

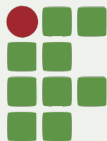
Assembler AVR

Sintaxe e Diretivas

Prof. Roberto de Matos

roberto.matos@ifsc.edu.br

09/2019



**INSTITUTO
FEDERAL**
Santa Catarina

Câmpus
São José

Objetivos

- Entender o formato do arquivo fonte esperado pelo montador do AVR
- Entender e praticar com diretivas do montador do AVR



Arquivo Fonte

■ Sintaxe do arquivo:

[Label:] diretiva [operandos] [comentários]
[Label:] instrução [operandos] [comentários]
Comentários
Linhas em branco

■ Todo texto após “;” ou “//” é comentário

■ Exemplo:

```
1  label: .EQU var1=100 ; associa a constante 100 à var1 (Diretiva)
2          .EQU var2=200 ; associa a constante 200 à var2 (Diretiva)
3
4  test:  rjmp test      ; laço infinito (Instrução)
5                          ; linha com comentário
6                          ; outro comentário
```



Diretivas do Montador

- Definição do seguimento de dados ou código: **.DSEG, .CSEG, .ESEG**
- Reserva de bytes na memória SRAM (variável): **.BYTE**
- Reserva de bytes/words (constantes): **.DB, .DW**
- Definir um nome simbólico para um registrador: **.DEF**
- Setar um símbolo a partir de uma expressão: **.EQU, .SET**
- Setar um endereço de origem: **.ORG**
- Incluir um arquivo fonte: **.INCLUDE**

¹Verificar lista completa de diretivas no [manual online do montador](#).



■ Diretivas de Segmentos

- **.CSEG** ;Segmento de Memória na Flash
- **.ESEG** ;Segmento de Memória na SRAM
- **.ESEG** ;Segmento de Memória na EEPROM

■ Exemplo:

```
1 .DSEG
2   var1: .BYTE 1           ; reserva 1 byte para var1
3   table: .BYTE tab_size ; reserva tab_size bytes
4
5 .CSEG
6   ldi r30, low(var1)      ; Carrega a parte baixa do ponteiro Z com a parte baixa do endereço de var1
7   ldi r31, high(var1)    ; Carrega a parte alta do ponteiro Z com a parte alta do endereço de var1
8   ld r1,Z                ; Carrega o conteúdo de var1 para r1 usando endereçamento indireto
```



Reserva de Bytes com Inicialização

- Reserva de bytes para variáveis (**.BYTE**) na SRAM com inicialização:

```
1 .DSEG
2   var1: .BYTE 1           ; reserva 1 byte para var1
3 .CSEG
4   ldi r30, low(var1)      ; carrega o ponteiro Z com o endereço de var1
5   ldi r31, high(var1)
6
7   ldi r16, 0xFE           ; Prepara o valor de inicialização para carregar em var1
8   st Z, r16               ; Carrega o conteúdo de var1 com r16
```

- Reserva de bytes constantes (**.DB**) na memória FLASH e EEPROM:

```
1 .CSEG:
2   consts: .DB 0, 255, 0b01010101, -128, 0xaa
3 .ESEG:
4   eeconst: .DB 0xff
```

- Reserva de *palavra* (**.DW**) constantes na memória FLASH e EEPROM:

```
1 .CSEG:
2   consts: .DW 0, 0xffff, 0b1001110001010101, -32768, 65535
3 .ESEG:
4   eeconst: .DW 0, 0xffff, 10
```



■ Sintaxe:

.DEF nome=reg

■ Exemplo:

```
1 .DEF temp=r16
2 .DEF ior=r0
3 .CSEG:
4     ldi temp, 0xf0 ; inicialize o reg. temp com 0xf0
5     in ior, 0x3f    ; carregue SREG no reg. ior
6     eor temp, ior   ; ou exclusivo de temp e ior
```



- **.EQU** NÃO permite redefinição no símbolo (*label*)
- Sintaxe:
 .EQU label = expressão
- Exemplo:

```
1 .EQU io_offset=0x23
2 .EQU porta=io_offset + 2
3 .CSEG:           ; Começo do segmento de Código
4   clr r2         ; Zera o R2
5   out porta, r2  ; Escreve no PORTB
```



- **.SET** permite redefinição no símbolo (*label*)
- Sintaxe:
 .SET label = expressão
- Exemplo:

```
1 .SET offset=0x23
2 .SET porta=offset + 2
3 .SET offset=0x11
4 .CSEG:           ; Começo do segmento de Código
5     clr r2       ; Zera o R2
6     out porta, r2 ; Escreve no PORTB
```



Especificando o endereço a ser usado pelo montador - .ORG

- **.ORG** permite definir o endereço para um valor absoluto. Pode ser usado em todos os segmentos de código (.CSEG, .DSEG e .ESEG).
- Sintaxe:
.ORG endereço
- Exemplo:

```
1
2 .DSEG                ; escrever no segmento de dados
3 .ORG 0x67            ; especificar a localizacao na SRAM
4   var1: .BYTE        ; reservar um byte na SRAM (end. 0x67)
5   var2: .BYTE        ; reservar um byte na SRAM (end. 0x68)
6
7 .ESEG                ; escrever no segmento da EEPROM
8 .ORG 0x20            ; especificar a localizacao na EEPROM
9   eevar: .DW 0xfeff ; inicialize uma palavra (end. 0x20)
10
11 .CSEG                ; escrever no segmento da Flash
12 .ORG 0x10            ; especificar a localizacao na Flash
13   mov r0,r1          ; faça algo(end.0x10)
```



Incluindo arquivos para montagem - .INCLUDE

- **.INCLUDE** informa ao montador para incluir o arquivo especificado na montagem.
- Sintaxe:
 .INCLUDE "filename"
 .INCLUDE <filename>
- No primeiro formato o montador primeiro procura no diretório local no segundo não.
- Exemplo:

```
1 .INCLUDE "m328pdef.inc" ; Arquivo de definições para o ATmega328p da Microchip:
2                           ; "xc8/v2.05/dfp/avrasm/inc/m328pdef.inc"
3 .DSEG
4     var1: .BYTE 1
5 .CSEG
6     ldi r16, 0xFF
7     ldi ZL, low(var1)   ; ZL e ZH são definidos no arquivo m328pdef.inc
8     ldi ZH, high(var1)  ; mais fácil do que lembrar que eles são R30 e R31.
9
10    st Z, r16
```



Expressões

- Rótulos definidos pelo usuário
- Variáveis definidas com a diretiva .SET
- Constantes definidas com a diretiva .EQU
- Números inteiros constantes nos seguintes formatos:
 - Decimal: 10, 255
 - Hexadecimal: 0x0a, \$0a, 0xff, \$ff
 - Binário: 0b00110101, 0b1100_1010
 - Octal (zero na frente): 010, 077
- PC localização atual do programa na memória
- Constantes ponto flutuante



- **LOW**(expression) returns the low byte of an expression
- **HIGH**(expression) returns the second byte of an expression
- **EXP2**(expression) returns 2 to the power of expression
- **LOG2**(expression) returns the integer part of $\log_2(\text{expression})$
- **ABS**() Returns the absolute value of a constant expression.
- **INT**(expression) Truncates a floating point expression to integer (ie discards fractional part)
- **FRAC**(expression) Extracts fractional part of a floating point expression (ie discards integer part).

²Verificar lista completa de funções no [manual online do montador](#).



Símbolo	Descrição
!	Logical not
~	Bitwise Not
-	Unary Minus
*	Multiplication
\	Division
%	Modulo
+	Addition
-	Subtraction
<<	Shift left
>>	Shift right
<	Less than

Símbolo	Descrição
<=	Less than or equal
>	Greater than
>=	Greater than or equal
==	Equal
!=	Not equal
&	Bitwise And
^	Bitwise Xor
	Bitwise Or
&&	Logical And
	Logical Or

³Verificar exemplos no [manual online do montador](#).

