znaku se příchozí symboly začnou ukládat do dalšího z připravených čtyř polí. Po znaku mezery je očekáváno číslo (případně čísla oddělena čárkou), které je automaticky uloženo do pole hodnota. Celý požadavek je ukončen klávesou enter. Některé příkazy můžeme vidět v tabulce 4.2.1., kompletní příkazová sada je součástí přílohy.

Tab. 4.2.1 Ukázka některých příkazů

POPIS FUNKCE	PŘÍKAZ
Zapnutí/vypnutí vstupu přístroje	INPUT:ON/OFF
Nastavení max. vstupního napětí	SET:MAX:VOLTage „číslo“
Čtení max. vstupního napětí	MEASure:MAX:VOLTage
Nastavení módu přístroje	SELECT:CCmode/CVmode/CWmode/CRmode
Nastavení proudu v CC módu	SET:CURRent „číslo“
Čtení proudu v CC módu	MEASure:CURRent
Nastavení transientní zk. v CW módu (mód continue)	SET:TRANWATT: CONTINUE „4x číslo“

4.2.3 POROVNÁNÍ OBRŽENÉ ZPRÁVY S PŘÍKAZOVOU SADOU

Po ukončení zadání požadavku klávesou enter, jsou všechna pole, tj. všechny parametry příkazu, porovnána s předpřipravenou příkazovou sadou. Proběhne-li shoda, program vygeneruje 26-bytovou zprávu, která odpovídá danému požadavku. Tento řetězec je uložen v poli s názvem odchod. Jako poslední 26. byte je odeslán kontrolní součet všech předchozích bytů, pro kontrolu přenosu. Jako 27. byte je do zprávy ještě umístěn znak enter, který značí konec zprávy (tento znak není odeslán elektronické zátěži). Je-li požadavek zadán chybně, uživateli se vrátí zpráva: „CHYBNE ZADANY PRIKAZ“. Celý tento proces probíhá ve funkci s názvem porovnaní.

5.2.4. ODESLÁNÍ VYGENEROVANÉ ZPRÁVY PŘÍSTROJI

Pokud celý proces proběhl správně, program vygeneruje zprávu odpovídající příkazu. Jak bylo zmíněno, zpráva má velikost 26 bytů a elektronické zátěži je poslána přenosovou rychlostí 9600 Bd. Příkaz je odeslán zátěži ve funkci s názvem posli. Do této funkce se řetězec dostane pomocí pointeru *message. Program nejprve vymaže bit signalizující příznak přenosu TI1, následně nakopíruje první adresu, na kterou odkazuje pointer *message, posune pointer

na další adresu a čeká, dokud není příznakový bit T11 nastaven na logickou jedničku. Tento bit signalizuje, že je přístroj připraven na přijetí dalšího bytu zprávy. Proces se opakuje do doby, než program narazí na znak enter. Znak enter značí konec zprávy, program se vrací z funkce zpět do hlavního programu a v závislosti na typu požadavku očekává další zprávu od uživatele, nebo odpověď od elektronické zátěže.

5.2.5. ODPOVĚĎ ELEKTRONICKÉ ZÁTĚŽE NA POŽADAVEK

Elektronická zátěž od uživatele dostává dva typy příkazů. Při prvním z nich příkaz provede, ale nic uživateli nevrací. Při druhém typu nic nevykonává, ale vrací uživateli například nastavené hodnoty. V tomto případě je tedy obdrženo od elektronické zátěže taktéž 26 bytů dlouhé slovo, která musí být programem zpracována. Užitečná část zprávy je poté zobrazena na terminálu PC. Elektronická zátěž je připojena na sériový port 1. V případě, že dojde k přenosu informace od elektronické zátěže, příznakový bit RI1 je nastaven a program pokračuje na adrese 0x23. Na této adrese se nachází obsluha přerušení pro sériový port 0.

V obsluze přerušení program nejprve vymaže příznakový bit RI1 a následně nakopíruje hodnotu z registru SBUF do pole jménem buff. Následně program čeká na znovunastavení bitu RI1 a uložení dalšího bytu zprávy. Jakmile je nakopírováno všech 26 bytů, je zpráva dále zpracovávána.

Při zpracování zprávy je nejprve nutné posoudit, zdali je užitečná informace slovo, nebo číslo. Slovem, nebo výrazem odpovídá program v případě, kdy se uživatel ptá například na pracovní režim elektronické zátěže. Naopak číselnou hodnotu vrací elektronická zátěž v momentech, kdy uživatel zjišťuje například maximální vstupní napětí. V případě, kdy je jako návratová hodnota očekáván výraz, je nejprve nutné zjistit typ příkazu. Typ příkazu je zjištěn pomocí 3. bytu zprávy, který je pro každý příkaz unikátní. V obsluze přerušení podle tohoto bytu a podle informační části zprávy (4. až 25. byte) vygeneruje odpověď. V případě, že návratovou hodnotu má být číslo, typ příkazu není důležitý. V obsluze přerušení je pouze číslo zpracováno, převedeno do dekadického tvaru a upraveno tak, aby jej bylo možno odeslat zpět na terminál PC.

5. Závěr

Hlavním cílem mé práce bylo vytvoření obslužného programu pro zřízení komunikace mezi elektronickou zátěží a terminálem PC. Nejprve bylo nutné seznámit se s parametry elektronické zátěže IT8512C. Pro splnění cílů diplomové práce bylo potřeba pochopit hlavně komunikaci zátěže přes I/O periférii, pracovními módy přístroje a testovací zkoušky. V další fázi jsem se zaměřil na zdokonalené komunikační zařízení, které slouží jako mezistupeň pro přenos informací mezi PC a elektronickou zátěží. Kompletní schéma zařízení nalezneme v příloze 2. Největší důraz jsem kladl na seznámení se s mikrokontrolérem DS89C440, který celé komunikační zařízení ovládá. Tento mikrokontrolér je postaven na architektuře mikroprocesorů 8051, ale má několik atypických vlastností, díky nimž byl do zařízení vybrán. Zejména se jedná o dva sériové porty a při stejném taktu i vyššímu výkonu procesoru. Dvě sériová rozhraní, v našem případě rozhraní UART, jsou pro tento typ zařízení nezbytnou podmínkou, neboť jeden port obsluhuje PC a druhý port elektronickou zátěž.

Při vytváření obslužného programu jsem nejprve zprovoznil komunikaci mezi oběma zařízeními a následně jsem do kódu přidal funkci, která přijatou zprávu zpracuje a vygeneruje z ní zprávu pro elektronickou zátěž. V poslední fázi jsem v kódu ošetřil chyby. Při vytváření jsem narazil na několik problémů, ve kterých se vesměs jednalo o nastavení sériových kanálů a výpočty přenosových rychlostí.

6. Literatura

[1] Elektronická zátěž série IT8500. [online]. 2012 [cit. 2014-05-06]. Dostupné z: <http://ebookbrowse.net/it8500-um-en-pdf-d429735173>

[2] Manuál IT8512. [online]. 2004 [cit. 2014-05-06]. Dostupné z: <http://www.rigolosziloskop.de/upload/info/2013031573197425.pdf>

[3] FT232R katalogový list. [online]. 2010 [cit. 2014-05-06]. Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

[4] *Optočlen PC817*. [online]. 2006 [cit. 2014-04-23]. Dostupné z: http://www.soselectronic.ro/a_info/resource/d/pc817.pdf

[5] DALLAS Ultra-High-Speed Flash Microcontroller. [online]. 2013 [cit. 2014-05-06]. Dostupné z: http://www.findthatpdf.com/search-96780566-hPDF/download-documents-ds89c420_uq.pdf.htm

[6] DS89C440 katalogový list. In: [online]. 2010 [cit. 2014-05-06]. Dostupné z: http://www.datasheet-pdf.com/datasheet-html/8/9/C/89C450_DallasSemiconducotr.pdf.html

[7] μ Vision 4. In: [online]. 2010 [cit. 2014-05-06]. Dostupné z: <http://www.keil.com/>

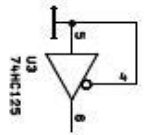
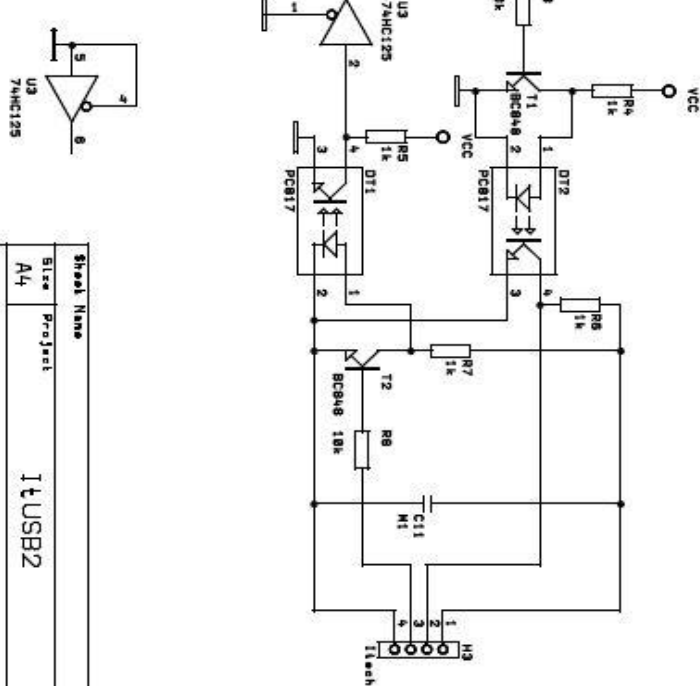
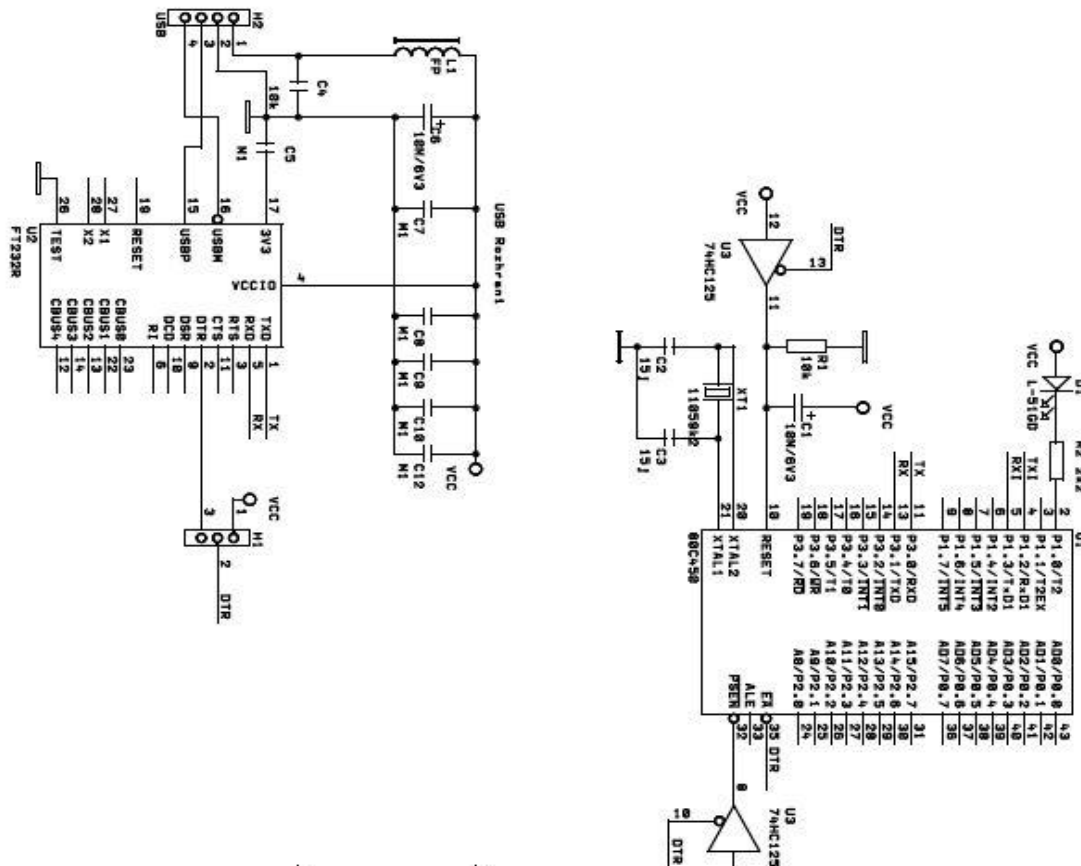
[8] Loader420. In: [online]. 2002 [cit. 2014-05-06]. Dostupné z: <http://www.silaresearch.com/data/loader420.zip>

[9] Hercules 3-2-6. In: [online]. 2010 [cit. 2014-05-06]. Dostupné z: http://www.hw-group.com/products/hercules/index_cz.html

7. Seznam příloh

- [1] Schéma komunikačního zařízení
- [2] Kompletní seznam příkazů
- [3] CD s obslužným programem

Příloha 1



Sheet Name			
Size	A4		
Project	I USB2		
Rev.	REV		
Date	APR-7-2008	Sheet	1 / 1

Příloha 2

POPIS FUNKCE	PŘÍKAZ
Ovládání na předním panelu/dálkové ovládání	REMOTECONTROL:ON/OFF
Zapnutí/vypnutí vstupu přístroje	INPUT:ON/OFF
Nastavení max. vstupního napětí	SET:MAX:VOLTage „číslo“
Nastavení max. vstupního proudu	SET:MAX:CURRent „číslo“
Nastavení max. vstupního výkonu	SET:MAX:POWER „číslo“
Čtení max. vstupního proudu	MEASure:MAX:CURRent
Čtení max. vstupního napětí	MEASure:MAX:VOLTage
Čtení max. vstupního výkonu	MEASure:MAX:POWER
Nastavení módu přístroje	SELECT:CCmode/CVmode/CWmode/CRmode
Nastavení proudu v CC módu	SET:CURRent „číslo“
Nastavení výkonu v CW módu	SET:POWER „číslo“
Nastavení odporu v CR módu	SET:RESISTance „číslo“
Nastavení napětí v CV módu	SET:VOLTage „číslo“
Čtení proudu v CC módu	MEASure:CURRent
Čtení napětí v CV módu	MEASure:VOLTage
Čtení výkonu v CW módu	MEASure:POWER
Čtení odporu v CR módu	MEASure:RESISTance
Nastavení transientní zk. v CC módu (mód continue)	SET:TRANCURRent: CONTINUE „4x číslo“
Nastavení transientní zk. v CV módu (mód continue)	SET:TRANVOLTage: CONTINUE „4x číslo“
Nastavení transientní zk. v CR módu (mód continue)	SET:TRANRESISTance: CONTINUE „4x číslo“
Nastavení transientní zk. v CW módu (mód continue)	SET:TRANWATT: CONTINUE „4x číslo“
Čtení transientních parametrů v libovolném módu	MEASure:TRANPARAMeter
Nastavení módu v operačním listu	SELECT:OPERATELIST:CCmode/CVmode/CWmode/CRmode
Provádění operačního listu jednou/vícekrát	SET:REPEATLIST:ONCE/REPEAT
Pojmenování operačního listu	SET:LISTNAME: "jméno"
Nastavení min. napětí v testu baterie	SET:BATT:MINVOLT "číslo"
Nastavení komunikační adresy	SET:ADDRESS: "adresa" (0x00 - 0xFE)
Nastavení funkčního módu	SELECT:FUNMODE:FIXED/SHORT/TRAN/LIST/BATTERY
Režim lokálního ovládání povolit/zakázat	LOCALCONTROL:ENABLE/DISABLE
Vybrání zdroje signálu trigger	SELECT:TRIGGER:KEYPAD/EXTERNAL/COMMAND
Odeslání trigger signálu	TRIGGERSIG
Vstup do kalibračního módu	CALIBRATE:ENABLE/DISABLE
*Odeslání aktuálního vst. napětí kalibračnímu módu	SEND:VOLTage
*Odeslání aktuálního vst. proudu kalibračnímu módu	SEND:CURRent
*Návrat do defaultního nastavení	SEND:CALIBRATE:DEFAULT

*příkaz funguje, je-li zařízení v kalibračním módu