Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

XMC1200 Boot Kit

Getting Started

Agenda (1/2)

Agenda (1/2)

Agenda (1/2)

Agenda (1/2)

Agenda (1/2)

Agenda (1/2)

Agenda (1/2)

Agenda (1/2)

Agenda (1/2)

› XMC1200 Boot Kit

- – Consists of an XMC1200 CPU Card
- – Supported Application Card examples: Colour LED Card, White LED Card

(Application Cards are orderable separately or as part of another Application Kit)
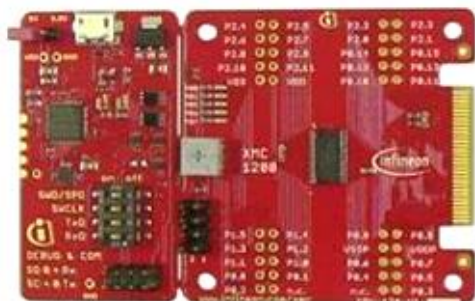
Colour LED Card

White LED Card

XMC1200 CPU Card

› ## XMC1200 CPU Card

Micro USB   5 User LEDs

On-board COM
and Segger
J-Link debugger

Connectors
according to pin-out

XMC1200   Edge connector for
Application cards

Home
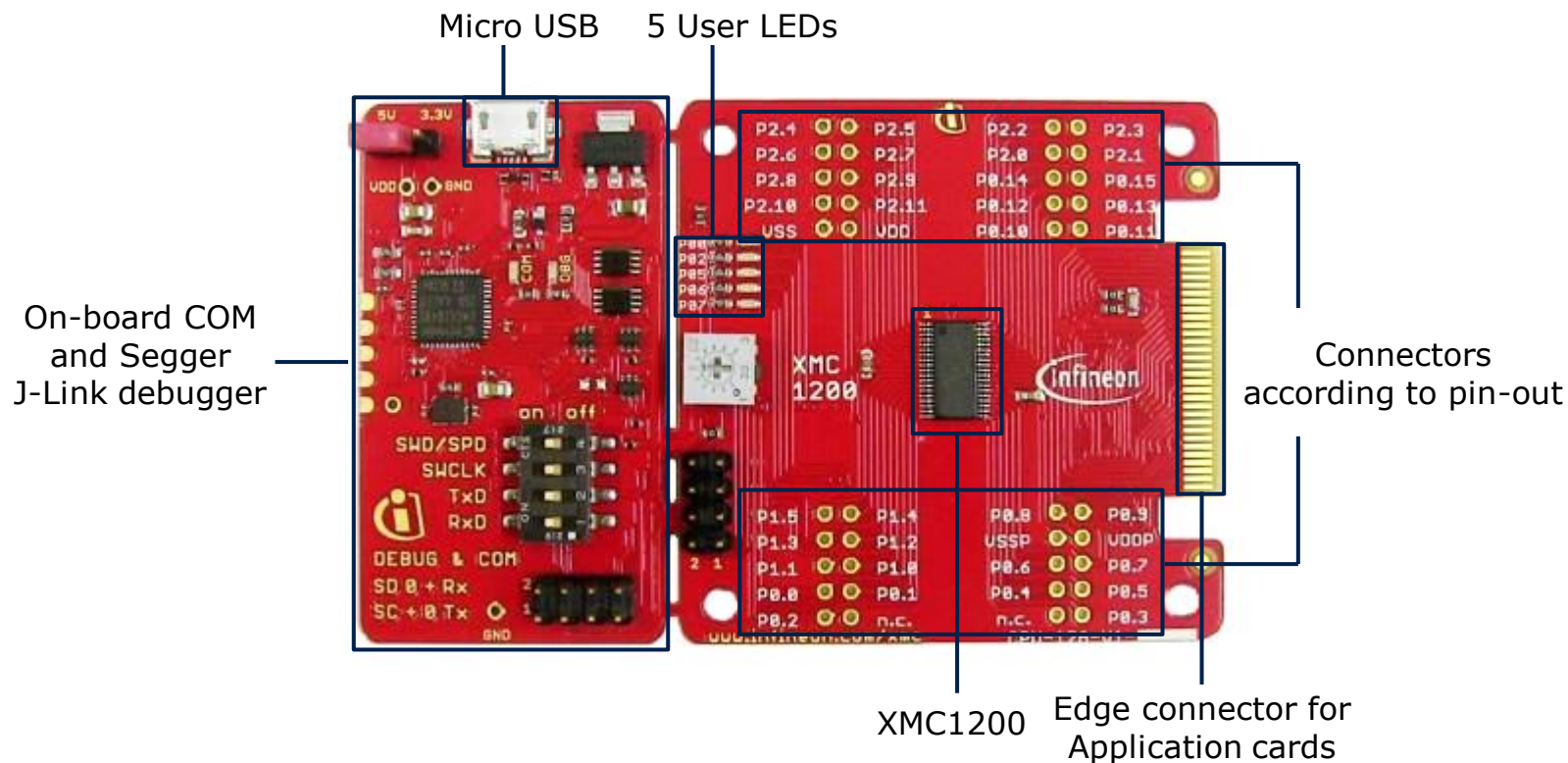
Hardware Overview

› Connect XMC1200 CPU Card to PC via USB cable

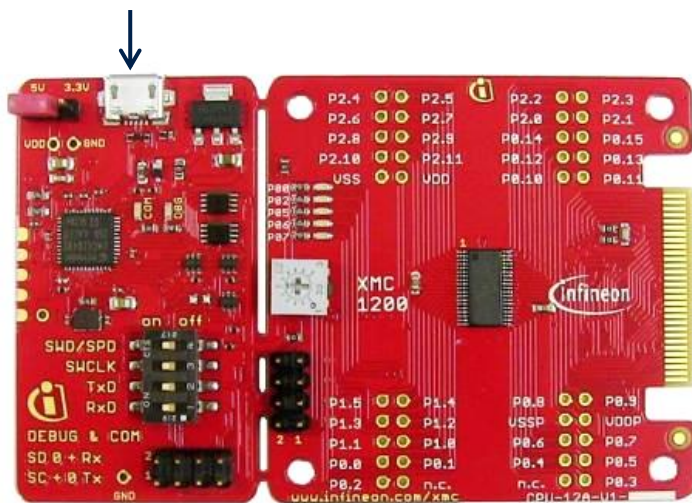› CPU Card is powered up (as indicated by LED on the card)

CPU Card powered
via USB cable

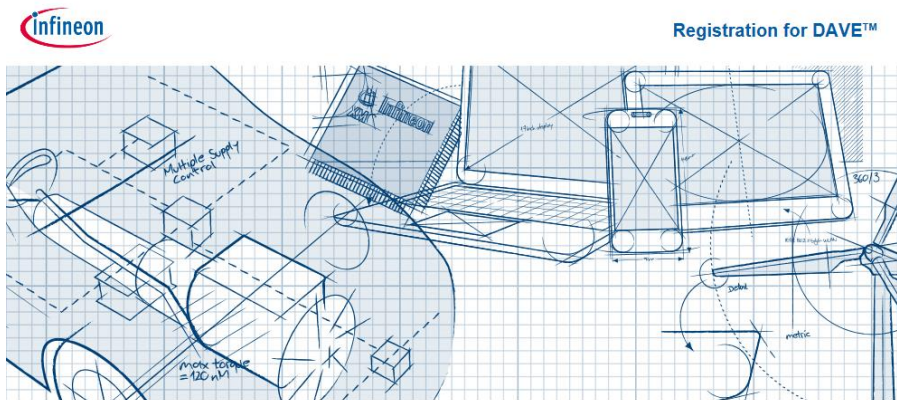› Note: Supported Application Card may be additionally connected to the CPU card

Home

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.

Home

Tooling Overview – DAVE™ (1/5)

› DAVE™ download package is available at:

http://infineon-community.com/LP=400

› After registration, download and unzip the installer package

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

› Open DAVE^{TM}

› Update DAVE^{TM} and DAVE^{TM} libraries
   – Help → Check for Updates
   – Help → Check for DAVE APP Updates
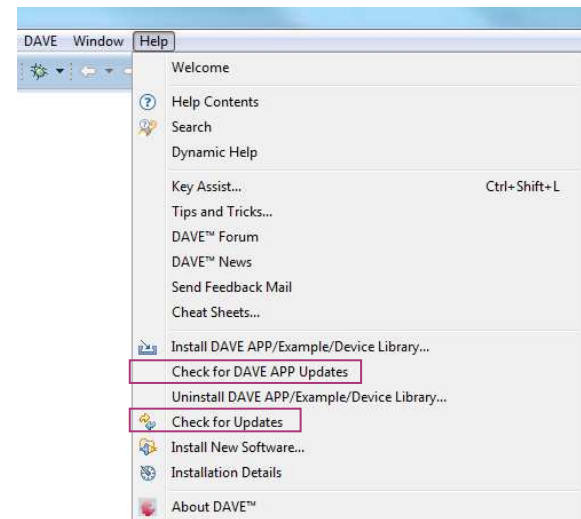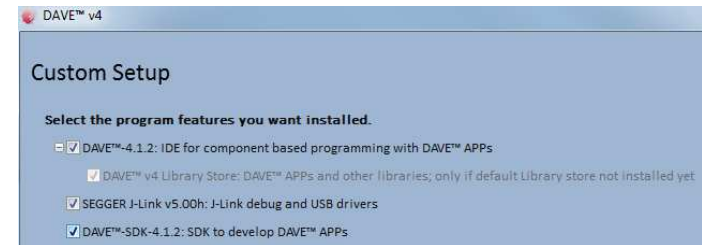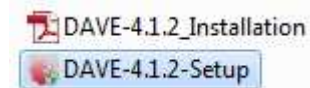
› Install DAVE™ APPs libraries and Device Description

– Help → Install DAVE APP/Example/Device Library
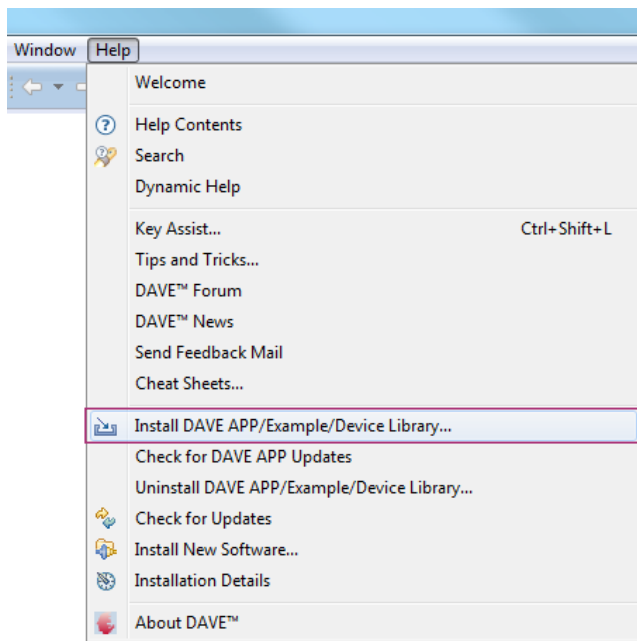
› Note: You may skip the above step if you are not using DAVE™ APPs

› Select DAVE Apps Library Manager in the drop-down menu
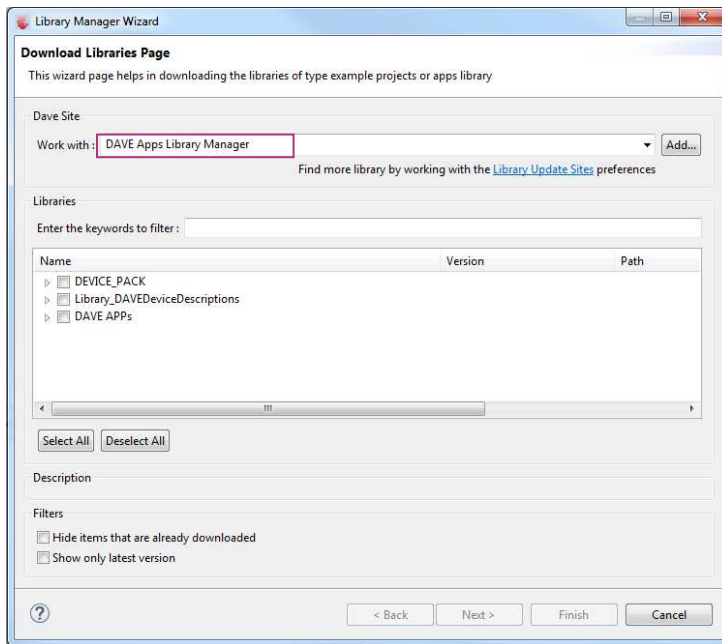
› Select DEVICE_PACK, Library_DAVEDeviceDescriptions
(XMC1200 Device) and DAVE APPs

› Accept terms of the license agreements and click Finish
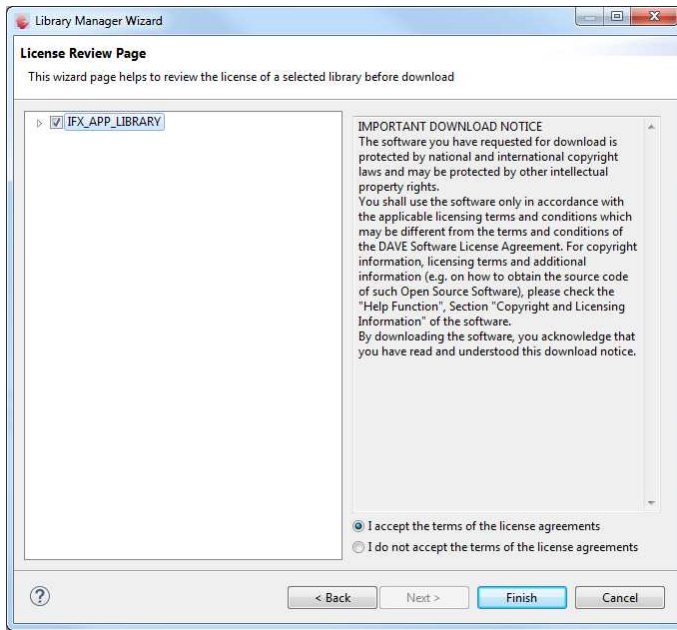
› DAVE^{TM} APPs libraries and Device Description are installed

Home

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. "XMC1200_Blinky"

   – Select "Simple Main Project" as Project Type

3. Select the device accordingly

› For this project, we will use

  – System clock frequency of 8MHz

  – LED on Port pin 0.0

  – 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

1. Set up System or Main Clock (MCLK) using XMCLib

    – Include the header files required for MCLK and GPIO configuration

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

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

    ```c
    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**

    ```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_GPIOC_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 P0.0 as general purpose output pin in **Main.c**

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

3. Configure SysTick and define its exception service routine

    – SysTick exception handler is defined in **startup_XMC1200.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;
  }
}
```

› Build project

1. Click ![hammer icon]

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
   2232      20    1040    3292     cdc XMC1200_Blinky.elf
'Finished building: XMC1200_Blinky.siz'
```

› Download code

1. Click ![debug icon]

2. Switch to Debug perspective ![DAVE IDE DAVE CE PinMapping Debug]

3. Click ![run icon] to run code

› LED blinks every 0.2s

![Home]