Programování mikrokontrolerů

Atmel AVRAS2

Quick Guide

```
#include <m128def.inc>

furt:    RJMP furt    ; furt dokola
```


Preprocessor

- Preprocesor
  - víceméně podle C (s nutnými výjimkami)
  - direktivy začínají **#**
  - operátory **#** a **##**
  - předdefinovaná makra **__NĚCO__**

Preprocessor directives

#define #undef
#error #warning #message
#if #else #elif #endif
#ifdef #ifndef
#include
#pragma
(empty directive)

(stringification)
(concatenation)

#pragma - general

#pragma warning range byte *option*
  integer / overflow / none
#pragma overlap *option*
  ignore / warning / error / default
#pragma error instruction
#pragma warning instruction

#pragma – AVR part related

#pragma AVRPART ADMIN PART_NAME *string*
  V0 / V0E / V1 / V2 / V2E
#pragma AVRPART CORE CORE_VERSION *version_string*
#pragma AVRPART CORE INSTRUCTIONS_NOT_SUPPORTED
  mnemonic[operand[,operand]][: …]
#pragma AVRPART CORE NEW_INSTRUCTIONS
  mnemonic[operand[,operand]][: …]
#pragma AVRPART MEMORY PROG_FLASH *size*
#pragma AVRPART MEMORY EEPROM *size*
#pragma AVRPART MEMORY INT_SRAM *size*
#pragma AVRPART MEMORY INT_SRAM START_ADDR *address*
  0x60 / 0x100
#pragma partinclude *num*
  0 / 1

AVRAS2 assembler

- preprocessor
- keywords

- operators
- directives
- expressions
- functions

Operands

- label
  - value of the location counter at that place
- variable
  - SET directive
- constant
  - user defined using EQU directive
  - integer
    decimal binary (0b…)
    hexadecimal (0x… $...) octal (0…)
  - floating-point

Operator Precedence

1.
2.
3. **? :** (conditional expression)
4. **||** (logical OR)
5. **&&** (logical AND)
6. **|** (bitwise OR)
7. **^** (bitwise XOR)
8. **&** (bitwise AND)
9. **==** (equal) **!=** (not equal)
10. **<** (less than) **<=** (less or equal) **>** (greater than) **>=** (greater or equal)
11. **<<** (shift left) **>>** (shift right)
12. **!** (unary logical NOT) **~** (unary bitwise NOT) **+** (addition) **-** (substraction)
13. ***** (multiplication) **/** (division) **%** (modulo)
14. **-** (unary minus)

Assembler directives

- umístění v paměti
- makra
- inicializace paměti
- podmíněný překlad
- proměnné a konstanty
- výstup

  - všechny direktivy začínají .
  - case-insensitive (lze přepnout, pak jsou klíčová slova lowercase)

Directives

- BYTE
- CSEG, DSEG, ESEG
- CSEGSIZE
- DB, DW, DD, DQ
- DEF, UNDEF, EQU, SET
- DEVICE
- EXIT
- ERROR, WARNING, MESSAGE
- IF, IFDEF, IFNDEF, ELSE, ELIF, ENDIF
- INCLUDE
- LIST, NOLIST, LISTMAC
- MACRO, ENDM, ENDMACRO
- ORG
- OVERLAP, NOOVERLAP

Pre-defined macros

```
__ARVASM_VERSION__
__CORE_VERSION__
__DATE__ __TIME__
__CENTURY__
__YEAR__ __MONTH__ __DATE__
__HOUR__ __MINUTE__ __SECOND__
__FILE__ __LINE__
__PART_NAME__ __partname__
__CORE_coreversion__
```

Expressions

- konstantní výrazy
  - interně 64bit
  - operandy
    - návěští, proměnné, konstanty; PC, int, float
  - operátory
  - funkce

Functions

- LOW, HIGH
- BYTE2, BYTE3, BYTE4
- LWRD, HWRD
- PAGE
- EXP2, LOG2
- INT, FRAC
- Q7, Q15
- ABS
- DEFINED
- STRLEN

Typický design

```
.include "m128def.inc"
.def TEMP = R19

.CSEG
.ORG 0
        RJMP RESET                ; Reset Handler
        ; ... other interrupt handlers

.ORG 0x23
RESET:
        LDI TEMP,LOW(RAMEND)      ; Initial Stack Pointer
        OUT SPL,TEMP
        LDI TEMP,HIGH(RAMEND)
        OUT SPH,TEMP
        ; ... whatever else needs to be initialized
MAIN:
        ; ... here the main code starts
        RJMP MAIN
```


Typický design

```
.include "m128def.inc"
.def TEMP = R19

.CSEG
.ORG 0
        RJMP RESET                ; Reset Handle
.ORG 34
        RETI
.ORG 35
RESET:
        LDI TEMP,LOW(RAMEND)      ; Initial Stack Pointer
        OUT SPL,TEMP
        LDI TEMP,HIGH(RAMEND)
        OUT SPH,TEMP
        ; ... whatever else needs to be initialized
MAIN:
        ; ... here the main code starts
        RJMP MAIN
```