Web51 - základní vývojový manuál

Co to je Web51

Web51 je komplexní projekt v rámci kterého je připojen procesor z rodiny Intel x51 k síťovému řadiči Realtek RTL8019AS. Tím vzniká velmi levné a snadno použitelné rozhraní do sítě Ethernet. Použitím velmi známého procesoru x51 se otevírá možnost i pro malé a střední firmy získat svým zařízením přístup do síťových technologií.

Na jaké aplikace je určeno řešení Web51

Popisovaný systém je navržen od samého počátku s optimalizací vzhledem k ceně s tím, že je určen pro hardwarově nenáročné např. telemetrické aplikace s tím, že prodloužení linky RS232 / 422 / 485 by mělo být zhruba horním stropem výkonnostních možností systému. Řešení stlačuje cenu na minimum s tím, že to je vykoupeno složitostí vývoje aplikací. Vhledem k ceně času vývojáře atd.. se Web51 hodí, pokud chcete použít řešení v cca 20 – 50 kusech konkrétní aplikace. Nevýhodou systému je tvorba programu SW řešení aplikace v assembleru, což však pro jednoduché aplikace nemusí být problém. S vývojovým kitem navíc získáte komplexní sbírku příkladů, která ukazuje některé možnosti systému Web51, ale hlavně vám umožní rychlé seznámení se systémem.

Web51 - VOLNÝ PROJEKT?
Lze na něm vyvíjet komerční aplikace?

Projekt Web51 existuje **ve dvou licencích** a tak je třeba na něj nahlížet. Základní rozdělení se liší typem licence a zdrojovými kódy jedné z knihoven. Část komunikace TCP/IP je uložena v knihovně **libk**, jejíž zdrojové kódy dodáváme pouze s naším DESIGN KITem. Ve volném balíku je ale obsažena přeložená verze této knihovny v několika verzích, takže je možný plný vývoj i na základě volné verze pod GPL.

Volná licence splňující podmínky GPL:

Volná verze Web51 je plně zdokumentována a můžete si ji stáhnout zdarma ve zdrojových kódech (mimo zdrojových kódů knihovny libk) v sekci DOWNLOAD, včetně zdrojových kódů příkladů a části kódu knihoven.

- Pod touto licencí je šířen celý projekt (včetně vývojových nástrojů z BINUTILS), **mimo** knihovnu **libk**.
- Pokud použijete zveřejněné zkompilované knihovny v binární formě, neměli by jste podle podmínek licence znepřístupnit vaše zdrojové kódy.
- Rádi bychom věděli o vašich následných projektech. Prosíme o upozornění, že jste na základě volné verze něco vytvořili. Nejlépe formou odkazu na WWW stránky.
- Máme zájem podporovat i odvozené projekty. Po dohodě vám můžeme nabídnout podporu projektu, zveřejnění na našem WEBu atd..
- **Veškeré následné projekty však musí zůstat nadále volné**. Volnou verzi je pod licencí GPL možné použít **bez další dohody** pouze pro osobní nebo nekomerční projekty.
- Při splnění uvedených podmínek, můžete projekt používat bez dalších omezení, nebo autorských poplatků z vyrobeného kusu.

Komerční licence s plnými zdrojovými kódy:

Volná část splňuje podmínky GNU licence GPL. Vzhledem k tomu, že jsme původními autory projektu, nevylučuje nám použití GPL licence šířit projekt i pod jinou, v tomto případě komerční licencí.

- Při zakoupení vývojového KITu s plným SW, získáváte komerční licenci projektu, **včetně** knihovny **libk** se všemi zdrojovými kódy.
- Komerční licence pokrývá celý projekt, včetně příkladů (mimo překladačů a podobných vývojových nástrojů).
- V rámci DESIGN VERZE kupujete licenci na prvních 20 kusů komerční aplikace. Nad těchto 20 kusů nám přísluší dle podmínek licence odměna za využití našeho kódu. Výše této částky byla stanovena na **1$** za každý vyrobený komerční výrobek, obsahující části projektu Web51!
- Po vzájemné dohodě vám v rámci poplatku za komerční použití můžeme garantovat vaši unikátní MAC adresu z námi zakoupeného prostoru MAC adres pro celý projekt Web51.
- Pro komerčně vyvinuté aplikace samozřejmě neplatí výše uvedená ustanovení GPL.

HW server bude nadále zveřejňovat nové verze příkladů a kódu systému Web51. Volně zveřejňovány však budou nadále pouze ovládací rutiny, moduly komunikace, řešení konkrétní problematiky atd.. Neuvažujeme o uvolnění našich ucelených aplikací, které paralelně distribuujeme v rámci aplikačních LITE verzí.

Distribuce Web51

Web51 je otevřený systém a můžete jej použít několika způsoby :

- Inspirujte se volným kóde a postavte si vlastní aplikaci na **vlastním HW a SW**.
- Kupte si náš vývojový systém s KITem, navrhněte vlastní HW a postavte si **svoji HW aplikaci**.
- Navrhněte svoji aplikaci na naši LITE VERZI (aplikační řešení) a nadále řešte **HW jen speciálních periférií.** Po dohodě vám dlouhodobě garantujeme dodávky modulů.
- **Potřebujete hotové řešení?** Kupte si naši LITE VERZI s kompletní aplikací (základní informace dále, podrobnosti zašleme, kontaktujte nás *Web51@HW.cz*)

Pro vývojové účely je určena DESIGN verze v několika modifikacích

- Web51 - **designer full** - plná verze včetně aplikační podpory
- Web51 - **designer support** - samostatná aplikační podpora (možnost zpětného dokoupení)
- Web51 - **designer lite** - HW kit + veškeré zdrojové kódy a dokumentace.
- **LPT ISP KIT** - programátor Procesorů ATMEL po ISP rozhraní.

Pro všechny produkty z **DESIGN** řady platí dodávka **plně otevřeného kódu** a **licence na 20 komerčních aplikací**. Nad tento rámec je cena licence cca 40 Kč / 1 ks zařízení (***včetně unikátní MAC ADRESY***).

Zákazníkům, kteří si postaví na Web51 jakýkoli vývoj můžeme po dohodě **GARANTOVAT DODÁVKY** a nadále dodávat pouze samotný hardware. Cena pouze HW řešení se pohybuje kolem

3.000 Kč + DPH.

Pro KONCOVÉ APLIKACE je řešením "lite" řada:

- **ETHERNET-RS232 konvertor** - Podpora RS485,TCP/IP nebo UDP/IP, Kryptovaná autorizace..
- **WWW I/O controll** - Vzdálené ovládání a monitorování teploměru, vstupů a výstupů přes WWW
- **WWW,EMAIL,SNMP I/O controll** - Vzdálené ovládání a monitorování přes WWW, EMAIL, SNMP a autorizovaný "telnet" klient.

Aplikační verze mimo DESIGN řadu jsou určeny jako konečná řešení konkrétních aplikací, jsou dodávány bez jakýchkoli zdrojových kódů.

Řešení hardwarové části Web51

Procesor x51 který je v našem případě zastoupen procesorem 89C8252 je zapojen pouze v portovém režimu a v aplikaci proto není vůbec použit LATCH 74 373 (74 573) který by byl jinak třeba pro vytvoření adresové sběrnice. Procesor 89C8252 je pak použit protože obsahuje 8 kB FLASH programové paměti programovatelné pomocí ISP, což zjednodušuje vývoj a interní sériovou EEPROM 2 kB. V zásadě je však možno použít libovolný procesor z x51/52 rodiny. Použití jiného procesoru však obvykle vyžaduje zasáhnout do obsluhy řadiče ethernetu a předpřipravených hlaviček protokolu TCP/IP. Rozhraní ethernetu využívá standardní konektor RJ45 pro kabeláž typu 10BaseT. Je možné však z desky vyvést i rozhraní AUI, např. pro připojení převodníků na optický kabel. Část 10BaseT je zapojena klasicky, včetně odrušovacích ochranných VN kondenzátorů proti napěťovým špičkám. Reset procesoru, případně celé desky je ovládán obvodem DS1833, software je hlídán interním watchdogem procesoru 89C8252.

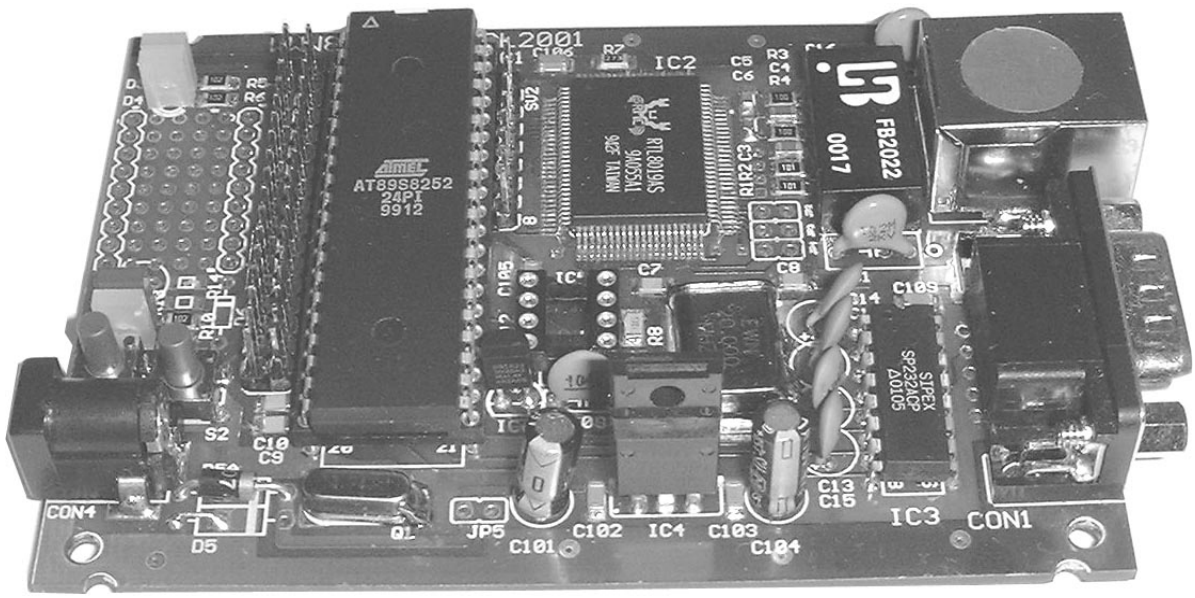
- Podrobnosti najdete na přiloženém schématu Web51 - (*zadní stránka tohoto manuálu*)
- Pokud budete programovat procesor po ISP rozhraní, počítejte s průběhem RESETu a programovacích pinů během procesu programování vzhledem k periferiím.
- Hardware projektu Web51 byl navrhován pro maximální cenovou optimalizaci projektu. Cílem bylo navrhnout **levné zařízení pro telemetrické účely**.

Základní parametry HardWaru Web51 :

- Síťový řadič RTL8019AS s výstupem přes transformátor na Twisted Pair vodiče.
- Procesor ATMEL 89C8252 = 8 kB FLASH, 256 B RAM, 2 kB EEPROM. Frekvence 22,1184 MHz.
- MAX 232 osazen na základní DPS. K programování lze použít ISP rozhraní.
- Vnější I^2C EEPROM od **24C32**(4k*8) do **24C1024**(64k*8).
- Plošný spoj DESIGN verze obsahuje také malé univerzální pole pro osazení malé jednoduché periférie a dvě LED diody se dvěma tlačítky pro základní ovládání, pokud to software podporuje.
- Uživatel Web51 má pro své aplikace k dispozici 16 I/O linek (celý port P1, část P3 a dva piny portu P2). Do těchto 16. pinů jsou však započítány i vyvedené vodiče I^2C sběrnice, RS232 nebo tlačítek a signalizačních LED na desce. Tyto porty jsou vyvedeny na konektor PFL34.

- Od verze 4.0 (Leden 2002) a verze 3.5 Design KITu je na tomto konektoru kompletní sběrnice procesoru pro možnost použití externích modulů rozšiřujících paměť v MOVX prostoru.

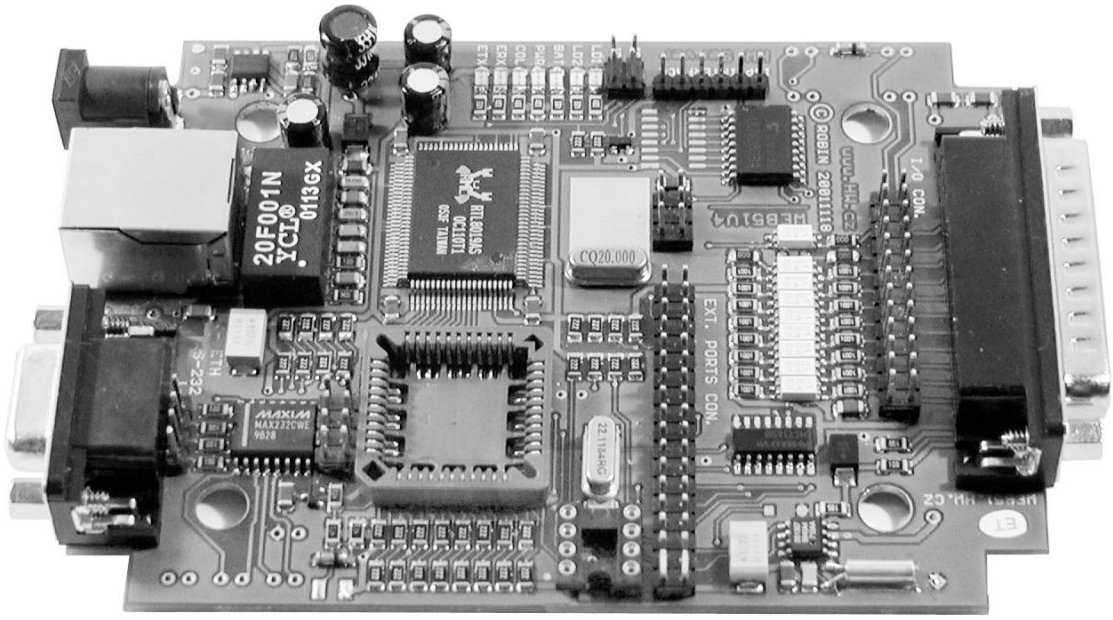
Fyzické řešení Web51 - DESIGN VERZE



Parametry řešení :

- Šířka DPS 72 mm pro možnost osazení do standardní lišty
- Polarita napájení ze souosého konektoru chráněna diodou, možno osadit v obou polaritách.
- Malé pájecí pole pro nenutnější přizpůsobení
- Provedeno klasickou montáží která umožňuje snadné uživatelské zásahy

Fyzické řešení Web51 - LITE VERZE 4.x *(aplikační řešení)*



Parametry řešení :

- Verze v plastikové i kovové krabičce
- Spínaný napájecí zdroj
- Možnost osazení RS 485 / RS 422 místo standardní RS 232
- Cannon 37 možno použít pro vlastní aplikaci, pomocí rozšíření horní deskou plošného spoje.

Řešení softwarové části Web51

Projekt Web51 je po softwarové stránce velmi rozsáhlý, rozdělený do mnoha zdrojových souborů a knihoven. Aby nebylo nutné vždy překládat vše znovu, je využito programu MAKE. MAKE řeší celkem složitý management všech zdrojových souborů pro kompilaci výsledného kódu. Jako kompilátor assembleru zdrojového kódu pro procesor 51 a linker byl pro projekt přenesen standardní překladač z prostředí Linuxu z balík nazývaného BINUTILS. Tento softwarový balík je však schopen i práce v prostředí Win32 tj. pod Windows,… a na řadě dalších systémů, včetně systému ze kterého vyšel tj. v Linuxu a celé řadě Unixů. Z prostředí Linuxu pochází i celá řada dalších použitých utilit

Adresářová struktura projektu Web51

Následující popis adresářové struktury slouží pro rychlé zorientování při prvním seznámení s projektem. Je pouze orientační, celý popis najdete v části Vývojový systém v rubrice Software.

Adresářová struktura projektu na dodávaném CD

Na dodávaném CD s projektem, najdete následující adresáře:

- **FIRMWARE** obsahuje archivy poslední verze distribuce, včetně komerčních zdrojových kódů.
- **SW** obsahuje výběr softwaru, který můžete potřebovat. (hlavně textové editory)
- **WWW** obsahuje kompletní WWW stránky projektu, platné v době aktualizace. Oproti WEBové verzi zde najdete podrobnější popisy komunikace po ethernetu (ARP, ICMP, TCP,...) a několik dalších zajímavostí.
- **WEB51** je adresář se všemi utilitami pro vývoj aplikací. Jeho obsah získáte také rozbalením balíků **BINUTILS** a **Web51 GNU development system** z distribuce Web51.

Popsané utility se vztahují k verzi 1.13 a v budoucnu se mohou měnit.

Dále popsaná adresářová struktura projektu Web51, se týká právě tohoto adresáře. Při běžném použití doporučujeme překopírovat si celý adresář WEB51 na HDD. Jak je uvedeno také v papírové dokumentaci, **důrazně doporučujeme nastavit adresář /BIN do systémové proměnné PATH**.

Directory: /bin

Adresář obsahuje všechny potřebné spustitelné soubory vývojového systému projektu Web51, s výjimkou editoru .ASM souborů a obsluhy programátoru obvodů.

- **mcs51-as** GNU as - GNU assembler. Podrobnější dokumentace je v souboru **as.pdf** z **gnudoc**.
- **mcs51-ld** GNU ld - GNU linker Podrobnější dokumentace je v souboru **ld.pdf** z **gnudoc**
- **mcs51-objcopy** GNU objcopy - kopírování a změny formátu přeložených souborů
- **bin2hex.exe** převod binárních souborů do tvaru Intel HEX
- **bash** GNU Bourne-Again SHell, DOS32 verze UNIXového příkazového interpretru. Podrobnější dokumentace je v souboru **bash.html** v adresáři /gnudoc

- **make** GNU make - DOS32 verze UNIXové utility make pro řízení překladu. Podrobnější dokumentace je v souboru **make.html** v adresáři /gnudoc
- **perl** Interpretr jazyka PERL
- **rm** GNU rm - DOS32 verze UNIXové utility pro mazání souborů a adresářů
- **html2db.pl** Perl skript pro konverzi souborů html/jpeg/gif... na tvar přeložitelný překladačem mcs51-as / a51 (Keil/Intel). Skript je popsán v WWW dokumentaci - část Konverze html stránek (jpeg, gif,...) pomocí html2db.pl
- **ip2hex.pl** Perl skript pro generování ip.inc. Skript je popsán v WWW dokumentaci - část Konverze IP a MAC adres pomocí mac2hex.pl a ip2hex.pl
- **mac2hex.pl** Perl skript pro generování ether.inc. Skript je popsán v WWW dokumentaci - část Konverze IP a MAC adres pomocí mac2hex.pl a ip2hex.pl

Adresář /bin, v závislosti na použitém programátoru může obsahovat i programy pro obsluhu ISP programátoru. Problematiku programování po ISP najdete podrobně rozebranou v kapitole HARDWARE v části ISP Programování.

Directory: /gnudoc

- **bashref.htm** GNU Bash Reference Manual
- **make.htm** The GNU Make Manual
- **ld.pdf** The GNU linker - V2.11 / ld version 2
- **as.pdf** The GNU Assembler - V2.11
- **gplcz.htm** český překlad GNU General Public Licence
- **gnu.jpg** GNU logo

Directory: /include

- **8019.inc** definice Ethernet, IP, TCP,... konstant, registrů RTL8019, ...
- **param.inc** základní definice HW a SW prostředí pro překlad

Directory: /lib

Adresář obsahuje knihovny **libk a libw**, využívané vývojovým systémem WEB51, základní jádro celého systému a skript pro sestavení jednotlivých modulů a knihoven do výsledného programu.

V závislosti na licenci GNU/komerční obsahuje i plné zdrojové texty knihoven a jádra systému.

- **libk23.a** Standardní knihovna **libk** dodávaná se systémem, přeložená pro podporu 2 TCP stacků, aktivní navazování spojení na stacku č. 1 a s podporou sériové linky.
- **libk80.a** Standardní knihovna **libk** dodávaná se systémem, přeložená pro podporu 1 TCP stacku, bez podporou sériové linky.
- **libw23.a** Standardní knihovna **libw** dodávaná se systémem, přeložená pro podporu 2 TCP stacků, aktivní navazování spojení na stacku č. 1 a s podporou sériové linky.
- **libw80.a** Standardní knihovna **libw** dodávaná se systémem, přeložená pro podporu 1 TCP stacku, bez podporou sériové linky.
- **web51_23.obj** Standardně přeložené základní jádro systému (2TCP/..)

- **web51_80.obj** Standardně přeložené základní jádro systému (1TCP/..)
- **www51.sc** Skript pro řízení linkeru mcs51-ld, řídicí sestavení modulů, knihoven a dalších prvků do výsledného programu

Directory: /cgi

Adresář obsahuje knihovnu CGI skriptů, používaných v HTML stránkách. Adresář obsahuje jak zdrojové texty, tak přeložené soubory, přeložené v distribuční verzi pro nastavení použité v knihovně lib$_x$80

Directory: /projekt

Adresář projekt obsahuje sadu vzorových projektů, demonstrujících programování WEB51. Do tohoto adresáře doporučujeme ukládat také vaše projekty. Jako základní sadu projektů zde najdete sadu příkladů, popsané v samostatné kapitole EXAMPLES.

Řízení překladu pomocí MAKEFILE

Celý projekt Web51 je rozdělen na mnoho částí a ty jsou rozvrženy do mnoha souborů. Celkově je při překladu nutné používat nějaký projektový management, z mnoha zdrojových souborů a knihoven vytvoří podle zadaných podmínek přeložený jediný soubor .HEX pro CPU (případně soubory pro sériové eeprom a další).

MAKE - Makefile

Při překladu projektu používáme z Unixu přenesenou **utilitu MAKEFILE**, která je obsažena v balíku **BINUTILS** a funguje samozřejmě i pod Win32. Funkčně by ji bylo možné přirovnat k velmi chytrému .BAT souboru z MS DOSu. MAKEFILE řeší celkem složitý management všech zdrojových souborů pro kompilaci výsledného kódu. - Podrobný popis utility najdete v DOWNLOADU v sekci *Dokumentace související s projektem*.

Utilita MAKEFILE je obsažena v adresáři **/bin**, měla by být v cestě a po svém spuštění vždy hledá v aktuálním adresáři soubor **"make"** *(bez jakékoli přípony)* který obsahuje dále podrobně popsaný předpis pro sestavení projektu.

Pokud pracujete pod Windows, dejte si **pozor na program Windows Commander**, který nevolá korektně příkazový řádek, takže doporučujeme makefile spouštět z spuštěné aplikace DOS shellu "Příkazový řádek".

MAKE - popis struktury

Dále popisovaný soubor je vždy postupně komentován. Skutečný výpis souboru make pro projekt LED2 proto doporučujeme vyhledat v projektech. (Obsah souboru Makefile je psán neproporcionálním písmem, pod kódem je vždy jeho vysvětlivka.)

```
PROJ = www8051
PROJFILE = makefile

CA  = ../../bin/mcs51-as
CAOPT =
LINKER = ../../bin/mcs51-ld
READELF = ../../bin/mcs51-readelf
PERL = perl
BASH = bash
SCRIPT = ../../bin
```

```
INCDIR = ../../include
LIBDIR = ../../lib
CGIDIR = ../../cgi
BIN2HEX = ../../bin/bin2hex
RM = ../../bin/rm
```

Nastavení systémových proměnných s umístěním jednotlivých souborů a případných globálních parametrů překladu, které budou používány v průběhu překladu. Makefile si cesty nastavuje, ale práce je výrazně pohodlnější, pokud cestu k adresáři /BIN z projektu Web51 nastavíte do vaší systémové proměnné cesty (PATH) v autoexec.bat. Můžete pak jednodušeji spouštět i další programy z vývojového systému, především ale samotný program MAKE.

```
FILES = www8051.asm index.html LEDsetup.asm
OBJS  = www8051.obj index.obj LEDsetup.obj
CGI   = $(CGIDIR)/testP3.obj
```

Výčet .ASM souborů, s nimiž bude projekt pracovat a které bude překladač překládat do .OBJ souborů v řádce pod nimi. Pokud váš projekt rozšíříte o použití nového .CGI scriptu z naší nabídky, nebo si napíšete svůj, nezapomeňte jej přidat do makefile na toto místo. Všimněte si, že jsou zde zahrnuty i HTML soubory, s nimiž musí makefile také pracovat. Pokud si používáte vlastní .CGI scripty, nechte je klidně v projektovém adresáři, ale nezapomeňte přidat názvy jejich souborů do proměnné OBJS a FILES.

Za těmito základními deklaracemi následuje řada příkazů pro zpracování v rámci makefile:

```
Co_chci_získat: Na_čem_to_závisí
    Jak_to_získat
```

Při volání MAKE je možno zadat cíl, aneb co_chci_získat, nicméně MAKE automaticky při neuvedení cíle vytváří cíl daný první definicí, takže když jako první definici uvedeme soubor ve formátu :

```
hex: www8051.hex
```

Pak při zpracování, bude mít MAKE snahu implicitně vytvořit soubor www8051.hex Co všechno MAKE přeloží, závisí na tom jaké je datování souborů ze kterých je výsledný soubor tvořen. MAKE vždy znovu přeloží zdrojové soubory, na kterých závisí výsledný soubor, pokud jsou tyto soubory novější než soubor, který má vytvořit. Lze vytvářet také soubory BIN apod. ale vzhledem k lepší možnosti kontroly neporušenosti souboru a bezproblémovému přenosu je preferován HEX

Soubor MAKE může obsahovat i několik výsledných cílů, jedním z může být i příkaz mazající pracovní soubory např.

```
.PHONY: clean
clean:
    $(BASH) -c "rm -f *.obj"
    $(BASH) -c "rm -f *.lst"
    $(BASH) -c "rm -f ether.inc"
    $(BASH) -c "rm -f ip.inc"
    $(BASH) -c "rm -f index.asm"
    $(BASH) -c "rm -f www8051.o"
    $(BASH) -c "rm -f www8051.rom"
    $(BASH) -c "rm -f www8051.eeprom"
    $(BASH) -c "rm -f www8051.eep"
    $(BASH) -c "rm -f map"
    $(BASH) -c "rm -f listing"
```

Pokud v příkazové řádce použijete syntaxi „`make clean`“ vykoná se tato část kódu ze souboru makefile. V tomto případě se smažou všechny vygenerované soubory. Tím vyčistíte adresář a zpřehledníte soubory projektu například před distribucí projektového adresáře dalším uživatelům.

```
.SUFFIXES: .obj .c .asm .inc .h .ina .html

www8051.obj : www8051.asm $(INCDIR)/8019.inc $(INCDIR)/param.inc ip ether
    $(BASH) -c "rm -f ether.inc"
    $(BASH) -c "rm -f ip.inc"
    $(PERL) $(SCRIPT)/mac2hex.pl -v -equ -binutils - <ether >ether.inc
    $(PERL) $(SCRIPT)/ip2hex.pl -v -equ -binutils - <ip >ip.inc
    $(CA) www8051.asm -o www8051.obj $(CAOPT) -a=www8051.lst
```

Tato část makefile zabezpečuje vlastní překlad souboru www8051.asm. Soubor www8051.obj získaný překladem však závisí na celé řade dalších souborů, při jejichž změně je nutno spustit překlad znovu. Tyto soubory jsou uvedeny na řádku za znakem „:“

Pokud MAKE zjistí novější datum některého ze zdrojových souborů, spustí tuto část scriptu, která postupně vytvoří jeden ze souborů ze kterých je složen výsledný “**www8052.hex**“. V tomto případě je do kódu je třeba vložit nezbytné definice názvů zkratek a proměnných, což zajišťují právě vkládané definice “**param.inc**“ “**8019.inc**“ (vložené do www8051.asm). Dalšími vloženými definicemi jsou soubory ether.inc a ip.inc. Tyto dva soubory jsou pro zlepšení čitelnosti generovány z textového konfiguračního souboru pomocí skriptů v jazyku Perl.

V tomto místě se použijí soubory „**ether**“ a „**ip**“ které mají syntaxi (ether):

```
00:00:E8:EE:10:34
```

Při překladu se z nich vytvoří pomocí utility PERL dále vypsaný soubor „**ether.inc**“ a soubor „**ip.inc**“ které již lze implementovat do .ASM zdrojového kódu.. Podrobnosti o **ip2hex.pl** a **mac2hex.pl** utilitě najdete v sekci SOFTWARE / IP -> HEX & MAC-> HEX v podrobné WWW dokumentaci.

```
.global MAC_MSB, MAC_NSB, MAC_LSB
.equ    MAC_MSB, 0x0000 ; MAC address
.equ    MAC_NSB, 0xE8EE
.equ    MAC_LSB, 0x1034
```

Zároveň si všimněte souboru „**param.inc**“ kterým se vkládá do zdrojového kódu základní nastavení pro krystalový oscilátor (používá pouze UART pro linku RS232). Zároveň jsou zde definována slova jako SDA, SCL což jsou piny pro připojení I^2C linky atd.. V tomto souboru však nenajdete definici „**.global reset**“ která definuje počáteční stav portů po resetu. To je občas nutné znát pro nastavení periférií. Tuto část kódu, stejně jako část průběžně běžící přerušení od časovače najdete v souboru „**web51.asm**“ v adresáři LIB.

Všimněte si generování .LST souborů. Otevřete si v přeloženém projektu například „**index.lst**“ a „**www8052.lst**“ ať víte, že zde najdete adresy kódu, pokud potřebujete lokalizovat chybu na konkrétní adrese..

```
LEDsetup.obj : LEDsetup.asm $(INCDIR)/param.inc
    $(CA) LEDsetup.asm -o LEDsetup.obj $(CAOPT) -a=LEDsetup.lst

index.obj : index.html
    $(BASH) -c "rm -f index.asm"
    $(PERL) $(SCRIPT)/html2db.pl -binutils -cpueeprom -index 0 index.html
    $(CA) index.asm -o index.obj $(CAOPT) -a=index.lst
```

HTML soubory je třeba nějak včlenit do výsledného .HEX souboru. Toto včlenění je zde zprostředkované přes assembler. Z .HTML souboru je vytvořen .ASM soubor, kde bude HTML převedeno do podoby tabulky v paměti programu nebo EEPROM. To lze opět udělat pomocí skriptu v PERLu (stejně jako v případě generování **ether.inc** a **ip.inc**)

Parametr „**cpueeprom**“ zde definuje kam bude výsledný kód přeložen. Nezapomeňte jej proto změnit, pokud budete potřebovat změnit organizaci dat v paměti. Podrobnosti o **html2db.pl** utilitě najdete v sekci SOFTWARE / HTML -> DB v podrobné WWW dokumentaci.

```
www8051.o : $(OBJS) $(CGI) $(LIBDIR)/libw80.a $(LIBDIR)/libk80.a $(LIBDIR)/web51.obj
    $(BASH) -c "rm -f www8051.o"
    $(LINKER) --script $(LIBDIR)/www51.sc -L $(LIBDIR) $(LIBDIR)/web51.obj $(OBJS)\
    $(CGI) -lk80 -lw80 -M -o www8051.o --no-check-sections >map
```

Na této řádce makefile jsou již připraveny .OBJ soubory a spouští se linkování celého projektu. Soubor „**www8051.o**“ je závislý na vstupních knihovnách **libw80.a libk80.a** nebo **libw23.a libk23.a** což je zde řečeno v prvním řádku. Podle pořadí kompilace jsou nyní připraveny všechny vstupní soubory, CGI scripty z knihovny atd. (viz. Pořadí linkování) Zde je třeba doplnit zbylé funkce z knihoven a vše slinkovat. První volá linker soubor **www51.sc** z adresáře **LIB**. Ten obsahuje něco jako definici pořadí linkování jednotlivých segmentů a jejich klíčových jmen. Parametr **–L** pak definuje kde se mají hledat knihovny. Linkování je nutné začít souborem **web51.obj** z adresáře **LIB**, protože obsahuje definice vektorů přerušení a proto musí začínat od nuly. Za tímto souborem se nalinkují následující .OBJ soubory v pořadí jak jsou definovány proměnou **OBJS.** Totéž je zopakováno s .OBJ soubory uvedenými v proměnné **CGI.** Obě proměnné jsou definovány na začátku makefile (nezapomínejte proto doplnit do tohoto seznamu překládaných souborů například CGI script, který chcete použít z nabídky..). Zde se nenechte zmást zpětným lomítkem, jedná se oznak pro pokračování řádku (respektive pro ignorování zbytku řádku, čímž příkaz plynule pokračuje na dalším řádku) . Po přilinkování souborů z **OBJS** a **CGI** se přesně podle definice linkování linkují zbylé funkce z knihoven **libk80** a **libw80. Plný název knihovny „libw80.a“ je na příkazovém řádku zkrácen na -lw80“** a označuje totéž, rozdíly v knihovnách viz. Pořadí linkování. Zpracování linkování probíhá zleva doprava a není rekurzivní, proto doporučujeme zachovat první **lk80** a teprve za ní uvádět **lw80**.

Parametr **–M** pak již pouze generuje soubor **MAP** který obsahuje informace o výsledném překladu. Doporučujeme vám otevřít MAP soubor a podívat se jak vypadá jeho struktura a co vše v něm lze nalézt.

Výsledek celého překladu se ukládá do souboru “**www8051.o**“ (Neplést se souborem “**www8051.obj**“ obsahujícím obslužné rutiny hlavního programu, přeložené ze souboru “**www8051.asm**“).

```
www8051.rom : www8051.o
    $(RM) -f www8051.rom
    $(READELF) -x 1 www8051.o | perl $(SCRIPT)/red2rom.pl > www8051.rom
    $(RM) -f www8051.eep
    $(READELF) -x 16 www8051.o | perl $(SCRIPT)/red2rom.pl > www8051.eep
```

Soubor **www8051.o** je obdobou souborů .obj, vypreparujeme z něj proto pomocí utility z BINUTILS a Perlového skriptu red2rom část obsahující programový kód a část obsahují obsah paměti EEPROM. Obě části nazvané www8051.rom a www8051.eep jsou binární. Nicméně programátor obvykle touží po souboru jednom s obsahem paměti EEPROM adresovaným v případě procesoru 8252 od 2000h aneb 8192. Takže mu jej pomocí utility BIN2HEX převedeme do formátu HEX a spojíme obě části dohromady. Obdobně lze generovat i část obsahující data v I2C paměti.

```
www8051.hex : www8051.rom
    $(RM) -f www8051.hex
    $(BIN2HEX) www8051.rom www8051.hex
    $(BIN2HEX) /O8192 /M www8051.eep www8051.hex
```

Pro urychlení programování je se souborů pro FLASH a EEPROM automaticky vygenerován soubor www8051.hex s posunem EEPROM od 2000H. Pokud potřebujete generovat .HEX soubor, hned na začátku nastavte **www8051.hex** jako cílový soubor a doplňte nakonec tento odstavec. Nezapomeňte na přidání souborů .EEP a .ROM do seznamu souborů pro spuštění části CLEAN z makefile.

Tím jsme probrali celý proces generování výsledného kódu pro procesor. Pokud nemáte adresář **BIN** nastaven v systémové proměnné PATH lze spustit zpracování makefile zadáním “**..\..\make**“ z adresáře projektu (pokud samozřejmě zachováváte doporučenou adresářovou strukturu). Pokud pracujete pod windows, jako že asi ano, doporučujeme spouštět MAKE s nastavenou PATH a z spuštěného DOS shellu „Příkazový řádek“ v adresáři projektu. Například oblíbený Volkov Commander má totiž přesměrovány standardní obrazovkové výstupy tak, že pokud se MAKE spustil pod ním, například do souboru **“ip.inc“** se zapsal nejen žádoucí kód, ale také kód který měl PERL z BINUTILS vypsat na obrazovku, což samozřejmě způsobilo chybu kompilace.

Pořadí linkování

*Popsaná struktura souboru **make** zajišťuje překlad celého projektu do .HEX souboru. Proto zde stručně shrneme, co je třeba udělat a pořadí jednotlivých kroků pro kompilaci jakéhokoli projektu. Věnujte pozornost Rozdělení do knihoven.*

Pořadí linkování

- Nastavit adresáře v nichž jsou uloženy použité proměnné s jejich relativními cestami
- Proměnné **FILES** a **OBJS** naplnit soubory které budou kompilovány do projektu, včetně .HTML souborům
- Do proměnou **CGI** nastavit použité .CGI soubory s jejich cestami
- Definovat název výsledného souboru, který se bude vždy kompilovat při změně vstupních souborů
- Připravit include soubory s IP a ETHERNET (MAC) adresou, vložit do kódu nezbytné deklarace
- Následuje výčet definic překladů jednotlivých vstupních souborů, včetně .HTML s jejich parametry
- Poslední se překládá soubor www8051.o do něhož se linkerem přilinkovávají všechny připravené .OBJ soubory, včetně nezbytných funkcí z knihoven
- Výsledný soubor www8051.o je rozdělen na binární soubory pro EEPROM a FLASH paměť procesoru a uložen do souborů .EEP a .ROM
- Případně je z těchto souborů vytvořen výsledný www8051.hex který umístí .ROM část od nuly a .EEP část od adresy 2000H (8kb) což většina programátorů akceptuje jako data pro interní EEPROM procesoru

Rozdělení do knihoven

Samotné komunikační rutiny, které obsluhují řadič ethernetu REALTEC 8019AS a další jsou složeny z mnoha zdrojových souborů, ale výsledně jsou uloženy celkem ve 2 knihovnách, pro použití s ukázkovými příklady, přeložených pro dvě hlavní konfigurace:

- První konfigurace je přeložena pro podporu 2 TCP stacků, aktivní navazování spojení na stacku č. 1 a podporu sériové linky. V názvu obsahuje označení 23.
- Druhá konfigurace je přeložena pro podporu 1 TCP stacku, a je bez podpory sériové linky. V názvu obsahuje označení 80.

Předpřeložené, přejmenované knihovny libw.a a libk.a jsou:

- **libw80.a** – Obecné rutiny jako je například matematika, výpisy kódu a podobně, určené pro jeden komunikační port. Předpokládá se port 80 (http) ale číslo portu lze nastavit.
- **libk80.a** – Síťové rutiny jako je například práce s virtuálním TCP/IP stackem atd..
- **libw23.a** – Obecné rutiny jako je například matematika, výpisy kódu a podobně, určené pro dva komunikační porty. Předpokládá se port 23 (telnet) a port 80 (http) ale čísla portů lze nastavit.
- **libk23.a** – Síťové rutiny jako je například práce s virtuálním TCP/IP stackem atd..

Bližší podrobnosti o knihovnách je možno najít v popisu vývojového systému a popisu knihoven libk a libw.

Linker pro zadávání jmen knihoven na příkazovém řádku používá několik konvencí:

- Knihovny se hledají v adresáři (adresářích) zadaných příkazem **–L**
- Jména knihoven jsou zkrácena, není uvedeno počáteční lib a koncové .a. Jméno knihovny je uvozeno příkazem **–l**. Tj. např. knihovnu libk80.a na příkazovém řádku linkeru zadáme pomocí příkazu **–lk80**

Jak bylo uvedeno, knihovny je nutné linkovat v pořadí libk a potom libw vzhledem k tomu, že síťová knihovna používá některé obecné funkce z knihovny libw a linkování **není rekurzivní**. (V rámci jedné knihovny však samozřejmě je, nezáleží proto na pořadí, v jakém jsou do knihoven přidávány jednotlivé moduly.)

Makefile se zpracovává odshora dolů, pokud make narazí na položku, která mu chybí a má ji proto vytvořit (přeložit) začne prohledávat Makefile znovu od počátku.

Nyní si ještě jednou přečtěte v popisu struktury Makefile odstavec deklarující vytvoření "**www8051.o**" – celá problematika by již měla být jasnější.

Rozdělení kódu projektu do jednotlivých souborů

Jak plyne z popisu makefile, každý projekt se při linkování skládá z několika souborů. Tento odstavec by vám měl objasnit co je uloženo v jednotlivých zdrojových souborech a jak skládat data do FLASH (případně EEPROM) paměti.

.HTML

Soubory uložené nakonec v paměti Web51 (FLASH programu, EEPROM interní nebo externí), jako předpřipravené HTML stránky, které jsou volány souborovým systémem. Soubory se konvertují pomocí direktivy v makefile na .ASM soubor pomocí "**html2db.pl**" scriptu. Tento perlový script likviduje CR LF znaky, neboli konce řádků tím, že je při převodu na definici tabulky v .ASM souboru prostě ignoruje. To nevadí čistému HTML. Bude-li však v kódu javascript, bude absence konců řádek znamenat nefunkčnost scriptu. V takovém případě je nutné do definice překladu tohoto .HTML souboru v makefile přidat parametr "**-appendIf**" (viz příklad LED3).

Na soubor "index.html" je nutné se dostat nejen voláním z prohlížeče "http://X.X.X.X/index.html" ale také jenom posledním lomítkem jako na hlavní soubor web serveru. To je zde řešeno v makefile při překladu, neboť pro překlad **index.obj** je zde uveden parametr "**-index 0 index.html**" kde část "**-index 0**" zajišťuje právě odkaz na soubor druhou hlavičkou, nejen jako název souboru, ale také jako na "/" - hlavní stránku zařízení.

Z HTML stránek můžete volat CGI scripty pomocí speciálního znaku apostrof " **'** ". Pokud tedy potřebujete použít v kódu HTML znak apostrof, je nutné jej zdvojit, což bude přeloženo jako jeden apostrof. Překladač nemá možnost zkontrolovat volání CGI skriptů z HTML kódu, jejich volání je až při interpretaci HTML kódu samotným WWW8051. Nezapomeňte proto všechny použité CGI skripty zahrnout do projektu a přidat do výsledného kódu.

.GIF .JPG .PNG

Obrázky lze do procesoru samozřejmě také ukládat do paměti Web51 (FLASH programu, EEPROM interní nebo externí). Příklad kde je to řešeno je například LED1 nebo teploměr. Ze souborů například .GIF je pro linkování linkerem potřeba samozřejmě vytvořit také soubor .OBJ Že se jedná o binární soubor si detekuje již samotný script "**html2db.pl**" takže je zde potřeba pouze nezapomenout uvést všechny použité obrázky v proměnných **FILES** a **OBJS** a doplnit do makefile potřebné řádky pro jejich přilinkování do výsledného kódu.

.CGI

CGI scripty jsou velmi vhodnou cestou pro předávání dat mezi programem, který běží ve Web51 a nemá s generováním stránek nic společného (měřící a vyhodnocovací rutiny) a HTML stránkami. Kód scriptu je obvykle psán v assembleru procesoru, ale script se volá vždy až když je třeba odeslat WWW stránku, proto by jeho vykonání nemělo trvat zbytečně dlouho. Většinou proto pouze odesílá ve vhodném formátu data uložená v nějaké proměnné. Podrobný popis struktury CGI souboru a jejich možností následuje dále v této dokumentaci. Druhou možností je napsat skript v Pkódu.

www8051.asm

Jedná se o hlavní program projektu. Na jeho začátku jsou vloženy .INC soubory obsahující nezbytné definice a alokována paměť pro systémové proměnné.

Následuje sekce "**Main Loop**" kde jsou definovány "**.section fast, #alloc**" a "**.section slow, #alloc**". V těchto sekcích by měly být umístěn kód konkrétní aplikace, který se zpracovává. Sekce **FAST** je spouštěna přibližně každou milisekundu, sekce **SLOW** každých 50 cyklů sekce **FAST**. Volání těchto sekcí se provádí z hlavního systémového kódu, umístěného v souboru "**web51.asm**" a definice konstant (to znamená i konstanty **slowtiming=50**) je uložena v souboru "**param.inc**". Do těchto dvou sekcí umístíte kód, který obsluhuje hardwarová rozhraní. Tyto sekce musí být průchozí, tj. nekončit příkazem ret a podobně (blíže viz. popis sekce reset_device). Potřebné podprogramy, volané většinou příkazem LCALL, můžete umístit do poslední části souboru "**www8051.asm**" která je standardně umístěna za sekcí **SLOW** a je uvozena deklarací "**.text**". Nepoužíváte-li sériový kanál, tak by konec této sekce měl obsahovat "**global intRITI**" ukazujícím na příkaz RETI. Na tuto globální proceduru se odkazují všechna přerušení, která nejsou použita. Zapomenete-li přilinkuje se z knihovny libw obsluha sériového kanálu, zabírající poměrně dost prostředků procesoru.

Pokud potřebujete nějaký kód spouštět pouze po startu zařízení (například inicializaci periferií) doplňte si mezi sekci **SLOW** a část "**.text**" další sekci "**.section reset_device, #alloc**" či "**.section reset_network, #alloc**"která bude vykonána pouze po startu programu.

Rozdíl mezi nimi spočívá v tom, že nejprve se vykonají všechny sekce **reset_device**, ze všech modulů a poté všechny sekce **reset_network**, opět ze všech modulů. V těchto sekcích je **zakázáno** přerušení. Tyto sekce musí obsahovat kód, který není ukončen příkazem **ret**, tj. jsou plně průchozí.

Nejjednodušším příkladem jednou provedené sekce pak může být uvedený kód, kde je použita instrukce NOP, místo inicializačního kódu:

```
.section reset_device, #alloc
nop
.text
test_cidla:
…..
```

Je nutno si rovněž uvědomit, že sekce končí na hranici další sekce, vykonávání programu po instrukci NOP v tomto příkladu pokračuje instrukcemi z jiného modulu obsahujícím sekci **reset_device** a nikoli instrukcemi za návěštím **test_cidla**.

Obdobou těchto dvou sekcí je sekce "**.section one_times, #alloc**", do které je soustředěn kód vykonávaný pouze jednou po startu a s **povoleným** přerušením. Kód této sekce opět nesmí končit příkazem **ret**.

Pokud budete tápat v assembleru, v sekci DOWNLOAD Najdete v kapitole *"Dokumentace související s projektem"* podrobný manuál k použitému assembleru..

Hardware Web 51 - ISP programování

Programování procesorů ATMEL pomocí rozhraní ISP je dnes již běžná věc. Šetří to čas při vývoji, není třeba stále přenášet CPU mezi programátorem a paticí v zařízení. Zde najdete kromě obecného popisu použití ISP také jedno z možných řešení interface, který vytváří ISP (In System Programming) pomocí LPT portu. Popsané řešení je možné realizovat podle dokumentace uveřejněné na www.HW.cz nebo objednat jako hotový kit.

Praktické použití ISP

Rozhraní pro sériové programování implementoval ATMEL hlavně do rodiny procesorů AVR. Zde je použití ISP celkem bez větších problémů, problémy mohou nastat pouze se strmostí hran, což lze částečně řešit snížením hodnoty PULL UP odporů a nepoužíváním připojovací kabelu delšího než cca 1m.

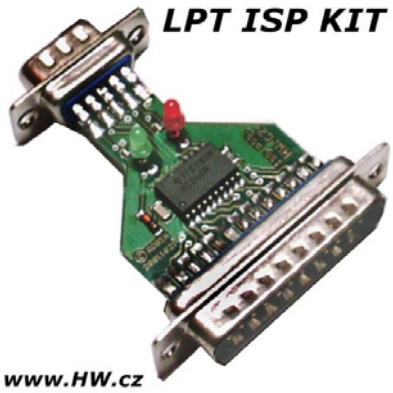
ISP je však také u dvou procesorů z rodiny x51 AT89C8252 a AT89S53. Použití nahrávání programu do CPU po ISP je jeden z velmi dobrých způsobů jak vyvíjet aplikace na staré rodině x51. Je zde však potřeba počítat s opačnou polaritou úrovně RESET oproti AVR rodině, což je potřeba u některého SW nastavit. Pozor také na resetovací zapojení v laděné aplikaci. Periferie musejí počítat s resetem z ISP programátoru.

Významným problémem je ale **špatná HW implementace podpory ISP rozhraní**. To se z důvodu nedostatku pinů připojuje na piny, sdílené s bránou P1. Při programování v aplikaci by mělo zapojení periférií počítat s možností přenosu dat po těchto pinech ve stavu RESET. Programování FLASH procesoru skrz ISP je řešeno doplněním bloku obsluhy ISP do struktury obvodu. Tento blok čeká po resetu na definovanou sekvenci, při níž odpojí zbytek CPU a naprogramuje paměť. Problém, který popisuje ATMEL i ve svých errata dokumentacích spočívá v tom, že v některých případech blok ISP považuje za sekvenci programování i přístup na port P1 při běhu programu v CPU. Pokud tento stav nastane, procesor se dostane do nedefinovatelného stavu, kdy programování přes ISP chvíli funguje, chvíli ne. CPU je potom nutné vymazat a znovu naprogramovat v klasickém paralelním programátoru.

Zde je třeba si uvědomit, že po každém ukončení programování přes ISP nastavuje RESET programovací SW do neaktivního režimu, čímž v podstatě okamžitě spouští program, který byl do CPU nahrán. Kombinací které přistupují na bránu P1 je docela hodně a proto je potřeba s touto HW chybou počítat. Atmel doporučuje nepřistupovat na piny ISP z P1 prvních cca 500 ms po resetu. Pokud tedy například nastavujete po RESETu všechny I/O piny do logické 0 použijte před nastavením ISP pinů (P1.5 - P1.7) čekací smyčku.
Web51 po startu nastavuje porty pro snížení spotřeby do stavu log. 0. Tím však manipuluje také se zmíněným P1 a proto soubor web51.asm v adresáři /lib/ obsahuje v sekci **.global_reset** čekací smyčku :

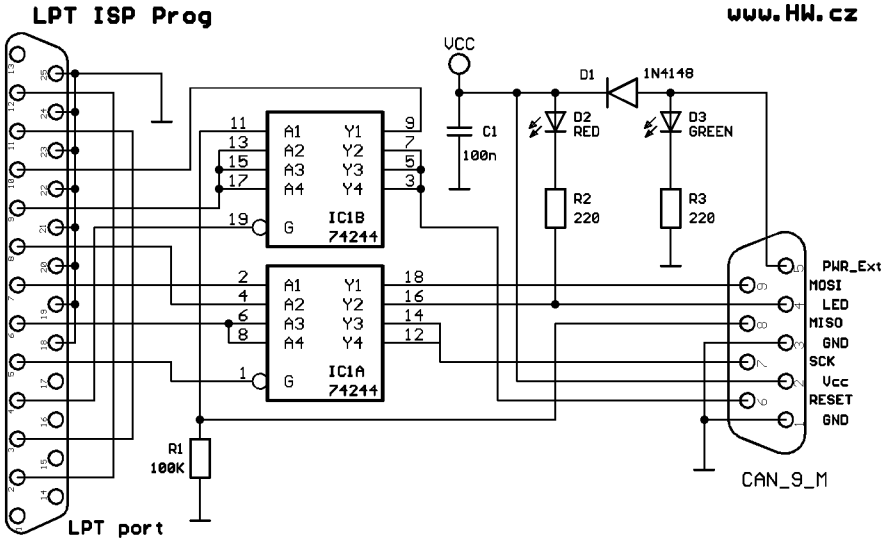
```
    .global reset
reset:
    MOV R7,#20      ;ISP DELAY for AT89S8252 CPU
ISP_Ra: MOV R6,#0xFF    ; 255 x 255 x 25 = 650 ms for 22.1184 MHz Xtal
ISP_Rb: MOV R5,#0xFF
ISP_Rc: DJNZ    R5,ISP_Rc
    DJNZ    R6,ISP_Rb
    DJNZ    R7,ISP_Ra
```

LPT ISP Prog

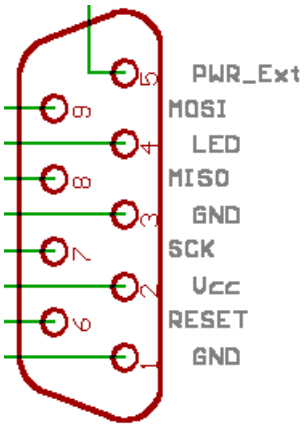


Programování procesorů ATMEL pomocí rozhraní ISP je dnes již běžná věc. Šetří to čas při vývoji. Chceme vám představit jednoduchý programátor, nebo spíš interface, který vytváří ISP (In System Programming) pomocí LPT portu.

LPT ISP Prog je spíš interface, nežli programátor v pravém slova smyslu. Neobsahuje žádné patice pro programování obvodů přímo na základní destičce. Jeho výstupem je pouze konektor ISP. Pokud potřebujete naprogramovat ISP procesor mimo aplikaci, lze samozřejmě připojit ISP přímo na procesor, doplnit k procesoru krystal s kondenzátory a to je vše..



Detail konektoru :



Napájení

ISP rozhraní obsahuje napájecí napětí. Protože zde existuje možnost napájení z aplikace, nebo z jiného zdroje, je na destičce programátoru oddělovací dioda D1. Pokuje ji zkratujete, je veškeré napájení propojeno. Dioda odděluje pouze napájení pro samotný oddělovací obvod, napájecí piny z ISP konektoru jsou propojeny se svorkovnicí pro případné externí napájení.

ISP rozhraní obsahuje kromě napájení ovládání resetování procesoru (**/RESET**), vodiče pro sériový zápis (**MOSI**) a čtení (**MISO**) obsahu interních pamětí FLASH a EEPROM + jeden vodič pro synchronizaci přenosu dat (**SCK**). Vodiče MISO a MOSI nejsou křížené = MISO z programátoru se připojuje na MISO CPU a MOSI na MOSI!

Výhody programátoru :

- Připojení na **Paralelní port – neblokuje sériový port**. Ten tak zůstává použitelný pro připojení laděného zařízení
- Software pro **Windows 95,98** i **Windows NT/2000** (SW obsahuje rychlejší drivery, které fungují pod W9x, nebo pomalejší i pro NT a W2000)
- **V ceně je zahrnut také poplatek pro autora SW C. Lanconelli za použití programátoru.**
- Na rozdíl o přímého připojení pinu paralelního portu obsahuje tento programátor budič sběrnice, který definuje logické úrovně výstupu, takže programátor funguje na všech paralelních portech korektně nemusí být připojen extrémně krátkým kabelem.
- Díky použití budiče, jsou hrany SCK dostatečně strmé a nehrozí problémy popisované v Errata dokumentech ATMELu. (pokud samozřejmě nepoužijete 10 metrů dlouhý kabel..)

- Software pro LPT ISP Prog je součástí standardního balíku „PonyProg“. Díky tomu je software průběžně inovován a aktualizován. Lze se tedy počítat s vyřešením problému, pokud například ATMEL upraví programovací algoritmy obvodů..
- Obslužný SW je velmi příjemný, je napsán pro průběžný vývoj. Obsahuje například volbu, kterou lze při každém programování .hex soubor načíst znovu z disku, což je nutné při jakémkoli vývoji a kompilaci kódu v jiném okně..
- Programátor obsahuje signalizaci průběhu programování a signalizaci napájení.

Aplikační tipy

- Pokud SW odmítá najít programátor na portu, zkuste změnit driver v nastavení hardwaru v PonyProg software.
- Pokud nesvítí LED napájení, hledejte chybu v napájení.“
- Pozor na resetovací zapojení v laděné aplikaci, musejí počítat s resetem z ISP programátoru.
- Při programování AT89S8252 respektujte omezení z ERRATA katalogových listů, hlavně část o nepoužívání pinů pro ISP prvních 500 ms po RESETu.

Ovládání programování skrz PonyProg z aplikace

Podle vyjádření autora PonyProgu lze volat ISP programování externě, přes PonyProg SW přes použití scriptů. Například z **MAKEFILE** tak lze zavolat Programovací SW a rovnou obvod naprogramovat. Přiložený příklad volá PonyProg z Makefile pomocí příkazu **make isp**. Potřebujete nejméně PonyProg2000 v 2.04h.

```
    ISP = c:/programmi/ponyprog2000/ponyprog2000.exe
    MCU = at89s8252

isp:    $(TRG).rom $(TRG).eep
    echo -e "SELECTDEVICE $(MCU)\nLOAD-PROG $(TRG).rom\nLOAD-DATA
$(TRG).eep\nERASE-ALL\nWRITE-ALL" >isp.e2s
    $(ISP) isp.e2s
    #$(RM) isp.e2s
```

Konstrukční podklady k programátoru

Veškeré konstrukční podklady najdete na HW serveru v článku LPT ISP Prog. Včetně zdrojových kódů pro plošné spoje a ovládacího softwaru.. Hotový ISP Programátor si můžete objednat s našimi dalšími produkty.

ISP programování jiných procesorů

Poslední dobou se objevuje možnost přechodu z CPU 89C8252 na procesor T89C51RD2. Ten obsahuje na čipu 1280 Bytů RAM (1024 bytů navíc proti 89C8252) a hlavně 64 kB FLASH.

V dokumentacích výrobce hovoří také o „ISP programování“, standardním rozhraním je zde však sériová linka a softwarový zavaděč. Jako fyzické prostředky pro nahrávání kódu do RD2 tedy stačí připojení na RS232, případně jednoduchý HW na definici RESETu pro Boatloader.

Web 51 - EXAMPLES - Vzorové projekty

*Adresář **/projekt** obsahuje ve svých podadresářích vzorové projekty, demonstrující použití vývojového systému Web 51.*

Pro kompilaci projektu stačí spustit make v adresáři s projektem voláním **..\..\bin\make**
Pro smazáni souborů vygenerovaných pomoci make spusťte make voláním **..\..\bin\make clean**
Pokud máte adresář s binutils v cestě, stačí volání **make**. Obdobně voláte MAKEFILE při generování knihoven programu. Uvedené příklady jsou obsaženy v balíku programu v sekci DOWNLOAD.

Welcome

Základní ukázka jednoduchého http serveru, bez použití CGI skriptů na straně serveru.

- **Makefile** - hlavní soubor pro MAKE
- **ether** - soubor se zadáním parametrů MAC adresy
- **ip** - soubor se zadáním parametrů IP adresy, masky gateway, ...
- **index.html** - html stránka uložená v procesoru
- **www8051.asm** - zdrojový kód, hlavní modul

Ether

"Diagnostika" obvodů řadiče Ethernet. Generuje ARP pakety do sítě. Vyslání popř. příjem paketu indikuje blikáním LED. Umožňuje ověření funkce obvodů CPU a řadiče Ethernetu s navazujícími obvody pomocí osciloskopu. V této verzi neodpovídá na PING.

- **Makefile** - hlavní soubor pro MAKE
- **ether** - soubor se zadáním parametrů MAC adresy
- **www8051.asm** - zdrojový kód, hlavní modul

Wjava

*Základní ukázka jednoduchého http serveru, bez použití CGI skriptů na straně serveru. Proti projektu **welcome** je ukázka na straně klienta doplněna o drobný JavaScript pro změnu barvy pozadí.*

- **Makefile** - hlavní soubor pro MAKE
- **ether, ip** - soubory se zadáním parametrů MAC adresy a IP adresy, masky a gateway, ...
- **index.html** - html stránka uložená v procesoru
- **www8051.asm** - zdrojový kód, hlavní modul

Led1

Základní ukázka jednoduchého http serveru, s CGI skriptem na straně serveru. CGI skript testuje svícení LED na desce Web 51. Na základě testu vloží server Web51 do stránky buď obrázek svíticí LED či LED zhasnuté. - Příklady LED jsou podrobněji popsány ve vlastní kapitole - Vyhodnocování stavu LED na desce Web 51 a jejich ovládání.

- **Makefile** - hlavní soubor pro MAKE
- **ether, ip** - soubory se zadáním parametrů MAC adresy a IP adresy, masky a gateway, ...
- **index.html** - html stránka uložená v procesoru
- **www8051.asm** - zdrojový kód, hlavní modul
- **black.gif, red.gif, yellow.gif** - obrázky zhasnuté, svítící červené a žluté LED

Led2

Složitější ukázka jednoduchého http serveru, s CGI skripty na straně serveru. První CGI skript zjišťuje svícení LED na desce Web51. Na základě testu vloží do stránky příslušný symbol (**). Druhý CGI skript vyhodnocuje vrácené údaje z html formuláře a na základě vrácených dat LED zhasíná, či rozsvěcuje. - Příklady LED jsou podrobněji popsány ve vlastní kapitole - Vyhodnocování stavu LED na desce Web 51 a jejich ovládání.*

- **Makefile** - hlavní soubor pro MAKE
- **ether, ip** - soubory se zadáním parametrů MAC adresy a IP adresy, masky a gateway, ...
- **index.html** - html stránka uložená v procesoru
- **LEDsetup.asm** - zdrojový kód cgi procedury LEDsetup.cgi
- **www8051.asm** - zdrojový kód, hlavní modul

Led3

Složitější ukázka jednoduchého http serveru, s CGI skripty na straně serveru a procesem na pozadí. První CGI skript zjišťuje svícení LED na desce Web 51. Na základě testu vloží do stránky příslušný symbol (**). Druhý CGI skript zjišťuje stav řídicích proměnných procesu na pozadí (nesviť, blikej, sviť). Třetí CGI skript vyhodnocuje vrácené údaje z html formuláře a na základě vrácených dat nastavuje řídicí proměnné procesu na pozadí (nesvícení, blikání, svícení LED).

Proces na pozadí, synchronizovaný timerem, na základě svých řídicích proměnných rozsvěcuje, zhasíná či bliká s LED. Jeho úkolem je rovněž na základě stisknutí tlačítek na desce Web 51 přepnutí režimu svícení. Pro zjednodušení obsluhy používá na klientské straně JavaScript.

Příklady LED jsou podrobněji popsány ve vlastní kapitole - Vyhodnocování stavu LED na desce Web 51 a jejich ovládání.

- **Makefile** - hlavní soubor pro MAKE
- **ether, ip** - soubory se zadáním parametrů MAC adresy a IP adresy, masky a gateway, ...
- **index.html** - html stránka uložená v procesoru
- **LEDsetup.asm** - zdrojový kód cgi procedury LEDsetup.cgi
- **testLED.asm** - zdrojový kód cgi procedury testLED.cgi
- **www8051.asm** - zdrojový kód, hlavní modul

Serial

Ukázková aplikace pro přenos dat ze sériového kanálu pomocí TCP/IP. Využívá již vyšší verzi knihovny s přeloženou podporou sériové linky a "telnet" klienta i HTTP serveru. Výsledkem je **konvertor ETHERNET-RS232** konfigurovatelný přes WWW rozhraní s přenosem pomocí plného TCP/IP. - viz popis příkladu "Telnet" Setup.

- **Makefile** - hlavní soubor pro MAKE
- **ether, ip** - soubory se zadáním parametrů MAC adresy a IP adresy, masky a gateway, ...
- **config.html, index.html, setup.html** - html stránky uložené v procesoru
- **ipsetup.asm** - zdrojový kód cgi procedury ipsetup.cgi
- **www8051.a51** - zdrojový kód, hlavní modul

DS1620

Ukázková aplikace WWW teploměru používající čip DS1620 pro měření teploty. Teplota se zobrazuje jednak číselně a jednak jako grafická stupnice - Viz. popis příkladu Teploměr DS1620.

- **Makefile** - hlavní soubor pro MAKE
- **ether, ip** - soubory se zadáním parametrů MAC adresy a IP adresy, masky a gateway, ...
- **temp.html** - html stránka uložená v procesoru
- **blue.gif, redblue.gif, stupnice.gif, whblue.gif** - obrázky pro grafické ztvárnění naměřené teploty uložené v procesoru
- **www8051.asm** - zdrojový kód, hlavní modul

V archivu aktuální verze postupně najdete některé další projekty. Všechny příklady jsou obsaženy v balíku programu **Web51 GNU development system** v sekci DOWNLOAD.

DESIGNER SUPPORT

Pokud jste si zaplatili službu „**Web51-designer support“** což je samostatná aplikační podpora, získáváte :

- Updaty softwaru zdarma 18 měsíců
- Emailové poradenství v řešení souvisejících problémů
- Zasílání aktuálních informací

Závěr:

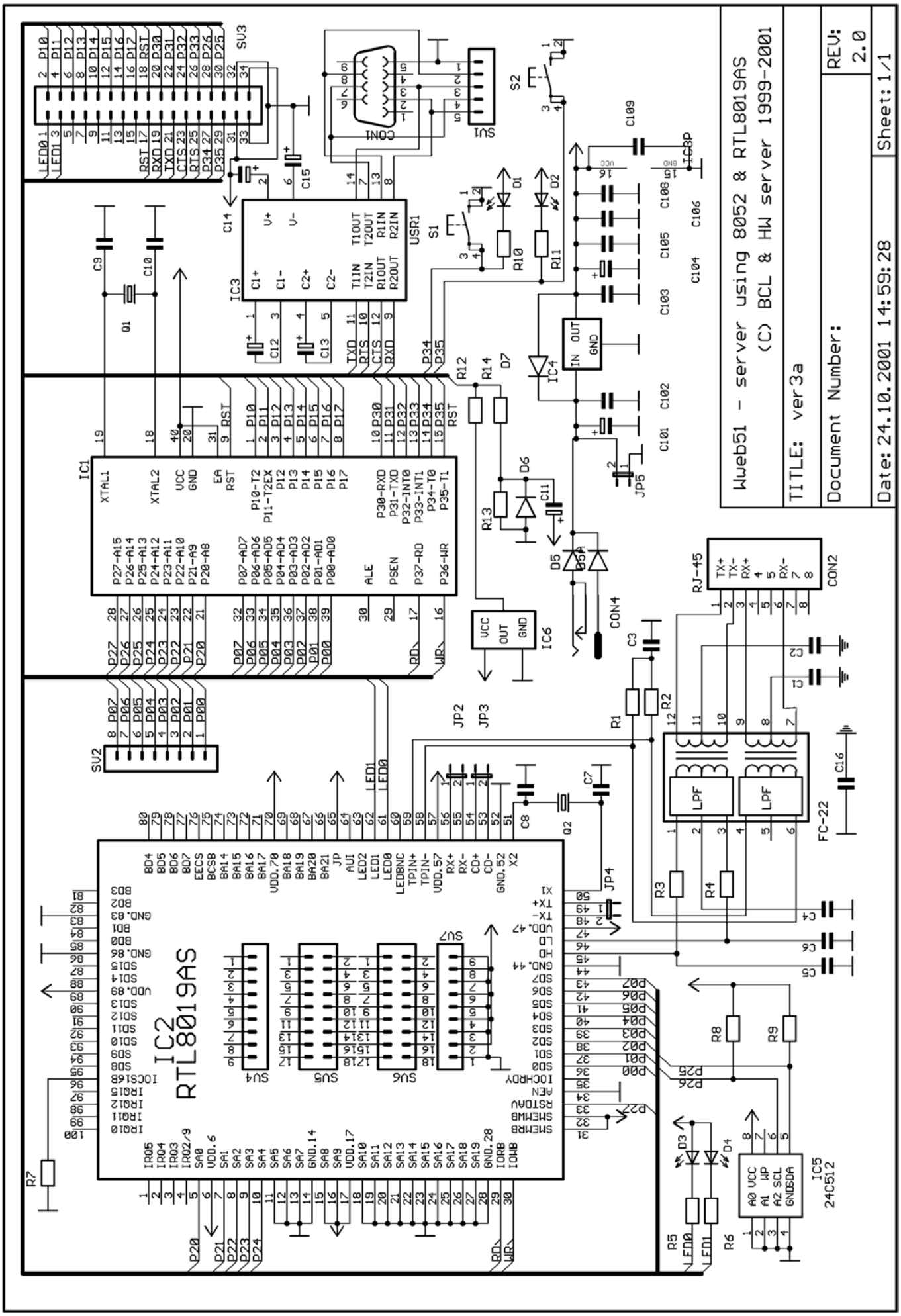
O dalších updatech a novinkách vás budeme nadále informovat.

Dokumentace a hlavně **přehled novinek** najdete na adrese **http://Web51.HW.cz**

únor 2002

Realizační team Web51

Web51@HW.cz


Web51 - server using 8052 & RTL8019AS
(C) BCL & HW server 1999-2001
TITLE: ver3a
Document Number:
REV: 2.0
Date: 24.10.2001 14:59:28
Sheet: 1/1
IC1
IC2
RTL8019AS
IC3
IC4
IC5
24C512
IC6
USR1
CON1
CON2
CON4
RJ-45
SV1
SV2
SV3
SU1
SU2
SU4
SU5
SU6
SU7
S1
S2
Q1
Q2
LPF
FC-22
JP1
JP2
JP3
JP4
JP5