

XMC1100 Boot Kit

Getting Started



Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

- Boot Modes

5

- DAVE™

6

Getting Started

7

- Example – Blinky based on XMC Lib

8

- Example – Blinky based on DAVE™ APPs

Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

– Boot Modes

5

– DAVE™

6

Getting Started

7

– Example – Blinky based on XMC Lib

8

– Example – Blinky based on DAVE™ APPs

Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

– Boot Modes

5

– DAVE™

6

Getting Started

7

– Example – Blinky based on XMC Lib

8

– Example – Blinky based on DAVE™ APPs

Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

– Boot Modes

5

– DAVE™

6

Getting Started

7

– Example – Blinky based on XMC Lib

8

– Example – Blinky based on DAVE™ APPs

Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

– Boot Modes

5

– DAVE™

6

Getting Started

7

– Example – Blinky based on XMC Lib

8

– Example – Blinky based on DAVE™ APPs

Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

– Boot Modes

5

– **DAVE™**

6

Getting Started

7

– Example – Blinky based on XMC Lib

8

– Example – Blinky based on DAVE™ APPs

Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

– Boot Modes

5

– DAVE™

6

Getting Started

7

– Example – Blinky based on XMC Lib

8

– Example – Blinky based on DAVE™ APPs

Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

– Boot Modes

5

– DAVE™

6

Getting Started

7

– Example – Blinky based on XMC Lib

8

– Example – Blinky based on DAVE™ APPs

Agenda (1/2)

1

Kit Overview

2

Hardware Overview

3

Tooling Overview

4

– Boot Modes

5

– DAVE™

6

Getting Started

7

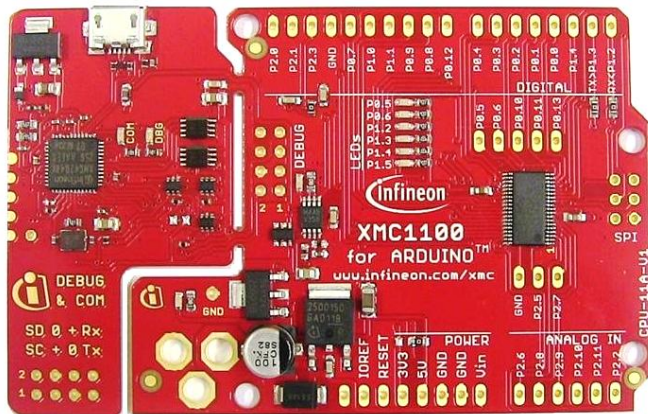
– Example – Blinky based on XMC Lib

8

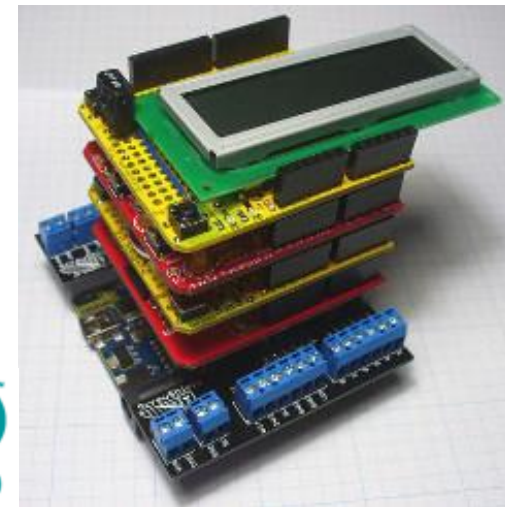
– Example – Blinky based on DAVE™ APPs

Kit Overview (1/2)

- › XMC1100 Boot Kit
 - Consists of an XMC1100 CPU Card
 - Compatible with Arduino™ Shields (<http://shieldlist.org>)



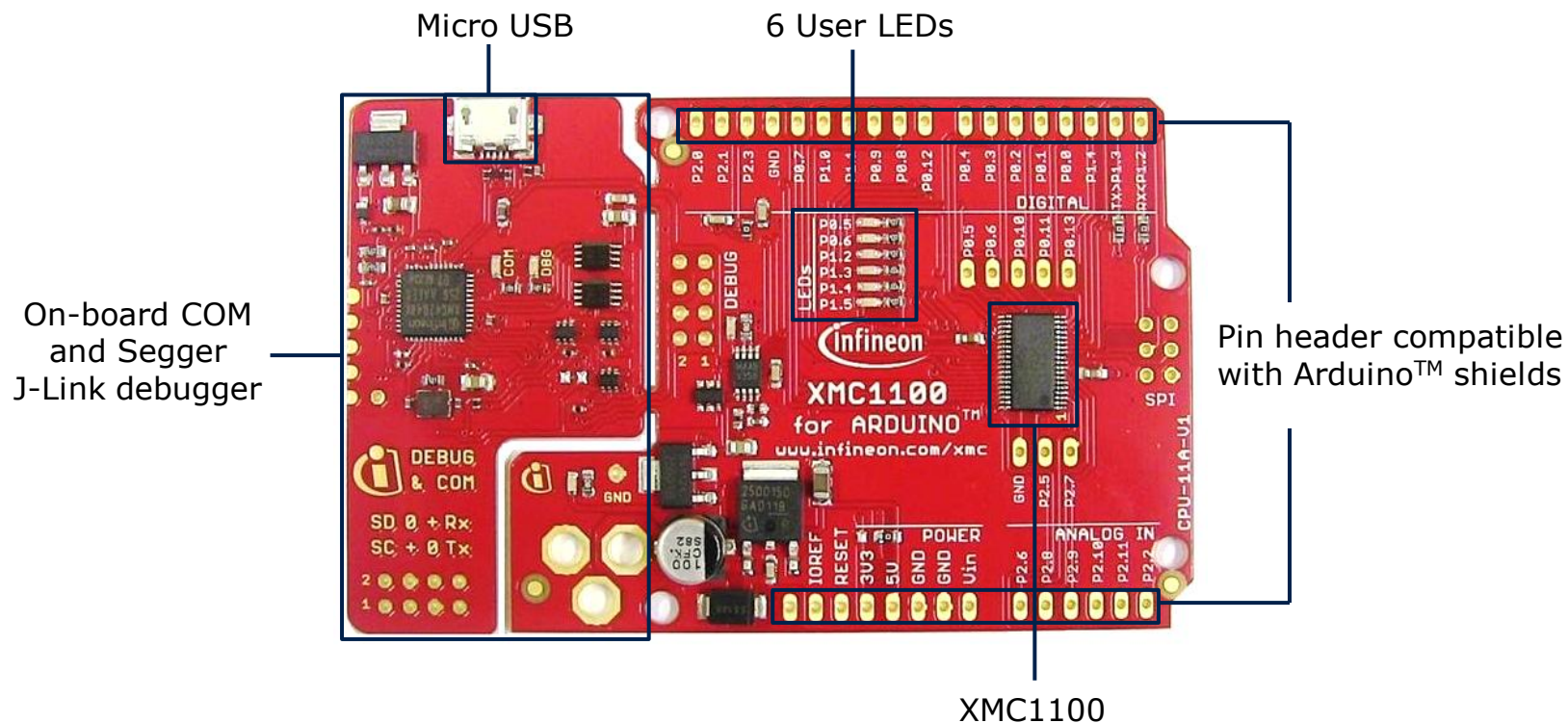
XMC1100 CPU Card



Arduino™ Shield

Kit Overview (2/2)

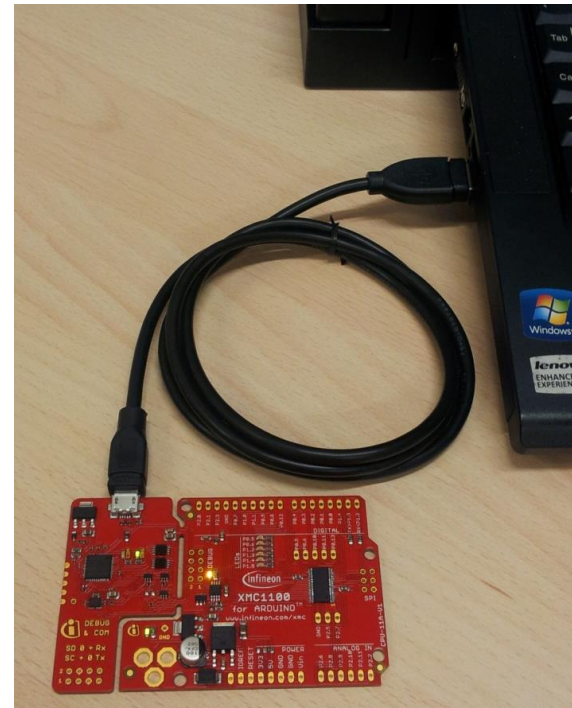
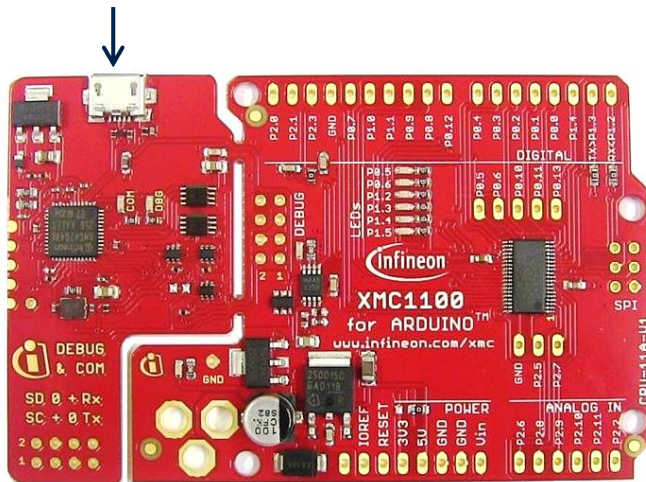
› XMC1100 CPU Card for Arduino™



Hardware Overview

- › Connect XMC1100 CPU Card to PC via USB cable
- › CPU Card is powered up (as indicated by LED on the card)

CPU Card powered
via USB cable



- › Note: Supported Application Card may be additionally connected to the CPU card

Tooling Overview – Boot Modes



- › Boot Modes available
 - UART Bootstrap-Loader Mode
 - User Mode (Halt After Reset)
 - User Mode (Debug) **Default Mode of device on Boot Kit**
 - User Mode (Productive)

- › Boot Modes can be configured via:
 - DAVE™
 - Download DAVE™
[DAVE™ v4.1.2 download](#)
 - MemTool
 - Download MemTool
[MemTool v4.65.exe download](#)

- › For more information on how to configure the BMI value, please refer to the XMC1000 Tooling Guide.



- 



First Name*	
Last Name*	
Email Address*	
Country*	--- Please select ---
Company*	
Business Phone	
Target Application	---please select---

Tooling Overview – DAVE™ (2/5)

› After registration, download and unzip the installer package

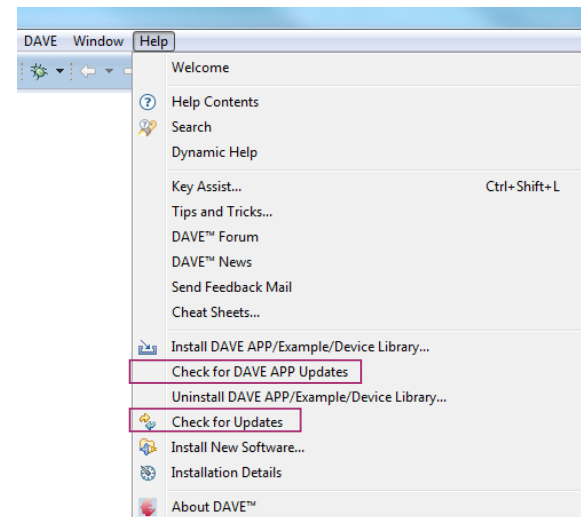
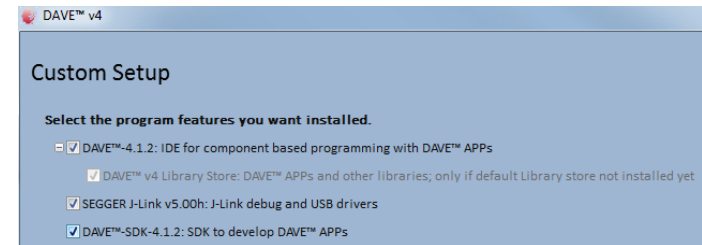
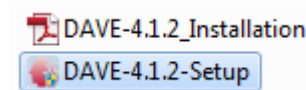
› Run DAVE-4.1.2-Setup.exe to install
DAVE™ IDE and SEGGER J-Link drivers

› Open DAVE™



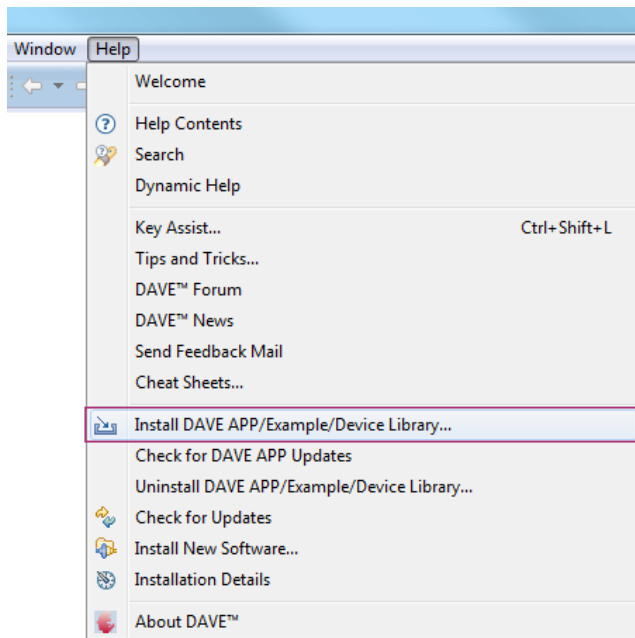
› Update DAVE™ and DAVE™ libraries

- Help → Check for Updates
- Help → Check for DAVE APP Updates



Tooling Overview – DAVE™ (3/5)

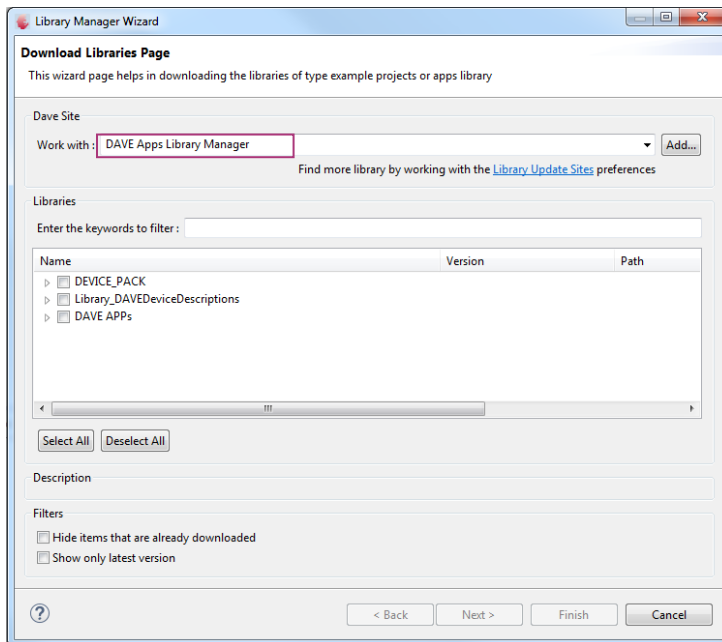
- › Install DAVE™ APPs libraries and Device Description
 - Help → Install DAVE APP/Example/Device Library



- › Note: You may skip the above step if you are not using DAVE™ APPs

Tooling Overview – DAVE™ (4/5)

- › Select DAVE Apps Library Manager in the drop-down menu

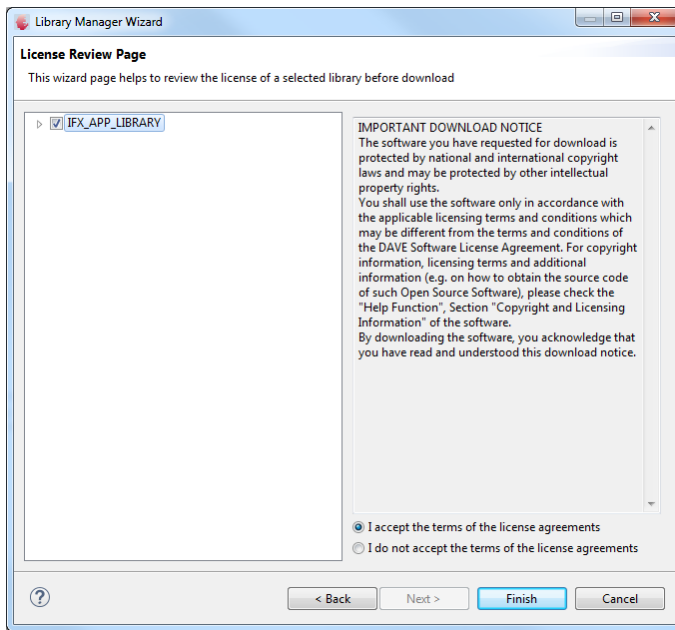


- › Select DEVICE_PACK, Library_DAVEDeviceDescriptions (XMC1100 Device) and DAVE APPs

- ▷ ☒ DEVICE_PACK
- ▷ ☒ Library_DAVEDeviceDescriptions
- ▷ ☒ DAVE APPs

Tooling Overview – DAVE™ (5/5)

- › Accept terms of the license agreements and click Finish



- › DAVE™ APPs libraries and Device Description are installed

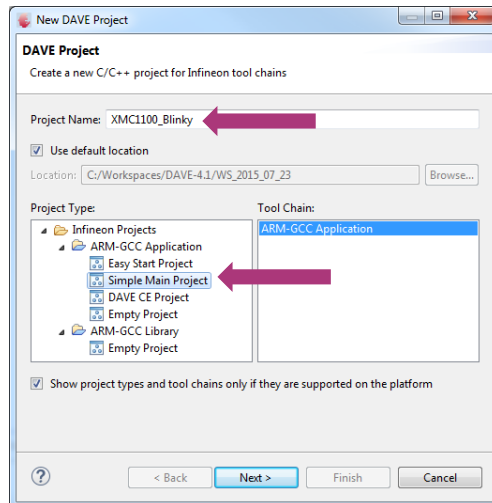
Getting Started – Example – Blinky based on XMC Lib (1/6)

1. Open DAVE™

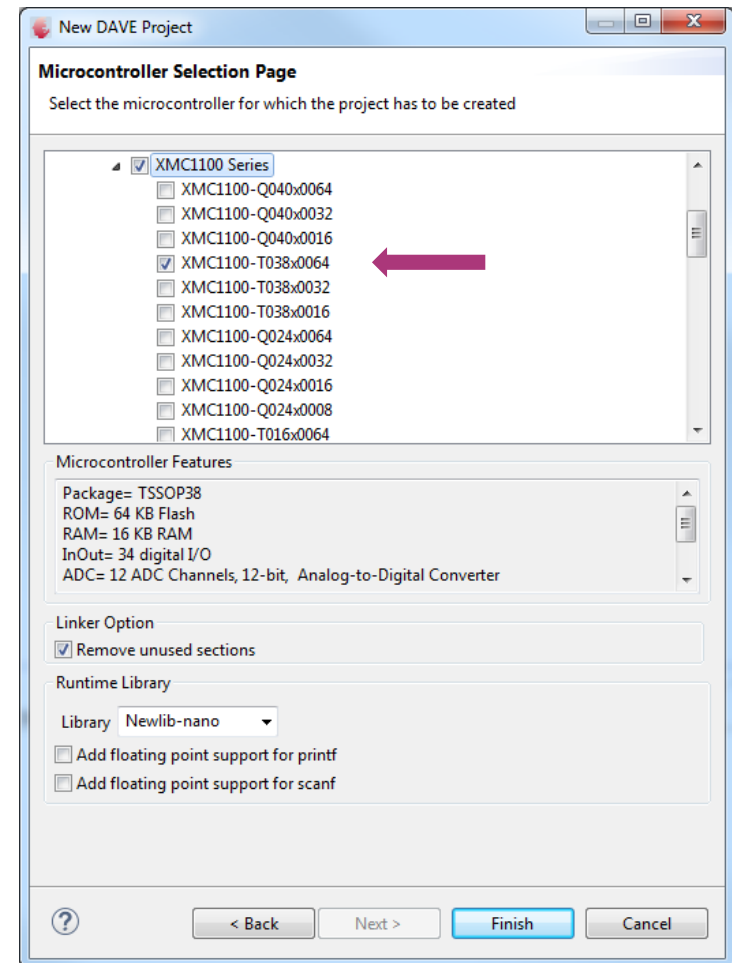


2. Create a new "Simple Main" project:

- File → New → DAVE Project
- Enter project name e.g. "XMC1100_Blinky"
- Select "Simple Main Project" as Project Type



3. Select the device accordingly



Getting Started – Example – Blinky based on XMC Lib (2/6)



- › For this project, we will use
 - System clock frequency of 8MHz
 - LED on Port pin 0.5
 - System timer, SysTick, as the time base for interrupt

- › Next, we will show you how to
 1. Set up the System or Main Clock (MCLK)
 2. Configure Port pin
 3. Configure SysTick and define its exception service routine

Getting Started – Example – Blinky based on XMC Lib (3/6)

1. Set up System or Main Clock (MCLK)

- Include the header files required for MCLK and GPIO configuration

```
#include "xmc_gpio.h"
#include "xmc_scu.h"
```

- MCLK configured via **IDIV** and **FDIV** bit fields in XMC_SCU_CLOCK_CONFIG data structure

```
XMC_SCU_CLOCK_CONFIG_t clock_config =
{
    .pclk_src = XMC_SCU_CLOCK_PCLKSRC_DOUBLE_MCLK, /*PCLK = 2*MCLK*/
    .rtc_src = XMC_SCU_CLOCK_RTCCLKSRC_DCO2,
    .fdiv = 0,  /**< Fractional divider */
    .idiv = 4,  /**MCLK = 8MHz */
};
```

- Initializes clock generators and clock tree in **Main.c**

```
int main(void)
{
    /* Ensure clock frequency is set at 8MHz (MCLK) */
    XMC_SCU_CLOCK_Init(&clock_config);
}
```

2. Configure Port pin

- GPIO to toggle the LED is configured via **mode** and **output_level** of XMC_GPIO_CONFIG structure.

```
XMC_GPIO_CONFIG_t gpio_output_config =  
{  
    .mode           = XMC_GPIO_MODE_OUTPUT_PUSH_PULL,  
    .output_level    = XMC_GPIO_OUTPUT_LEVEL_HIGH,  
};
```

- Initializes port pin P0.5 as general purpose output pin in **Main.c**

```
/* Initialise P0.0 as an output pin */  
XMC_GPIO_Init(LED1, &gpio_output_config);
```

Getting Started – Example – Blinky based on XMC Lib (5/6)

3. Configure SysTick and define its exception service routine

- SysTick exception handler is defined in **startup_XMC1100.s**

```
/* ===== */
.globl SysTick_Veneer
SysTick_Veneer:
    LDR R0, =SysTick_Handler
    MOV PC,R0
/* ===== */
```

- Initialize the SysTick in **Main.c**

```
/* System timer configuration */
SysTick_Config(SystemCoreClock / TICKS_PER_SECOND);
```

- Define the SysTick exception handler routine in Main.c

```
void SysTick_Handler(void)
{
    static uint32_t ticks = 0;

    ticks++;
    if (ticks == TICKS_WAIT)
    {
        XMC_GPIO_ToggleOutput(LED1);
        ticks = 0;
    }
}
```

Getting Started – Example – Blinky based on XMC Lib (6/6)

› Build project

1. Click 

2. Wait for Build to finish

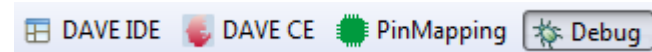
```
'Invoking: ARM-GCC Print Size'  
"C:\DAVEv4\DAVE-4.1.2\eclipse\ARM-GCC-49/bin/arm-none-eabi-  
text    data    bss     dec     hex filename  
2204      20    1044    3268     cc4 XMC1100_Blinky.elf  
'Finished building: XMC1100_Blinky.siz'
```

› Download code

1. Click 

2. Switch to Debug perspective

3. Click  to run code



› LED blinks every 0.2s



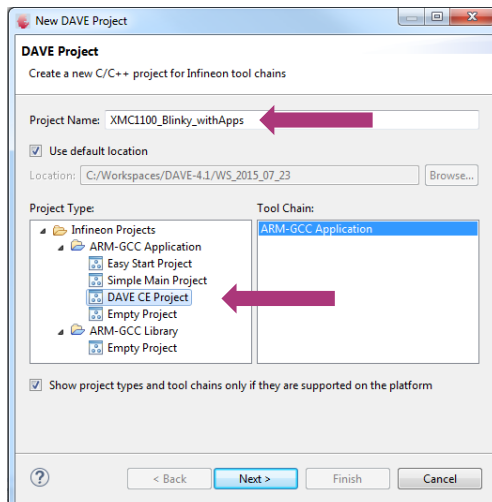
Getting Started – Example – Blinky based on DAVE™ APPs (1/7)

1. Open DAVE™

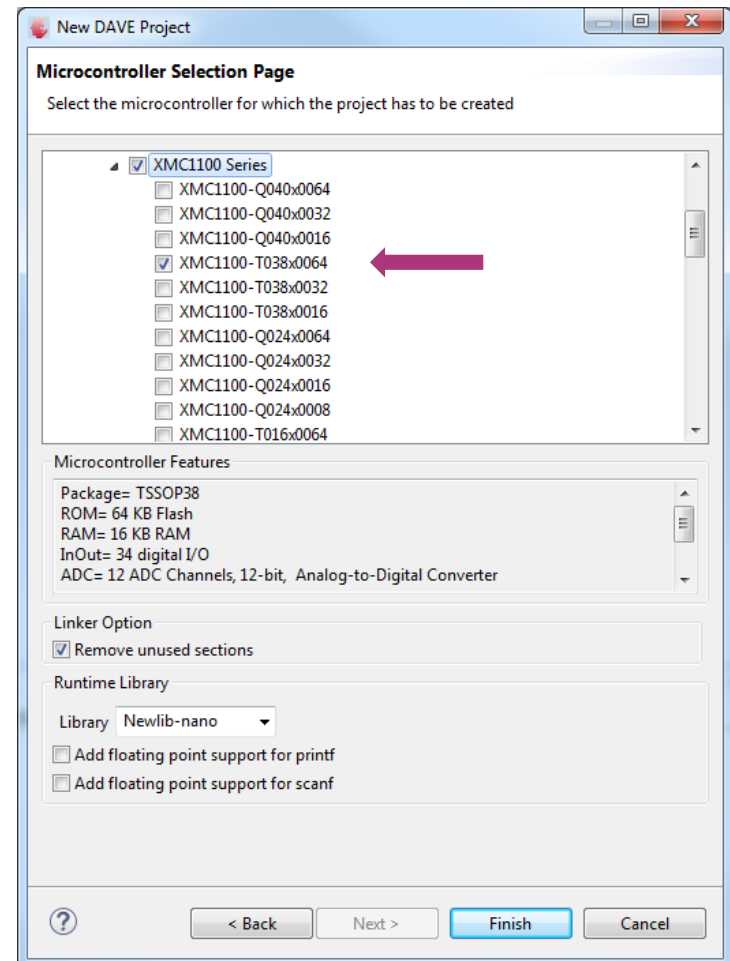


2. Create a new "DAVE CE" project:

- File → New → DAVE Project
- Enter project name e.g. "XMC1100_Blinky_withApps"
- Select "DAVE CE Project" as Project Type



3. Select the device accordingly



Getting Started – Example – Blinky based on DAVETM APPs (2/7)



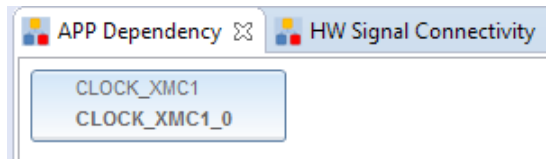
- › For this project, we will use
 - System clock frequency of 8MHz
 - LED on Port pin 0.5
 - System timer as the time base for interrupt
 - Time base of 0.2s

- › Next, we will show you how to
 1. Set up the System or Main Clock (MCLK)
 2. Configure Port pin
 3. Configure System Timer and define its exception service routine

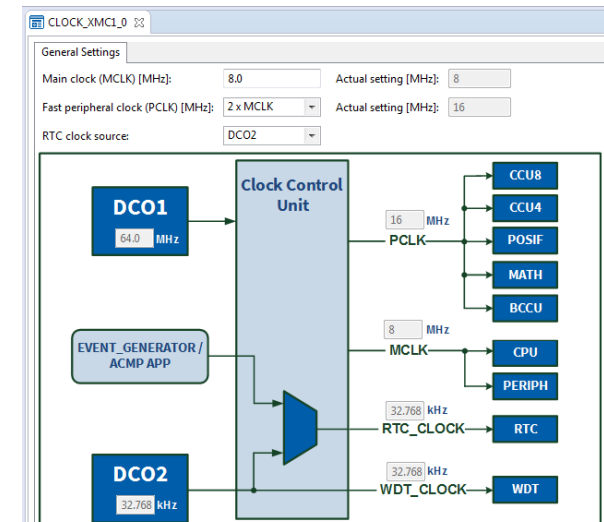
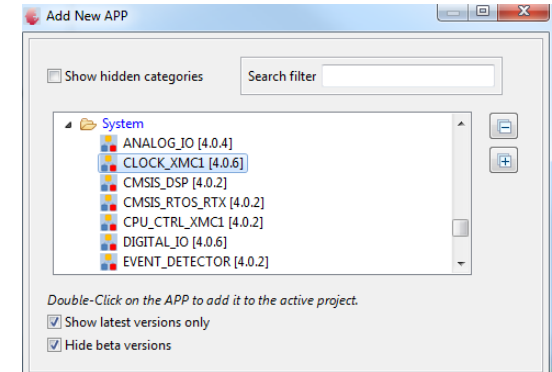
Getting Started – Example – Blinky based on DAVETM APPs (3/7)

1. Set up System or Main Clock (MCLK)

- Click  to add new APP
- Double-click **CLOCK_XMC1** APP and close window
- Open APP configuration editor
 - In **APP Dependency** view, double-click **CLOCK_XMC1**



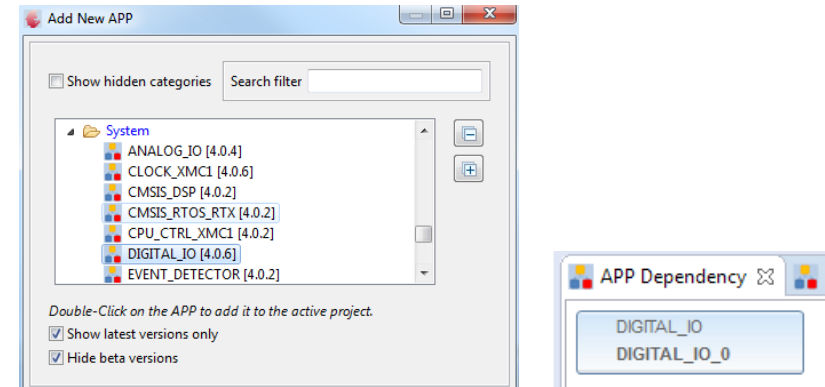
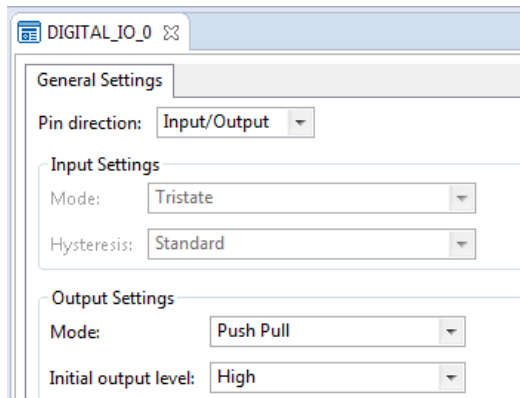
- Configure APP instance
 - In APP configuration window, set **Main clock (MCLK)** to 8MHz



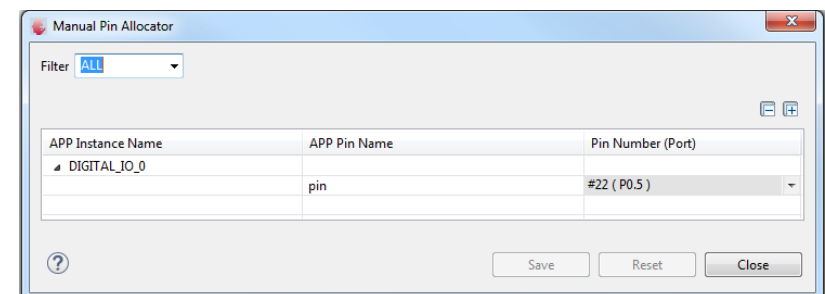
Getting Started – Example – Blinky based on DAVETM APPs (4/7)

2. Configure Port pin

- Click  to add new APP
- Double-click **DIGITAL_IO** APP and close window
- Open APP configuration editor
 - In **APP Dependency** view, double-click DIGITAL_IO
- Configure APP instance
 - In APP configuration window, set **Pin direction** to Input/Output and set **Initial output level** to High



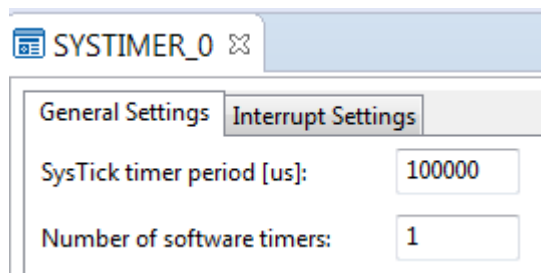
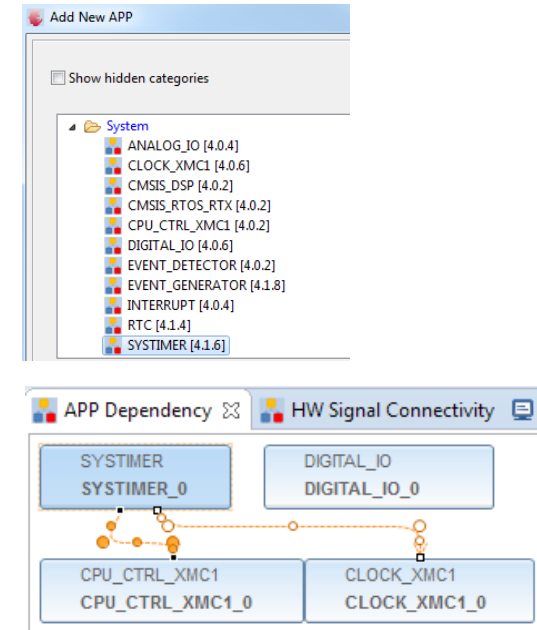
- Assign pin to P0.5
 - Click  to open **Manual Pin Allocator**
 - Set Pin Number (Port) to **#22 (P0.5)**
 - **Solve and Save**



Getting Started – Example – Blinky based on DAVETM APPs (5/7)

3. Configure System Timer and define its exception service routine

- Click  to add new APP
- Double-click **SYSTIMER** APP and close window
- Open APP configuration editor
 - In **APP Dependency** view, double-click SYSTIMER
- Configure APP instance
 - In APP configuration window, under **General Settings** tab, set **System timer tick interval** to 100000us (0.1s)



Getting Started – Example – Blinky based on DAVETM APPs (6/7)



- Create software timer using SYSTIMER Apps.

```
TimerId = (uint32_t)SYSTIMER_CreateTimer(Point2SEC,SYSTIMER_MODE_PERIODIC,(void*)LED_Toggle_EveryPoint2Sec,NULL);  
if (TimerId != 0U)  
{  
    //timer is created successfully, now start/run software timer  
    status = SYSTIMER_StartTimer(TimerId);  
}
```

- Define exception handler routine in **Main.c**
 - Define the toggle interval (in usec)

```
#define Point2SEC 200000U  
void LED_Toggle_EveryPoint2Sec(void)  
{  
    DIGITAL_IO_ToggleOutput(&DIGITAL_IO_0); //toggles : 1 -> 0 (if initial output level is logic 1)  
}
```

Getting Started – Example – Blinky based on DAVETM APPs (7/7)

› Generate code

1. Click 

```
'Invoking: ARM-GCC Print Size'
"C:\DAVEv4\DAVE-4.1.2\eclipse\ARM-GCC-49\bin/arm-none-eabi-size" --
    text    data    bss    dec    hex filename
   3908      20   1100   5028   13a4 XMC1100_Blinky_withApps.elf
'Finished building: XMC1100_Blinky_withApps.siz'
```

› Build project

1. Click 
2. Wait for Build to finish

› Download code

1. Click 
2. Switch to Debug perspective
3. Click  to run code

› LED blinks every 0.2s



General Information (1/2)

- › Where to buy kit:
 - <http://ehitex.com/starter-kits/for-xmc1000>
 - Order Number: KIT_XMC11_BOOT_001

- › Infineon parts utilized on kit:

Infineon Parts	Order Number
XMC1100 Microcontroller	XMC1100-T038F0064
XMC4200 Microcontroller	XMC4200-Q48F256
5V regulator	IFX25001TFV50
3V3 regulator	IFX25001MEV33

General Information (2/2)

- › Kit documentation:
 - [Boot Kit XMC1100](#)

- Video Series: XMC1000 Boot Kit Getting Started
 - [Introduction](#)
 - [DAVE™ Setup](#)
 - [Boot Mode Index Configuration via DAVE or MemTool](#)
 - [XMC1100 Hardware Setup](#)
 - [Simple Blinky Example](#)
 - [Blinky Example based on DAVE™ Apps](#)
 - [Example Projects Download](#)

References – Where to find XMC Lib documentation?

1. Go to DAVE™ Version 4 website

4. Click on **Modules** or **Files**

<http://www.infineon.com/dave/v4>

2. Download XMC Lib and unzip file

XMC™ Lib

Ready to use APIs for peripherals which are tested for GNU-, ARM-, IAR- und TASKING- compiler, and released for Altium, ARM/KEIL, Atollic, IAR Systems und Rowley compiler IDEs.
Low level driver libraries for XMC™ peripherals (APIs), CMSIS / MISRA 2004 compliant including documentation. XMC™ Lib – [Release Note](#)

System:	Timer/PWM:	Analog-mixed Signal:	Communication:	Application specific:	Examples:
- DMA - ERU - FCE - FLASH - GPIO - MATH - PAU - PRNG	- CCU4 - CCU8 - HRPWM - POSIF	- ACMP - ADC - DAC - DSD	- CAN - I2C - SPI - UART - USB - USIC - Ethernet	- BCCU - LEDTS - MATH - POSIF - HRPWM	Examples for all peripherals drivers and ARM, GCC, IAR, and Tasking

3. Open XMC1_Peripheral_Library

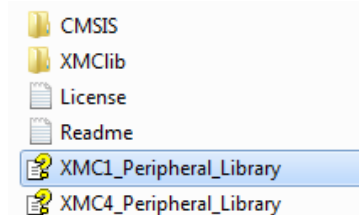
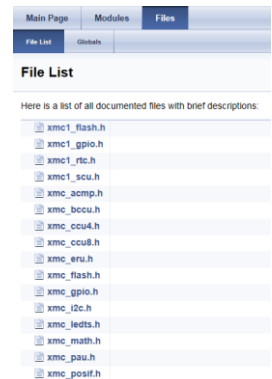
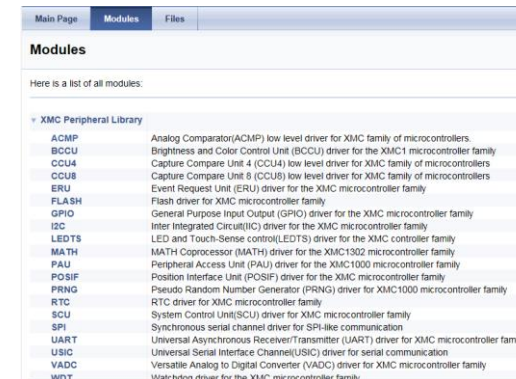


Table of Contents

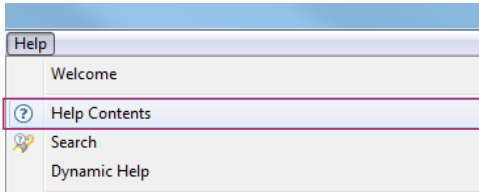
- Supported devices and toolchains
- Overview
- Coding Rules and Conventions
- How to use the XMC Peripheral Library
 - Device Names
- Directories and Files
 - XMC Lib examples
 - Keil MDK-ARM
 - IAR Embedded Workbench for ARM
 - DAVE
- MISRA-C 2004 Compliance Exceptions
- Test conditions
- XMC Peripheral Library Licensing

The XMC Peripheral Library (XMC Lib) consists of low-level drivers for the XMC product family peripherals. Built on top the Cortex Microcontroller Software Interface Standard (CMSIS) and MISRA-C 2004 compliant, it provides access to all peripheral features.

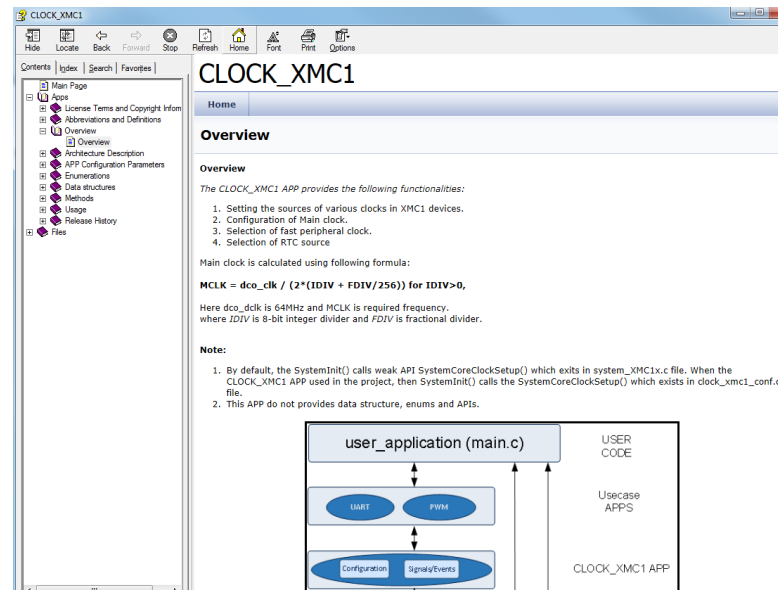
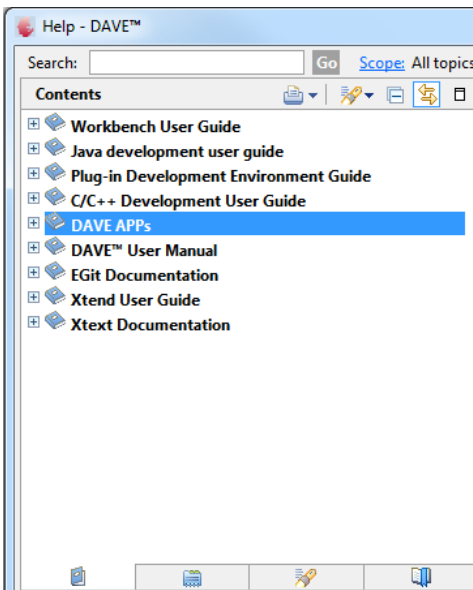


References – Where to find DAVE™ APP documentation?

1. In DAVE™, go to Help → Help Contents



2. Expand DAVE Apps → Click on **CLOCK_XMC1** → Overview



References – Where to download DAVE™ Example Projects?



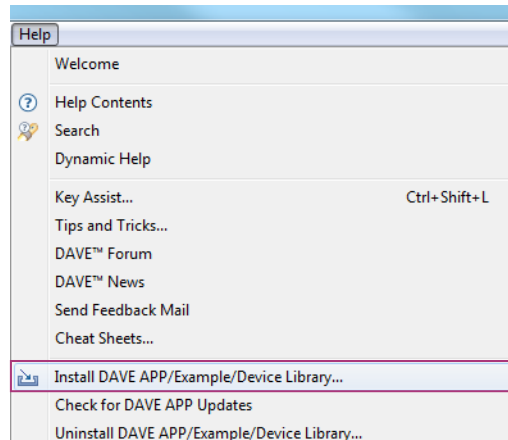
1. Example Project library within DAVE™
2. DAVE™ website
3. Example from XMC Lib package

References –

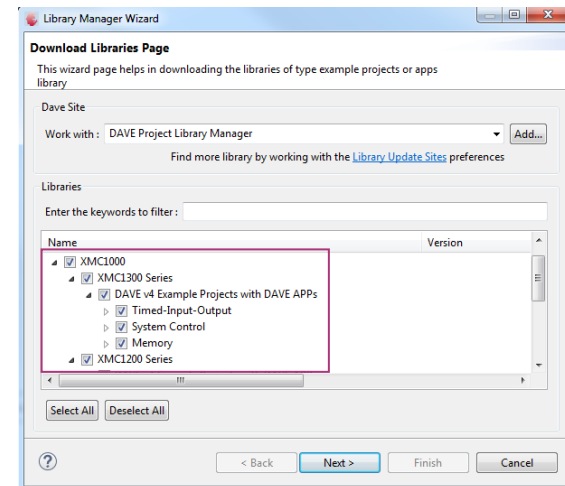
How to load Example Project in DAVE™? (1/4)

› Example Project library within DAVE™

1. Help → Install DAVE APP/Example/Device Library

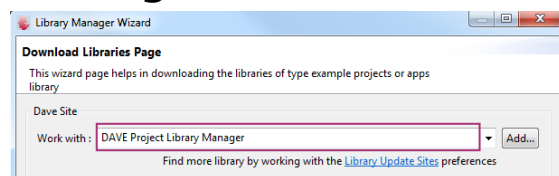


3. Select Examples in the Libraries window → Click Next



4. Accept terms of the license agreements → Click Finish

2. Select DAVE Project Library Manager



References – How to load Example Project in DAVE™? (2/4)

› DAVE™ website

1. Go to DAVE™ Version 4 website

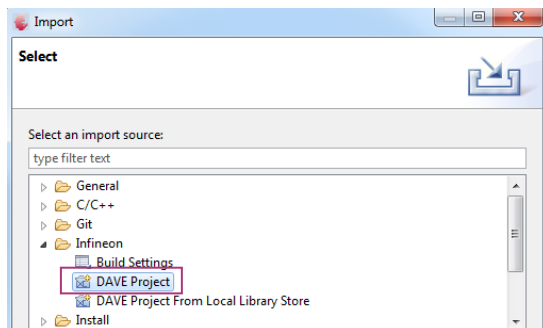
<http://www.infineon.com/dave/v4>

2. Download DAVE™ EXAMPLES

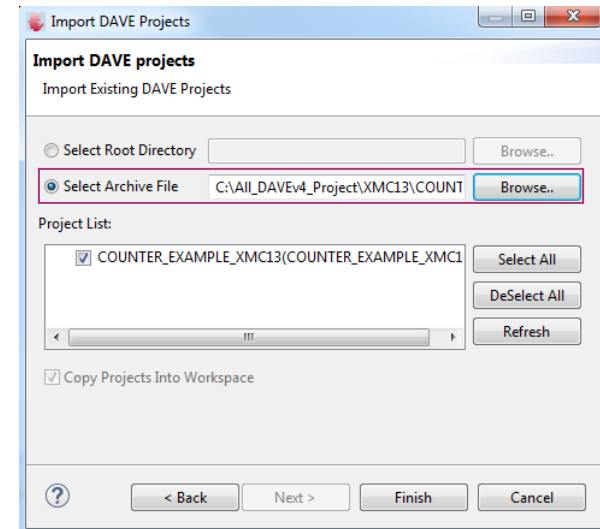


3. In DAVE™, go to File → Import

4. Select DAVE Project → Next



5. Select Archive File → Browse to downloaded project zip file



6. Click Finish

References –

How to load Example Project in DAVE™? (3/4)

› Example from XMC Lib package

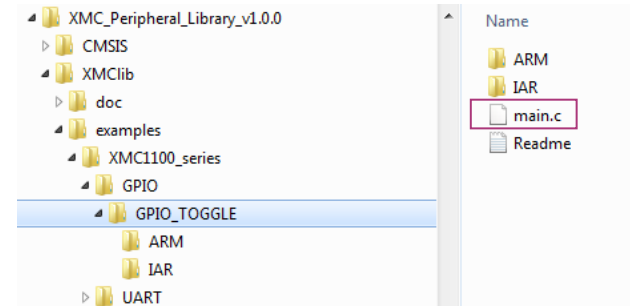
1. Go to DAVE™ Version 4 website

<http://www.infineon.com/dave/v4>

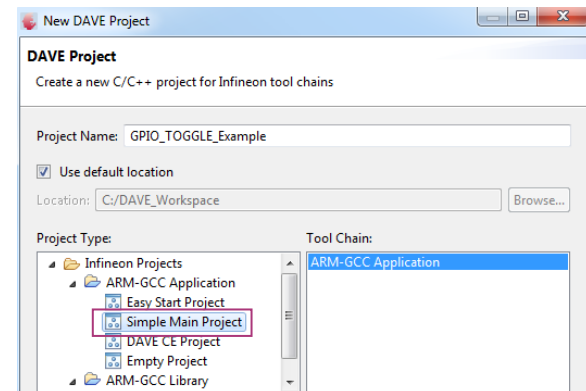
2. Download XMC Lib and unzip file

XMC Lib					
Low level driver libraries for XMC peripherals (APIs), CMSIS / MISRA 2004 compliant including documentation					
System:	Timer/PWM:	Analog-mixed Signal:	Communication:	Application specific:	Examples:
■ DMA ■ ERU ■ FCE ■ FLASH ■ GPIO ■ MATH ■ PAU ■ PRNG	■ CCU4 ■ CCU8 ■ HRPWM ■ POSIF	■ ACMP ■ ADC ■ DAC	■ CAN ■ I2C ■ MATH ■ SPI ■ UART ■ USB ■ USIC	■ BCCU ■ LEDTS ■ MATH ■ POSIF ■ HRPWM	■ Examples for all peripherals drivers and ARM, GCC, IAR, and Tasking

3. Example code (main.c) can be found within XMC Lib package



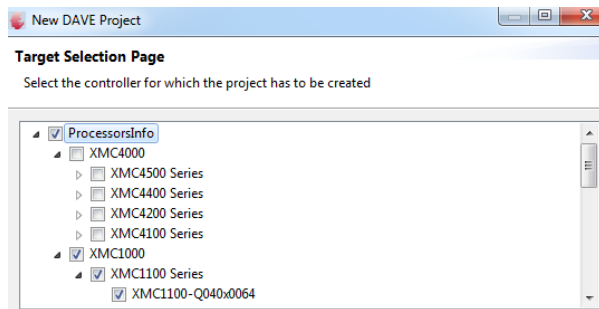
4. Create new “Simple Main Project” in DAVE™



References –

How to load Example Project in DAVE™? (4/4)

5. Select target device of selected main.c example

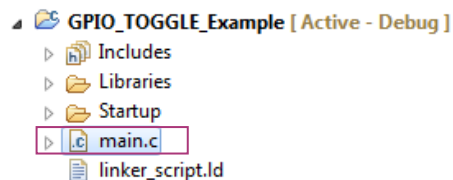


7. Click  to Build project

8. Click  to download and run project on target board

6. Delete main.c in the newly created DAVE project

7. Copy main.c from XMC Lib example into DAVE project





Part of your life. Part of tomorrow.

