

Obsah

Skripta jsou určena především pro střední „Číslicové techniky a elektronické měření“ bakalářského studia a velkou část má i charakter zpracování signálů pro magisterské studium. Navazuje na studijní pomůcky [1], vydávané pro využití při a zaktivnění laboratorní práce studentů. Alespoň skriptům předchází výklad architektury a struktury jednodívkového mikroprocesoru a jeho periferního řazení.

Úvodem	1
1. Organizační operace nad daty: nastavení hodnoty, kopírování, přesuny, čtení dat, používání zásobníku a vyrovnávací paměti	4
2. Řízení běhu programu: nepodmíněné a podmíněné větvení, větvení řízené tabulkou, transport dat přes zásobník, podprogramy a podprogramy pro obsluhu přerušení, obsluha vnějšího přerušení	25
3. Aritmetické a logické operace nad daty: aritmetika 8 bit a 16 bit, převody mezi číselnými soustavami, převody mezi různým zobrazením čísel, zrcadlový obraz, kontrolní součet crc, hashing funkce	36
4. Odměřování času: zpožďovací smyčky, časovače a čítače, obsluha časovače a čítače podsystémem přerušení, generování periodických signálů	82
5. Operace nad vstupními a výstupními registry portů: světelná indikace diodami LED a segmentovkami, lcd displej, tlačítka a spínače, klávesnice, programová emulace nestandardních rozhraní	93
6. Zpracování analogových napětí: analogový komparátor, A/D převodník, D/A převodník, nestandardní použití vestavěných analogových modulů	113
7. Programová podpora standardních komunikačních sběrnic: seriové sběrnice RS232C, SPI, TWI (I ² C)	122
8. Vytváření a používání makro instrukcí: kdy a jak	137
Přílohy:	
Příloha A1: Pomocný program AVR Fp CALC and Timer Tool	146
Příloha A2: Pomocný program AVR DL	148
Příloha A3: Pomocný program AVR Stimuli File Generator	150
Literatura	151

Rovněž nelze pominout uživatelskou příručku IDE, relativně snadné - téměř intuitivní - ovládání pro obsluhu se znalostmi ovládání počítačů a počítačových operačních systémů typu MS Windows a vysoké didaktické hodnoty IDE.

Skripta se soustředí na jazyk smíšených adres ISA assembler, který je pro dosažení cílů výuky ve výše uvedených představitel vhodnější než vyšší programovací jazyk. Plánovaný druhý díl skriptů se bude naopak soustředit na jazyk C. Rovněž u vyššího programovacího jazyka se může vyskytnout potřeba dobré znalosti JSA, typickým případem je pochopení funkce speciálních vestavěných modulů jednodívkového mikroprocesoru, komunikačních protokolů a jejich časování, varianta nastavení funkčních modulů jednodívkového mikroprocesoru prostřednictvím io registrů, zápis do nejvyššího prováděných časově kritických segmentů aplikačního programu, vytváření vlastních ovladačů do knihovny jazyka std. Z tohoto pohledu je studium předložených skript vhodnou přípravou ke studiu mnohem obtížnějšího jazyka C - dokumentujeme srovnáním zápisu téhož jednoduchého problému

assembler	jazyk C
<pre>.include "alldef.inc" ;ser r16,0xff out ddrb,r16 ;ports -out ;clr r16 out ddrb,r16 ;portb -in ser r16 out portb,r16 ;portb pull up loop: in r16,pinb ;read portb</pre>	<pre>#include <avr/io.h> int main () { ddrb = 0xff; ddrb = 0x00; portb = 0xff; while (1) {</pre>