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

# Define uVision settings for C251 source code files

#

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

# Version : 1.0.0

# Product : uVision

# Author : EG

#

#

# (A) C251 Keyword sections

# 1) C251 ANSI C Keywords

# 2) C251 Operator

# 3) C251 Listing Directive

# 4) C251 Object Directive

# 5) C251 Source Directive

# 6) C251 New Keyword

# 7) C251 Memory Areas

# 8) C251 Memory Models

# 9) C251 Memory Type

# 10) C251 Data Types

# 11) C251 Storage Classes

# 12) C251 Predefined Macros

# 13) C251 Directive

# 14) C251 Optimizer Options

# 15) C251 Optimizer Options for cx51 code (backward compatibility)

# 16) C251 absolute access macros

#

# (B) C251 Keyword assignement

#

#

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

# (A) C251 Keyword sections

#

# Nr 1) C251 ANSI C Keywords

c251\_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) C251 Operator

# c251\_operator=( ) [ ] . -> - ! ++ -- - + \* .. sizeof / % << >> < > <= > >= \

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

# Nr 3) C251 Listing Directive

c251\_listing\_directive=code cond eject listinclude nocond noprint pagelength pagewidth \

preprint print symbols tabs warning warninglevel

# Nr 4) C251 Object Directive

c251\_object\_directive=browse debug disable fixdrk float64 floatfuzzy functions hptr hold \

inerval intr2 intvector large maxargs modbin name noalias noamake nvm\_const noaasmexpand nocase \

nointvector noobject noregparms object objectadvanced optimize order pack \

parm251 parm51 regfile regparms renamecode restore rom save small src string \

tiny usigned\_char userclass xsmall xtiny

# Nr 5) C251 Source Directive

c251\_source\_directive=asm asmexpand define endasm incdir noextend preprintonly

# Nr 6) C251 New Keyword

c251\_new\_keyword=\_at\_ alien bdata bit code compact data \

far idata interrupt large pdata \_priority\_ reentrant \

sbit sfr sfr16 small \_task\_ using xdata \

# Nr 7) C251 Memory Areas

c251\_memory\_area=code const data bdata ebdata xdata near far huge sfr

# Nr 8) C251 Memory Models

c251\_memory\_model=tiny xtiny small xsmall large

# Nr 9) C251 Memory Type

c251\_memory\_type=code data bdata ebdata near xdata far huge idata

# Nr 10) C251 Data Types

c251\_data\_type=bit char enum int float double sbit sfr sfr16

# Nr 11) C251 Storage Classes

c251\_storage\_class=auto register static extern

# Nr 12) C251 Predefined Macros

c251\_predefined\_macro=\_\_c251\_\_ \_\_date\_\_ \_\_date2\_\_ \_\_file\_\_ \_\_line\_\_ \_\_model\_\_ \_\_time\_\_ \_\_stdc\_\_ \_\_float64\_\_ \_\_modsrc\_\_

# Nr 13) C251 Directive

c251\_directive=

# Nr 14) C251 Optimizer Options

c251\_optimizer\_option=regfile noalias optimize size objectadvanced

# Nr 15) C251 directives for cx51 code

c251\_c51\_compatibility\_directives=aregs noaregs register

# Nr 16) C251 absolute access macros

c251\_absolute\_access\_macros=cbyte dbyte hbyte nbyte xbyte \

cvar dvar hvar nvar xvar carray darray harray narray xarray

# (B) Keyword assignement for C251

keywords6.$(file.patterns.cpp)=$(c251\_new\_keyword) $(c251\_optimizer\_option) $(c251\_storage\_class) $(c251\_data\_type) $(c251\_memory\_type) $(c251\_memory\_model) \

$(c251\_directive) $(ext\_instruction) $(c251\_source\_directive) $(c251\_listing\_directive) $(c251\_object\_directive) $(c251\_predefined\_macro) $(c251\_c51\_compatibility\_directives) $(c251\_absolute\_access\_macros)