XMC1000 LED lighting application kit

XMC™ microcontrollers
July 2016
<table>
<thead>
<tr>
<th>Number</th>
<th>Agenda Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Kit overview</td>
</tr>
<tr>
<td>2</td>
<td>Hardware overview</td>
</tr>
<tr>
<td>3</td>
<td>Tooling overview – boot modes</td>
</tr>
<tr>
<td>4</td>
<td>Tooling overview – DAVE™</td>
</tr>
<tr>
<td>5</td>
<td>Getting started - examples</td>
</tr>
<tr>
<td>6</td>
<td>General information</td>
</tr>
<tr>
<td>7</td>
<td>References</td>
</tr>
</tbody>
</table>
Agenda

1. Kit overview
2. Hardware overview
3. Tooling overview – boot modes
4. Tooling overview – DAVE™
5. Getting started - examples
6. General information
7. References
Kit overview (1/3)

› XMC1200 CPU Card

- Micro USB
- 5 User LEDs
- On-board COM and Segger J-Link debugger
- Connectors according to pin-out
- Edge connector for application cards
- XMC1200
Kit overview (2/3)

› Color LED card
  - Showcases color control

- Low power RGB LEDs (10 mA per channel)
- Ambient light sensor
- RF connectivity
- DALI connectivity
- DMX512 connectivity
Kit overview (3/3)

- White LED card
  - Showcases brightness control

White LED strings (5 LEDs per string)
24 V powered, 20 mA per channel

Temperature sensor

Ambient light sensor

RF connectivity

DALI connectivity
Agenda

1. Kit overview
2. Hardware overview
3. Tooling overview – boot modes
4. Tooling overview – DAVE™
5. Getting started - examples
6. General information
7. References
Hardware overview

› Attach color LED or white LED card to XMC1200 CPU card
› Connect XMC1200 CPU card to PC via USB cable
› CPU card is powered up (as indicated by LED on the card)
Agenda

1. Kit overview
2. Hardware overview
3. Tooling overview – boot modes
4. Tooling overview – DAVE™
5. Getting started - examples
6. General information
7. References
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™
      [http://www.infineon.com/dave/v4](http://www.infineon.com/dave/v4)
  - MemTool
    - Download MemTool

› For more information on how to configure the BMI value, please refer to the XMC1000 tooling guide
Agenda

1. Kit overview
2. Hardware overview
3. Tooling overview – boot modes
4. Tooling overview – DAVE™
5. Getting started - examples
6. General information
7. References
DAVE™ is a free development platform for code generation by Infineon.

It can be downloaded from:

- [http://www.infineon.com/dave/v4](http://www.infineon.com/dave/v4)

For a guide on setting up DAVE™, please refer to XMC1x00 boot kit getting started.
Agenda

1. Kit overview
2. Hardware overview
3. Tooling overview – boot modes
4. Tooling overview – DAVE™
5. Getting started - examples
6. General information
7. References
Getting started – Example 1
RGB lamp using LED_LAMP APP (1/20)

Example 1: RGB lamp using LED_LAMP APP

Connect to PC

XMC1200/XMC1300 CPU card

Color LED card
Getting started – Example 1
RGB lamp using LED_LAMP APP (2/20)

1. Open DAVE™

2. In DAVE™ workspace, create a new “DAVE™ CE” project:
   › File->New->DAVE™ Project
   › Give the project a name e.g. “RGB_LAMP_EXAMPLE”
   › Select “DAVE™ CE Project” as project type

3. Select the device accordingly
This example demonstrates RGB lamp functionality using LED_LAMP APP

We will use the system timer (SysTick) as the time base for the interrupt
- Time base of 1 s
- In the interrupt, a new target dimming level or target color is regularly set with a 7 s transition time

Next, we will show you the steps to creating this project:
1. Instantiate LED_LAMP APP
2. Configure LED_LAMP APP
3. Configure BCCU Channels
4. Assign PDM_BCCU APPs to the right channels
5. Configure Brightness and Color Control Unit (BCCU) global settings
6. Configure Port Pins
7. Configure SysTick
8. Define the SYSTIMER callback function
1. Instantiate LED_LAMP APP
   › Click to add new APP
   › Select the LED_LAMP APP
   › LED_LAMP APP automatically aggregates a BCCU channel app (PDM_BCCU), a BCCU dimming engine app (DIM_BCCU) and a BCCU global app (GLOBAL_BCCU)
2. Configure LED_LAMP APP
   › Double-click **LED_LAMP_0** to open UI

   › Under **General Settings** tab,
     - set **Number of LED channels** to 3
     - select **Dimming Engine** as **Dimming Source**
2. Configure LED_LAMP APP (continued)

- Under **Dimming and Intensities Settings** tab
  - set initial **Dimming Level** to **1024**
  - set initial **Channel Intensities** to **1365**
  - set initial **Intensity linear walk time** to **0 ms**
  - Set initial 0-100% dimming transition time to **0 ms**
2. Configure LED_LAMP APP (continued)

› Rename Instance Label

- **Right-click** LED_LAMP APP
- Select **Rename Instance Label**...
- Rename as **RGB_LAMP**
Getting started – Example 1
RGB lamp using LED_LAMP APP (8/20)

3. Configure BCCU Channels
   › Double-click a **PDM_BCCU APP**

   › Select **Flicker Watchdog (WD)** to enable

   › Repeat for the other 2 PDM_BCCU APP instances
4. Assign PDM_BCCU APPs to the right channels
   › Hover mouse cursor over the connecting arrow to a PDM_BCCU APP
   › A label will appear momentarily e.g. LED0/LED1/LED2
4. Assign PDM_BCCU APPs to the right channels (continued)
   ❖ The labels correspond to the LED channels in the UI

   ❖ Rename the PDM_BCCU instance label according to the table below
     - Right-click PDM_BCCU APP
     - Select “Rename Instance Label”

     | Label   | New Label |
     |---------|-----------|
     | LED0    | R_LED1    |
     | LED1    | G_LED1    |
     | LED2    | B_LED1    |

   - Repeat the above steps with the other 2 PDM_BCCU APP instances
Getting started – Example 1
RGB lamp using LED_LAMP APP (11/20)

4. Assign PDM_BCCU APPs to the right channels (continued)
   › Click to assign pins to PDM_BCCU APPs
   › Assign pins as shown:
5. Configure BCCU global settings
   › Double-click GLOBAL_BCCU_0 in APP Dependency tab

   › Under **Clock Settings** tab,
     - to get a bit time of 5 us
     - change the **Desired Fast Clock Frequency** to **0.8 MHz**
5. Configure BCCU global settings (continued)
   › Under **Functional Settings** tab,
     - limit the maximum possible off time to approx. 5ms (no flicker)
       - change **ON-bit insertion threshold** to **1024**

![Screenshot of Functional Settings tab with ON-bit insertion threshold set to 1024]
6. Configure PORT Pins

- The intention of this step is to ensure that the unused pins (to the LED2 and LED3) are not left in a floating state.

- Add 6 instances of DIGITAL_IO APP to the project.
Getting started – Example 1
RGB lamp using LED_LAMP APP (15/20)

6. Configure PORT Pins (continued)
   › Double-click a **DIGITAL_IO** APP to open UI
   › Set **Pin Direction** to **Input/Output**
   › Repeat for other 5 instances of DIGITAL_IO APP
Getting started – Example 1
RGB lamp using LED_LAMP APP (16/20)

6. Configure PORT Pins (continued)

› Click to assign pins to DIGITAL_IO APPs
› Assign pins as shown:

<table>
<thead>
<tr>
<th>DIGITAL_IO_0</th>
<th>pin</th>
<th>#22 (P0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIGITAL_IO_1</td>
<td>pin</td>
<td>#23 (P0.6)</td>
</tr>
<tr>
<td>DIGITAL_IO_2</td>
<td>pin</td>
<td>#24 (P0.7)</td>
</tr>
<tr>
<td>DIGITAL_IO_3</td>
<td>pin</td>
<td>#27 (P0.8)</td>
</tr>
<tr>
<td>DIGITAL_IO_4</td>
<td>pin</td>
<td>#28 (P0.9)</td>
</tr>
<tr>
<td>DIGITAL_IO_5</td>
<td>pin</td>
<td>#29 (P0.10)</td>
</tr>
</tbody>
</table>
7. Configure SysTick

› Add **SYSTIMER** to the project

› Double-click **SYSTIMER** APP to open UI
  
  – Set **SysTick timer period** to **1000 us**
  
  – Set **Number of software timers** to **1**

› Click ![file_icon] to generate code
8. Define SYSTIMER callback function

› Purpose of callback function is to change the colour and brightness of LED every 7 seconds

› Initialize callback function

```c
void OneSecTick(void);
```

› Create software timer and start timer

```c
timer_id;
TimerId = SYSTIMER_CreateTimer(1000000, SYSTIMER_MODE_PERIODIC, OneSecTick, NULL);
SYSTIMER_StartTimer(TimerId);
```
8. Define SYSTIMER callback function (continued)

```c
void OneSecTick(void)
{
    static uint8_t step = 0;
    if (++step==1) {// change color to red
        RGB_LAMP_config.led_intensity[0] = 4095;
        RGB_LAMP_config.led_intensity[1] = 0;
        RGB_LAMP_config.led_intensity[2] = 0;
        LED_LAMP_SetColorAdv(&RGB_LAMP, 0x2AC);
    }
    else if (step==9) {// change color to green
        RGB_LAMP_config.led_intensity[0] = 0;
        RGB_LAMP_config.led_intensity[1] = 4095;
        RGB_LAMP_config.led_intensity[2] = 0;
        LED_LAMP_SetColorAdv(&RGB_LAMP, 0x2AC);
    }
    else if (step==17) {// change color to blue
        RGB_LAMP_config.led_intensity[0] = 0;
        RGB_LAMP_config.led_intensity[1] = 0;
        RGB_LAMP_config.led_intensity[2] = 4095;
        LED_LAMP_SetColorAdv(&RGB_LAMP, 0x2AC);
    }
    else if (step==25) {// change color to white
        RGB_LAMP_config.led_intensity[0] = 1365;
        RGB_LAMP_config.led_intensity[1] = 1365;
        RGB_LAMP_config.led_intensity[2] = 1365;
        LED_LAMP_SetColorAdv(&RGB_LAMP, 0x2AC);
    }
    else if (step==33) {// dim down slowly to 0%
        RGB_LAMP_config.dim_level = 0;
        LED_LAMP_SetDimLevelExponentialAdv(&RGB_LAMP,0x64,0xDB);
    }
    else if (step==40) {// dim up slowly to 25%
        RGB_LAMP_config.dim_level = 1024;
        LED_LAMP_SetDimLevelExponentialAdv(&RGB_LAMP,0x64,0xDB);
    }
    else if (step==47) {
        step = 0;
    }
}
```

Copyright © Infineon Technologies AG 2016. All rights reserved.
Getting started – Example 1
RGB lamp using LED_LAMP APP (20/20)

› Build project
   1. Click
   2. Wait for Build to finish

› Download code
   1. Click
   2. Switch to Debug view
   3. Click to run code

› LED1 regularly changes color and brightness
Example 2: White Lamp using LED_LAMP APP

Connect to PC

XMC1200/XMC1300 CPU Card

White LED Card

24V Power Supply
1. Open DAVE™

2. In DAVE™ workspace, create a new “DAVE™ CE” project:
   - File->New->DAVE™ Project
   - Give the project a name e.g. “WHITE_LAMP_EXAMPLE”
   - Select “DAVE™ CE Project” as Project Type

3. Select the device accordingly
This example demonstrates White Lamp functionality using LED_LAMP APP.

We will use the System Timer (SysTick) as the time base for the interrupt:
- Time base of 1 s
- In the interrupt, a new target dimming level is set and the dimming process is started.

Next, we will show you the steps to creating this project:
1. Instantiate LED_LAMP APP
2. Configure LED_LAMP APP
3. Assign PDM_BCCU APPs to the right channels
4. Configure BCCU Channels
5. Configure Brightness and Color Control Unit (BCCU) global settings
6. Configure SysTick
7. Define the SYSTIMER callback function
1. Instantiate LED_LAMP APP
   - Click + to add new APP
   - Select the LED_LAMP APP
   - LED_LAMP APP automatically aggregates a BCCU channel app (PDM_BCCU), a BCCU dimming engine app (DIM_BCCU) and a BCCU global app (GLOBAL_BCCU)
Getting started – Example 2
White lamp using LED_LAMP APP (5/19)

2. Configure LED_LAMP APP
   › Double-click **LED_LAMP_0** to open UI
   › Under **General Settings** tab,
     - set **Number of LED channels** to 4
     - select **Dimming Engine** as **Dimming Source**
2. Configure LED_LAMP APP (continued)

› Under **Dimming and Intensities Settings** tab
- set initial **Dimming Level** to 0
- set initial **Channel Intensities** to 4095
- set initial **Intensity linear walk time** to 0 ms
- Set initial **0-100% dimming transition time to 0 ms**
3. Assign PDM_BCCU APPs to the right channels

› Hover mouse cursor over the connecting arrow to a PDM_BCCU APP

› A label will appear momentarily e.g. LED0/LED1/LED2/LED3
3. Assign PDM_BCCU APPs to the right channels (continued)
   – The labels correspond to the LED channels in the UI

   ![Diagram of LED channels and settings](image)

   – Rename the PDM_BCCU instance label according to the table below
     - Right-click PDM_BCCU APP
     - Select “Rename Instance Label”

     | Label   | New Label |
     |---------|-----------|
     | LED0    | D_LED1    |
     | LED1    | D_LED2    |
     | LED2    | D_LED3    |
     | LED3    | D_LED4    |

   – Repeat the above steps with the other 2 PDM_BCCU APP instances
3. Assign PDM_BCCU APPs to the right channels (continued)

› Click ![Image](image) to assign pins to PDM_BCCU APPs

› Assign pins as shown:

![Manual Pin Allocator](image)
4. Configure BCCU Channels

› Double-click PDM_BCCU instance **D_LED1**

› Select **Flicker Watchdog (WD)** to enable

› Select **Packer** to enable

› Set **Number of ON-bits grouped** to **3**

› Set **Number of OFF-bits grouped** to **50**
Getting started – Example 2
White lamp using LED_LAMP APP (11/19)

4. Configure BCCU Channels (continued)
   › Double-click PDM_BCCU instance **D_LED2**
   › Select **Flicker Watchdog (WD)** to enable
   › Select **Packer** to enable
   › Set **Packer OFF-bit counter value** to 12
   › Set **Number of ON-bits grouped** to 3
   › Set **Number of OFF-bits grouped** to 50
Getting started – Example 2
White lamp using LED_LAMP APP (12/19)

4. Configure BCCU Channels (continued)
   › Double-click PDM_BCCU instance D_LED3
   › Select Flicker Watchdog (WD) to enable
   › Select Packer to enable
   › Set Packer OFF-bit counter value to 25
   › Set Number of ON-bits grouped to 3
   › Set Number of OFF-bits grouped to 50
4. Configure BCCU Channels (continued)

› Double-click PDM_BCCU instance D_LED4

› Select Flicker Watchdog (WD) to enable

› Select Packer to enable

› Set Packer OFF-bit counter value to 37

› Set Number of ON-bits grouped to 3

› Set Number of OFF-bits grouped to 50
5. Configure BCCU global settings
   › Double-click GLOBAL_BCCU_0 in APP Dependency tab
   › Under Clock Settings tab,
     - to get a bit time of 4 us
     - change the Desired Fast Clock Frequency to 1 MHz
5. Configure BCCU global settings (continued)

› Under **Functional Settings** tab,
  - limit the minimum brightness to 1%
  - change **ON-bit insertion threshold** to 100
6. Configure SysTick
   › Add **SYSTIMER** to the project
   
   › Double-click **SYSTIMER** APP to open UI
      - Set **SysTick timer period** to **1000 us**
      - Set **Number of software timers** to **1**

   › Click ![File icon] to generate code
7. Define SYSTIMER callback function
   › Purpose of callback function is to change the brightness of lamp every 10 seconds
   › Initialize callback function

```c
void OneSecTick(void);
```

› Create software timer and start timer

```c
uint32_t timer_id;
TimerId = SYSTIMER_CreateTimer(1000000, SYSTIMER_MODE_PERIODIC, OneSecTick, NULL);
SYSTIMER_StartTimer(TimerId);
```
7. Define SYSTIMER callback function (continued)

```c
void OneSecTick(void)
{
    static uint8_t step = 0;

    if (++step==1) {
        /* Dim up to 10% slowly */
        LED_LAMP_0_config.dim_level = 410;
        LED_LAMP_SetDimLevelExponentialAdv(&LED_LAMP_0, 0x64, 0xDB);
    }
    else if (step==10) {
        /* Dim up to 100% slowly */
        LED_LAMP_0_config.dim_level = 4095;
        LED_LAMP_SetDimLevelExponentialAdv(&LED_LAMP_0, 0x64, 0xDB);
    }
    else if (step==20) {
        /* Dim down to 0% slowly */
        LED_LAMP_0_config.dim_level = 0;
        LED_LAMP_SetDimLevelExponentialAdv(&LED_LAMP_0, 0x64, 0xDB);
    }
    else if (step==30) {
        step = 0;
    }
}
```
Getting started – Example 2
White lamp using LED_LAMP APP (19/19)

› Build project
  1. Click
  2. Wait for Build to finish

› Download code
  1. Click
  2. Switch to Debug view
  3. Click to run code

› LEDs regularly change brightness
General information (1/2)

› Where to buy kit:
  - [http://ehitex.com/starter-kits/for-xmc1000](http://ehitex.com/starter-kits/for-xmc1000)
  - Order Number: KIT_XMC1x_AK_LED_001

› Infineon parts utilized on kit:

<table>
<thead>
<tr>
<th>Infineon parts</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMC1200 Microcontroller</td>
<td>XMC1200-T038F0200</td>
</tr>
<tr>
<td>XMC4200 Microcontroller</td>
<td>XMC4200-Q48F256</td>
</tr>
<tr>
<td>5 V regulator</td>
<td>IFX25001TFV50</td>
</tr>
<tr>
<td>3V3 regulator</td>
<td>IFX25001MEV33</td>
</tr>
<tr>
<td>BCR421/SC74 LED Driver</td>
<td>BCR421UE6327HTSA1</td>
</tr>
<tr>
<td>BCR450 LED Driver</td>
<td>BCR450E6327HTSA1</td>
</tr>
<tr>
<td>TDA7200 RF Receiver</td>
<td>TDA7200XUMA1</td>
</tr>
</tbody>
</table>
General information (2/2)

› Kit documentation:
  - LED Lighting Application Kit

› Video series: XMC1000 boot kit getting started
  - Introduction
  - DAVE™ (Version 4) - Project Management
  - Boot Mode Index Configuration via DAVE™ or MemTool
  - Example Projects Download
Agenda

1. Kit overview
2. Hardware overview
3. Tooling overview – boot modes
4. Tooling overview – DAVE™
5. Getting started - examples
6. General information
7. References
Go to Help → Help Contents
Click DAVE™ APPs
Click APP_Name (e.g. LED_LAMP)
Usage information can be found under **Usage** section.
Where to download example projects?

- 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
Additional application examples available

- RGB Lamps Example with Apps
  (LED_LAMP_3RGB_EXAMPLE_XMC12.zip)
  - Demonstrates 3 RGB Lamps functionality using 3 LED_LAMP APPs
    (9 Channels, 3 Dimming Engines)
- RGB Lamp Example
  (BCCU_RGB_LAMP_EXAMPLE.zip)
  - Demonstrates 1 RGB Lamp functionality using XMC™ Lib
Where to download example projects?

› Additional application examples available
  – White Lamp Example
    (BCCU_WHITE_LAMP_EXAMPLE.zip)
    – Demonstrates white lamp functionality using XMC™ Lib
› Can be downloaded from the web HERE
Download example projects via DAVE™ library store

- Help → Install DAVE™ APP/Example/Device Library...
Select **DAVE™ Project Library Manager** in the drop-down menu.
Select examples in the **Libraries** window and click Next.
Accept terms of the license agreement and click Finish.

DAVE™ example projects are installed.
Download Example Projects from the web

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

- Download the project zip file and unzip to a known location
- Open DAVE™ and go to File → Import → Infineon → DAVE™ Project
- Select Select Archive File
- Browse to the downloaded DAVE™ project zip file
- Click Open
- Click Finish
## Support material

### Collaterals and Brochures
- Product Briefs
- Selection Guides
- Application Brochures
- Presentations
- Press Releases, Ads

### Technical Material
- Application Notes
- Technical Articles
- Simulation Models
- Datasheets, MCDS Files
- PCB Design Data

### Videos
- Technical Videos
- Product Information Videos

### Contact
- Forums
- Product Support

---

**www.infineon.com/XMC**

**Kits and Boards**

**DAVE™**

**Software and Tool Ecosystem**

**Infineon Media Center**

**XMC Mediathek**

**Infineon Forums**

**Technical Assistance Center (TAC)**
The information given in this training materials is given as a hint for the implementation of the Infineon Technologies component only and shall not be regarded as any description or warranty of a certain functionality, condition or quality of the Infineon Technologies component.

Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this training material.
Part of your life. Part of tomorrow.