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

# Define uVision settings for C166 source code files

#

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

# Version : 1.0.0

# Product : uVision

# Author : EG

#

#

# (A) C166 Keyword sections

# 1) C166 ANSI C Keywords

# 2) C166 Operator

# 3) C166 Listing Directive

# 4) C166 Object Directive

# 5) C166 Source Directive

# 6) C166 Linker Directive

# 7) C166 New Keyword

# 8) C166 Classes

# 9) C166 Memory Areas

# 10) C166 Memory Models

# 11) C166 Memory Type

# 12) C166 Data Types

# 13) C166 Predefined Macros

# 14) C166 Directive

# 15) C166 Optimizer Options

# 16) C166 absolute access macros

#

# (B) C166 Keyword assignement

#

#

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

# (A) C166 Keyword sections

#

# Nr 1) C166 ANSI C Keywords

c166\_ansi\_c\_keyword=auto break case char const continue default do double else enum extern float for goto \

if int long register return short signed sizeof static struct switch typedef \

union unsigned void volatile while

# Nr 2) C166 Operator

# c166\_operator=( ) [ ] . -> - ! ++ -- - + \* .. sizeof / % << >> < > <= > >= \

# == != & ^ | && || ?: = \*= /= %= += -= <<= >>= &= |= ^= ,

# Nr 3) C166 Listing Directive

c166\_listing\_directive=code1 cond eject2 listinclude nocond noprint1 pagelength1 pagewidth1 preprint1 \

preprintonly1 print1 symbols1 tabs1 warning2 warninglevel1

# Nr 4) C166 Object Directive

c166\_object\_directive=browse1 bytealign commonret compact1 debug1 disable2 dynamicusrstk1 extins extmac \

fix166 fix167 fix272 fix32f fixbfld fix\_br03 fixmdu fixpec float64 hcompact1 hlarge1 hold \

init2 large1 mac medium1 mod167 modv2 noalias1 noamake1 noasmexpand nodppsave1 \

nofixdpp1 noframe noinit2 noobject1 nosavemac object1 order1 pack pecdef1 \

reentrant regfile1 renameclass reorder restore2 save2 savemac savesys saveusr small1 \

src1 static tiny1 unsigned\_char1 userstackdpp31 xlarge1 \

# Nr 5) C166 Source Directive

c166\_source\_directive=asm2 asmexpand define3 endasm2 incdir1 noextend1

# Nr 6) C166 Linker Directive

c166\_linker\_directive=assign cinittab classes disablewarning dppuse elf groups intno ixref linkonly name \

noamake nocase nocomments nodefaultlibrary noinit nolines nomap noprint nopublics \

nosortsize nosymbols notype novectab objectcontrols pagelength pagewidth print \

printcontrols publicsonly purge regbank regfile reserve rtx166 rtx166tiny secsize \

sections vectab warninglevel

# Nr 7) C166 New Keyword

c166\_new\_keyword=\_\_asm bdata bit ebdata far huge idata interrupt \_\_inline near \

sbit sdata sfr \_task\_ using xhuge

# Nr 8) C166 Classes

c166\_classes=ncode nconst fconst xconst fcode ndata ndata0 sdata sdata0 \

idata idata0 bit bit0 bdata bdata0 fdata fdata0 hdata hdata0 xdata xdata0 icode

# Nr 9) C166 Memory Areas

c166\_memory\_area=code const data bdata ebdata xdata near far huge sfr

# Nr 10) C166 Memory Models

c166\_memory\_model=tiny small medium compact large hcompact hlarge xlarge

# Nr 11) C166 Memory Type

c166\_memory\_type=near bdata idata sdata far huge xhuge

# Nr 12) C166 Data Types

c166\_data\_type=char enum short int long float double bit sbit sfr

# Nr 13) C166 Predefined Macros

c166\_predefined\_macro=\_\_c166\_\_ \_\_date\_\_ \_\_date2\_\_ \_\_dus\_\_ \_\_file\_\_ \_\_float64\_\_ \_\_line\_\_ \

\_\_mac\_\_ \_\_mod167\_\_ \_\_model\_\_ \_\_modv2\_\_ \_\_savemac\_\_ \_\_stdc\_\_ \_\_time\_\_

# Nr 14) C166 Directive

c166\_directive=

# Nr 15) C166 Optimizer Options

c166\_optimizer\_option=noalias nodppsave nofixdpp optimize size static

# Nr 16) C166 absolute access macros

c166\_absolute\_access\_macros=hvar mvar xvar harray marray xarray

# (B) Keyword assignement for C166

keywords6.$(file.patterns.cpp)=$(c166\_new\_keyword) $(c166\_optimizer\_option) $(c166\_classes) $(c166\_data\_type) $(c166\_memory\_type) $(c166\_memory\_model) $(c166\_memory\_area) \

$(c166\_directive) $(c166\_source\_directive) $(c166\_listing\_directive) $(c166\_object\_directive) $(c166\_linker\_directive) $(ext\_instruction) $(c166\_predefined\_macro) $(c166\_absolute\_access\_macros)