

XMC1100 Boot Kit

Getting Started



Contents

- DAVE™ Setup
- Hardware Setup
- Boot Mode Index Configuration
- Getting Started Examples
 - Simple Blinky
(Simple XMC1100 Blinky.zip)
 - Blinky based on DAVE apps
(XMC1100 Blinky.zip)
- Example Projects Download
- Getting Started Videos

Contents

- **DAVE™ Setup**
- Hardware Setup
- Boot Mode Index Configuration
- Getting Started Examples
 - Simple Blinky
(Simple_XMC1100_Blinky.zip)
 - Blinky based on DAVE apps
(XMC1100_Blinky.zip)
- Example Projects Download
- Getting Started Videos



DAVE™ Setup

■ Download DAVE™ installer package from:

http://www.infineon.com/cms/en/product/promopages/aim-mc/DAVE_3_Download.html

DAVE™ Download



DAVE™ version 3.1.6 Download Options

There are two download options available:

- **DAVE™ version 3.1.6 as regular installer package**

This is a regular installer (setup.exe) that install DAVE™ and the SEGGER J-Link drivers in a user defined target folder. In case an earlier version of DAVE™ is already installed, the installer will uninstall the old version and install the new version in the same target folder. This ensures that workspaces from earlier versions of DAVE™ can be used with the new version. The installer package includes also an installation of a complete set of DAVE™ Apps and device descriptions if there are no DAVE™ Apps installed yet.

[DOWNLOAD](#) the installer package
- **DAVE™ version 3.1.6 as zipped file package**

This package contains a zip file with all required files to run DAVE™ on a PC plus the installation setup for the SEGGER J-Link drivers. The zip file can be unzipped anywhere and DAVE™ can be started from the ... \eclipse folder. The package is wrapped in an exe that users can accept the licenses conditions.

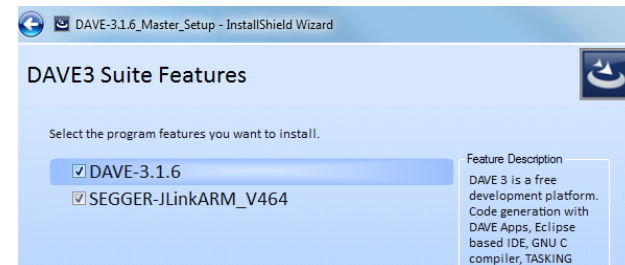
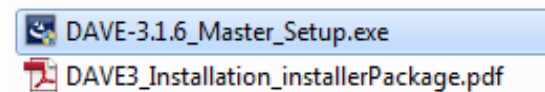
[DOWNLOAD](#) the zipped file package

- **Note:** For users who have downloaded DAVE™ as a zipped file package, DAVE™ can be started via DAVE-*.exe in the eclipse folder.

DAVE™ Setup

■ Download and unzip the installer package

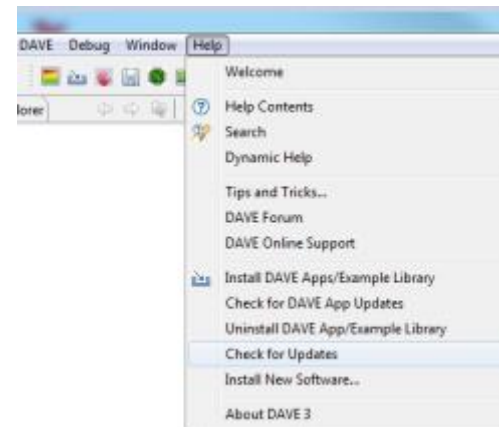
■ Run *_Setup.exe file to install
DAVE and Segger J-Link drivers



■ Open DAVE™ program



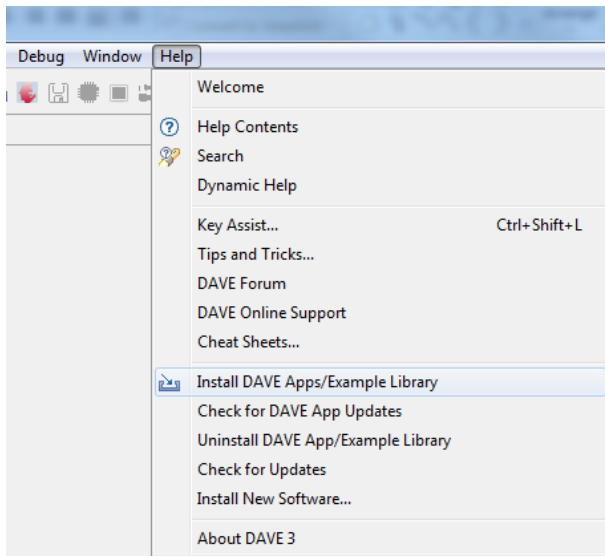
■ Check for DAVE updates
Help → Check for Updates



DAVE™ Setup

■ Install DAVE Apps and Device Descriptions

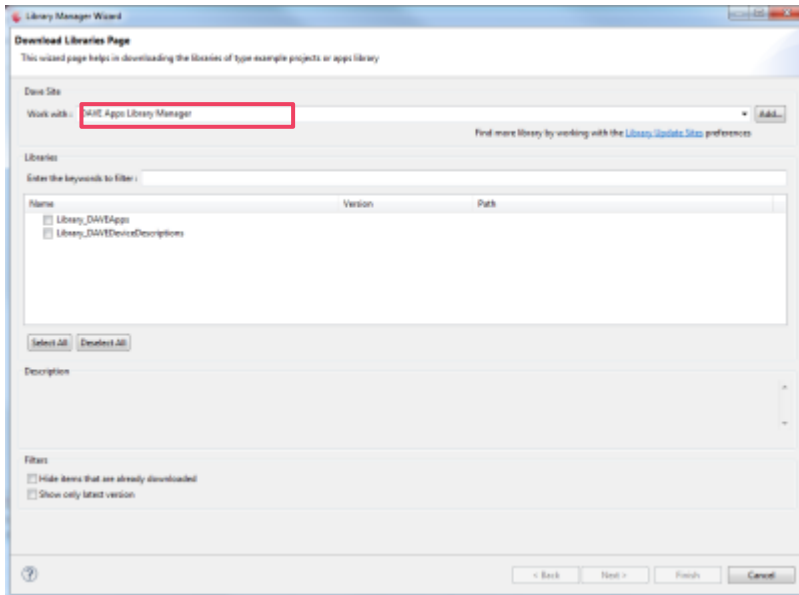
Help → Install DAVE Apps Library



■ Note: You may skip the above step if you are not using DAVE Apps

DAVE™ Setup

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



- Select Library_DAVEApps and Library_DAVEDeviceDescriptions (for XMC1100 Device) and click Next

- ▶ Library_DAVEApps
- ▶ Library_DAVEDeviceDescriptions

DAVE™ Setup

- Accept terms of the license agreements and click Finish

IMPORTANT DOWNLOAD NOTICE

The software you have requested for download is protected by national and international copyright laws and may be protected by other intellectual property rights. You shall use the software only in accordance with the applicable licensing terms and conditions which may be different from the terms and conditions of the DAVE 3 Software License Agreement. For copyright information, licensing terms and additional information (e.g. on how to obtain the source code of such Open Source Software), please check the "Help Function", Section "Copyright and Licensing Information" of the software. By downloading the software, you acknowledge that you have read and understood this download notice.

I accept the terms of the license agreements

I do not accept the terms of the license agreements

- DAVE Apps and DAVE device descriptions are installed

Contents

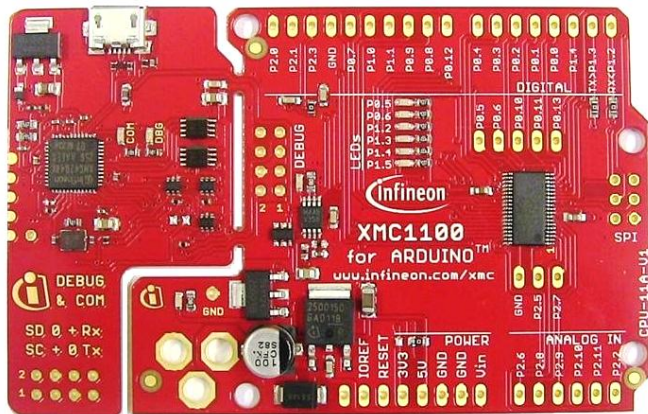
- DAVE™ Setup
- Hardware Setup
- Boot Mode Index Configuration
- Getting Started Examples
 - Simple Blinky
(Simple_XMC1100_Blinky.zip)
 - Blinky based on DAVE apps
(XMC1100_Blinky.zip)
- Example Projects Download
- Getting Started Videos



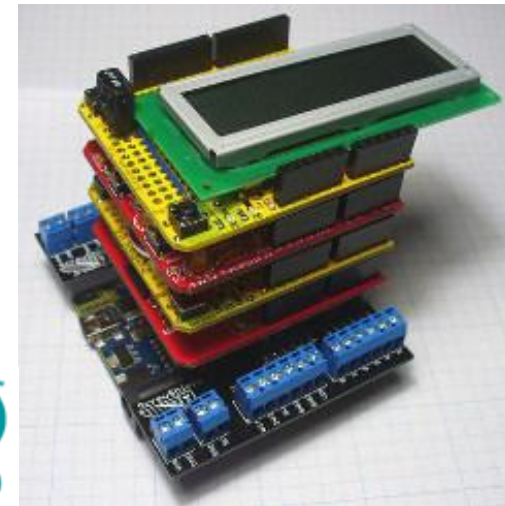
Hardware Setup

- XMC1100 Boot Kit

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



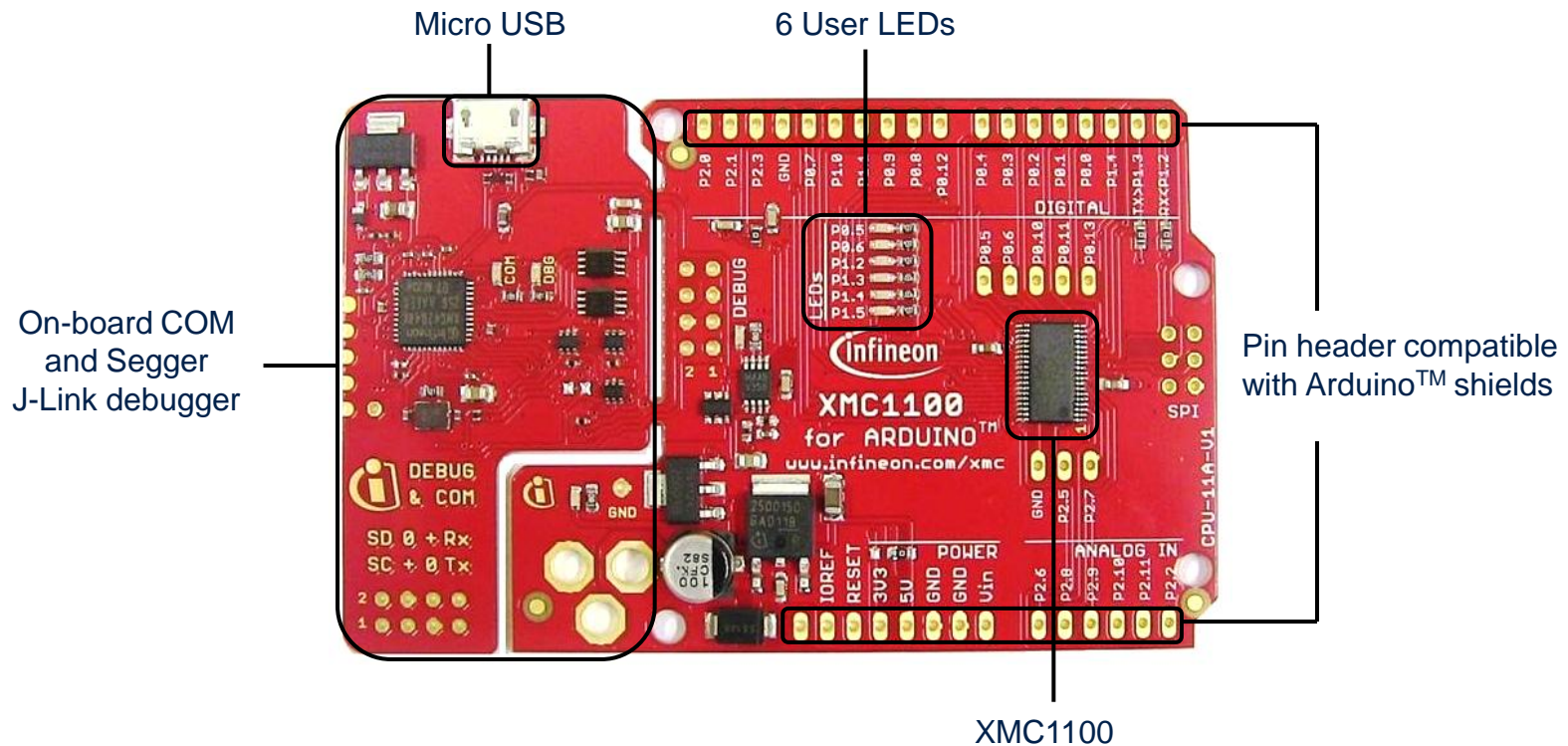
XMC1100 CPU Card



Arduino™ Shield

Hardware Setup

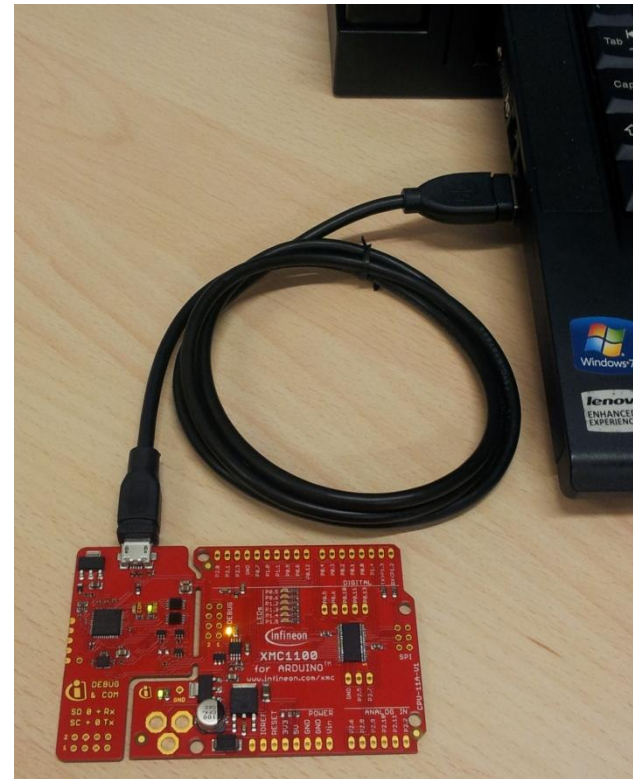
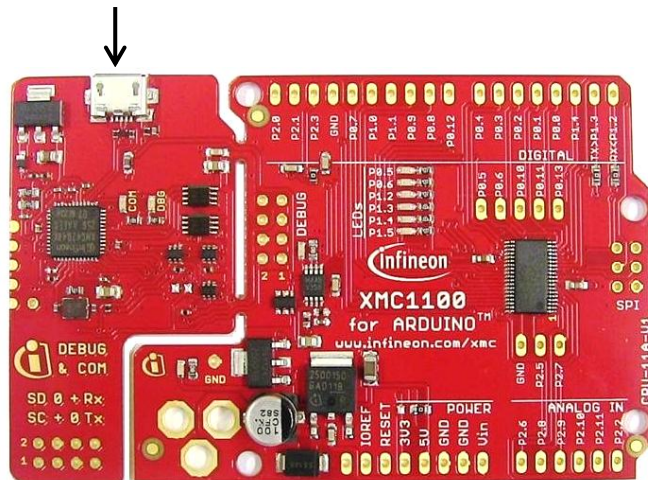
■ XMC1100 CPU Card for Arduino™



Hardware Setup

- 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

Contents

- DAVE™ Setup
- Hardware Setup
- **Boot Mode Index Configuration**
- Getting Started Examples
 - Simple Blinky
(Simple_XMC1100_Blinky.zip)
 - Blinky based on DAVE apps
(XMC1100_Blinky.zip)
- Example Projects Download
- Getting Started Videos



Boot Mode Index Configuration

- 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
 - http://www.infineon.com/cms/en/product/promopages/aim-mc/DAVE_3_Download.html
 - MemTool
 - Download MemTool
 - <http://www.infineon.com/cms/en/product/channel.html?channel=ff80808112ab681d0112ab6b50fe07c9>

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

Contents

- DAVE™ Setup
- Hardware Setup
- Boot Mode Index Configuration
- Getting Started Examples
 - Simple Blinky
(Simple_XMC1100_Blinky.zip)
 - Blinky based on DAVE apps
(XMC1100_Blinky.zip)
- Example Projects Download
- Getting Started Videos



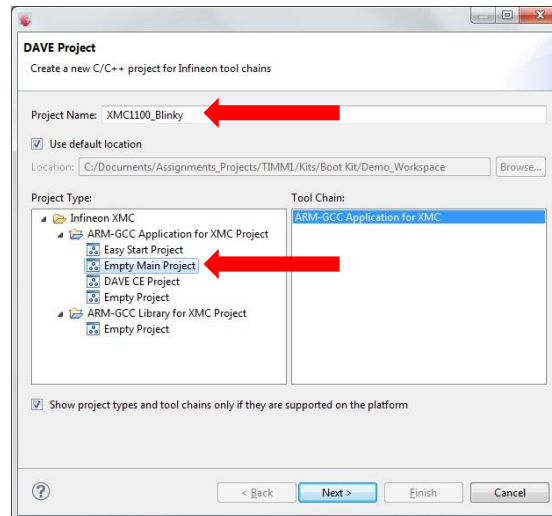
Getting Started Example Simple Blinky

1. Open DAVE™

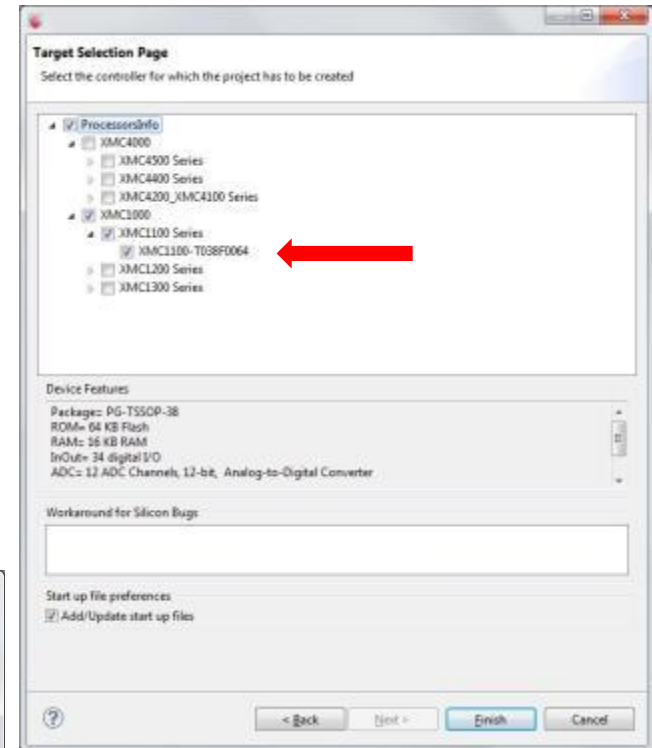


2. In DAVE™ workspace, create a new “Empty Main” project:

- ❑ File->New->DAVE Project
- ❑ Give the project a name e.g. “XMC1100_Blinky”
- ❑ Select “Empty Main Project” as Project Type



3. Select the device accordingly



Getting Started Example

Simple Blinky



- For this project, we will use
 - System clock frequency of 8MHz
 - LED on Port pin 0.5
 - └ LED is toggled in the SysTick interrupt service routine
 - System timer, SysTick, as the time base for the 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 SysTick and define its exception service routine

Getting Started Example Simple Blinky



1. Set up System or Main Clock (MCLK)

- ❑ MCLK configured via **IDIV** and **FDIV** bit fields in register **CLKCR**
- ❑ CLKCR and all registers of the XMC1100 are defined in **XMC1100.h**



```
483 /*-----*/
484 /*----- SCU_CLK -----*/
485 /*-----*/
486
487
488 /**
489  * @brief System Control Unit (SCU_CLK)
490  */
491
492 typedef struct {
493     __IO uint32_t CLKCR;          /*!< (@ 0x40010300) Clock Control Register
494     __IO uint32_t PWRSCVR;       /*!< (@ 0x40010304) Power Save Control Register
495     __I  uint32_t CGATSTAT0;     /*!< (@ 0x40010308) Peripheral 0 Clock Gating Status
496     __O  uint32_t CGATSETB;      /*!< (@ 0x4001030C) Peripheral 0 Clock Gating Set
497     __O  uint32_t CGATCLR0;     /*!< (@ 0x40010310) Peripheral 0 Clock Gating Clear
498     __IO uint32_t OSCCSR;       /*!< (@ 0x40010314) Oscillator Control and Status Register
499 } SCU_CLK_TypeDef;
```

- ❑ IDIV and FDIV are protected bits, therefore access has to be opened prior to configuration via register **PASSWD**
- ❑ Configuration in **Main.c**

```
67 //----CLOCK-SETUP-----
68 SCU_GENERAL->PASSWD = 0x000000C0UL; // disable bit protection scheme
69 SCU_CLK->CLKCR = 0x3FF00400UL; // 8 MHz MCLK, 8 MHz PCLK
70 while((SCU_CLK->CLKCR)&0x40000000UL); // wait for VDDC to stabilize
71 SCU_GENERAL->PASSWD = 0x000000C3UL; // enable bit protection scheme
72 //-----
```

Getting Started Example Simple Blinky



2. Configure Port pin

- Port registers are defined in **XMC1100.h**
- Ports-related macros and functions are provided in the header file, **gpio_xmc1100_tssop38.h**

```
22 #define OUTPUT_PP_AF6 0x80U
23 #define OUTPUT_PP_AF7 0x88U
24 #define OUTPUT_OD_GP 0xC0U
25 #define OUTPUT_OD_AF1 0xC8U
26 #define OUTPUT_OD_AF2 0xD0U

37 __STATIC_INLINE void P0_0_set_mode(uint8_t mode){
38     PORT0->IOCR0 &= ~0x000000f8UL;
39     PORT0->IOCR0 |= mode << 0;
40 }
```

```
672 /**
673  * @brief Port 0 (PORT0)
674  */
675
676 typedef struct {
677     __IO uint32_t OUT;
678     __IO uint32_t OMR;
679     __IO uint32_t RESERVED0[2];
680     __IO uint32_t IOCR0;
681     __IO uint32_t IOCR4;
682     __IO uint32_t IOCR8;
683     __IO uint32_t IOCR12;
684     __IO uint32_t RESERVED1;
685     __IO uint32_t IN;
686     __IO uint32_t RESERVED2[6];
687     __IO uint32_t PHCR0;
688     __IO uint32_t PHCR1;
689     __IO uint32_t RESERVED3[6];
690     __IO uint32_t PDISC;
691     __IO uint32_t RESERVED4[3];
692     __IO uint32_t PPS;
693     __IO uint32_t HWSEL;
694 } PORT0_Type;
```

- With this header file, we can easily initialize the port pin 0.5 as a general purpose output pin in **Main.c**, using the provided macro

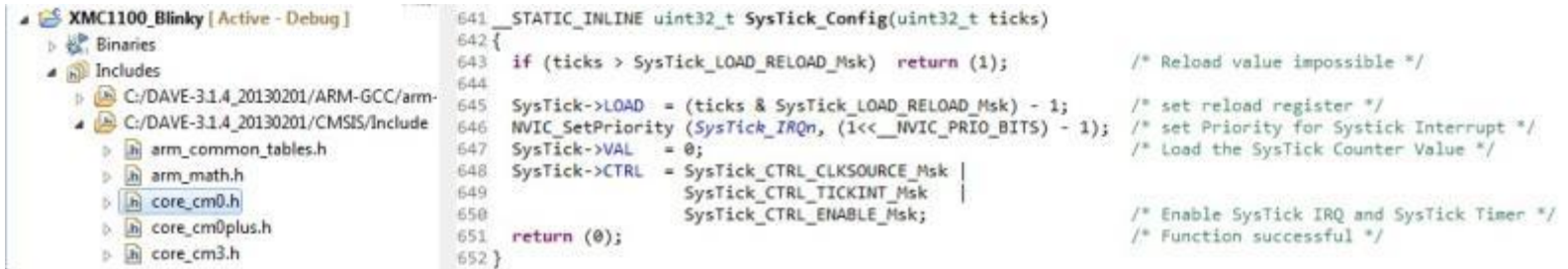
```
49 #include "gpio_XMC1100_tssop38.h"

63 //-----PIN-SETUP-----
64 P0_5_set_mode(OUTPUT_OD_GP);
65 //-----
```

Getting Started Example Simple Blinky

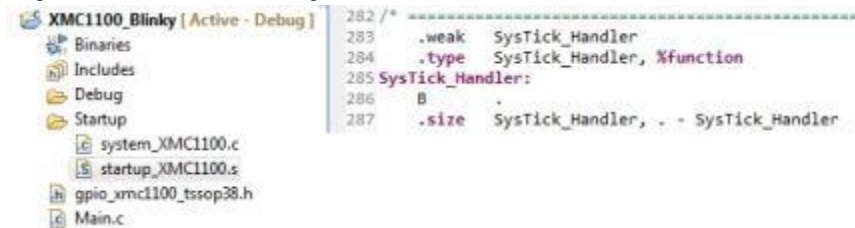
3. Configure SysTick and define its exception service routine

Initialization function is defined in **core_cm0.h**



```
641 __STATIC_INLINE uint32_t SysTick_Config(uint32_t ticks)
642 {
643     if (ticks > SysTick_LOAD_RELOAD_Msk) return (1); /* Reload value impossible */
644
645     SysTick->LOAD = (ticks & SysTick_LOAD_RELOAD_Msk) - 1; /* set reload register */
646     NVIC_SetPriority (SysTick_IRQn, (1<<__NVIC_PRIO_BITS) - 1); /* set Priority for SysTick Interrupt */
647     SysTick->VAL = 0; /* Load the SysTick Counter Value */
648     SysTick->CTRL = SysTick_CTRL_CLKSOURCE_Msk |
649                   SysTick_CTRL_TICKINT_Msk |
650                   SysTick_CTRL_ENABLE_Msk; /* Enable SysTick IRQ and SysTick Timer */
651     return (0); /* Function successful */
652 }
```

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



```
282 /*-----*/
283 .weak SysTick_Handler
284 .type SysTick_Handler, %function
285 SysTick_Handler:
286     B     .
287     .size SysTick_Handler, . - SysTick_Handler
```

Initialize the SysTick in **Main.c**

```
74 //-----SYSTICK-SETUP-----
75 SystemCoreClockUpdate();
76 SysTick_Config(SystemCoreClock / 5); // 0.2s interrupt base
77 //-----
```

Define the SysTick exception handler routine in **Main.c**

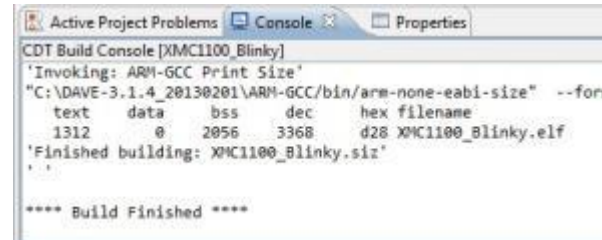
→ Toggle the LED

```
55 void SysTick_Handler(void) // 0.2s interrupt base
56 {
57     P0_5_toggle();
58 }
```

Getting Started Example Simple Blinky

■ Build project



1. Click 
2. Wait for Build to finish

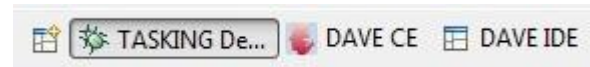


```
CDT Build Console [XMC1100_Blinky]
'Invoking: ARM-GCC Print Size'
"C:\DAVE-3.1.4_20130201\ARM-GCC\bin\arm-none-eabi-size" --forr
text  data  bss   dec   hex filename
1312   0     2056  3368  d28 XMC1100_Blinky.elf
'Finished building: XMC1100_Blinky.siz'

**** Build Finished ****
```

■ Download code

1. Click 
2. Switch to TASKING Debug view
3. Click  to run code



■ LED blinks every 0.2s



Contents

- DAVE™ Setup
- Hardware Setup
- Boot Mode Index Configuration
- Getting Started Examples
 - Simple Blinky
(Simple_XMC1100_Blinky.zip)
 - Blinky based on DAVE apps
(XMC1100_Blinky.zip)
- Example Projects Download
- Getting Started Videos



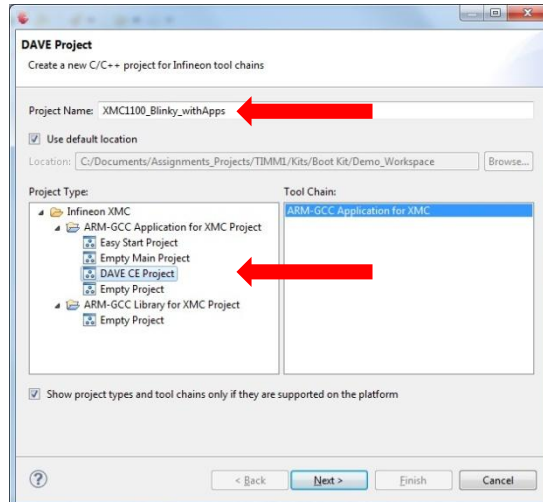
Getting Started Example Blinky based on DAVE apps

1. Open DAVE™

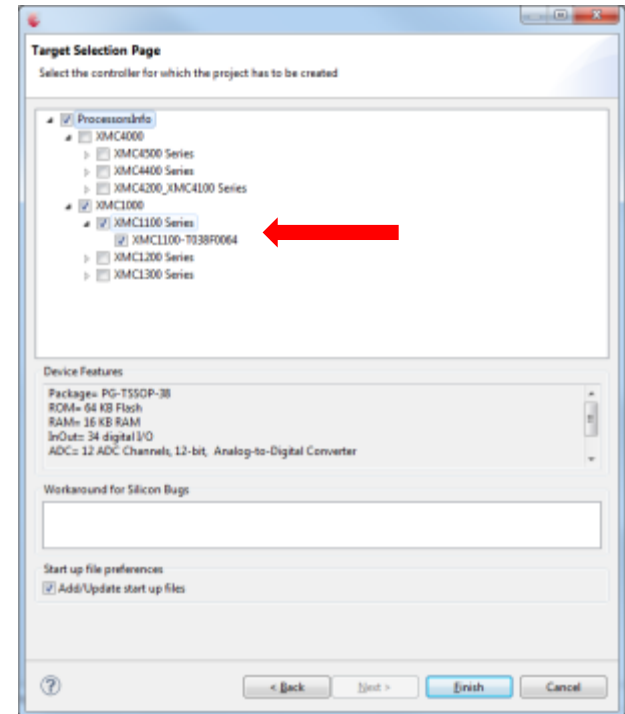


2. In DAVE™ workspace, create a new “DAVE CE” project:

- ❑ File->New->DAVE Project
- ❑ Give the project a name e.g. “XMC1100_Blinky_withApps”
- ❑ Select “DAVE CE Project” as Project Type



3. Select the device accordingly



Getting Started Example

Blinky based on DAVE apps



- For this project, we will use
 - System clock frequency of 8MHz
 - LED on Port pin 0.5
 - └ LED is toggled in the SysTick interrupt service routine
 - System timer, SysTick, as the time base for the 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 SysTick and define its exception service routine

Getting Started Example

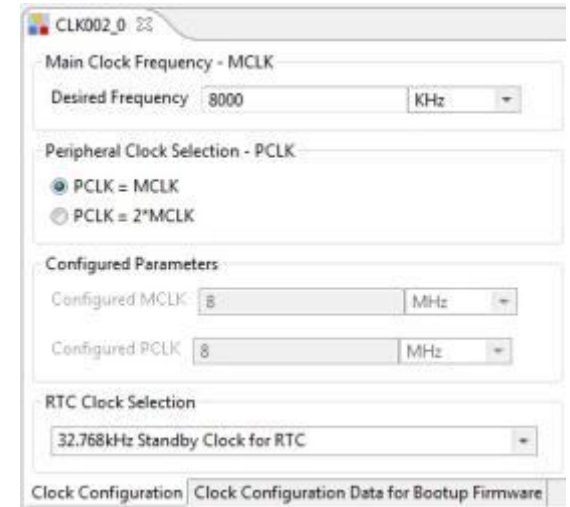
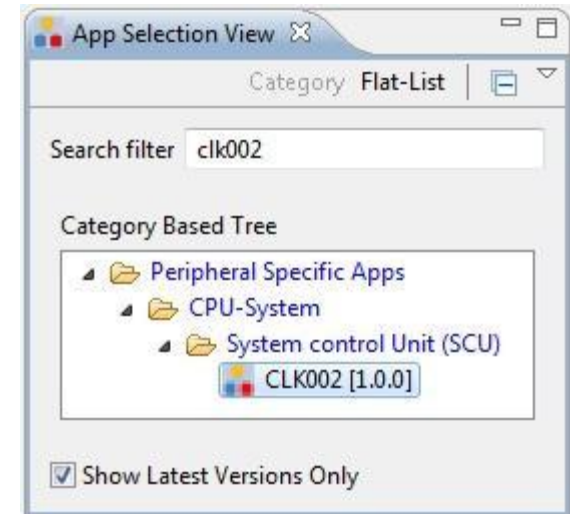
Blinky based on DAVE apps

1. Set up System or Main Clock (MCLK)

- ❑ Select **CLK002** app from the **App Selection View** window
- ❑ Open CLK002 **UIEditor** by double-clicking or right-click->UIEditor on the app in **S/W Connectivity View**



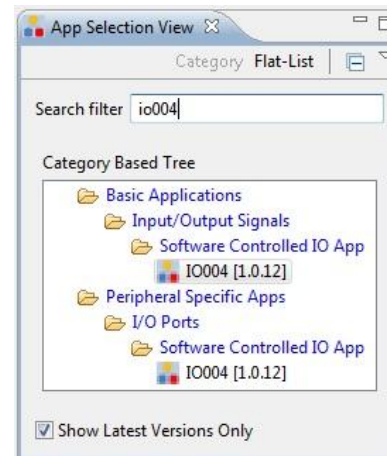
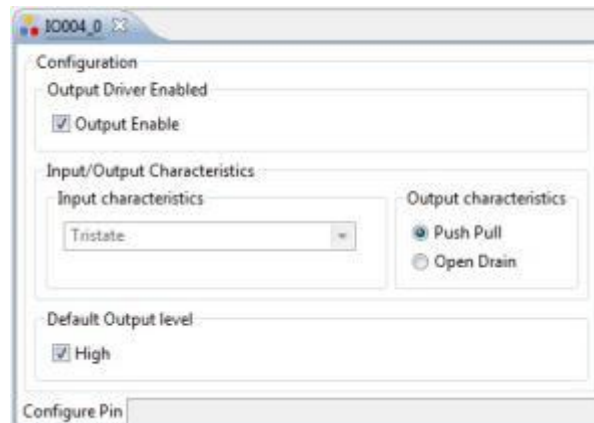
- ❑ In UIEditor, under the **Clock Configuration** tab, change the **Desired Frequency** of the MCLK to 8MHz



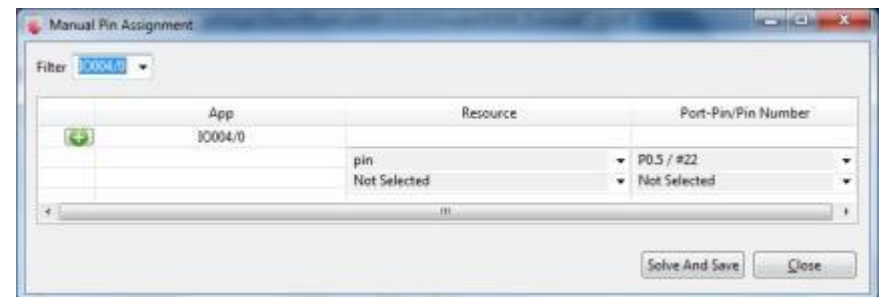
Getting Started Example Blinky based on DAVE apps

2. Configure Port pin

- ❑ Select **IO004** app from the **App Selection View** window
- ❑ Open IO004 **UIEditor** by double-clicking or right-click->UIEditor on the app in **S/W Connectivity View**
- ❑ In UIEditor, under the **Configure Pin** tab, enable the **Output Driver** and set **Output Level** to High by checking the respective check-boxes



- ❑ Assign pin to P0.5
 - Right-click on app->**Manual Pin Assignment**
 - In Manual Pin Assignment window, set **pin** as Resource, **P0.5** as Port-Pin
 - **Solve And Save**



Getting Started Example Blinky based on DAVE apps



3. Configure SysTick and define its exception service routine

Initialization function is defined in **core_cm0.h**



```
641 _STATIC_INLINE uint32_t SysTick_Config(uint32_t ticks)
642 {
643     if (ticks > SysTick_LOAD_RELOAD_Msk) return (1); /* Reload value impossible */
644
645     SysTick->LOAD = (ticks & SysTick_LOAD_RELOAD_Msk) - 1; /* set reload register */
646     NVIC_SetPriority (SysTick_IRQn, (1<< __NVIC_PRIO_BITS) - 1); /* set Priority for Systick Interrupt */
647     SysTick->VAL = 0; /* Load the SysTick Counter Value */
648     SysTick->CTRL = SysTick_CTRL_CLKSOURCE_Msk |
649                   SysTick_CTRL_TICKINT_Msk |
650                   SysTick_CTRL_ENABLE_Msk; /* Enable SysTick IRQ and SysTick Timer */
651     return (0); /* Function successful */
652 }
```

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

```
291     Insert_ExceptionHandler SysTick_Handler
```

Initialize the SysTick in **Main.c**

```
74     //-----SYSTICK-SETUP-----
75     SystemCoreClockUpdate();
76     SysTick_Config(SystemCoreClock / 5); // 0.2s interrupt base
77     //-----
```

Define the SysTick exception handler routine in **Main.c**

Toggle the LED

```
53 void SysTick_Handler(void) // 0.2s interrupt base
54 {
55     I0004_TogglePin(I0004_Handle0);
56 }
```

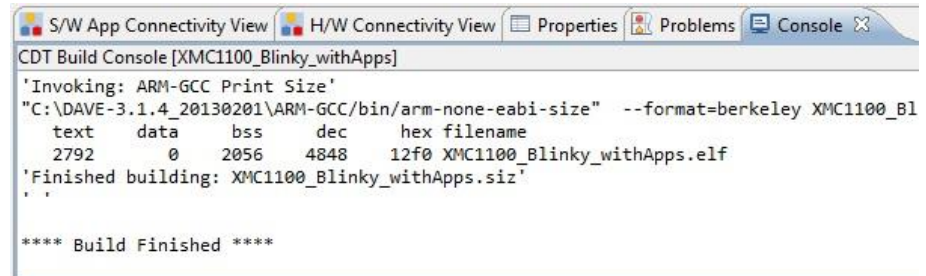
Getting Started Example Blinky based on DAVE apps

■ Generate code

1. Click 

■ Build project

1. Click 
2. Wait for Build to finish



```
CDT Build Console [XMC1100_Blinky_withApps]
'Invoking: ARM-GCC Print Size'
"C:\DAVE-3.1.4_20130201\ARM-GCC/bin/arm-none-eabi-size" --format=berkeley XMC1100_Bl
  text  data  bss  dec  hex filename
 2792   0   2056  4848  12f0 XMC1100_Blinky_withApps.elf
'Finished building: XMC1100_Blinky_withApps.siz'

**** Build Finished ****
```

■ Download code

1. Click 
2. Switch to TASKING Debug view



3. Click  to run code

■ LED blinks every 0.2s



Contents

- DAVE™ Setup
- Hardware Setup
- Boot Mode Index Configuration
- Getting Started Examples
 - Simple Blinky
(Simple_XMC1100_Blinky.zip)
 - Blinky based on DAVE apps
(XMC1100_Blinky.zip)
- Example Projects Download
- Getting Started Videos



Example Projects Download

- Two sets of Example Projects available
 - Additional Application Examples
 - Can be downloaded directly from the web
 - DAVE™ Project Library Examples
 - Can be downloaded from library in DAVE™
 - Can also be downloaded directly from the web

Example Projects Download

Additional Application Examples

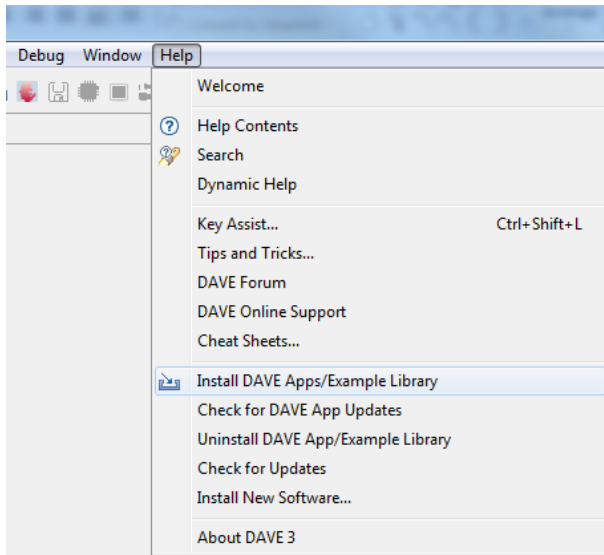


- Additional Application Examples available
 - Running LEDs Example
(Simple_XMC1100_RunningLEDs.zip)
 - UART Example
(Simple_XMC1100_UART.zip)

- Can be downloaded from the web [HERE](#)

Example Projects Download DAVE™ Example Library

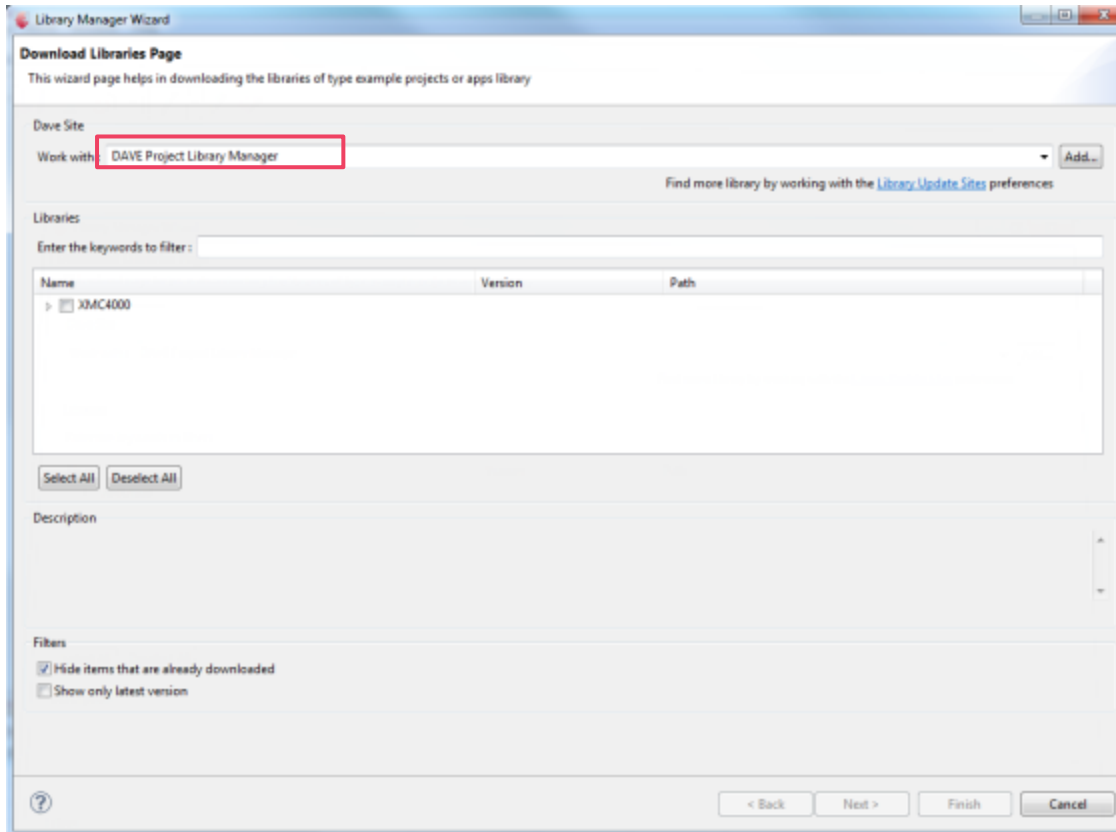
- Download Example Projects via DAVE™ library store
- Help → Install DAVE Example Library



Example Projects Download DAVE™ Example Library

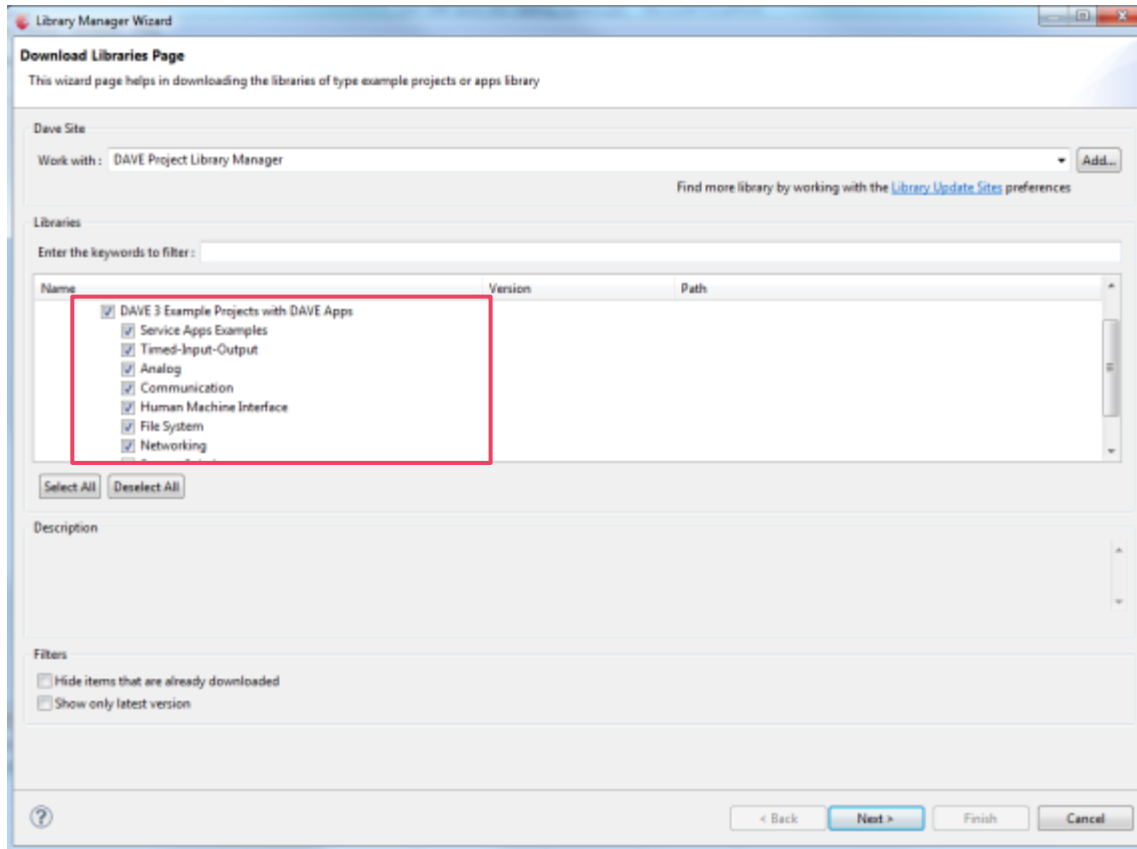


- Select DAVE Project Library Manager in the drop-down menu



Example Projects Download DAVE™ Example Library

- Select Examples in the Libraries window and click Next



Example Projects Download DAVE™ Example Library



- Accept terms of the license agreements and click Finish

IMPORTANT DOWNLOAD NOTICE

The software you have requested for download is protected by national and international copyright laws and may be protected by other intellectual property rights. You shall use the software only in accordance with the applicable licensing terms and conditions which may be different from the terms and conditions of the DAVE 3 Software License Agreement. For copyright information, licensing terms and additional information (e.g. on how to obtain the source code of such Open Source Software), please check the "Help Function", Section "Copyright and Licensing Information" of the software. By downloading the software, you acknowledge that you have read and understood this download notice.

I accept the terms of the license agreements

I do not accept the terms of the license agreements

- DAVE Example Projects are installed

Example Projects Download

DAVE™ Example Library



■ Download Example Projects from the web

http://www.infineon.com/cms/en/product/promopages/aim-mc/DAVE_3_Support_Portal/DAVE_Example_Project_Download.html

- Download the project zip file
- Open DAVE™ and go to File → Import → Infineon → DAVE Project
- Check “Select Archive File”
- Browse to the downloaded DAVE project zip file
- Press “Open”

Contents

- DAVE™ Setup
- Hardware Setup
- Boot Mode Index Configuration
- Getting Started Examples
 - Simple Blinky
(Simple_XMC1100_Blinky.zip)
 - Blinky based on DAVE apps
(XMC1100_Blinky.zip)
- Example Projects Download
- Getting Started Videos



Getting Started Videos

- 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



ENERGY EFFICIENCY MOBILITY SECURITY

Innovative semiconductor solutions for energy efficiency, mobility and security.

