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

# Define uVision editor settings for A51 Assembler files

#

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

# Version : 1.0.0

# Product : uVision

# Author : EG

#

#

# (A) A51 Keyword sections

# 1) A51 Instructions

# 2) A51 Operator

# 3) A51 Dircectives

# 4) A51 Dircectives Operand

# 5) A51 Assembler Statements

# 6) Ax51 Memory Classes

# 7) A51 Assembler Macro Processor Statements

# 8) A51 Assembler Macro Processor Operators

# 9) A51 Predefined Macros

# 10) A51 Register

# 11) A51 Linker Relocation

# 12) A51 Linker Alignment

#

# (B) A51 Keyword assignement

#

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

# (A) A51 Keyword sections

#

# 1) A51 Instructions

a51\_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

# 2) A51 Operators

a51\_operator=() \* + - / < <= <> == >= \

AND BYTE0 BYTE1 BYTE2 BYTE3 EQ GT GTE HIGH LOW LT LTE MBYTE MOD NE NOT OR SHL SHR WORD0 WORD2 XOR

# 3) A51 Directives

a51\_directive=$case $cond $date $debug $define $ecrm $eject $else $elseif $endif $errorprint \

$gen $genonly $if $incdir $include $list $macro $mod\_cont $mod\_mx51 $mod51 $mpl $mx \

$noamake $nocase $nocond $nodebug $noerrorprint $nogen $nolines $nolist $nomacro $nomod51 \

$nompl $noobject $noprint $noregisterbank $nosymbols $nosymlist $noxref $object \

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

# 4) A51 Directives Operand

a51\_directive\_operand=$

# 5) A51 Assembler Statements

a51\_statement=\_\_error\_\_ \_\_warning\_\_ bseg const cseg edata db dbit dd ds dsb dsd dseg dsw dw \

ebit econst else elseif end endif endp equ even extern extrn \

hconst if iseg label lit name org proc public rseg \

sbit segment set sfr sfr16 using xseg

# 6) Ax51 Memory Classes

a51\_memory\_class=bit data idata xdata code bank0 bank1 bank2 bank3 bank4 bank5 bank6 bank7 bank8 bank9 bank10 \

bank11 bank12 bank13 bank14 bank15 bank16 bank17 bank18 bank19 bank20 bank21 bank22 bank23 bank24 \

bank25 bank26 bank27 bank28 bank29 bank30 bank31

a51\_extended\_memory\_class=hdata ecode

ax51\_memory\_class=hconst

# 7) A51 Assembler Macro Processor Statements

a51\_macro=macro exitm endm local rept irp irpc

# 8) A51 Assembler Macro Processor Operators

a51\_macro\_operator=nul & < > % ;; !

# 9) A51 Predefined Macros

a51\_predef\_macro=\_\_a51\_\_ \_\_ax51\_\_ \_\_date\_\_ \_\_date2\_\_ \_\_file\_\_ \_\_keil\_\_ \_\_line\_\_ \_\_mod\_cont\_\_ \_\_time\_\_

# 10) A51 Register

a51\_register=a b r0 r1 r2 r3 r4 r5 r6 r7 dph dpl pc psw dptr sp

# 11) A51 Linker Relocation

a51\_linker\_relocation=at bitaddressable inblock inpage inseg offs overlayable

# 12) A51 Linker Alignment

a51\_linker\_alignment=bit byte word dword page block seg

# (B) Keyword assignement for A51

#

keywords.$(file.patterns.asm)=$(a51\_instruction) $(ax51\_instruction) $(a51\_operator)

keywords2.$(file.patterns.asm)=

keywords3.$(file.patterns.asm)=$(a51\_register)

keywords4.$(file.patterns.asm)=$(a51\_directive) $(a51\_directive\_operand) $(a51\_statement) $(a51\_linker\_relocation) $(a51\_linker\_alignment) $(a51\_memory\_class) $(a51\_extended\_memory\_class) $(ax51\_memory\_class)

keywords5.$(file.patterns.asm)=$(a51\_macro) $(a51\_predef\_macro) $(a51\_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