#------------------------------------------------------------------------------

# Define uVision editor settings for A251 Assembler files

#

# Copyright : (c) 2012 - 2016 ARM Ltd. and ARM Germany GmbH

# Version : 1.0.0

# Product : uVision

# Author : EG

#

#

# (A) A251 Keyword sections

# 1) A251 Instructions

# 2) A251 Operator

# 3) A251 Dircectives

# 4) A251 Dircectives Operand

# 5) A251 Assembler Statements

# 6) A251 Assembler Macro Processor Statements

# 7) A251 Assembler Macro Processor Operators

# 8) A251 Predefined Macros

# 9) A251 Register

# 10) A251 Linker Relocation

# 11) A251 Linker Alignment

#

# (B) A251 Keyword assignement

#

#------------------------------------------------------------------------------

# (A) A251 Keyword sections

#

# 1) A251 Instructions

a251\_instruction=acall add addc ajmp anl cjne clr cpl da dec div djnz \

inc jb jbc jc jmp jnb jnc jnz jz lcall ljmp \

mov movc movx mul nop orl pop push \

ret reti rl rlc rr rrc setb sjmp subb swap xch xchd xrl \

jsle jsg jle jg jsl jsge je jne movs movz bit trap ejmp ecall eret cmp sra srl sll

# 2) A251 Operators

a251\_operator=() \* + - / < <= <> == >= \

and byte0 byte1 byte2 byte3 eq ge gt high le low lt mbyte mod ne not or shl shr word0 word2 xor

# 3) A251 Directives

a251\_directive=$case $cond $date $debug $define $eject $else $elseif $endif $errorprint \

$fixdrk $gen $genonly $if $incdir $include $intr2 $list $macro $modsrc $mpl $noamake \

$nocase $nocond $nodebug $noerrorprint $nogen $nolines $nolist $nomacro $nompl $noobject \

$noprint $noregisterbank $nosymbols $nosymlist $noxref $object $pagelength $pagewidth \

$print $registerbank $reguse $reset $restore $save $set $symbols $symlist $title $xref

# 4) A251 Directives Operand

a251\_directive\_operand=$

# 5) A251 Assembler Statements

a251\_statement=\_\_error\_\_ \_\_warning\_\_ bit bseg code const cseg data db dbit dd ds dsb dsd dseg dsw dw \

ebit edata else elseif end endif endp equ esfr even extern extrn \

hdata hconst idata if iseg label lit name org proc public rseg \

sbit segment set sfr sfr16 using xdata xseg

# 6) A251 Assembler Macro Processor Statements

a251\_macro=macro exitm endm local rept irp irpc

# 7) A251 Assembler Macro Processor Operators

a251\_macro\_operator=nul & < > % ;; !

# 8) A251 Predefined Macros

a251\_predef\_macro=\_\_a251\_\_ \_\_date\_\_ \_\_date2\_\_ \_\_file\_\_ \_\_keil\_\_ \_\_line\_\_ \_\_mod\_src\_\_ \_\_time\_\_

# 9) A251 Register

a251\_register=a b ab dptr r0 r1 r2 r3 r4 r5 r6 r7 r8 r9 r10 r11 r12 r13 r14 r15 \

wr0 wr2 wr4 wr8 wr10 wr12 wr14 wr16 wr18 wr20 wr22 wr24 wr26 wr28 wr30 \

dr0 dr4 dr8 dr12 dr16 dr20 dr24 dr28 dr56 dr60 \

pc psw sp

# 10) A251 Linker Relocation

a251\_linker\_relocation=at bitaddressable inblock inpage inseg offs overlayable

# 11) A251 Linker Alignment

a251\_linker\_alignment=bit byte word dword page block seg align

# (B) A251 Keyword assignement

#

keywords.$(file.patterns.asm)=$(a251\_instruction) $(a251\_operator)

keywords2.$(file.patterns.asm)=

keywords3.$(file.patterns.asm)=$(a251\_register)

keywords4.$(file.patterns.asm)=$(a251\_directive) $(a251\_directive\_operand) $(a251\_statement) $(a251\_linker\_relocation) $(a251\_linker\_alignment)

keywords5.$(file.patterns.asm)=$(a251\_macro) $(a251\_predef\_macro) $(a251\_macro\_operator)

keywords6.$(file.patterns.asm)=$(ext\_instruction)

# (C)

#

comment.block.asm=;

command.compile.$(file.patterns.asm)=masm $(FileNameExt)

command.name.0.\*.asm=Link

command.0.\*.asm=link $(FileName)

statement.indent.$(file.patterns.asm)=9 .if