Keil® MDK Version 5
Component-based Software Development

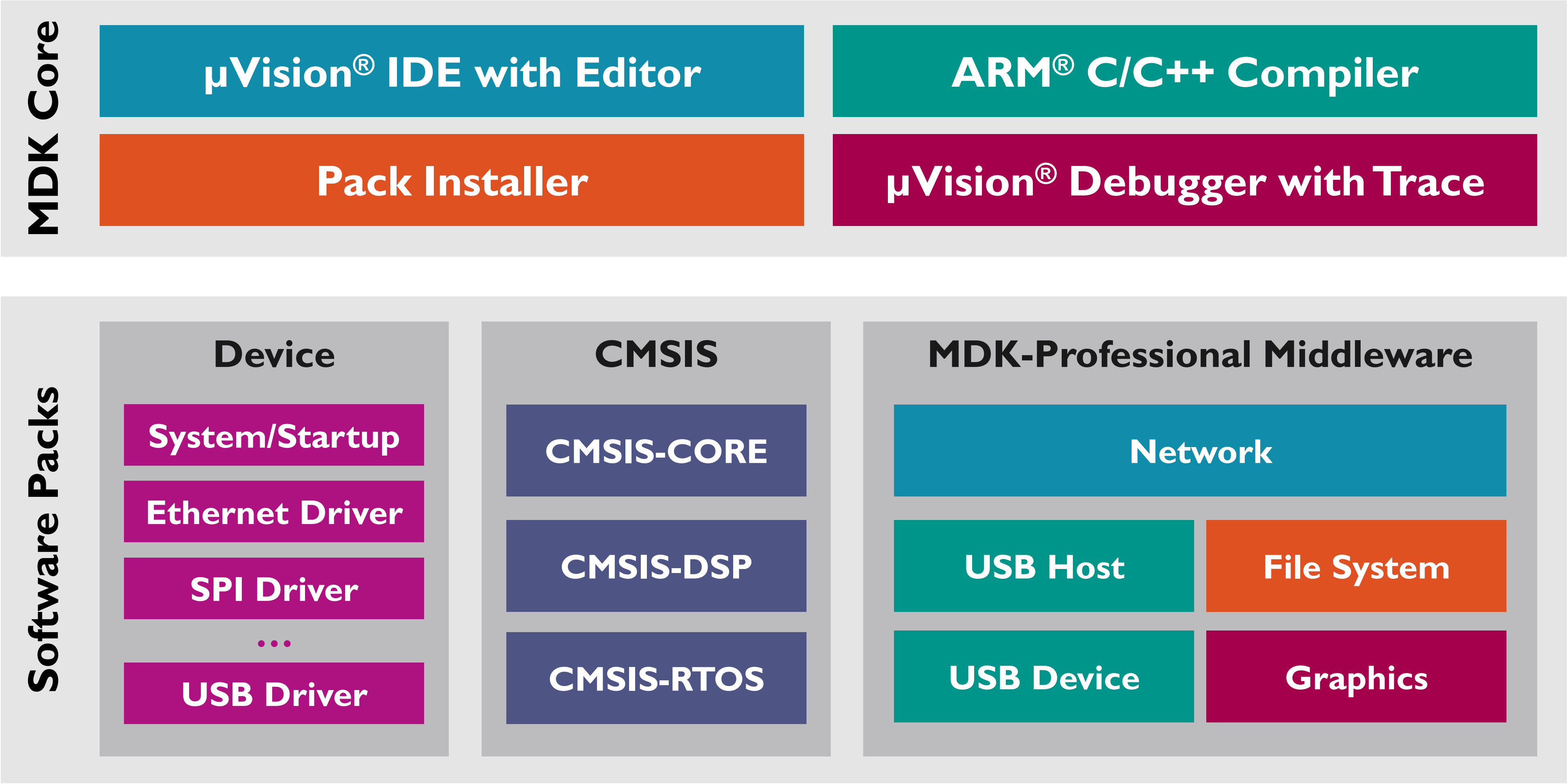
Christopher Seidl
Technical Marketing Manager, DSG

Installing the Hands-On Material

- Required hardware: XMC1200 Boot Kit and XMC4500 Relax Kit
- Required software:
  - DAVE™ 4.0.0 Build 2015-02-21(www.inifineon.com/dave)
  - MDK Version 5.14: Download from www.keil.com/demo/eval/arm.htm
  - MDK-Professional license: Free 7-Day Trial License available from within the tool (see www.keil.com/mdk5/activation)
  - Infineon XMC1000 Device Family Pack: Infineon.XMC1000_DFP.2.0.0.pack
  - Infineon XMC4000 Device Family Pack: Infineon.XMC4500_DFP.2.2.0.pack
  - ARM CMSIS Pack: ARM.CMSIS.4.3.0.pack
  - Keil Middleware Pack: Keil.MDK-Middleware.6.3.0.pack
  - Keil ARM Compiler Pack: Keil.ARM_Compiler.1.0.0.pack
- DAVE™ Project:
  - Project\XMC1200_dev_days_2015_v2.zip

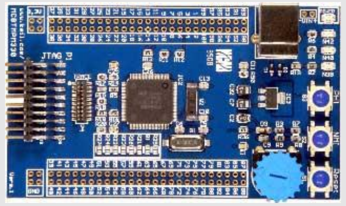
Keil® MDK Version 5 Development System

ARM® KEIL®
Microcontroller Tools



ULINK™ Debug Adapters

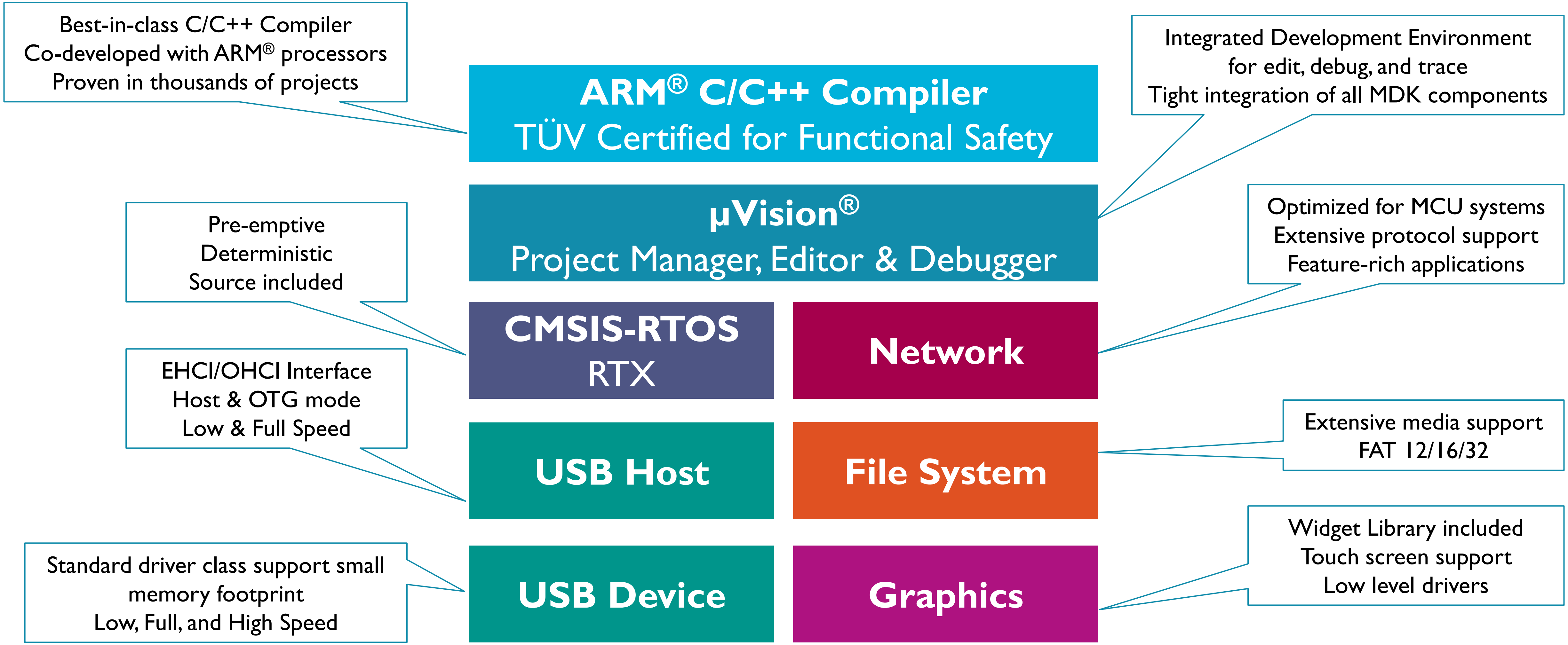
Evaluation Boards



ARM

MDK Professional

ARM® KEIL®
Microcontroller Tools



Best-in-class C/C++ Compiler
Co-developed with ARM® processors
Proven in thousands of projects

ARM® C/C++ Compiler
TÜV Certified for Functional Safety

Integrated Development Environment
for edit, debug, and trace
Tight integration of all MDK components

µVision®
Project Manager, Editor & Debugger

Pre-emptive
Deterministic
Source included

CMSIS-RTOS
RTX

Network

Optimized for MCU systems
Extensive protocol support
Feature-rich applications

EHCI/OHCI Interface
Host & OTG mode
Low & Full Speed

USB Host

File System

Extensive media support
FAT 12/16/32

Standard driver class support small memory footprint
Low, Full, and High Speed

USB Device

Graphics

Widget Library included
Touch screen support
Low level drivers

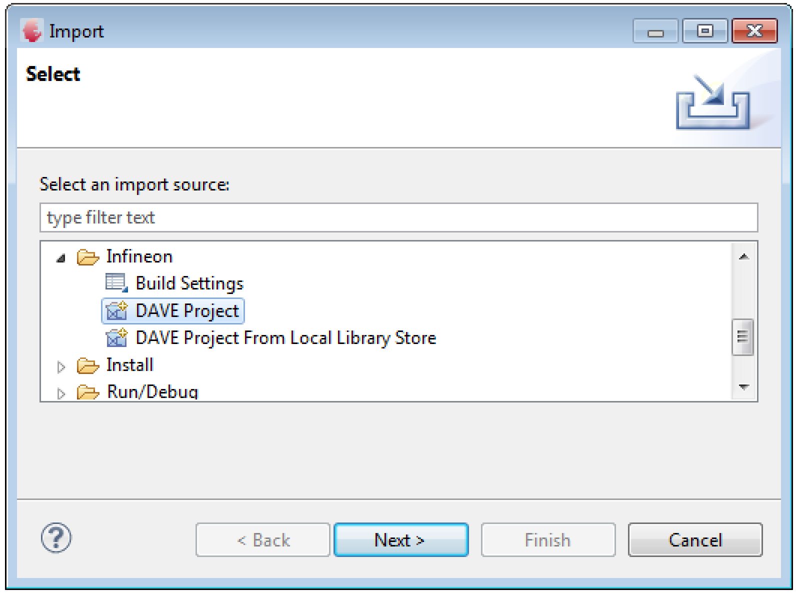
MDK Editions

	Professional	Standard	Cortex-M	XMC1000 Edition	Lite
µVision®					
IDE	✓	✓	✓	✓	✓
Debugger, Simulator	✓	✓	✓	✓	32 KB
ARM® Compiler					
C/C++ Compilation Tools	✓	✓	✓	✓	32 KB
Compiler Qualification Kit	✓				
Extended Compiler Maintenance	✓				
Device Support					
Cortex®-M	✓	✓	✓	**XMC1000 only**	✓
SecurCore® (SC000, SC300)	✓	✓			✓
ARM7™, ARM9™, Cortex®-R4	✓	✓			✓
RTOS and Middleware					
CMSIS-RTOS RTX (with full source)	✓	✓	✓	✓	✓
Middleware Libraries	✓				
Pricing	8260 €	4260 €	3340 €	**free**	free

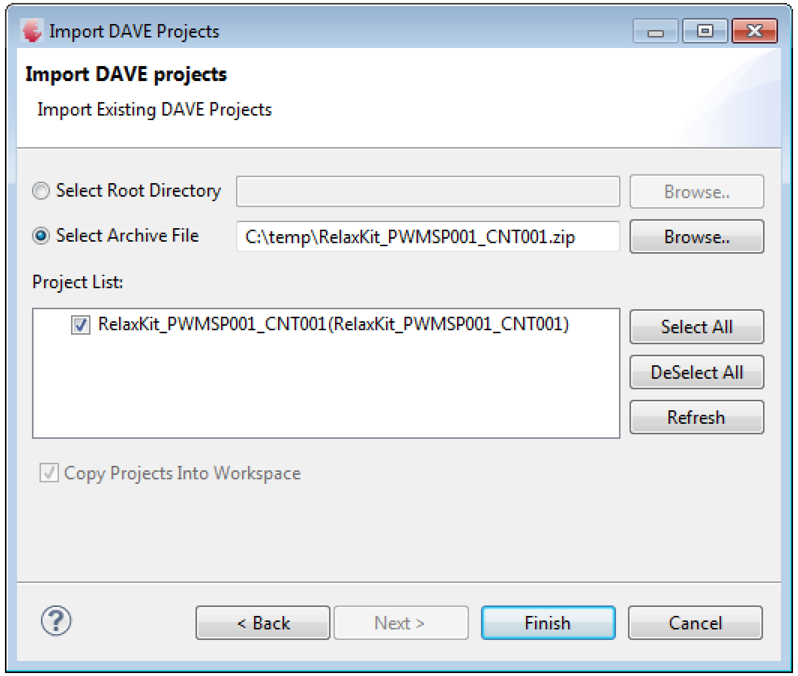
Importing DAVE™ projects in MDK Version 5
XMC1200 Boot Kit Hands-On

Import the Project into DAVE™

- Go to **File → Import → DAVE Project** and click **Next >**



- On the USB stick, browse to the file **Project\ XMC1200_dev_days_2015_v2.zip** and click **Finish**



○ Select Root Directory

● Select Archive File C:\temp\RelaxKit_PWMSP001_CNT001.zip

Project List:

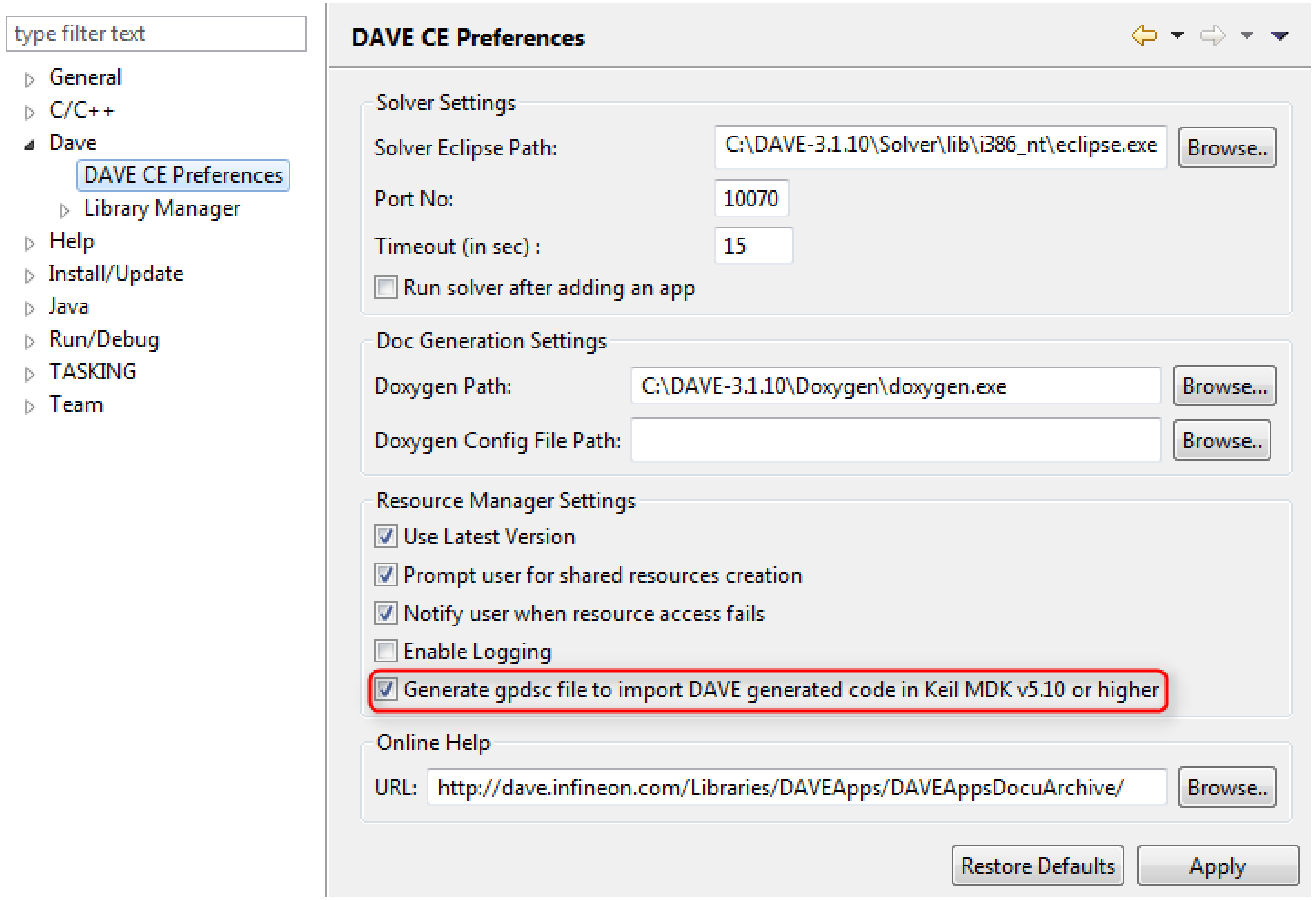
☑ RelaxKit_PWMSP001_CNT001(RelaxKit_PWMSP001_CNT001)

☑ Copy Projects Into Workspace

- Project will be imported, opened and set active automatically

Required Setting for GPDSC Generation

- Go to **Window → Preferences** and **Dave → DAVE CE Preferences**
- Make sure that **Generate gpdsc file** is enabled:



DAVE CE Preferences

Solver Settings

Solver Eclipse Path: C:\DAVE-3.1.10\Solver\lib\i386_nt\eclipse.exe

Port No: 10070

Timeout (in sec) : 15

☐ Run solver after adding an app

Doc Generation Settings

Doxygen Path: C:\DAVE-3.1.10\Doxygen\doxygen.exe

Doxygen Config File Path:

Resource Manager Settings

☑ Use Latest Version

☑ Prompt user for shared resources creation

☑ Notify user when resource access fails

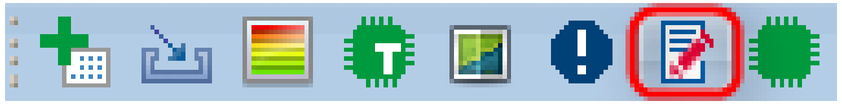
☐ Enable Logging

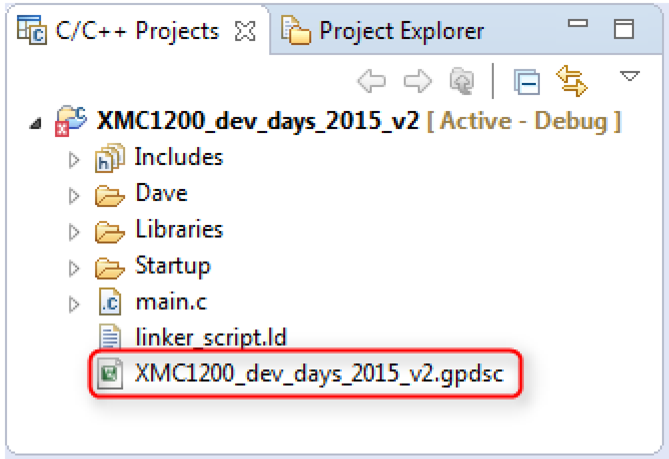
☑ Generate gpdsc file to import DAVE generated code in Keil MDK v5.10 or higher

Online Help

URL: http://dave.infineon.com/Libraries/DAVEApps/DAVEAppsDocuArchive/

Code Generation

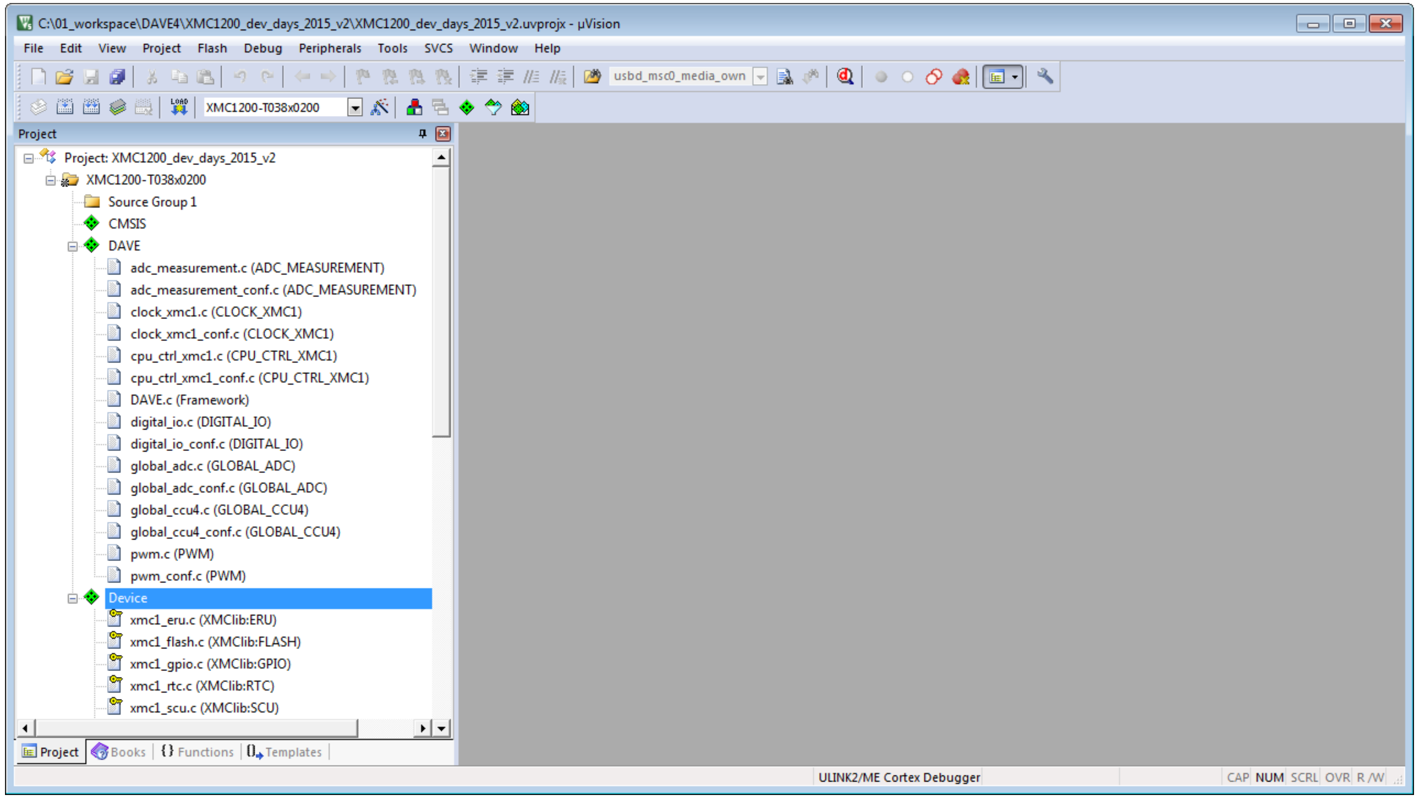
- The library code of the DAVE™ Apps needs to be generated. This will also trigger the creation of the GPDSC file.
- Click the **Generate Code** button: 
- Check if the GPDSC file has been generated:



XMC1200_dev_days_2015_v2 [Active - Debug]
- Includes
- Dave
- Libraries
- Startup
- main.c
- linker_script.ld
- XMC1200_dev_days_2015_v2.gpdsc

Invoking µVision® from DAVE™

- To import the project to µVision, simply double-click the **XMC1200_dev_days_2015_v2.gpdsc** file in the **C/C++ Projects** view:



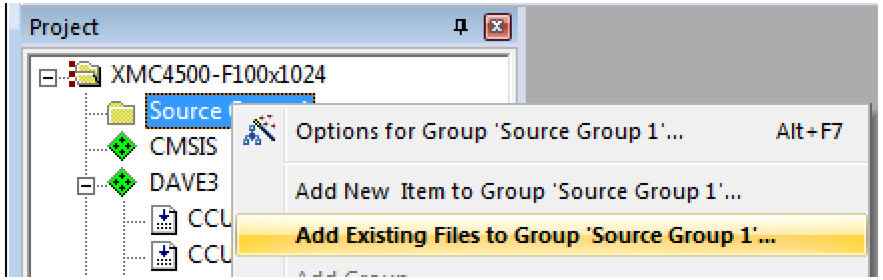
C:\01_workspace\DAVE4\XMC1200_dev_days_2015_v2\XMC1200_dev_days_2015_v2.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
usbd_msc0_media_own
XMC1200-T038x0200
Project
Project Books Functions Templates
ULINK2/ME Cortex Debugger
CAP NUM SCRL OVR R /W

- Project: XMC1200_dev_days_2015_v2
  - XMC1200-T038x0200
    - Source Group 1
    - CMSIS
    - DAVE
      - adc_measurement.c (ADC_MEASUREMENT)
      - adc_measurement_conf.c (ADC_MEASUREMENT)
      - clock_xmc1.c (CLOCK_XMC1)
      - clock_xmc1_conf.c (CLOCK_XMC1)
      - cpu_ctrl_xmc1.c (CPU_CTRL_XMC1)
      - cpu_ctrl_xmc1_conf.c (CPU_CTRL_XMC1)
      - DAVE.c (Framework)
      - digital_io.c (DIGITAL_IO)
      - digital_io_conf.c (DIGITAL_IO)
      - global_adc.c (GLOBAL_ADC)
      - global_adc_conf.c (GLOBAL_ADC)
      - global_ccu4.c (GLOBAL_CCU4)
      - global_ccu4_conf.c (GLOBAL_CCU4)
      - pwm.c (PWM)
      - pwm_conf.c (PWM)
    - Device
      - xmc1_eru.c (XMClib:ERU)
      - xmc1_flash.c (XMClib:FLASH)
      - xmc1_gpio.c (XMClib:GPIO)
      - xmc1_rtc.c (XMClib:RTC)
      - xmc1_scu.c (XMClib:SCU)

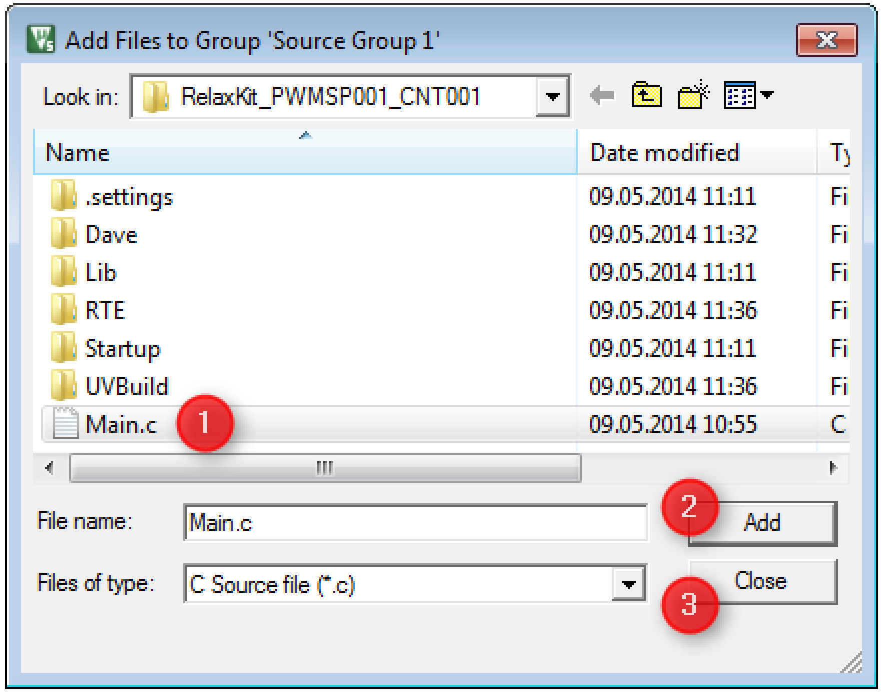
- **Note**: µVision will create a project in the same folder as the DAVE™ project

Add a Main.c File to the µVision® Project

- There is no main file in the project, so we need to add one manually
- Right click on **Source Group 1**, select **Add Existing files to 'Source Group 1'…**


Project
XMC4500-F100x1024
Source G…
CMSIS
DAVE3
CCU
CCU
Options for Group 'Source Group 1'… Alt+F7
Add New Item to Group 'Source Group 1'…
Add Existing Files to Group 'Source Group 1'…
Add Group…

1. Click on **main.c**
2. Click **Add**
3. Click **Close**



Look in: RelaxKit_PWMSP001_CNT001

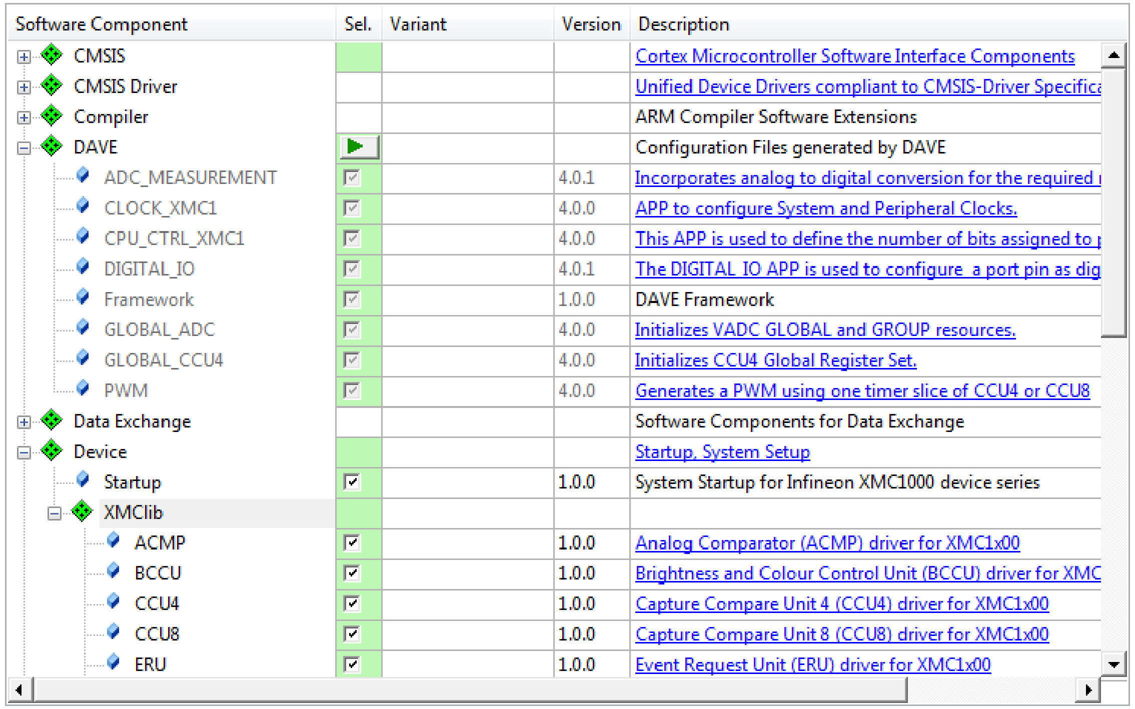
Name	Date modified	Ty
.settings	09.05.2014 11:11	Fi
Dave	09.05.2014 11:32	Fi
Lib	09.05.2014 11:11	Fi
RTE	09.05.2014 11:36	Fi
Startup	09.05.2014 11:11	Fi
UVBuild	09.05.2014 11:36	Fi
Main.c	09.05.2014 10:55	C

File name: Main.c

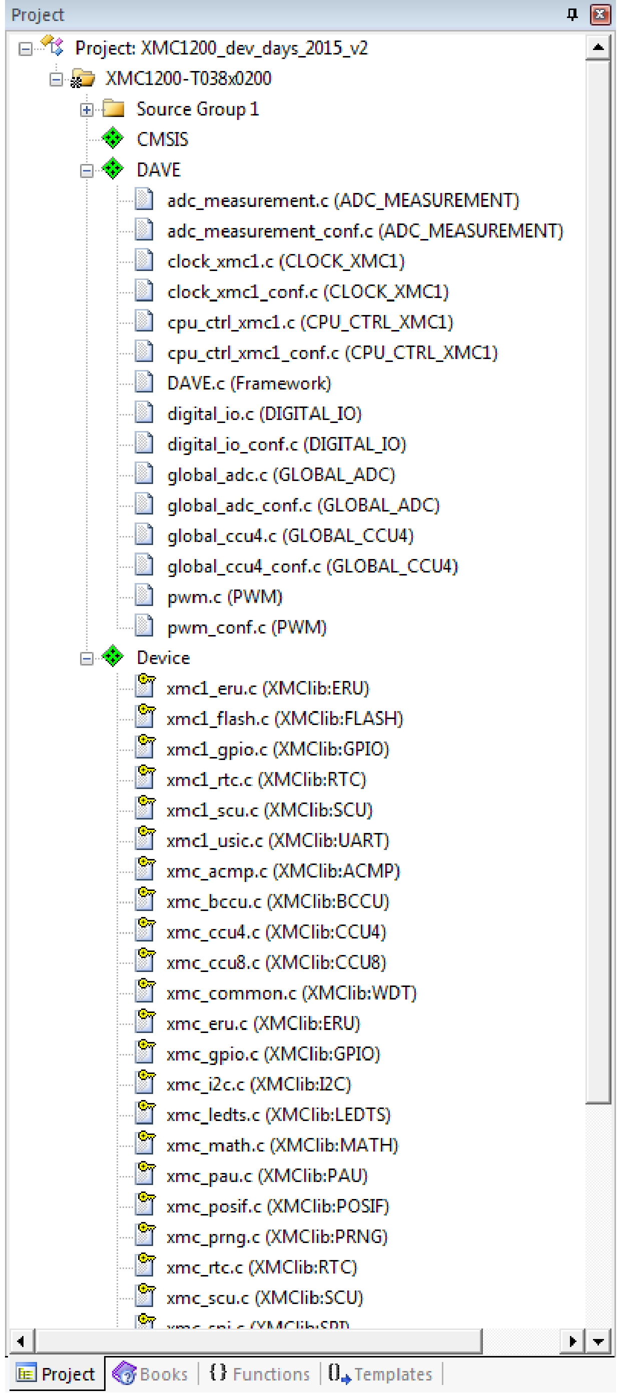
Files of type: C Source file (*.c)

Explore the Project

- The **Project** window shows you the software components:
  - **CMSIS** is representing the CMSIS-CORE framework
  - **DAVE** contains generated files which are part of the GPDSC
  - **Device** contains startup files and the XMClib low-level drivers
- The **Manage Run-Time Environment** window reflects this:



Software Component	Sel.	Variant	Version	Description
CMSIS				Cortex Microcontroller Software Interface Components
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifica
Compiler				ARM Compiler Software Extensions
DAVE	▶			Configuration Files generated by DAVE
ADC_MEASUREMENT	☑		4.0.1	Incorporates analog to digital conversion for the required
CLOCK_XMC1	☑		4.0.0	APP to configure System and Peripheral Clocks.
CPU_CTRL_XMC1	☑		4.0.0	This APP is used to define the number of bits assigned to
DIGITAL_IO	☑		4.0.1	The DIGITAL_IO APP is used to configure a port pin as dig
Framework	☑		1.0.0	DAVE Framework
GLOBAL_ADC	☑		4.0.0	Initializes VADC GLOBAL and GROUP resources.
GLOBAL_CCU4	☑		4.0.0	Initializes CCU4 Global Register Set.
PWM	☑		4.0.0	Generates a PWM using one timer slice of CCU4 or CCU8
Data Exchange				Software Components for Data Exchange
Device				Startup, System Setup
Startup	☑		1.0.0	System Startup for Infineon XMC1000 device series
XMClib				
ACMP	☑		1.0.0	Analog Comparator (ACMP) driver for XMC1x00
BCCU	☑		1.0.0	Brightness and Colour Control Unit (BCCU) driver for XMC
CCU4	☑		1.0.0	Capture Compare Unit 4 (CCU4) driver for XMC1x00
CCU8	☑		1.0.0	Capture Compare Unit 8 (CCU8) driver for XMC1x00
ERU	☑		1.0.0	Event Request Unit (ERU) driver for XMC1x00

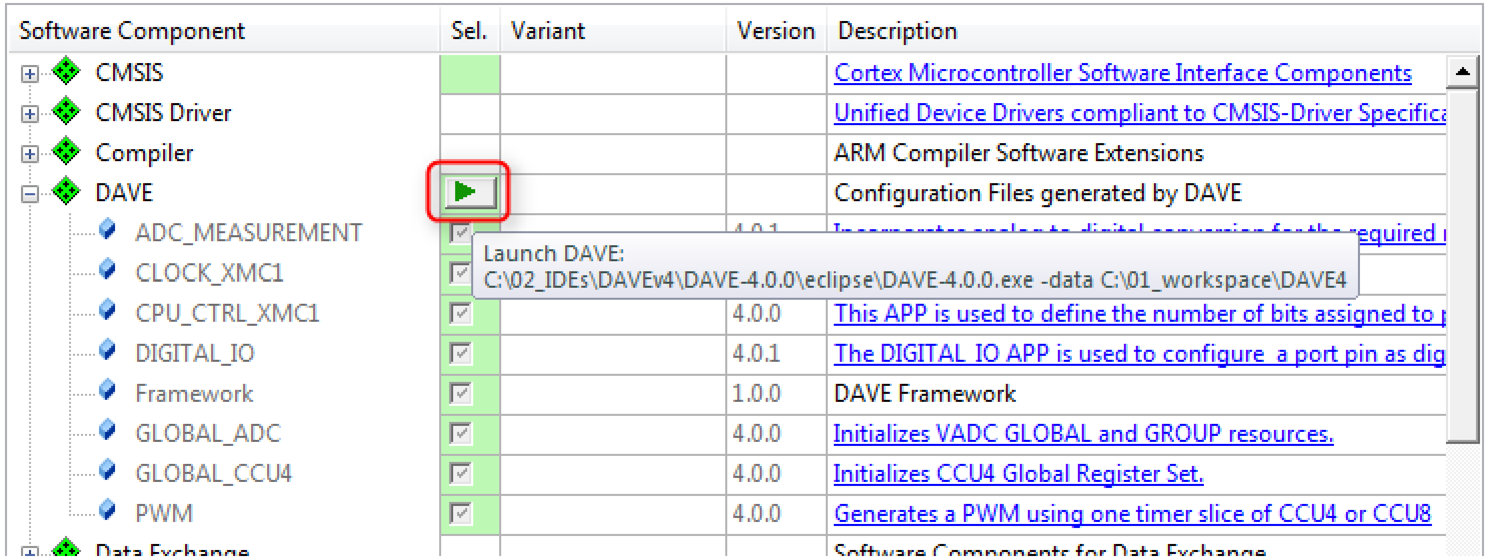


Project

- Project: XMC1200_dev_days_2015_v2
  - XMC1200-T038x0200
    - Source Group 1
    - CMSIS
    - DAVE
      - adc_measurement.c (ADC_MEASUREMENT)
      - adc_measurement_conf.c (ADC_MEASUREMENT)
      - clock_xmc1.c (CLOCK_XMC1)
      - clock_xmc1_conf.c (CLOCK_XMC1)
      - cpu_ctrl_xmc1.c (CPU_CTRL_XMC1)
      - cpu_ctrl_xmc1_conf.c (CPU_CTRL_XMC1)
      - DAVE.c (Framework)
      - digital_io.c (DIGITAL_IO)
      - digital_io_conf.c (DIGITAL_IO)
      - global_adc.c (GLOBAL_ADC)
      - global_adc_conf.c (GLOBAL_ADC)
      - global_ccu4.c (GLOBAL_CCU4)
      - global_ccu4_conf.c (GLOBAL_CCU4)
      - pwm.c (PWM)
      - pwm_conf.c (PWM)
    - Device
      - xmc1_eru.c (XMClib:ERU)
      - xmc1_flash.c (XMClib:FLASH)
      - xmc1_gpio.c (XMClib:GPIO)
      - xmc1_rtc.c (XMClib:RTC)
      - xmc1_scu.c (XMClib:SCU)
      - xmc1_usic.c (XMClib:UART)
      - xmc_acmp.c (XMClib:ACMP)
      - xmc_bccu.c (XMClib:BCCU)
      - xmc_ccu4.c (XMClib:CCU4)
      - xmc_ccu8.c (XMClib:CCU8)
      - xmc_common.c (XMClib:WDT)
      - xmc_eru.c (XMClib:ERU)
      - xmc_gpio.c (XMClib:GPIO)
      - xmc_i2c.c (XMClib:I2C)
      - xmc_ledts.c (XMClib:LEDTS)
      - xmc_math.c (XMClib:MATH)
      - xmc_pau.c (XMClib:PAU)
      - xmc_posif.c (XMClib:POSIF)
      - xmc_prng.c (XMClib:PRNG)
      - xmc_rtc.c (XMClib:RTC)
      - xmc_scu.c (XMClib:SCU)
      - xmc_spi.c (XMClib:SPI)

Going Back to DAVE™

- Changing the DAVE™ project (add/remove Apps or adjust App settings) needs to be done in DAVE™.
- The **Play** button in the Manage Run-Time Environment window invokes DAVE™ with the correct project:

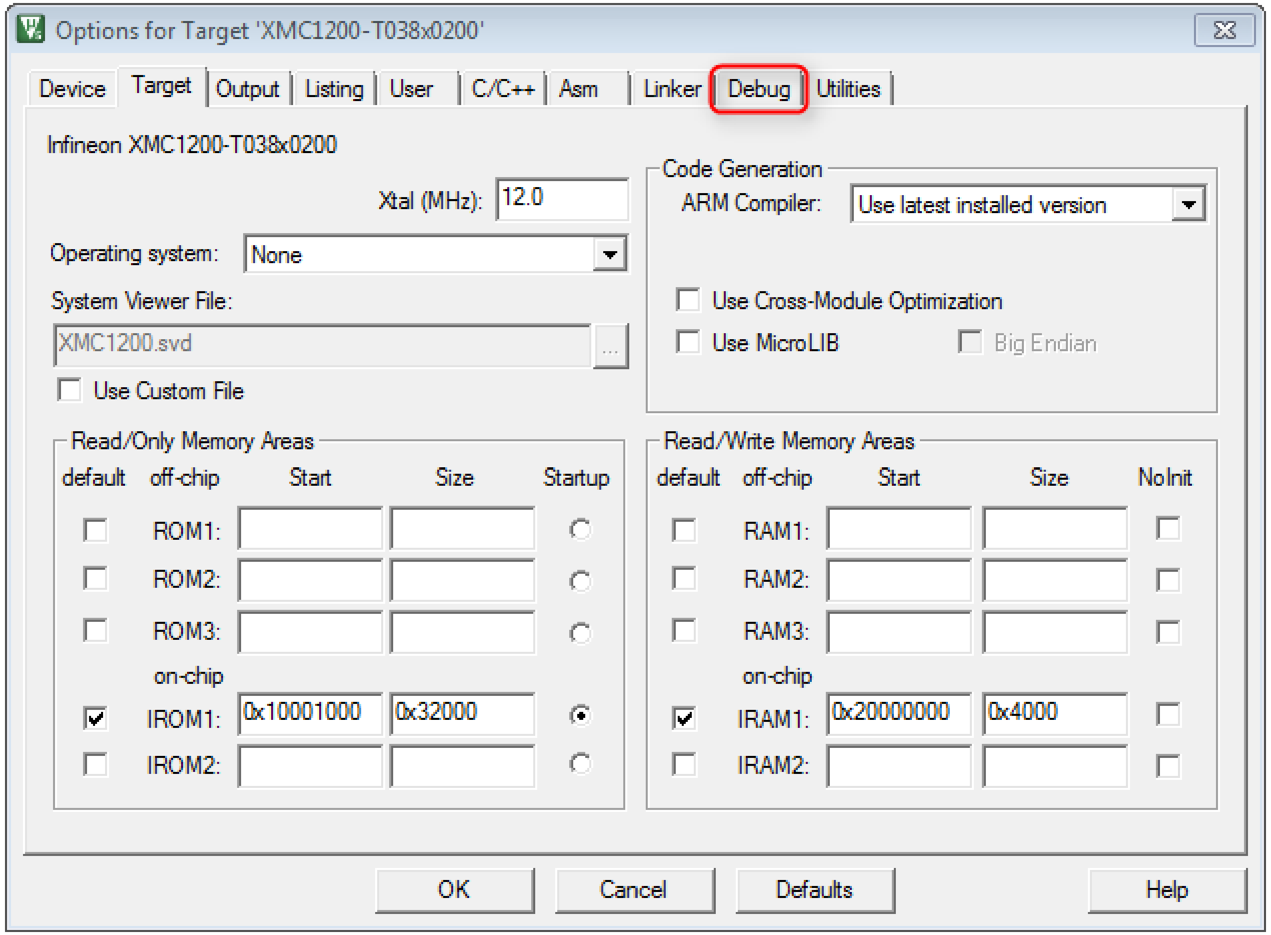


Software Component	Sel.	Variant	Version	Description
CMSIS				Cortex Microcontroller Software Interface Components
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifica
Compiler				ARM Compiler Software Extensions
DAVE	▶			Configuration Files generated by DAVE
ADC_MEASUREMENT	☑		4.0.1	Incorporates analog to digital conversion for the required r
CLOCK_XMC1	☑			
CPU_CTRL_XMC1	☑		4.0.0	This APP is used to define the number of bits assigned to p
DIGITAL_IO	☑		4.0.1	The DIGITAL IO APP is used to configure a port pin as dig
Framework	☑		1.0.0	DAVE Framework
GLOBAL_ADC	☑		4.0.0	Initializes VADC GLOBAL and GROUP resources.
GLOBAL_CCU4	☑		4.0.0	Initializes CCU4 Global Register Set.
PWM	☑		4.0.0	Generates a PWM using one timer slice of CCU4 or CCU8
Data Exchange				Software Components for Data Exchange

Launch DAVE:
C:\02_IDEs\DAVEv4\DAVE-4.0.0\eclipse\DAVE-4.0.0.exe -data C:\01_workspace\DAVE4

Open Target Options Dialog

- Go to **Project → Options for Target 'XMC1200-T038x0200'** (or press ALT+F7) and click on the **Debug** tab:



Infineon XMC1200-T038x0200

Xtal (MHz): 12.0

Operating system: None

System Viewer File: XMC1200.svd

☐ Use Custom File

Code Generation

ARM Compiler: Use latest installed version

☐ Use Cross-Module Optimization

☐ Use MicroLIB ☐ Big Endian

Read/Only Memory Areas

default	off-chip	Start	Size	Startup
☐	ROM1:			○
☐	ROM2:			○
☐	ROM3:			○
	on-chip			
☑	IROM1:	0x10001000	0x32000	●
☐	IROM2:			○

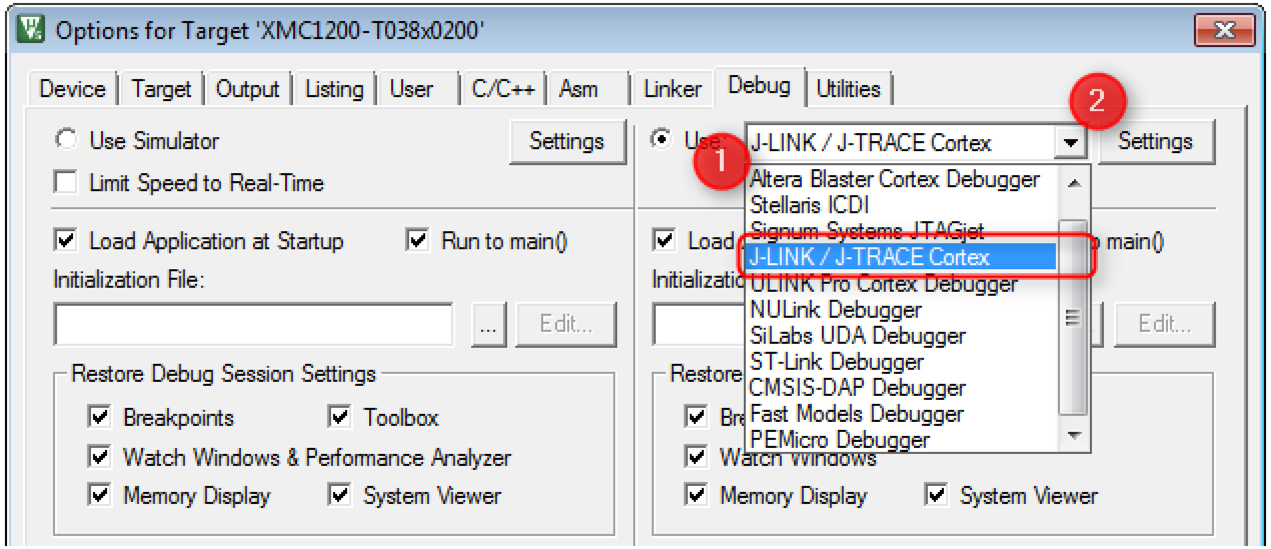
Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
☐	RAM1:			☐
☐	RAM2:			☐
☐	RAM3:			☐
	on-chip			
☑	IRAM1:	0x20000000	0x4000	☐
☐	IRAM2:			☐

Select the J-LINK Debugger

- The XMC1200 Boot Kit has an integrated J-LINK debug adapter

1. To change default adapter, click on **ULINK2/ME Cortex Debugger** and scroll down until **J-LINK/J-TRACE Cortex**
2. Click on **Settings**, to check the connectivity between the target and your PC



Options for Target 'XMC1200-T038x0200'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

○ Use Simulator [Settings]

☐ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

[...] [Edit...]

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows & Performance Analyzer

☑ Memory Display ☑ System Viewer

● Use: J-LINK / J-TRACE Cortex [Settings]

☑ Load ... main()

Initializatio... [Edit...]

Restore ...

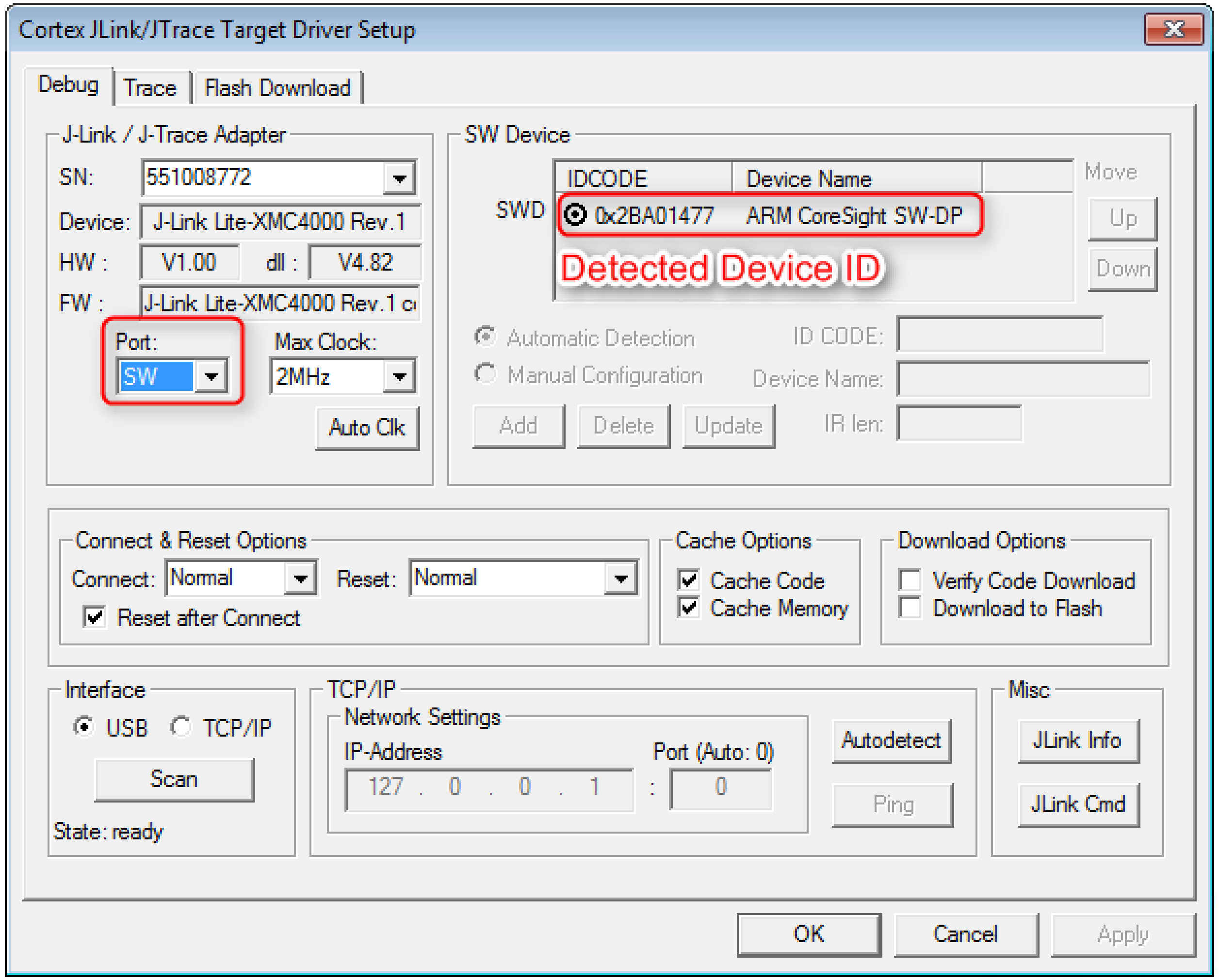
☑ Bre...

☑ Watch Windows

☑ Memory Display ☑ System Viewer

Enable the SW Port

- µVision will try to connect to the J-LINK using a JTAG port. This is not available on the Relax Kit
- Click on **Port:** JTAG and set to **SW**
- The connected device will be detected automatically
- Click **OK** twice



Cortex JLink/JTrace Target Driver Setup

J-Link / J-Trace Adapter

SN: 551008772

Device: J-Link Lite-XMC4000 Rev.1

HW : V1.00 dll : V4.82

FW : J-Link Lite-XMC4000 Rev.1 c

Port: SW

Max Clock: 2MHz

SW Device

	IDCODE	Device Name
SWD	● 0x2BA01477	ARM CoreSight SW-DP

Detected Device ID

○ Automatic Detection

○ Manual Configuration

ID CODE:

Device Name:

IR len:

Connect & Reset Options

Connect: Normal

Reset: Normal

☑ Reset after Connect

Cache Options

☑ Cache Code

☑ Cache Memory

Download Options

□ Verify Code Download

□ Download to Flash

Interface

● USB ○ TCP/IP

State: ready

TCP/IP

Network Settings

IP-Address: 127 . 0 . 0 . 1

Port (Auto: 0): 0

Build the Project and Download to Target

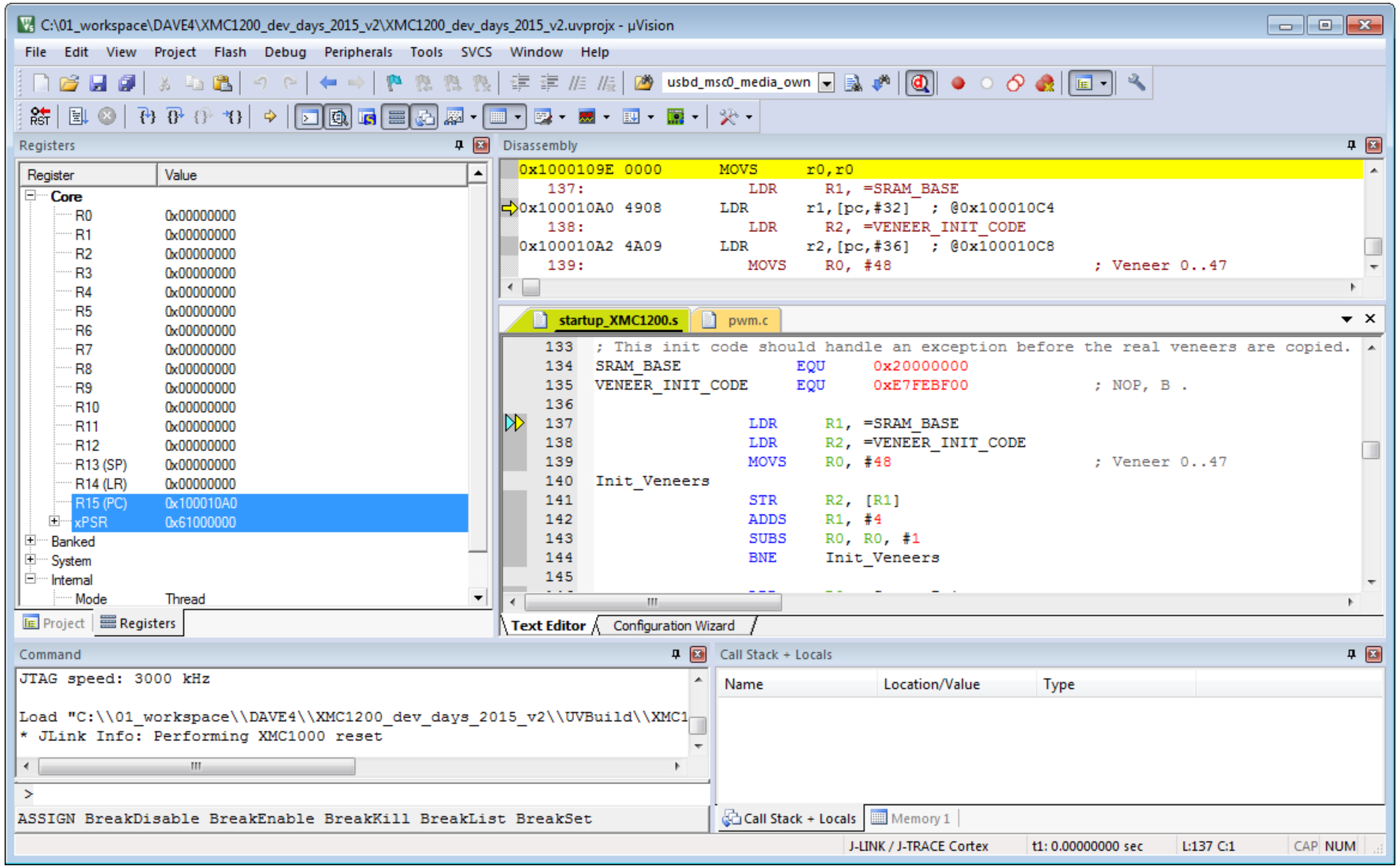
- Go to **Project → Build** (or press F7) to start the build process

- You will see a #111-D warning which can be safely ignored.

- Go to **Flash → Download** (or click on LOAD) to download the program into the target's Flash memory

Start a Debug Session

- Go to **Debug → Start/Stop Debug Session** (or press CTRL+F5) to switch to the µVision debugger.
- During the start of the debug session, µVision loads the application, executes startup code, and stops at the main C function
- Click **Run** 📄 on the toolbar. The LED connected to P00 will start flashing. Play with the on-board potentiometer to change the flashing frequency



Register	Value
Core	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x100010A0
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread

```
0x1000109E 0000      MOVS     r0,r0
   137:                 LDR      R1, =SRAM_BASE
0x100010A0 4908      LDR      r1,[pc,#32]  ; @0x100010C4
   138:                 LDR      R2, =VENEER_INIT_CODE
0x100010A2 4A09      LDR      r2,[pc,#36]  ; @0x100010C8
   139:                 MOVS     R0, #48                        ; Veneer 0..47
```

```
; This init code should handle an exception before the real veneers are copied.
SRAM_BASE           EQU     0x20000000
VENEER_INIT_CODE    EQU     0xE7FEBF00                 ; NOP, B .

                    LDR     R1, =SRAM_BASE
                    LDR     R2, =VENEER_INIT_CODE
                    MOVS    R0, #48                    ; Veneer 0..47
Init_Veneers
                    STR     R2, [R1]
                    ADDS    R1, #4
                    SUBS    R0, R0, #1
                    BNE     Init_Veneers

```

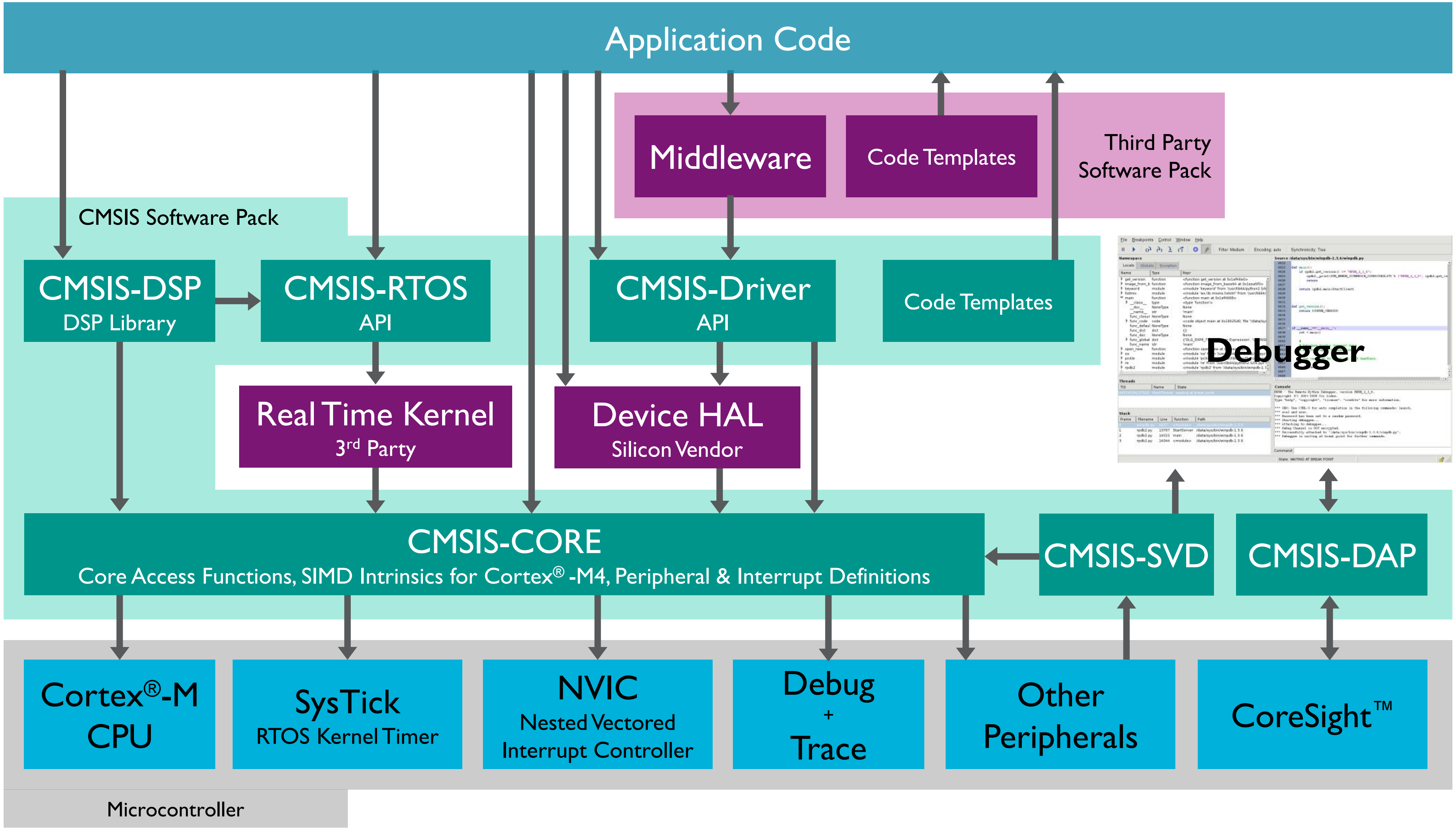
```
JTAG speed: 3000 kHz

Load "C:\\01_workspace\\DAVE4\\XMC1200_dev_days_2015_v2\\UVBuild\\XMC1
* JLink Info: Performing XMC1000 reset
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

Name	Location/Value	Type

J-LINK / J-TRACE Cortex　t1: 0.00000000 sec　L:137 C:1　CAP NUM

Cortex® Microcontroller Software Interface Standard (CMSIS)

CMSIS Version 4



Application Code

Third Party Software Pack

Middleware

Code Templates

CMSIS Software Pack

CMSIS-DSP
DSP Library

CMSIS-RTOS
API

CMSIS-Driver
API

Code Templates

Real Time Kernel
3rd Party

Device HAL
Silicon Vendor

Debugger

CMSIS-CORE
Core Access Functions, SIMD Intrinsics for Cortex® -M4, Peripheral & Interrupt Definitions

CMSIS-SVD

CMSIS-DAP

Cortex®-M CPU

SysTick
RTOS Kernel Timer

NVIC
Nested Vectored Interrupt Controller

Debug + Trace

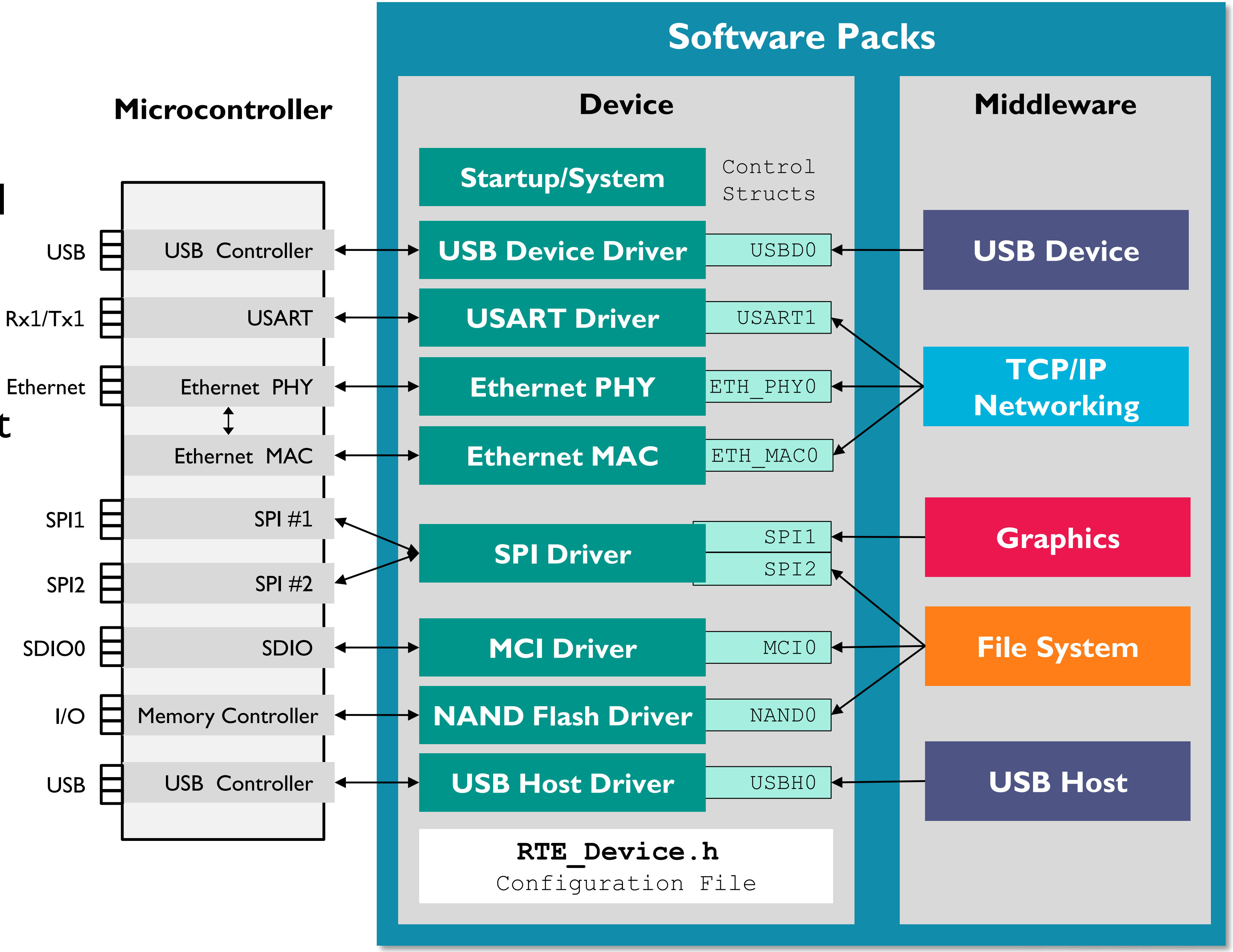
Other Peripherals

CoreSight™

Microcontroller

CMSIS-Driver 2.0

- API describing peripheral driver interfaces for middleware stacks and user applications
- Generic and independent of a specific RTOS
- Covers a wide range of use cases for the supported peripheral types

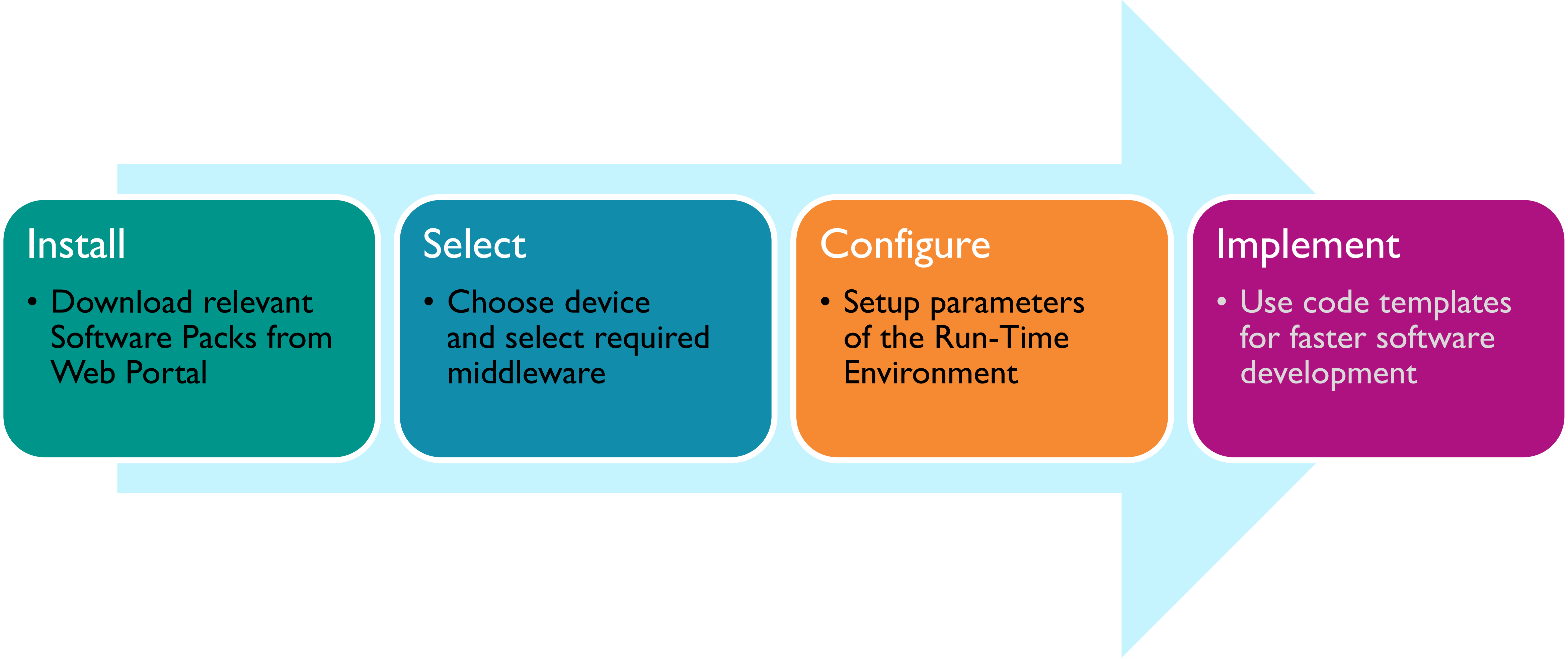


CMSIS-Pack: Use Cases

Note that a Software Pack can address multiple use cases

Variant	Device Family Pack	CMSIS Pack	Middleware Pack	Board Support Pack	In-house Software Pack
Source	Silicon Vendor, Tool Vendor	ARM	Silicon Vendor, Tool Vendor, 3rd Party	Board Vendor	Tool User
Use Case	Deploy support for new MCU families	Standard delivery of CMSIS components	Simplify integration of pre-build middleware	Support of evaluation boards with interfaces and example projects	Supply and update software components within a company

Keil MDK Workflow using Software Packs



Install

- Download relevant Software Packs from Web Portal

Select

- Choose device and select required middleware

Configure

- Setup parameters of the Run-Time Environment

Implement

- Use code templates for faster software development

Advantages of Software Components in Keil MDK

Enhanced Productivity

- Convenient selection of software components
- Easy access to documentation
- Code templates and examples to kick-start development

Long-term Project Maintenance

- Software Packs with update facility and version management
- Simplifies the replacement of the target device

Improved Flexibility

- Support for new devices is provided by Software Packs
- Open standard allows adding third party components

Creating a USB Data Logger Application using Middleware and CMSIS-Driver
XMC4500 Relax Kit Hands-On

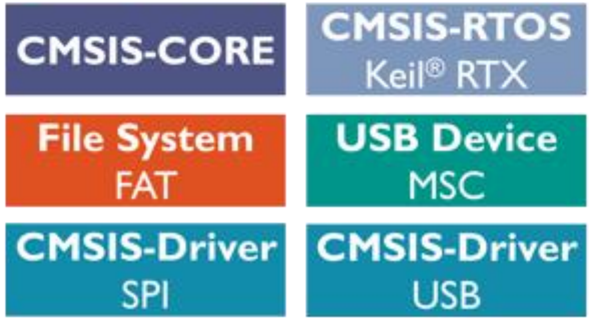
Objective: USB Data Logger

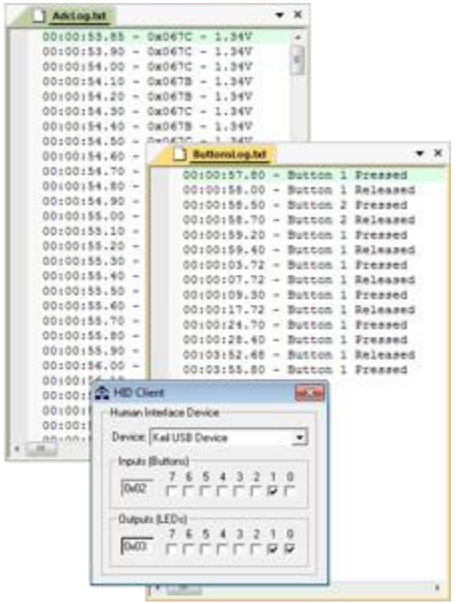
- Record analog and digital data on a MicroSD card

- Read the data on a PC

- These middleware components are used:
    - USB for communication with the PC
    - File System for data storage on the MicroSD card

- For the USB communication with the PC, a USB CMSIS-Driver is required, whereas the MicroSD card is connected to the application via an SPI CMSIS-Driver.


CMSIS-CORE
CMSIS-RTOS Keil® RTX
File System FAT
USB Device MSC
CMSIS-Driver SPI
CMSIS-Driver USB



```
00:00:53.85 - 0x067C - 1.34V
00:00:53.90 - 0x067C - 1.34V
00:00:54.00 - 0x067C - 1.34V
00:00:54.10 - 0x067B - 1.34V
00:00:54.20 - 0x067B - 1.34V
00:00:54.30 - 0x067C - 1.34V
00:00:54.40 - 0x067B - 1.34V
00:00:54.50 - 0x067C - 1.34V
00:00:54.60 -
00:00:54.70 -
00:00:54.80 -
00:00:54.90 -
00:00:55.00 -
00:00:55.10 -
00:00:55.20 -
00:00:55.30 -
00:00:55.40 -
00:00:55.50 -
00:00:55.60 -
00:00:55.70 -
00:00:55.80 -
00:00:55.90 -
00:00:56.00 -
```

```
00:00:57.80 - Button 1 Pressed
00:00:58.00 - Button 1 Released
00:00:58.50 - Button 2 Pressed
00:00:58.70 - Button 2 Released
00:00:59.20 - Button 1 Pressed
00:00:59.40 - Button 1 Released
00:00:03.72 - Button 1 Pressed
00:00:07.72 - Button 1 Released
00:00:09.30 - Button 1 Pressed
00:00:17.72 - Button 1 Released
00:00:24.70 - Button 1 Pressed
00:00:28.40 - Button 1 Pressed
00:03:52.48 - Button 1 Released
00:03:55.80 - Button 1 Pressed
```

Device: Keil USB Device

Inputs (Buttons): 0x02 — 7 □ 6 □ 5 □ 4 □ 3 □ 2 □ 1 ☑ 0 □

Outputs (LEDs): 0x03 — 7 □ 6 □ 5 □ 4 □ 3 □ 2 □ 1 ☑ 0 ☑

Hands-On: Application Note 273

- Application note 273 provides a step-by-step instruction on how to create the project

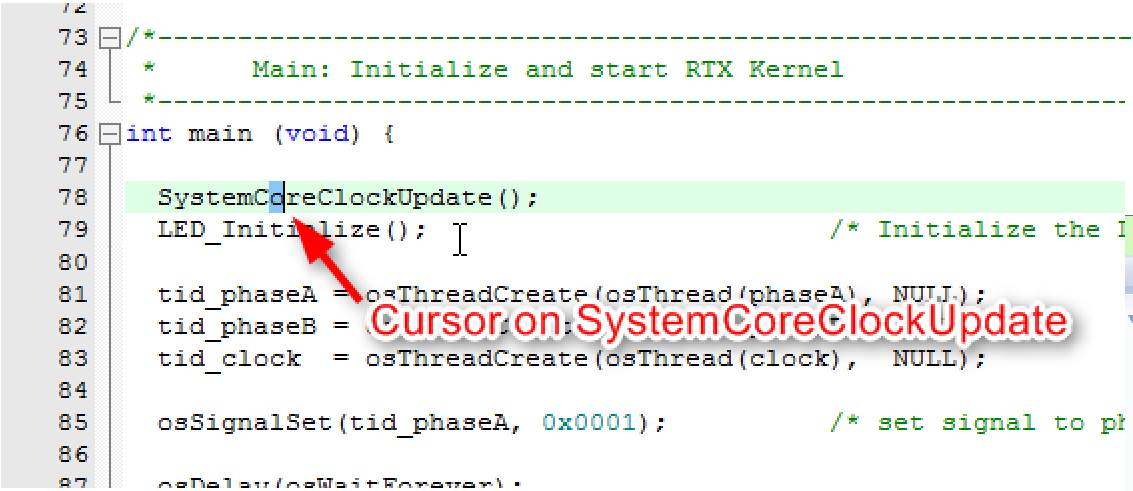
- If you are lost at any point, the apnt_273.zip file contains separate µVision projects for every step.

- You will find AN273 on the USB stick in the **Collaterals** folder or online at www.keil.com/appnotes/docs/apnt_273.asp

µVision® IDE Features

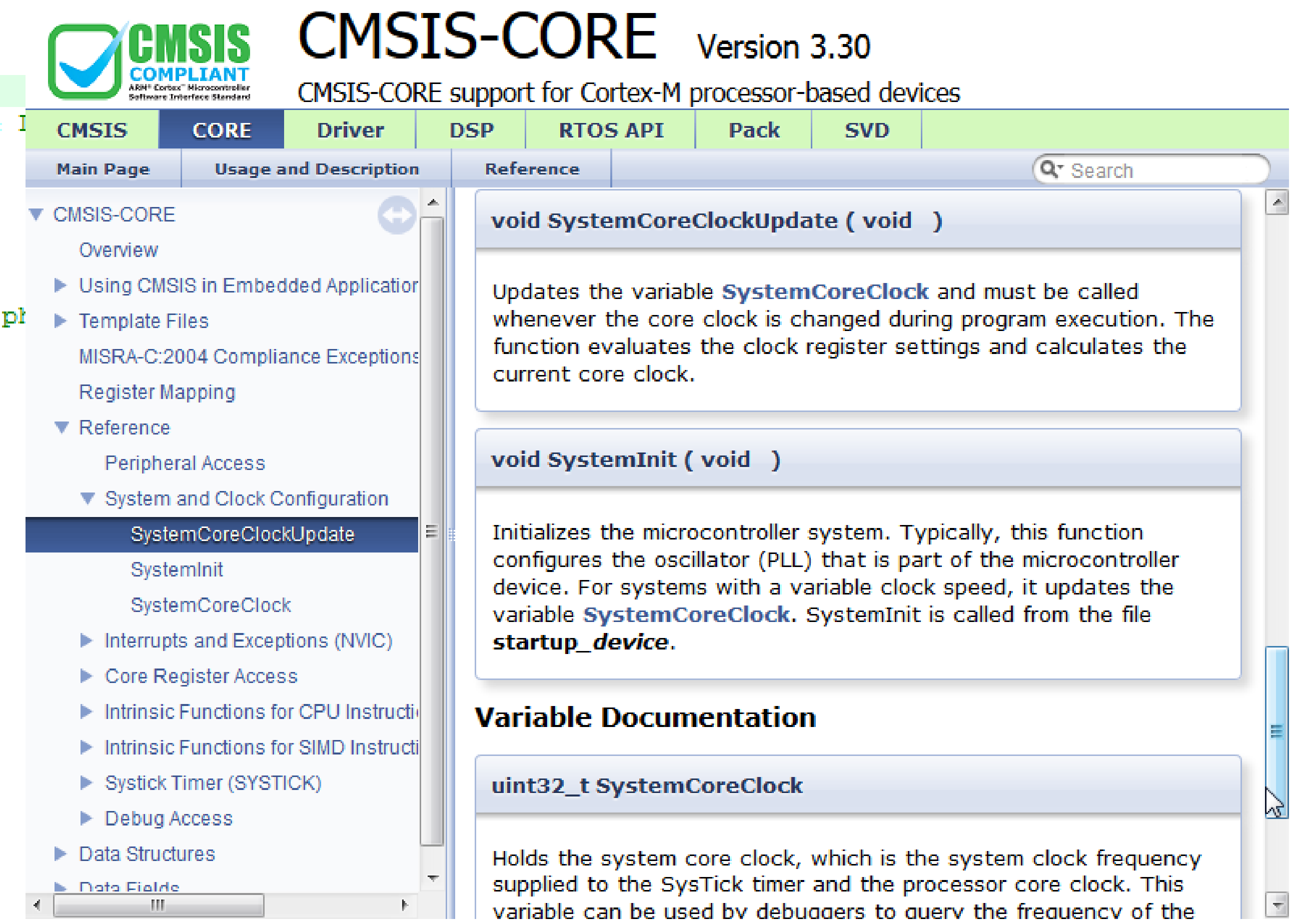
Context Sensitive F1 Help

- Pressing F1 with cursor on any keyword will bring up the related help page:



```
/*----------------------------------------------------------------------------
 *      Main: Initialize and start RTX Kernel
 *---------------------------------------------------------------------------
int main (void) {

  SystemCoreClockUpdate();
  LED_Initialize();                        /* Initialize the I

  tid_phaseA = osThreadCreate(osThread(phaseA), NULL);
  tid_phaseB =
  tid_clock  = osThreadCreate(osThread(clock),  NULL);

  osSignalSet(tid_phaseA, 0x0001);         /* set signal to ph

  osDelay(osWaitForever);
```

Cursor on SystemCoreClockUpdate



CMSIS-CORE Version 3.30

CMSIS-CORE support for Cortex-M processor-based devices

void SystemCoreClockUpdate (void)

Updates the variable **SystemCoreClock** and must be called whenever the core clock is changed during program execution. The function evaluates the clock register settings and calculates the current core clock.

void SystemInit (void)

Initializes the microcontroller system. Typically, this function configures the oscillator (PLL) that is part of the microcontroller device. For systems with a variable clock speed, it updates the variable **SystemCoreClock**. SystemInit is called from the file **startup_*device***.

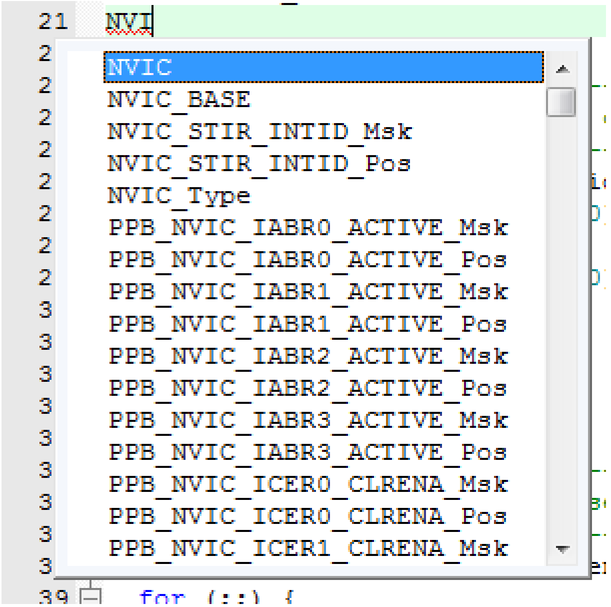
Variable Documentation

uint32_t SystemCoreClock

Holds the system core clock, which is the system clock frequency supplied to the SysTick timer and the processor core clock. This variable can be used by debuggers to query the frequency of the

Code Completion

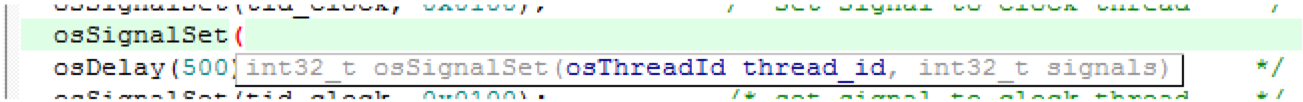
- List showing all program symbols that contain the currently typed characters
- List appears after typing:
  - 3 characters (default)
  - A **trigger character**:
    - . For structures or classes
    - -> For pointer structures
    - :: For symbols within a specific scope
- **CTRL+<space>**
- Insert the highlighted list-item into the code by pressing:
  - Tab, space, or enter
  - Typing a bracket
  - Any trigger character



```
NVI
NVIC
NVIC_BASE
NVIC_STIR_INTID_Msk
NVIC_STIR_INTID_Pos
NVIC_Type
PPB_NVIC_IABR0_ACTIVE_Msk
PPB_NVIC_IABR0_ACTIVE_Pos
PPB_NVIC_IABR1_ACTIVE_Msk
PPB_NVIC_IABR1_ACTIVE_Pos
PPB_NVIC_IABR2_ACTIVE_Msk
PPB_NVIC_IABR2_ACTIVE_Pos
PPB_NVIC_IABR3_ACTIVE_Msk
PPB_NVIC_IABR3_ACTIVE_Pos
PPB_NVIC_ICER0_CLRENA_Msk
PPB_NVIC_ICER0_CLRENA_Pos
PPB_NVIC_ICER1_CLRENA_Msk
  for (;;) {
```

Parameter Information

- For a function or method Parameter Information shows:
  - Parameter names
  - Amount of parameters
  - Parameter types

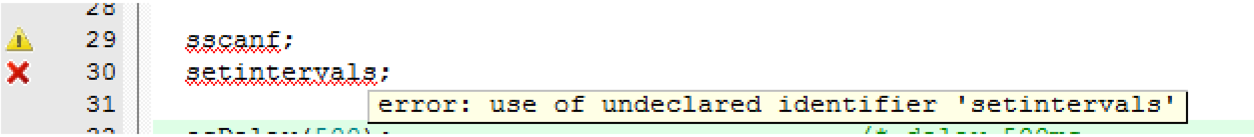


```
osSignalSet(tid_clock, 0x0100);          /* set signal to clock thread     */
osSignalSet(
osDelay(500) int32_t osSignalSet(osThreadId thread_id, int32_t signals)      */
osSignalSet(tid_clock, 0x0100);          /* set signal to clock thread     */
```

- While typing, the bolded text indicates the next required parameter

Dynamic Syntax Checking while Typing

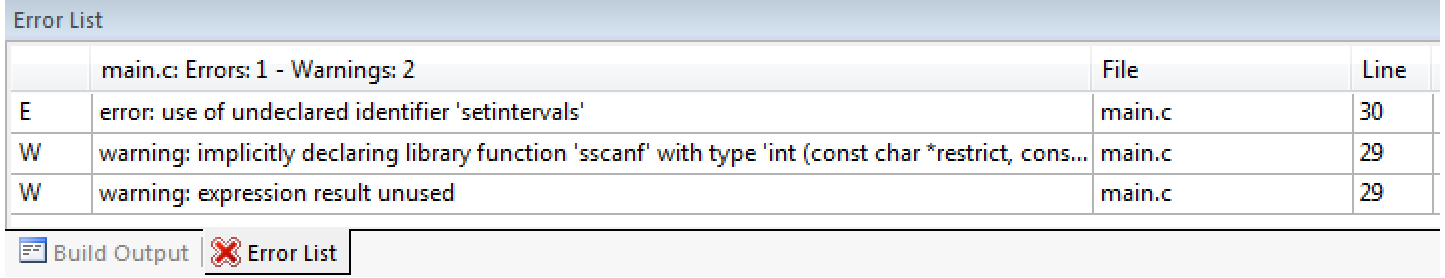
- Validates program syntax
  - Alerts to potential code violations before compilation
- Errors/warnings shown by:
  - Squiggly red lines in the editor
  - Icons next to the line number



```
sscanf;
setintervals;
```

error: use of undeclared identifier 'setintervals'

- Hover the mouse on an icon for details about the syntax violation
- Also in:
  - View → Error List Window

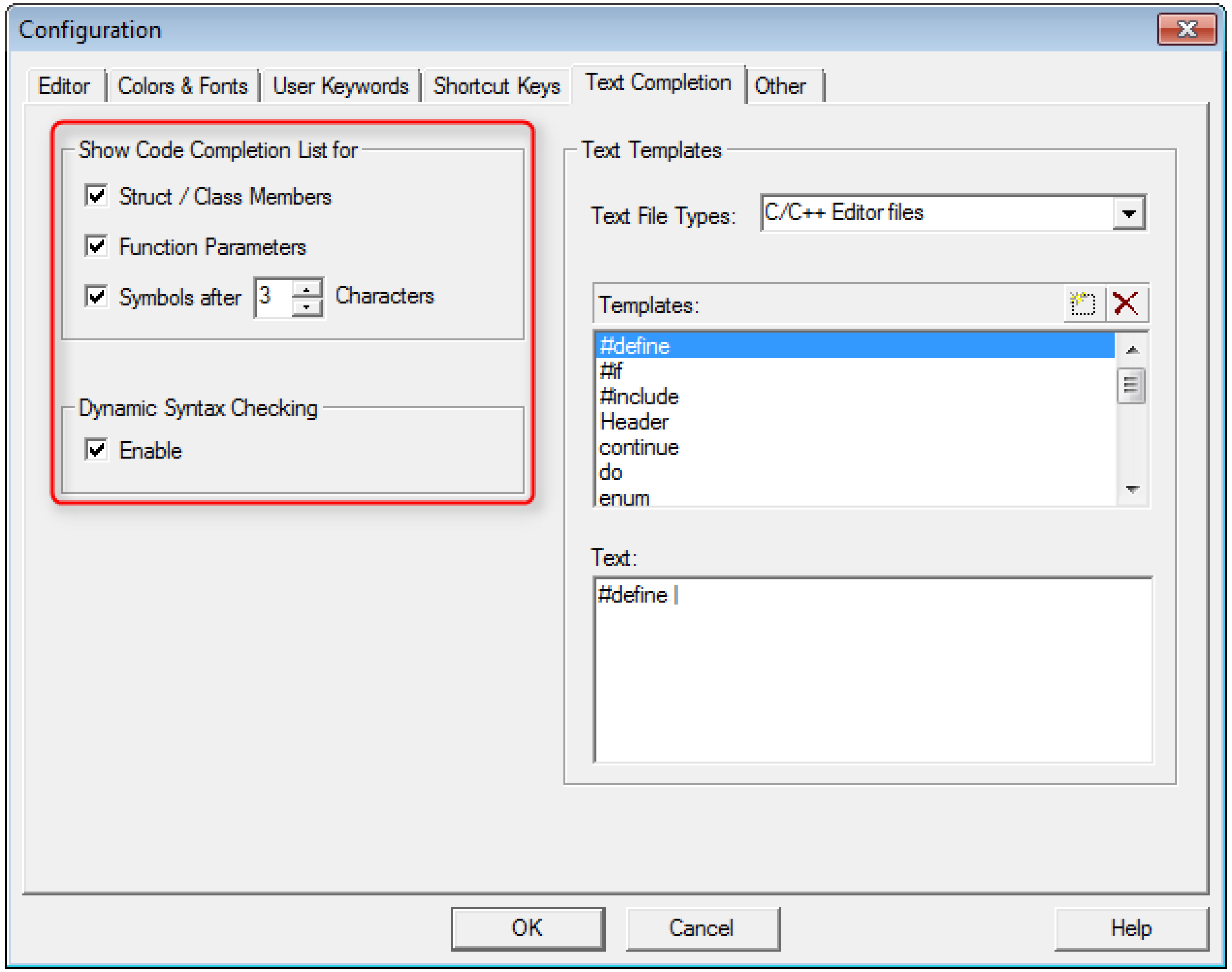


Error List

	main.c: Errors: 1 - Warnings: 2	File	Line
E	error: use of undeclared identifier 'setintervals'	main.c	30
W	warning: implicitly declaring library function 'sscanf' with type 'int (const char *restrict, cons...	main.c	29
W	warning: expression result unused	main.c	29

Feature Configuration

- Go to **Edit → Configuration**, click on the **Text Completion** tab



Show Code Completion List for
☑ Struct / Class Members
☑ Function Parameters
☑ Symbols after 3 Characters

Dynamic Syntax Checking
☑ Enable

Text Templates
Text File Types: C/C++ Editor files

Templates:
#define
#if
#include
Header
continue
do
enum

Text:
```
#define |
```

Collaterals on USB Stick

- AN258 - Using DAVE3 with MDK Version 5 (applies to DAVE 4 as well)
- AN260 - Infineon XMC1100 2Go Lab for uVision V5
- AN263 - Infineon XMC1200 Boot Kit Lab for uVision V5
- AN273 - Creating a USB Data Logger Application using Middleware and CMSIS
- Getting Started – Create Applications with MDK Version 5
- µVision Keyboard Shortcuts

Links on USB Stick

▪ MDK Version 5 Overview:	keil.com/mdk5
▪ MDK Middleware:	keil.com/mdk5/middleware
▪ Infineon on keil.com:	keil.com/infineon
▪ Keil MDK for Infineon XMC1000:	keil.com/infineon/mdk
▪ Cortex-M Learning Platform:	keil.com/learn
▪ CMSIS:	www.arm.com/cmsis and keil.com/cmsis
▪ CMSIS Pack Tutorial:	keil.com/cmsis/pack
▪ Support:	keil.com/support
▪ Forums:	keil.com/forum
▪ Application Notes:	keil.com/appnotes
▪ Keil Tools on ARM Connected Community:	cc.arm.com/groups/tools