

Mikrokontoléry

Atmel AVR

ASSEMBLER

Vladimír Váňa

Praha 2003



Publikace poskytuje základní informace důležité pro aplikaci jednočipových mikrokontrolérů AVR firmy ATMEL. Klade si za cíl seznámit především začátečníky s tvorbou programového vybavení pro mikrokontroléry AVR. Volně navazuje na knihu "Mikrokontroléry Atmel AVR – popis procesoru a instrukční soubor".

V této knize se budeme věnovat pouze jednomu assembleru a to ATMEL AVR assembleru pro mikrokontroléry řady AT90S. Tento assembler poskytuje zdarma firma ATMEL.

Nejprve se seznámíme se strukturou AVR programů v assembleru, práci s registry, porty, použitím SRAM, řízením chodu programů a prováděním výpočtů v assemblerech AVR. Rovněž je uveden popis ATMEL assembleru AVR tak, jak ho uveřejnil výrobce v helpu k tomuto assembleru. Tyto kapitoly obsahují jen fragmenty kódů. Proto je na konci knihy uvedeno pro začátečníky několik jednoduchých, avšak úplných programů, odzkoušených s AT90S8515 ve startkitu uvedeném v příloze.

Na doprovodném CD jsou kromě výpisů programů a softwarových balíčků umístěna i klíše plošných spojů všech čtyř přípravků (programátoru ISP a tři startkitů).

Vladimír Váňa

Mikrokontroléry Atmel AVR assembler

Bez předchozího písemného svolení nakladatelství nesmí být kterákoli část kopírována nebo rozmnožována jakoukoli formou (tisk, fotokopie, mikrofilm nebo jiný postup), zadána do informačního systému nebo přenášena v jiné formě či jinými prostředky.

Autor a nakladatelství nepřijímají záruku za správnost tištěných materiálů. Předkládané informace jsou zveřejněny bez ohledu na případné patenty třetích osob. Nároky na odškodnění na základě změn, chyb nebo vynechání jsou zásadně vyloučeny.

Všechny registrované nebo jiné obchodní známky použité v této knize jsou majetkem jejich vlastníků. Uvedením nejsou zpochybněna z toho vyplývající vlastnická práva.

Veškerá práva vyhrazena

© Vladimír Váňa, Praha 2003

© Nakladatelství BEN – technická literatura, Věšínova 5, Praha 10

Vladimír Váňa: Mikrokontroléry Atmel AVR assembler

BEN – technická literatura, Praha 2003

1. vydání

ISBN 80-7300-093-8

OBSAH

Co najdete na doprovodném CD-ROM	7
Ještě než se začnete	8
Assemblery	9
1 Struktura AVR programů v assembleru	11
Komentáře	11
Začátek programu	12
Struktura programového kódu	13
2 Registry	17
Rozdílné registry	18
Registry – ukazatele	19
Shrnutí o používání registrů	22
3 Porty procesorů AVR	23
Některé důležité porty u AVR	25
Stavový registr	26
Vstupní/výstupní brány AVR	27
4 Použití SRAM v AVR assembleru	29
SRAM	29
Důvody používání SRAM	30
Ukázky používání SRAM	30

Použití SRAM jako zásobníku	32
Definice SRAM jako zásobníku	32
Použití zásobníku	33
Chyby při zásobníkových operacích	34

5 Řízení chodu programu 35

Reset	35
Lineární provádění programu a skoky	37
Časování při běhu programu	38
Makra a běh programu	39
Podprogramy	40
Přerušeni a běh programu	42

6 Výpočty v AVR assembleru 47

Základní datové typy v assembleru	47
BCD (Binary Coded Digits)	48
Zhuštěná BCD	49
Čísla v ASCII formátu	49
Bitové operace	50
Posuv a rotace	51
Sčítání, odčítání a porovnávání	53
Konverze číselných formátů	56

7 AVR assembler 59

Obecné informace	59
Zdrojový kód assembleru	59
Mnemonické instrukce	60

Direktivy assembleru	65
Výrazy	74
Funkce	79
Obecné použití	79

8 První program

Střídavé rozsvěcení LED programované pomocí časových smyček	85
---	----

Příklad 2

Střídavé rozsvěcení LED programované pomocí maker	93
---	----

Příklad 3

Střídavé rozsvěcení LED programované pomocí podprogramu	95
---	----

Příklad 4

Ovládání rozsvěcení LED pomocí tlačítek	97
---	----

Příklad 5

Ovládání rozsvěcení LED pomocí tlačítek s použitím různých programátorských obrátů	99
---	----

Příklad 6

Programování hardwarového čítače v poling módu	104
--	-----

Příklad 7

Programování hardwarového čítače s použitím přerušeni	107
---	-----

Příklad 8

Využití hardwarového čítače ke generování sekundových tiků a jejich čítání s převodem výsledku do BCD kódu	110
---	-----

Příklad 9

Programování s využitím tabulky pevných hodnot	116
--	-----

Příklad 10

Ukázka programátorských „fint“, volání podprogramu pomocí zásobníku ... 119

Příklad 11

Vysílání a příjem sériových znaků pomocí UARTu 121

Příloha

Příloha 1

Popisek zapojení tlačítek a LED připojených k portům 125

Příloha 2

Příklady zapojení s MCU AVR – konstrukce startkitů 126

Programovací jazyky pro AVR 126

Příloha 3

AVR ISP programátor 135

Literatura 137

Přehled přípravků ATMEL a plošné spoje 138

Knihy nakladatelství BEN – technická literatura 139

Kontaktní adresy na firmu BEN – technická literatura 143

Pár slov o nakladatelství BEN – technická literatura 144

CO NAJDETE NA DOPROVODNÉM CD-ROM

Doprovodné CD-ROM obsahuje všechny informace potřebné pro snadnou práci s knihou. Tyto informace lze rozdělit do logických celků, které se nacházejí v oddělených adresářích:

- adresář **BEN** obsahuje off-line verzi www stránek nakladatelství BEN – technická literatura (aktualizováno k počátku léta 2003), jejichž součástí je počítačová verze tištěného katalogu – Edičního plánu „jaro a léto 2003“ a samostatného přehledu naší produkce „BEN 2003“.
- adresář **DATASHEET** obsahuje dokumentaci ve formátu PDF vybraných integrovaných obvodů ATMEL, které jsou v knize používány. Najdete zde též samorozbalitelný archiv programu Adobe Acrobat Reader verze 5.0, který slouží k prohlížení PDF souborů,
- adresář **PRIKLADY** obsahuje zdrojové i přeložené formy všech programů realizovaných v knize,
- adresář **SPOJE** obsahuje klíše plošných spojů všech přípravků popsanych v knize ve formátu TIF, aby si čtenáři případně mohli plošné spoje upravit podle vlastních představ. Některé spoje jsme ještě v redakci dodatečně upravovali (tvar úchytných plošek pro součástky, ...), takže finální verze je uložena pouze ve formátu TIF (je shodná s klíše otištěnými v knize).
- adresář **SW** obsahuje samostatné složky s volně šiřitelnými verzemi nebo demoverzemi vývojového prostředí určeného pro procesory ATMEL AVR.

ATMEL – obsahuje především více verzí vývojového prostředí AVR Studio v3.20, v3.56 a v4.07. Všechny verze pracují pod operačním systémem Windows. Starší verze (3.xx) jsme uvedli proto, že pracují téměř na každém PC s prostředím alespoň Windows 95. Navíc je na CD program WAVRASM v1.30, který rovněž umožňuje kompletní vývoj programů pro ATMEL AVR v assembleru. Pro čtenáře bude jistě i užitečný ovládací program pro programátor ATMEL AVR ISP 3.30, který je rovněž ve složce ATMEL.

BASCOM – vývojové prostředí včetně překladače z jazyka, který se podobá známému Visual Basicu 6.0. Je produktem firmy MCS Electronics. Omezení je na maximálně 2 kB výsledného kódu (HEX). Výhodou jsou speciální příkazy podporující práci s LCD displeji, komunikací I²C, 1WIRE atd.

CVAVR – výborným kompilátorem C pro AVR, včetně vývojového prostředí, je CodeVision AVR. Rovněž tento překladač C lze nainstalovat jako součást AVR Studia. Zdarma je jeho školní verze (CodeVisionAVR C Compiler v1.23.5 Evaluation), jejímž jediným omezením je velikost výsledného kódu do 2 kB.

GNU_C – Kompilátor C, který lze nainstalovat jako součást AVR Studia. Na tento překladač není žádné časové omezení nebo omezení velikosti kódu. Je k dispozici zcela zdarma. Pro jeho užití je pouze nutné dodržet licenci GNU.

IAR – obsahuje časově omezená vývojová prostředí firmy IAR. Jedná se především o assembler a překladač z jazyka C/C++. Konkrétně se jedná o IAR Embedded Workbench Evaluation version for Atmel AVR v2.27B a IAR Embedded Workbench Assembler Edition for Atmel AVR v1.50B. Navíc je zde umístěn i produkt IAR MakeApp for Atmel AVR v3.01.

JAVA – klasická Java, ke které jsou přidány knihovny JEPES dánské firmy Mjølner Informatics. Demoverze umožňuje programovat pouze AT90S8515.

PASCAL – ideální prostředek pro programování, jedná se o školní verzi produktu (demo) německé firmy E-LAB Computers. Omezení je na maximálně 4 kB výsledného kódu (HEX), což pro většinu aplikací stačí. V assembleru to představuje cca 6000 řádků kódu.

PONYPROG – volně šiřitelný ovládací program firmy LancOS, který je určen pro programátory mnoha typů včetně mikrokontrolérů ATMEL AVR. Program je lokalizován do mnoha světových jazyků včetně slovenštiny.

PROG910 – ovládací sw pro programátor s AT90S1200. Autor knihy jej upravil i pro mikrokontrolér AT90S8535.

JEŠTĚ NEŽ SE ZAČTETE

Publikace poskytuje základní informace důležité pro aplikaci jednočipových mikrokontrolérů AVR firmy ATMEL. Navazuje na knihu „Mikrokontroléry Atmel AVR, popis procesoru a instrukční soubor“, která seznamuje čtenáře především s RISCovou architekturou mikrokontrolérů ATMEL řady AT90 a se souborem instrukcí – popisuje tedy především hardware a instrukční soubor. Jeho znalost se nám bude hodit v této další publikaci, jež si klade za cíl seznámit především začátečníky s tvorbou programového vybavení pro mikrokontroléry AVR.

K tvorbě programového vybavení je potřeba mít vhodné nástroje a znalosti, jak tyto nástroje používat. Mají vývojářům co nejvíce usnadnit práci, zlevnit a zrychlit vývoj konkrétních aplikací. Těmito nástroji jsou překladače z kódu nějakého programovacího jazyka do strojového kódu MCU AVR, simulátory, emulátory a vlastní vývojové prostředí. Pokud jde o programovací jazyky pro mikrokontroléry AVR, máme k dispozici jednak různé **jazyky symbolických adres**, jednak vyšší programovací jazyky C/C++, Pascal a Basic.

U „velkých“ počítačů (mezi ně můžeme počítat i počítače PC, pokud je budeme srovnávat s „jednočipáky“) byl dlouhou dobu nejrozšířenějším a nejžádanějším programovacím jazykem C++ a C. Podle některých studií byl v poslední době předstížen jazykem Java. Jde o objektový jazyk odvozený od C a C++, původně vyvíjený pro „vestavěné (Embedded) systémy“, což je odborný termín pro běžná elektronická zařízení (typu pračky, mikrovlnné trouby atd.) ovládaná zabudovaným mikroprocesorem. Firma SUN, která tento jazyk vyvíjela, si brzy uvědomila vzrůstající důležitost WWW a možnosti využít Javu pro programování aplikací pro WWW. Proto si většinou spojujeme Javu právě s Internetem. Rozšířenost Javy a její výhody vedly v poslední době k tomu, že si ji všimli ti, kterým byla původně určena – vývojáři embedded zařízení. Jako příklad si můžeme uvést programování mobilních telefonů v Javě. Ze vzrůstající oblíbenosti Javy i při vývoji programového vybavení pro embedded zařízení lze usuzovat, že zasáhne i vývoj pro MCU AVR (viz např. projekt/knihovna Javy JEPES dánské firmy Mjolner Informatics z jejichž [www stránek](#) si lze zdarma stáhnout funkční demo verzi umožňující programovat AT90S8515 v Javě).

Kromě ulehčení práce vývojáře je další výhodou použití vyšších programovacích jazyků při tvorbě programového vybavení pro MCU, či jednočipové mikropočítače i to, že zdrojový kód programu určeného původně pro určitý typ MCU se dá snadněji přepsat na zdrojový kód jiného typu MCU, jiného výrobce než přepisovat kód, který je více závislý na hardware (assembler). Jako příklad mohu uvést např. IAR Embedded Workbench, což je jednotné vývojové prostředí s velkým počtem překladačů z jazyka C++ do strojových kódů různých jednočipových mikropočítačů a mikrokontrolérů, samozřejmě včetně ATMEL AVR, x51, PIC atd. Stále však existuje jistá třída úloh, kterou ve vyšších programovacích jazycích nelze realizovat, nebo

k nimž nemáme v příslušném vyšším jazyce zabudovanou podporu. V tom případě musíme použít jazyk symbolických adres – assembler. Nejeefektivnější v takovém případě je napsat aplikaci ve vyšším programovacím jazyce a jen její část v assembleru. Proto i v případě, že používáme vyšší programovací jazyky, je dobré mít alespoň *základní znalosti programování v assembleru*. K jejich získání má posloužit tato kniha. Programování AVR aplikací ve vyšších programovacích jazycích budou předmětem dalších publikací.

Assemblery

Jazyky symbolických adres patří mezi nejstarší programovací jazyky, se kterými se setkáváme již u počítačů první generace, tedy již před několika desítkami let. Dříve, než se u počítačů objevily, programovalo se přímo ve strojovém kódu, tj. většinou v binární, oktálové nebo hexadecimální reprezentaci instrukcí počítače. Všechny objekty, s nimiž počítač pracoval, byly pochopitelně označovány pouze číselně. Např. naplnění registru 16 konstantou má ve strojovém jazyce mikrokontroléru ATMEL AVR tvar:

```
1100 1111 0000 0110
```

Poznámka: *Že jde o binární reprezentaci, budeme zapisovat
0b1100111100000110*

Je zřejmé, že programování ve strojovém jazyce je obtížné a nepřehledné a mohlo vyhovovat jen v úplných začátcích, kdy programy byly krátké a jednoduché.

Se zdokonalováním technického vybavení rostly nároky na programy a programování ve strojovém jazyce začalo být neúnosné. Brzo se však zjistilo, že pracnost programování lze značně snížit, zrušíme-li nutnost označovat objekty v instrukcích číselně (vyhovuje stroji) a zavedeme-li symbolické označování objektů (vyhovuje člověku) s tím, že vazbu symbolů na jejich číselné vyjádření nebude provádět programátor, ale zajistí ji specializovaný program – překladač. Tak vznikly jazyky symbolických adres, v nichž se mohly operační znaky instrukcí a jejich operandy označovat symboly, a překladače jazyků symbolických adres (assemblery), které překládaly symbolický jazyk do strojového jazyka.

V dalším vývoji byly do jazyků symbolických adres doplňovány nové prostředky (např. makrojazyk, makra), přičemž vývoj byl obvykle spjat s rozšířením symbolických objektů. Makrojazyk je např. založen na možnosti pojmenovat posloupnost instrukcí, bez makrojazyka bylo možné pojmenovávat pouze objekty v instrukcích.

Brzy se ukázalo, že programátorovi lze práci ještě více usnadnit, zbavíme-li ho závislosti na instrukční síti počítače, se kterou byl spjat i při programování v jazyce symbolických adres. Objevily se proto jazyky nezávislé na počítači, které jsou na-

vrženy tak, aby programátor instrukční část počítače vůbec nemusel znát. Jazyky symbolických adres však nebyly zcela nahrazeny, neboť stále existují úlohy, které ve vyšších programovacích jazycích nelze vyřešit. Při programování v jazyce symbolických adres se totiž dostáváme do nejužšího styku se systémem, s hardware.

Velké počítače obvykle řeší úlohy, snadno řešitelné pomocí vyšších programovacích jazyků, jednočipové mikropočítače a mikrokontroléry naopak úlohy, řešitelné assembly. Vyšší programovací jazyky jsou implementovány řadou výrobců a pro mnoho různých procesorů pracujících pod různými operačními systémy, často pro určitý operační systém a procesor existuje i dosti velký počet různých překladačů z téhož jazyka. Různé verze překladačů téhož jazyka se liší rychlostí překladačů, velikostí a rychlostí výsledného kódu, komfortem vývojového prostředí, knihovnami atd. Definice vlastního vyššího programovacího jazyka, jeho lexikální symboly, syntaxe, sémantika, je však často dána nějakou normou, ať mezinárodní (např. ANSI C, ANSI C++) nebo podnikovou (Java u SUNu). Proto různé příručky či učebnice např. jazyka C, C++, Javy jsou použitelné při práci s různými překladači. Můžeme říci: programuji v C++, Javě, Pascalu, SQL, C#. Prohlásit „programuji v assembleru“ již není tak jednoznačné. Z toho, co jsme o assemblerech zatím uvedli je zřejmé, že máme mnoho různých assemblerů. Jednak různé procesory mají různé instrukční soubory, jednak jsou rozdíly i mezi assembly pro určitý konkrétní typ procesoru. Vždy je nutné prostudovat dokumentaci k příslušnému assembleru. My se v této knize budeme věnovat pouze jednomu assembleru a to ATMEL AVR assembleru pro mikrokontroléry řady AT90S. Tento assembler poskytuje zdarma firma ATMEL. Je naprosto odlišný od assemblerů např. pro 8086 či x51 nebo PIC. Na druhé straně bude mít hodně společného s jinými assembly pro MCU řady AT90S, např. s IAR AVR assemblerem, takže informace z této knížky můžeme s určitou dávkou opatrnosti použít i při práci s jinými assembly pro AVR.

V následujících kapitolách se nejprve seznámíme se strukturou AVR programů v assembleru, práci s registry, porty, použitím SRAM, řízením chodu programů a prováděním výpočtů v AVR assembleru. Rovněž je uveden popis ATMEL AVR assembleru tak, jak ho uveřejnil výrobce v helpu k tomuto assembleru. Tyto kapitoly obsahují jen fragmenty kódů. Proto na konci knihy je pro začátečníky uvedeno několik jednoduchých, ale úplných programů, odzkoušených s AT90S8515 v startkitu uvedeném v příloze této publikace.