# XMC1000 LED lighting application kit

XMC$^{TM}$ microcontrollers
July 2016

# 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

# Agenda

1 Kit overview

2 Hardware overview

3 Tooling overview – boot modes

4 Tooling overview – DAVE™

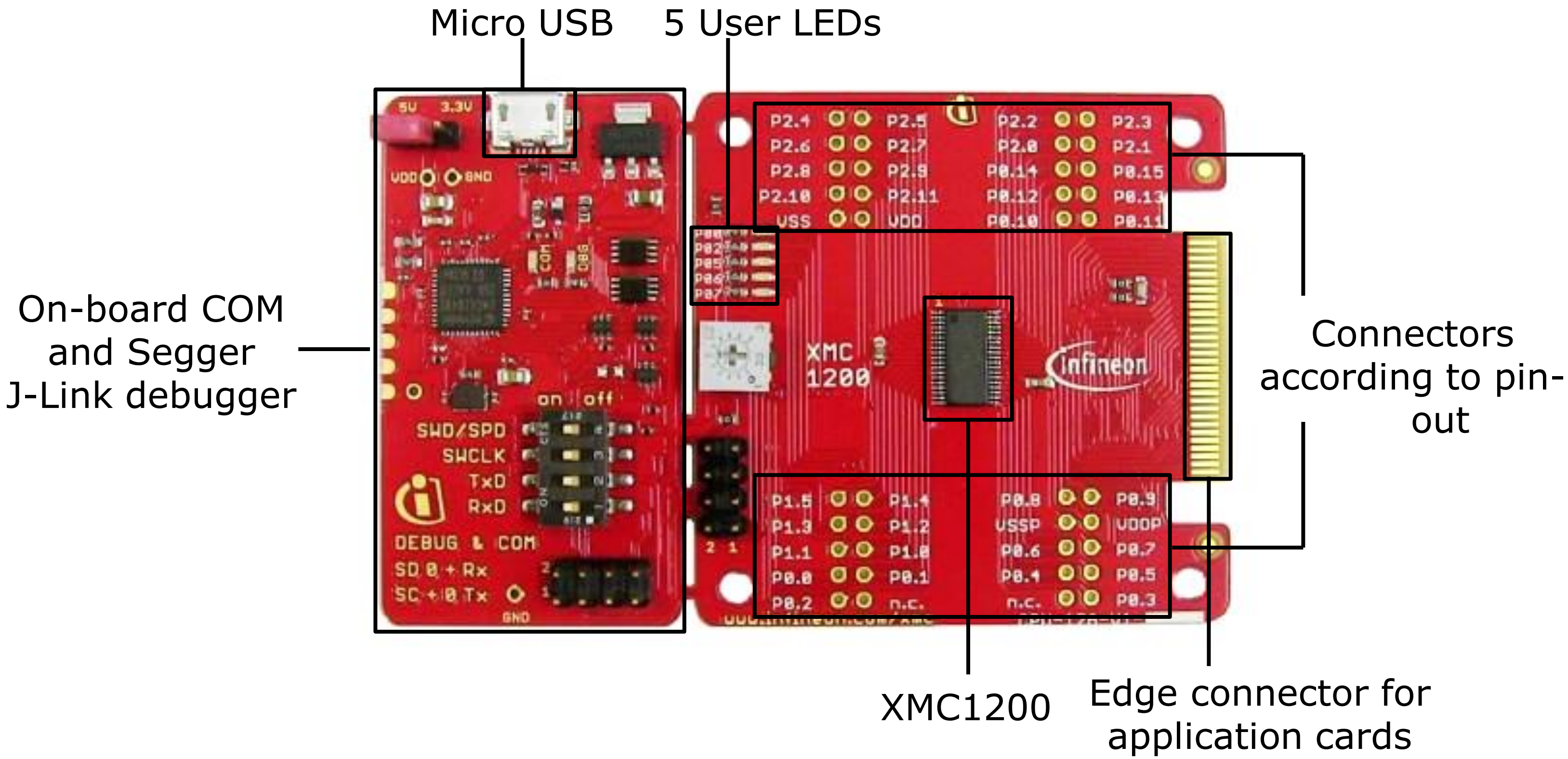
5 Getting started - examples

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

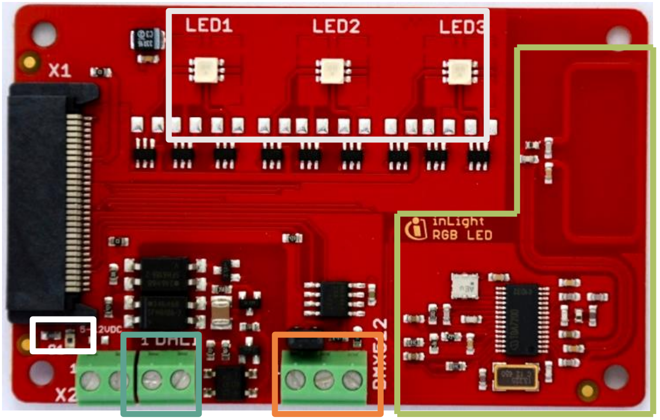
Connectors according to pin-out

XMC1200

Edge connector for application cards

# Kit overview (2/3)

- Color LED card
  - Showcases color control



Low power RGB LEDs (10 mA per channel)

Ambient light sensor

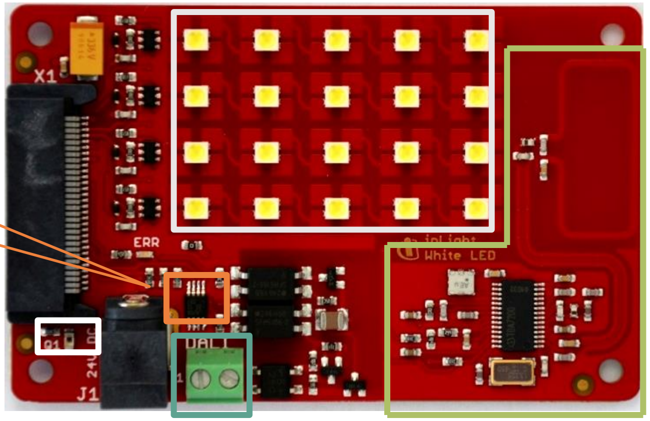
RF connectivity

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

RF connectivity

Ambient light sensor

DALI connectivity

# Agenda

1 Kit overview

2 Hardware overview

3 Tooling overview – boot modes

4 Tooling overview – DAVE™

5 Getting started - examples

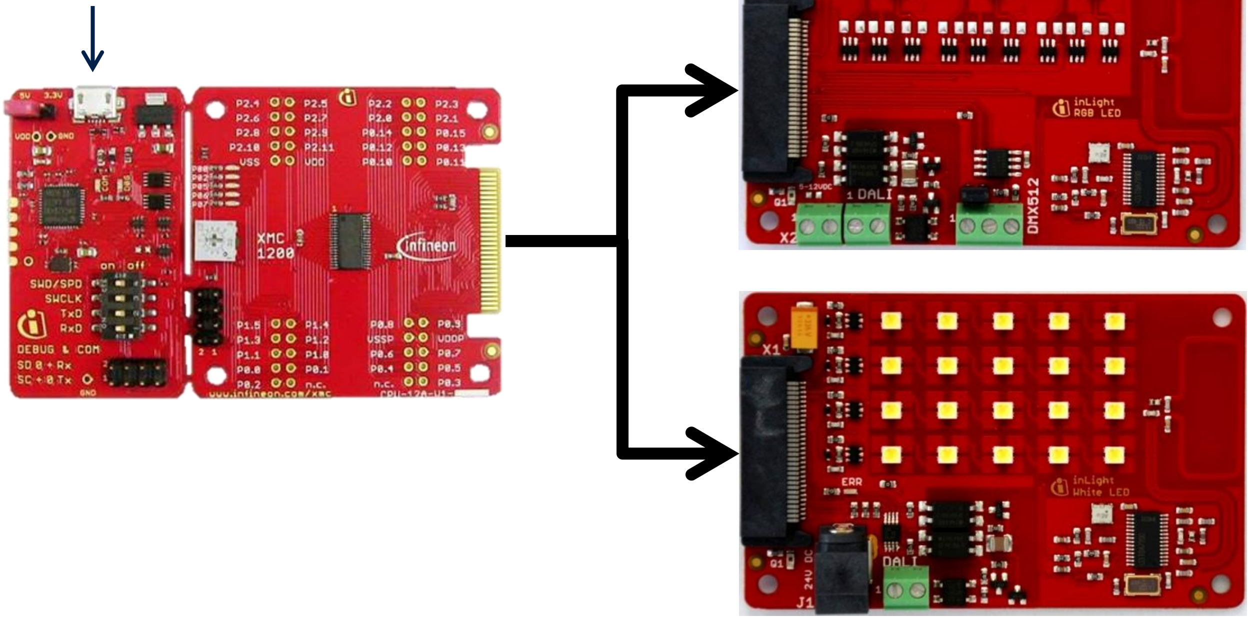
6 General information

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)

CPU card powered
via USB cable



# 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
  - 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

# 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
# DAVE$^{TM}$

- DAVE$^{TM}$ is a free development platform for code generation by Infineon
- It can be downloaded from:
  - http://www.infineon.com/dave/v4
- For a guide on setting up DAVE$^{TM}$, 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

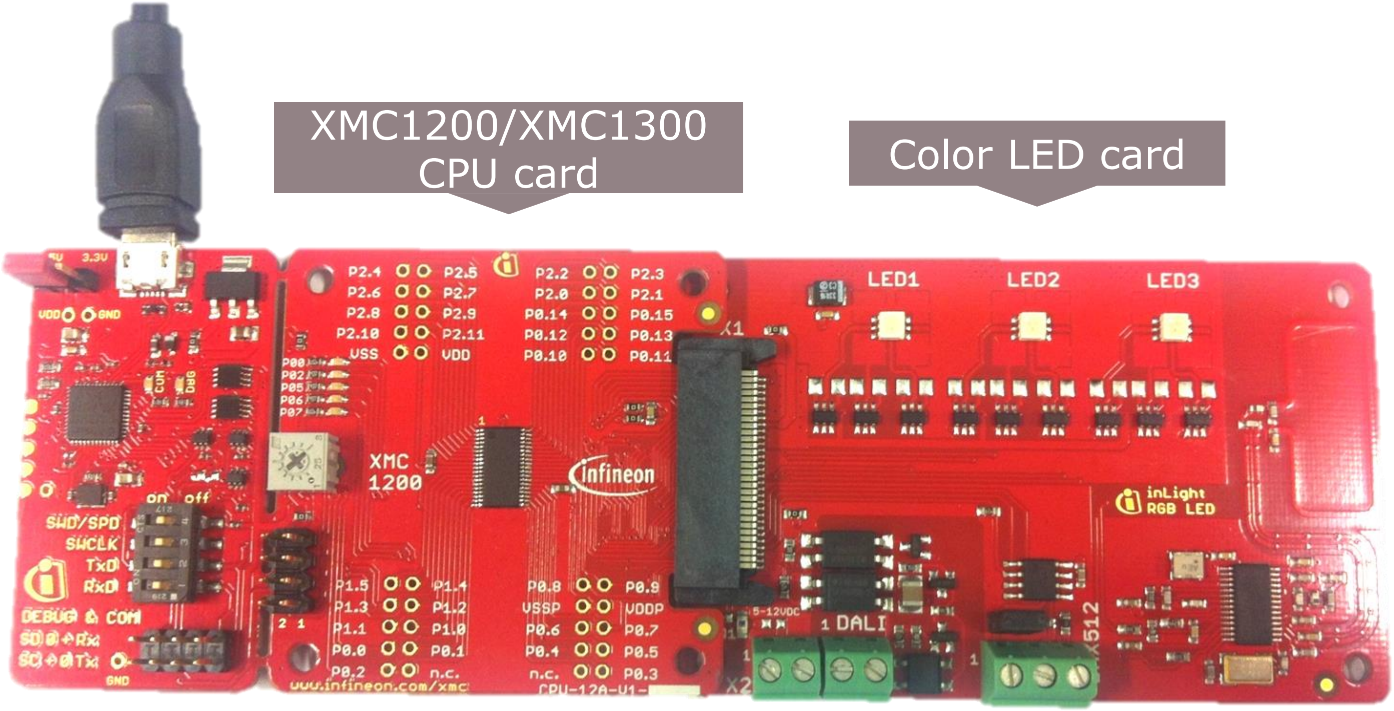
# Getting started – Example 1
# RGB lamp using LED_LAMP APP (1/20)

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)

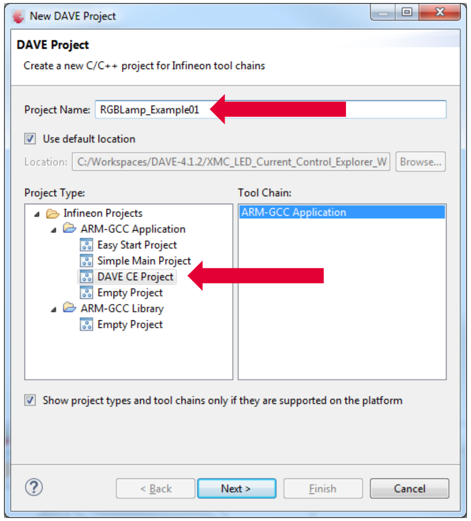
1. Open DAVE™

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



New DAVE Project

**DAVE Project**

Create a new C/C++ project for Infineon tool chains

Project Name: RGBLamp_Example01

☑ Use default location

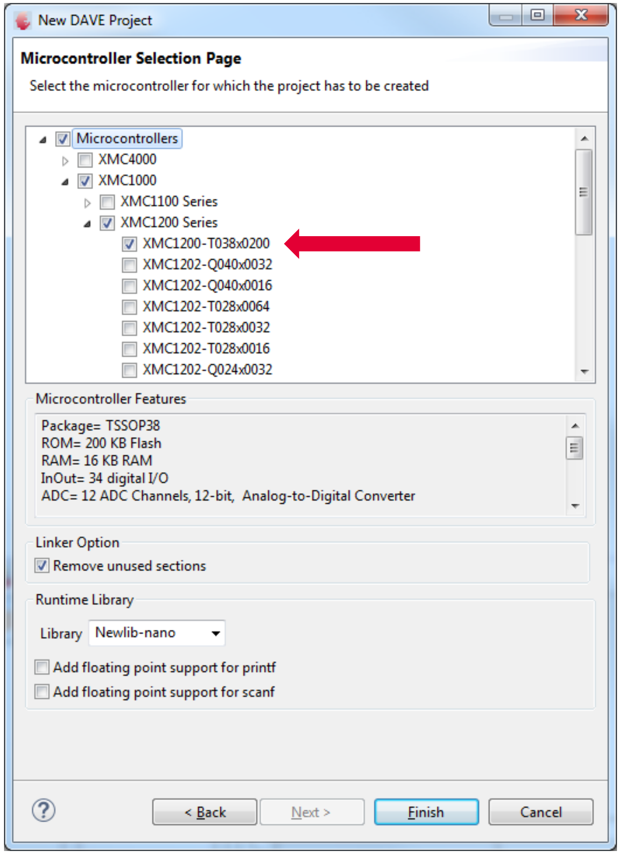
Location: C:/Workspaces/DAVE-4.1.2/XMC_LED_Current_Control_Explorer_W Browse...

Project Type:
- Infineon Projects
  - ARM-GCC Application
    - Easy Start Project
    - Simple Main Project
    - DAVE CE Project
    - Empty Project
  - ARM-GCC Library
    - Empty Project

Tool Chain: ARM-GCC Application

☑ Show project types and tool chains only if they are supported on the platform

3. Select the device accordingly



New DAVE Project

**Microcontroller Selection Page**

Select the microcontroller for which the project has to be created

- ☑ Microcontrollers
  - □ XMC4000
  - ☑ XMC1000
    - □ XMC1100 Series
    - ☑ XMC1200 Series
      - ☑ XMC1200-T038x0200
      - □ XMC1202-Q040x0032
      - □ XMC1202-Q040x0016
      - □ XMC1202-T028x0064
      - □ XMC1202-T028x0032
      - □ XMC1202-T028x0016
      - □ XMC1202-Q024x0032

Microcontroller Features

Package= TSSOP38
ROM= 200 KB Flash
RAM= 16 KB RAM
InOut= 34 digital I/O
ADC= 12 ADC Channels, 12-bit, Analog-to-Digital Converter

Linker Option

☑ Remove unused sections

Runtime Library

Library: Newlib-nano

□ Add floating point support for printf

□ Add floating point support for scanf

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (3/20)

- 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

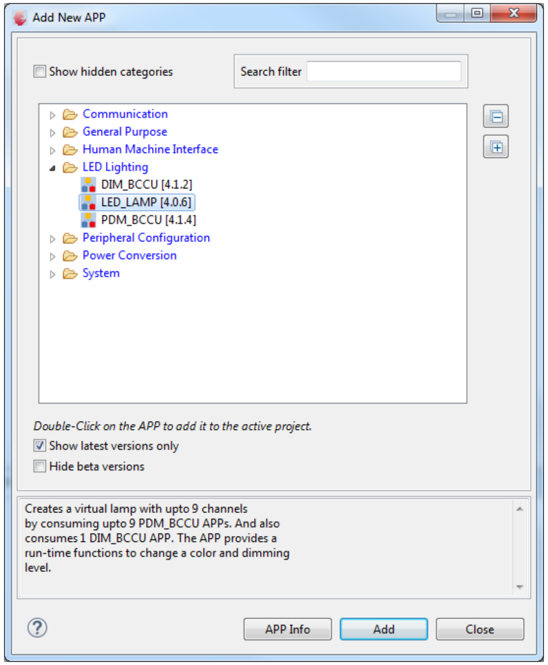
# Getting started – Example 1
# RGB lamp using LED_LAMP APP (4/20)

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**)



Add New APP

□ Show hidden categories    Search filter

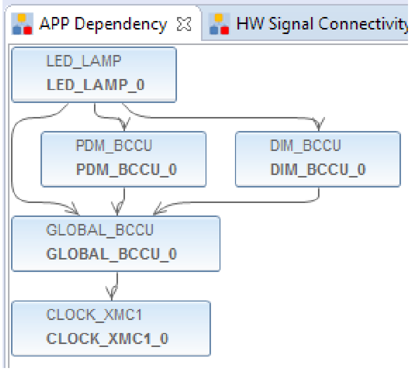
- Communication
- General Purpose
- Human Machine Interface
- LED Lighting
  - DIM_BCCU [4.1.2]
  - LED_LAMP [4.0.6]
  - PDM_BCCU [4.1.4]
- Peripheral Configuration
- Power Conversion
- System

*Double-Click on the APP to add it to the active project.*

☑ Show latest versions only

□ Hide beta versions

Creates a virtual lamp with upto 9 channels by consuming upto 9 PDM_BCCU APPs. And also consumes 1 DIM_BCCU APP. The APP provides a run-time functions to change a color and dimming level.



# Getting started – Example 1
# RGB lamp using LED_LAMP APP (5/20)

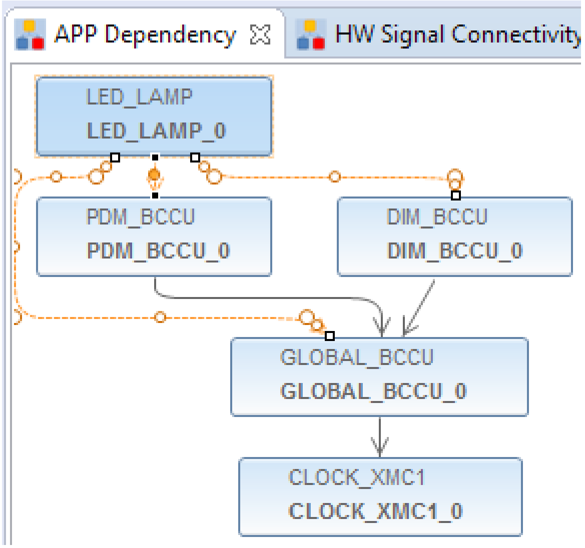
2. Configure LED_LAMP APP

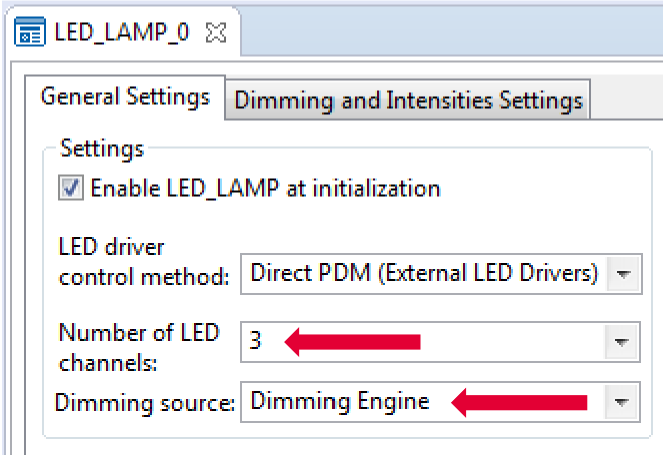
› Double-click **LED_LAMP_0** to open UI

› Under **General Settings** tab,

  – set **Number of LED channels** to **3**

  – select **Dimming Engine** as **Dimming Source**





Settings

☑ Enable LED_LAMP at initialization

LED driver control method: Direct PDM (External LED Drivers)

Number of LED channels: 3

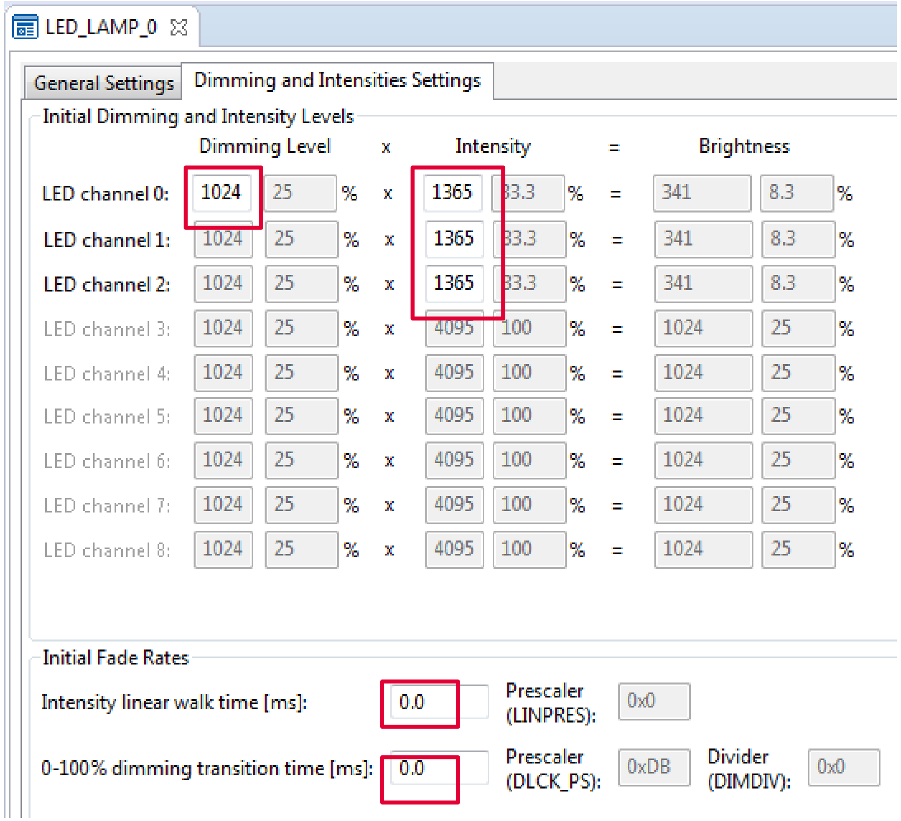
Dimming source: Dimming Engine

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (6/20)

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**



LED_LAMP_0

General Settings | Dimming and Intensities Settings

Initial Dimming and Intensity Levels

| | Dimming Level | | | x | Intensity | | | = | Brightness | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LED channel 0: | 1024 | 25 | % | x | 1365 | 33.3 | % | = | 341 | 8.3 | % |
| LED channel 1: | 1024 | 25 | % | x | 1365 | 33.3 | % | = | 341 | 8.3 | % |
| LED channel 2: | 1024 | 25 | % | x | 1365 | 33.3 | % | = | 341 | 8.3 | % |
| LED channel 3: | 1024 | 25 | % | x | 4095 | 100 | % | = | 1024 | 25 | % |
| LED channel 4: | 1024 | 25 | % | x | 4095 | 100 | % | = | 1024 | 25 | % |
| LED channel 5: | 1024 | 25 | % | x | 4095 | 100 | % | = | 1024 | 25 | % |
| LED channel 6: | 1024 | 25 | % | x | 4095 | 100 | % | = | 1024 | 25 | % |
| LED channel 7: | 1024 | 25 | % | x | 4095 | 100 | % | = | 1024 | 25 | % |
| LED channel 8: | 1024 | 25 | % | x | 4095 | 100 | % | = | 1024 | 25 | % |

Initial Fade Rates

Intensity linear walk time [ms]: 0.0 Prescaler (LINPRES): 0x0

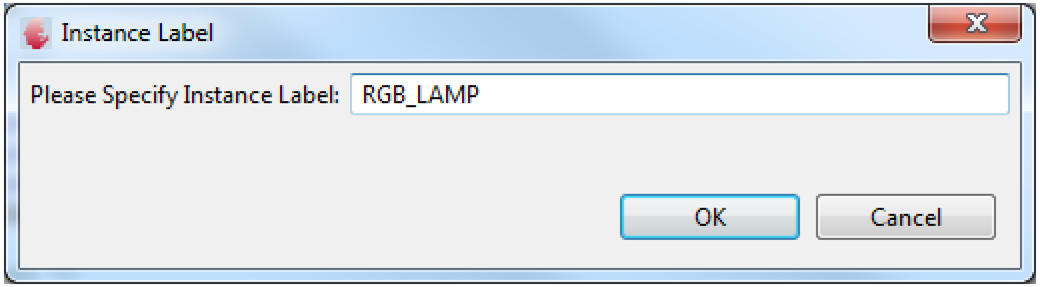
0-100% dimming transition time [ms]: 0.0 Prescaler (DLCK_PS): 0xDB Divider (DIMDIV): 0x0

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (7/20)

2. Configure LED_LAMP APP (continued)

› Rename Instance Label

- **Right-click** LED_LAMP APP
- Select **Rename Instance Label…**
- Rename as **RGB_LAMP**

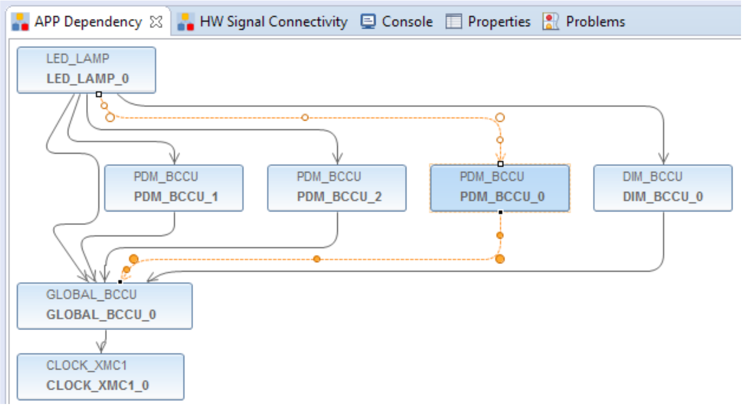


Please Specify Instance Label: RGB_LAMP

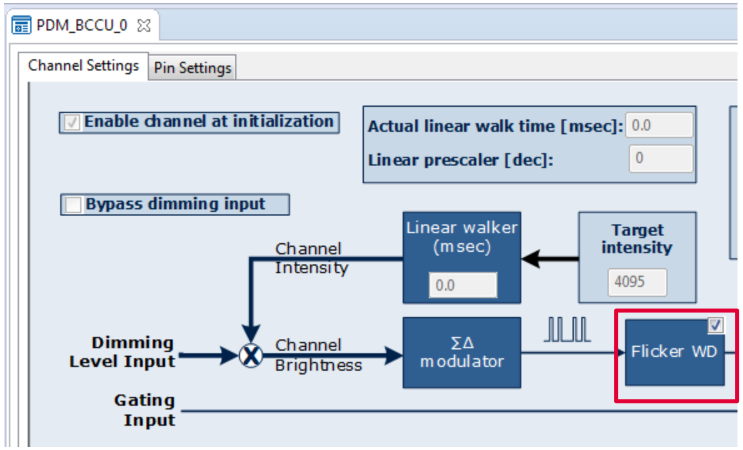
# Getting started – Example 1
# RGB lamp using LED_LAMP APP (8/20)

3. Configure BCCU Channels

- Double-click a **PDM_BCCU APP**



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



☑ Enable channel at initialization

Actual linear walk time [msec]: 0.0

Linear prescaler [dec]: 0

□ Bypass dimming input

Linear walker (msec): 0.0

Target intensity: 4095

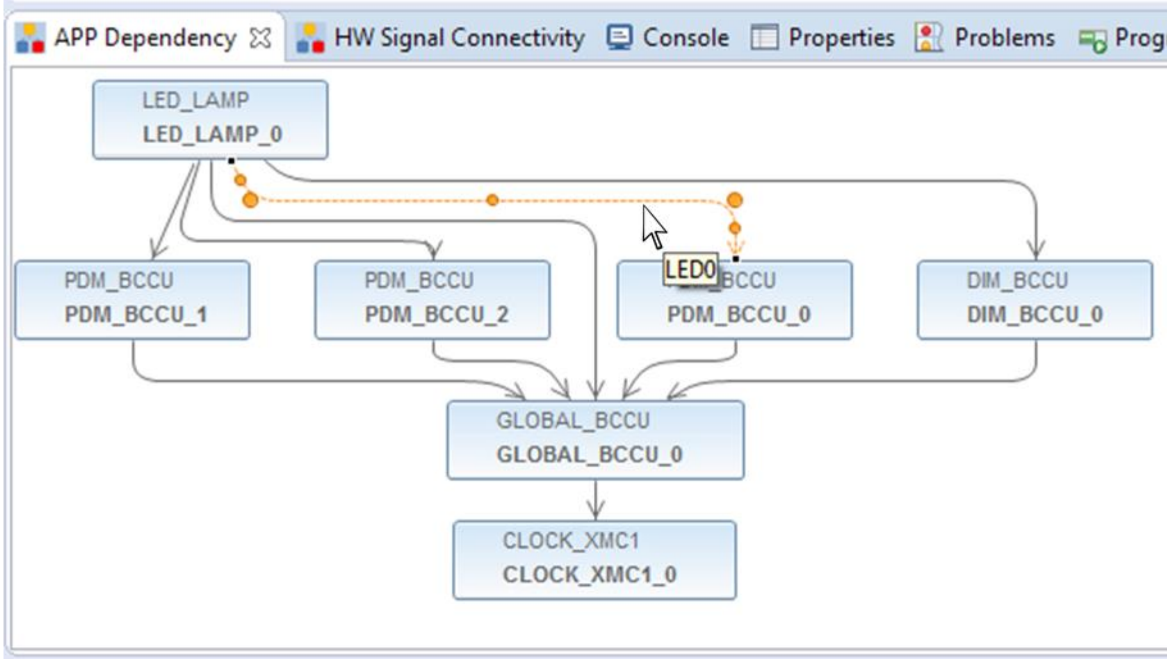
☑ Flicker WD

- Repeat for the other 2 PDM_BCCU APP instances

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (9/20)

4. Assign PDM_BCCU APPs to the right channels

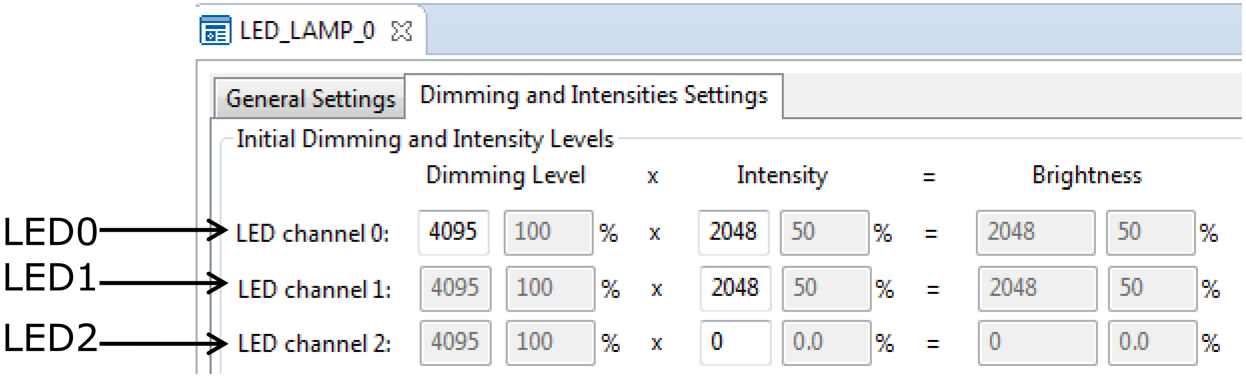
- Hover mouse cursor over the connecting arrow to a PDM_BCCU APP
- A label will appear momentarily e.g. LED0/LED1/LED2



# Getting started – Example 1
# RGB lamp using LED_LAMP APP (10/20)

## 4. Assign PDM_BCCU APPs to the right channels (continued)

› The labels correspond to the LED channels in the UI



LED_LAMP_0

General Settings | Dimming and Intensities Settings

Initial Dimming and Intensity Levels

| | Dimming Level | | | x | Intensity | | | = | Brightness | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LED channel 0: | 4095 | 100 | % | x | 2048 | 50 | % | = | 2048 | 50 | % |
| LED channel 1: | 4095 | 100 | % | x | 2048 | 50 | % | = | 2048 | 50 | % |
| LED channel 2: | 4095 | 100 | % | x | 0 | 0.0 | % | = | 0 | 0.0 | % |

› Rename the PDM_BCCU instance label according to the table below

– Right-click PDM_BCCU APP

– Select “Rename Instance Label”

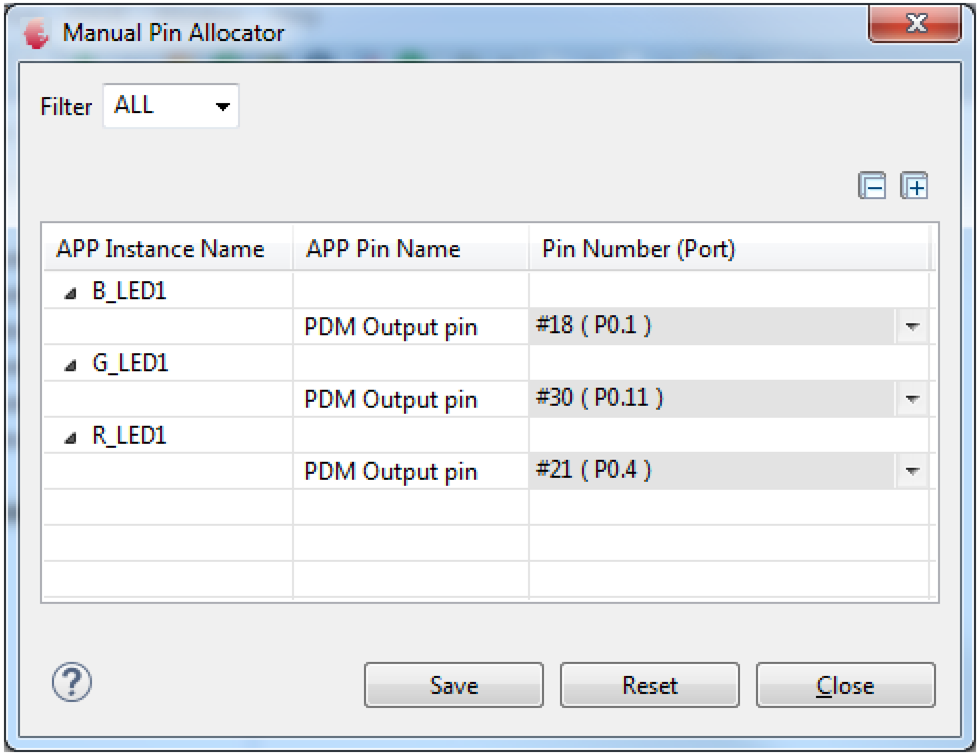
| Label | New Label |
|---|---|
| LED0 | R_LED1 |
| LED1 | G_LED1 |
| LED2 | B_LED1 |

– 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 T to assign pins to PDM_BCCU APPs
- Assign pins as shown:



Manual Pin Allocator

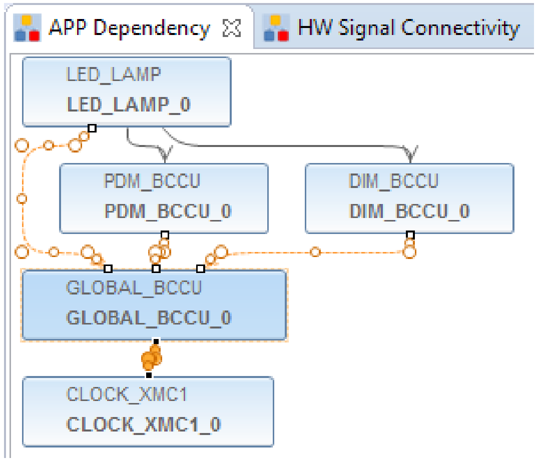
Filter: ALL

| APP Instance Name | APP Pin Name | Pin Number (Port) |
|---|---|---|
| B_LED1 | | |
| | PDM Output pin | #18 ( P0.1 ) |
| G_LED1 | | |
| | PDM Output pin | #30 ( P0.11 ) |
| R_LED1 | | |
| | PDM Output pin | #21 ( P0.4 ) |

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (12/20)

5. Configure BCCU global settings

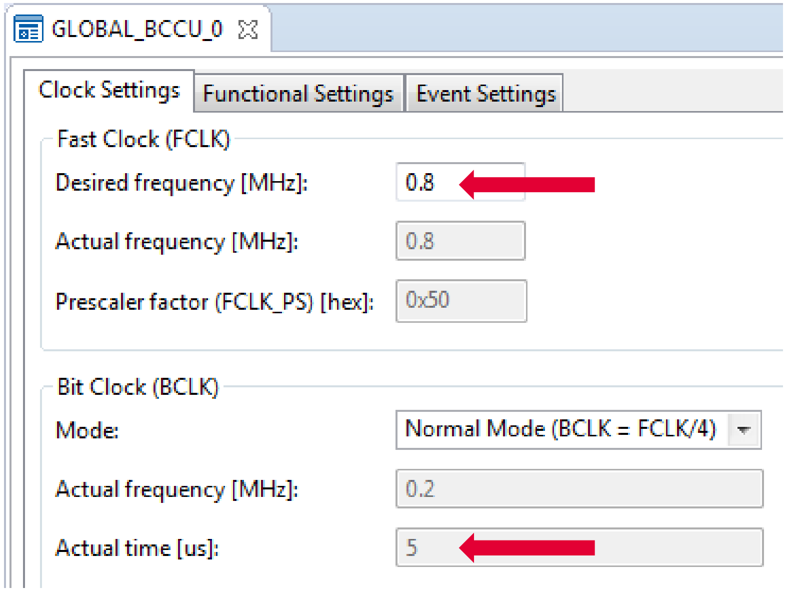
› Double-click **GLOBAL_BCCU_0** in **APP Dependency tab**



› Under **Clock Settings** tab,

– to get a bit time of 5 us

– change the **Desired Fast Clock Frequency** to **0.8 MHz**



Fast Clock (FCLK)

Desired frequency [MHz]: 0.8

Actual frequency [MHz]: 0.8

Prescaler factor (FCLK_PS) [hex]: 0x50

Bit Clock (BCLK)

Mode: Normal Mode (BCLK = FCLK/4)

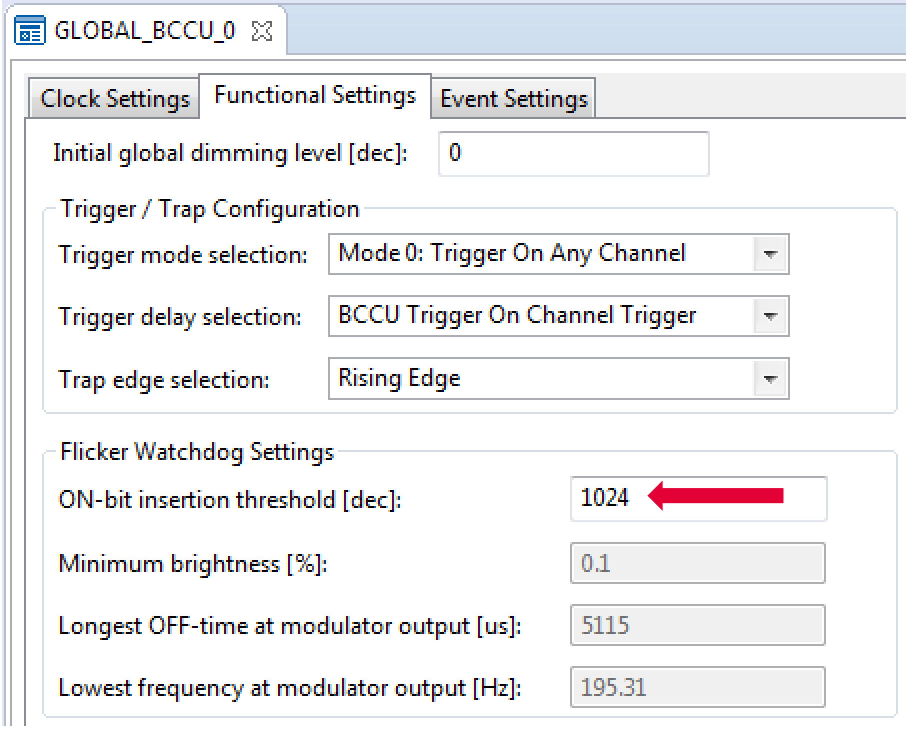
Actual frequency [MHz]: 0.2

Actual time [us]: 5

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (13/20)

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**



Initial global dimming level [dec]: 0

Trigger / Trap Configuration

Trigger mode selection: Mode 0: Trigger On Any Channel

Trigger delay selection: BCCU Trigger On Channel Trigger

Trap edge selection: Rising Edge

Flicker Watchdog Settings

ON-bit insertion threshold [dec]: 1024

Minimum brightness [%]: 0.1

Longest OFF-time at modulator output [us]: 5115

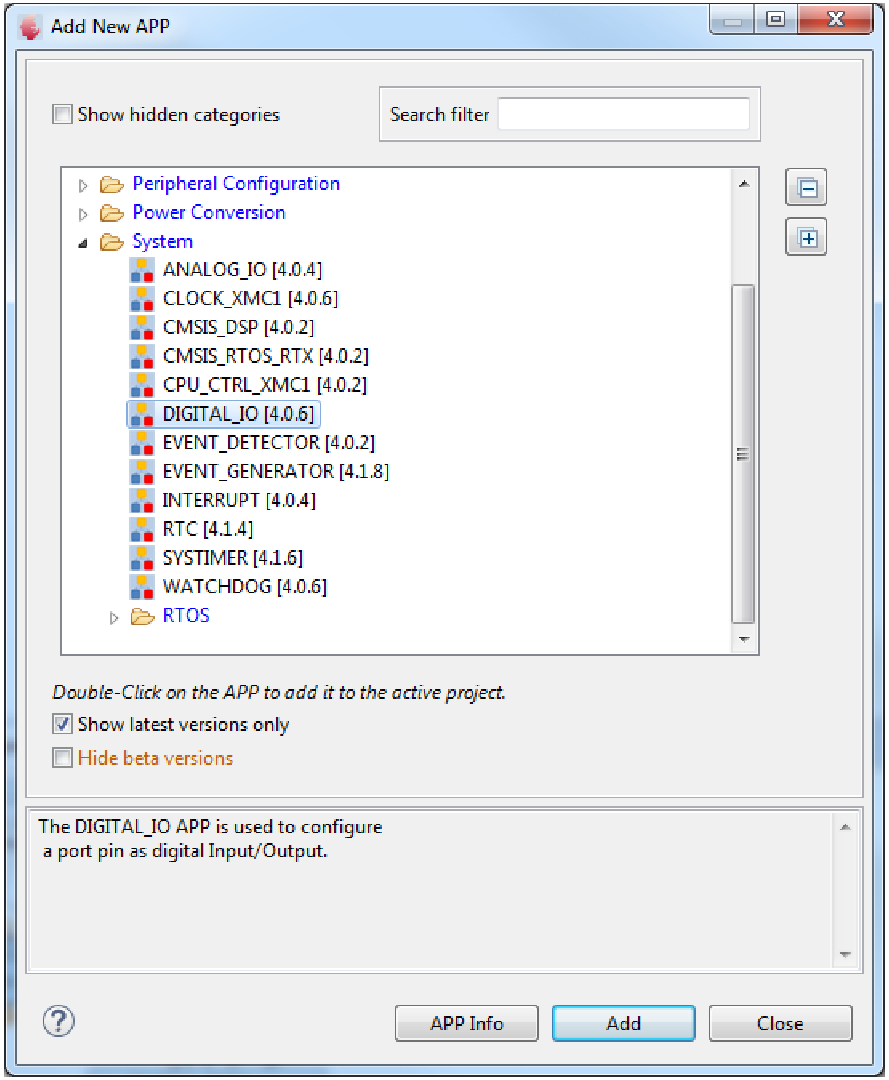
Lowest frequency at modulator output [Hz]: 195.31

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (14/20)

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



Add New APP

□ Show hidden categories

Search filter

- Peripheral Configuration
- Power Conversion
- System
  - ANALOG_IO [4.0.4]
  - CLOCK_XMC1 [4.0.6]
  - CMSIS_DSP [4.0.2]
  - CMSIS_RTOS_RTX [4.0.2]
  - CPU_CTRL_XMC1 [4.0.2]
  - DIGITAL_IO [4.0.6]
  - EVENT_DETECTOR [4.0.2]
  - EVENT_GENERATOR [4.1.8]
  - INTERRUPT [4.0.4]
  - RTC [4.1.4]
  - SYSTIMER [4.1.6]
  - WATCHDOG [4.0.6]
  - RTOS

*Double-Click on the APP to add it to the active project.*

☑ Show latest versions only

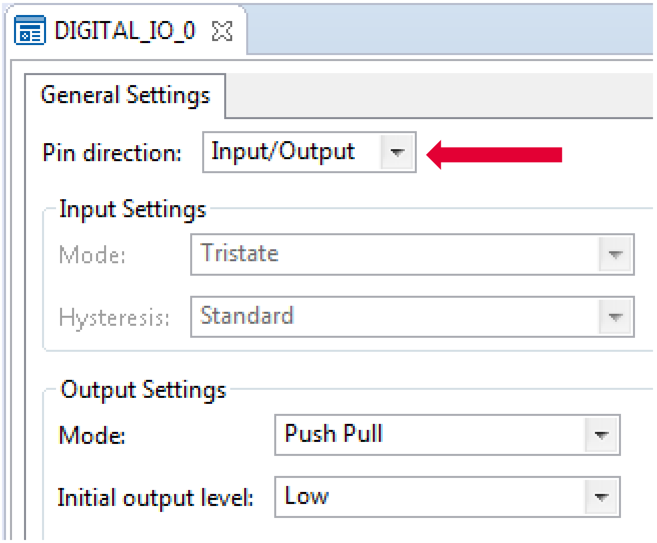
□ Hide beta versions

The DIGITAL_IO APP is used to configure a port pin as digital Input/Output.

# 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**



Pin direction: Input/Output

Input Settings

Mode: Tristate

Hysteresis: Standard

Output Settings

Mode: Push Pull

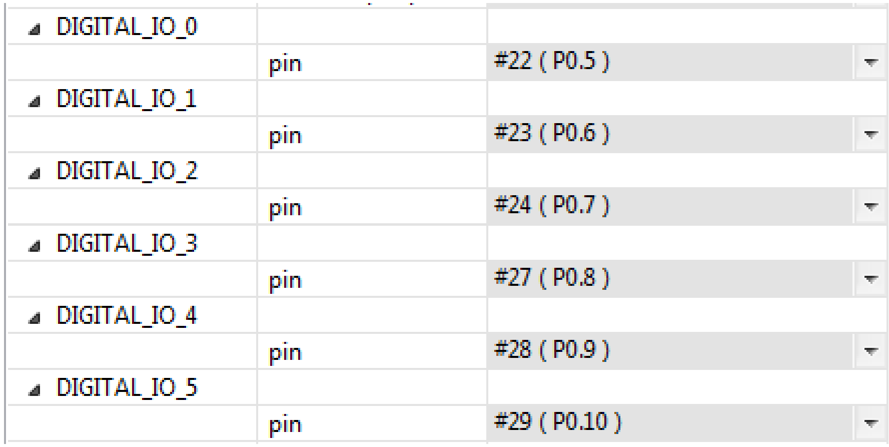
Initial output level: Low

- 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:



| | | |
|---|---|---|
| DIGITAL_IO_0 | | |
| | pin | #22 ( P0.5 ) |
| DIGITAL_IO_1 | | |
| | pin | #23 ( P0.6 ) |
| DIGITAL_IO_2 | | |
| | pin | #24 ( P0.7 ) |
| DIGITAL_IO_3 | | |
| | pin | #27 ( P0.8 ) |
| DIGITAL_IO_4 | | |
| | pin | #28 ( P0.9 ) |
| DIGITAL_IO_5 | | |
| | pin | #29 ( P0.10 ) |

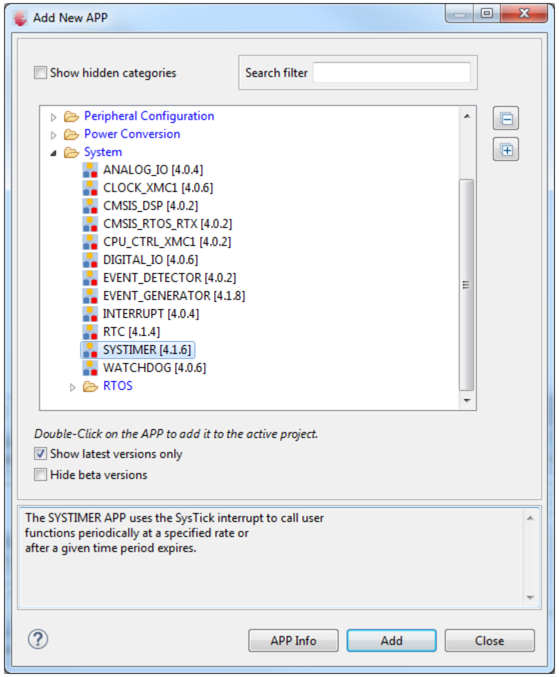
# Getting started – Example 1
# RGB lamp using LED_LAMP APP (17/20)

## 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 📝 to generate code



Add New APP

□ Show hidden categories    Search filter

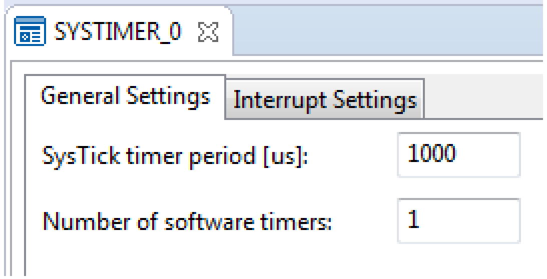
- Peripheral Configuration
- Power Conversion
- System
  - ANALOG_IO [4.0.4]
  - CLOCK_XMC1 [4.0.6]
  - CMSIS_DSP [4.0.2]
  - CMSIS_RTOS_RTX [4.0.2]
  - CPU_CTRL_XMC1 [4.0.2]
  - DIGITAL_IO [4.0.6]
  - EVENT_DETECTOR [4.0.2]
  - EVENT_GENERATOR [4.1.8]
  - INTERRUPT [4.0.4]
  - RTC [4.1.4]
  - SYSTIMER [4.1.6]
  - WATCHDOG [4.0.6]
  - RTOS

*Double-Click on the APP to add it to the active project.*

☑ Show latest versions only

□ Hide beta versions

The SYSTIMER APP uses the SysTick interrupt to call user functions periodically at a specified rate or after a given time period expires.



SYSTIMER_0

General Settings | Interrupt Settings

SysTick timer period [us]: 1000

Number of software timers: 1

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (18/20)

8. Define SYSTIMER callback function

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

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

- Create software timer and start timer

```
uint32_t timer_id;
TimerId = SYSTIMER_CreateTimer(1000000,SYSTIMER_MODE_PERIODIC,OneSecTick,NULL);
SYSTIMER_StartTimer(TimerId);
```

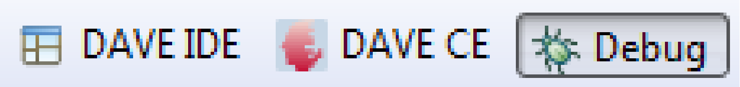
# Getting started – Example 1
# RGB lamp using LED_LAMP APP (19/20)

## 8. Define SYSTIMER callback function (continued)

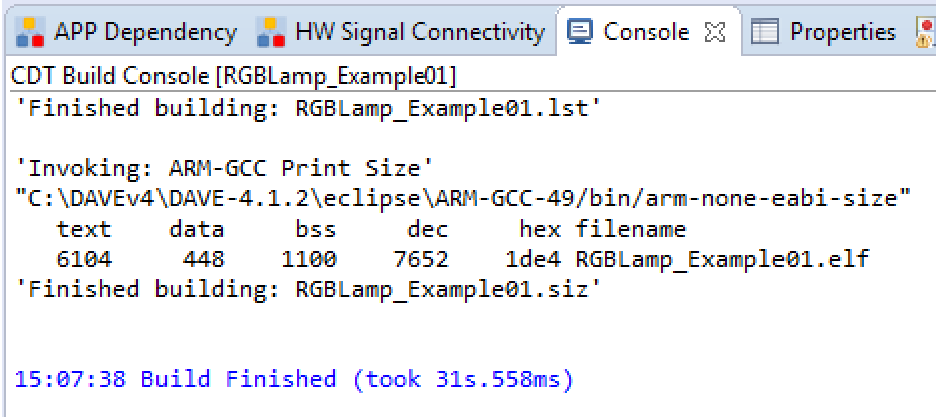
```c
void OneSecTick(void)
{
  static uint8_t step = 0;
  if (++step==1) {// change color to red
    RGB_LAMP_config.led_intensity[0] = 4095;
    RGB_LAMP_config.led_intensity[1] = 0;
    RGB_LAMP_config.led_intensity[2] = 0;
    LED_LAMP_SetColorAdv(&RGB_LAMP, 0x2AC);
  }
  else if (step==9) {// change 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;
  }
}
```

# Getting started – Example 1
# RGB lamp using LED_LAMP APP (20/20)

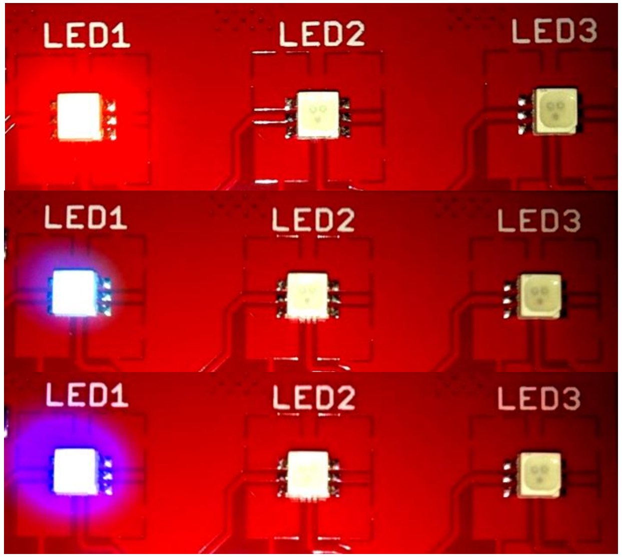
- Build project
  1. Click
  2. Wait for Build to finish
- Download code
  1. Click
  2. Switch to Debug view



  3. Click to run code
- LED1 regularly changes color and brightness

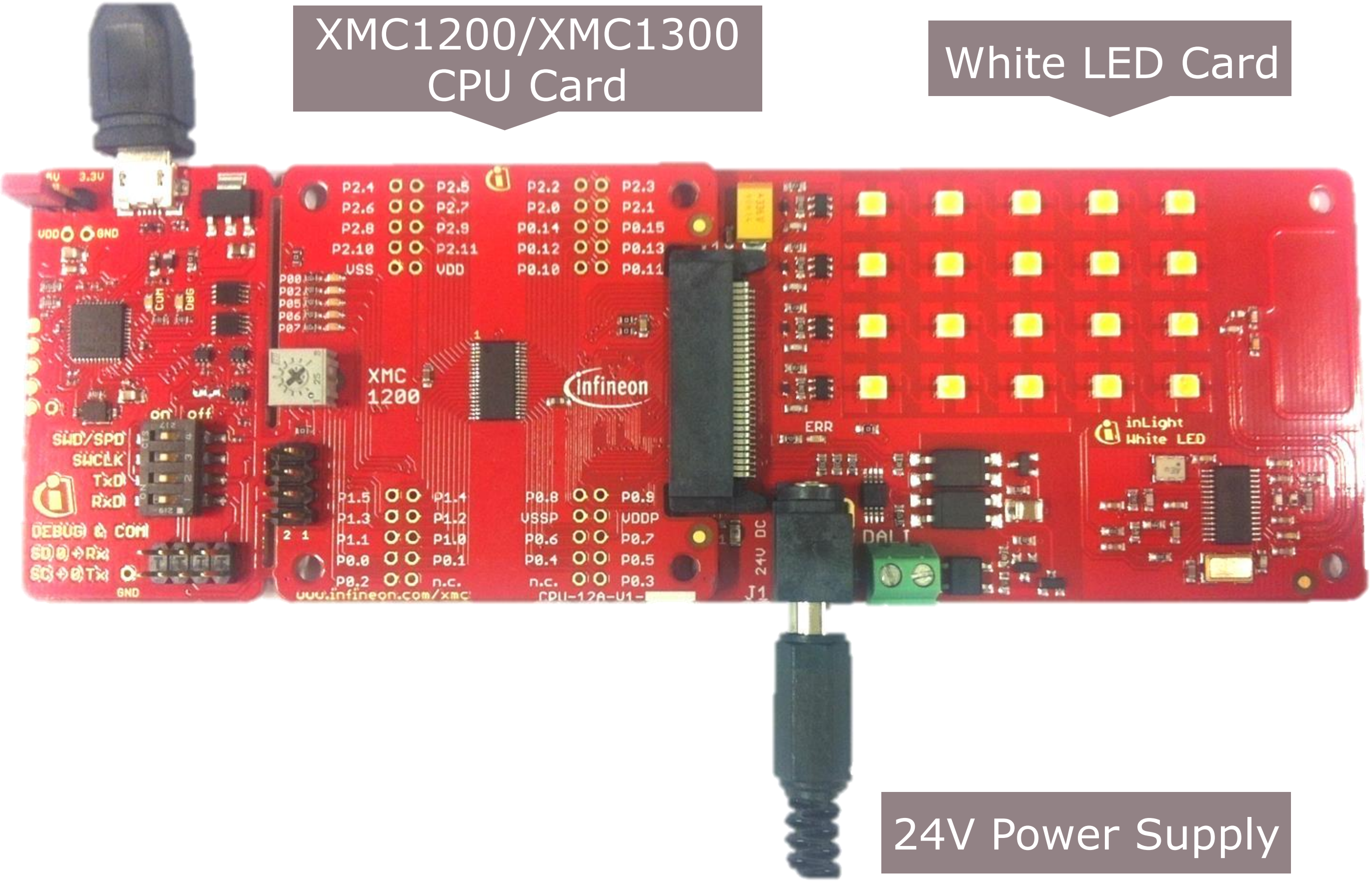


```
CDT Build Console [RGBLamp_Example01]
'Finished building: RGBLamp_Example01.lst'

'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
   6104     448    1100    7652    1de4 RGBLamp_Example01.elf
'Finished building: RGBLamp_Example01.siz'


15:07:38 Build Finished (took 31s.558ms)
```



# Getting started – Example 2
# White lamp using LED_LAMP APP (1/19)

## Example 2: White Lamp using LED_LAMP APP



Connect to PC

XMC1200/XMC1300
CPU Card

White LED Card

24V Power Supply

# Getting started – Example 2
# White lamp using LED_LAMP APP (2/19)

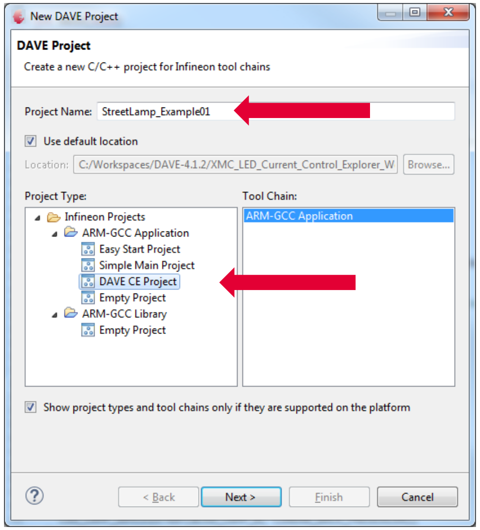
1. Open DAVE™

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



New DAVE Project

**DAVE Project**

Create a new C/C++ project for Infineon tool chains

Project Name: StreetLamp_Example01

☑ Use default location

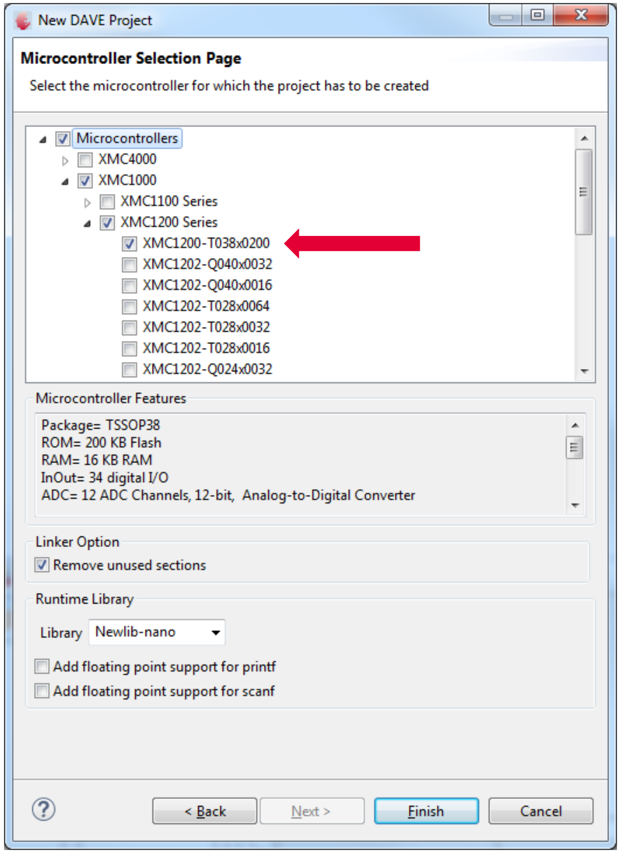
Location: C:/Workspaces/DAVE-4.1.2/XMC_LED_Current_Control_Explorer_W

Project Type:
- Infineon Projects
  - ARM-GCC Application
    - Easy Start Project
    - Simple Main Project
    - DAVE CE Project
    - Empty Project
  - ARM-GCC Library
    - Empty Project

Tool Chain: ARM-GCC Application

☑ Show project types and tool chains only if they are supported on the platform

3. Select the device accordingly



New DAVE Project

**Microcontroller Selection Page**

Select the microcontroller for which the project has to be created

- ☑ Microcontrollers
  - □ XMC4000
  - ☑ XMC1000
    - □ XMC1100 Series
    - ☑ XMC1200 Series
      - ☑ XMC1200-T038x0200
      - □ XMC1202-Q040x0032
      - □ XMC1202-Q040x0016
      - □ XMC1202-T028x0064
      - □ XMC1202-T028x0032
      - □ XMC1202-T028x0016
      - □ XMC1202-Q024x0032

Microcontroller Features

Package= TSSOP38
ROM= 200 KB Flash
RAM= 16 KB RAM
InOut= 34 digital I/O
ADC= 12 ADC Channels, 12-bit, Analog-to-Digital Converter

Linker Option

☑ Remove unused sections

Runtime Library

Library: Newlib-nano

□ Add floating point support for printf

□ Add floating point support for scanf

# Getting started – Example 2
# White lamp using LED_LAMP APP (3/19)

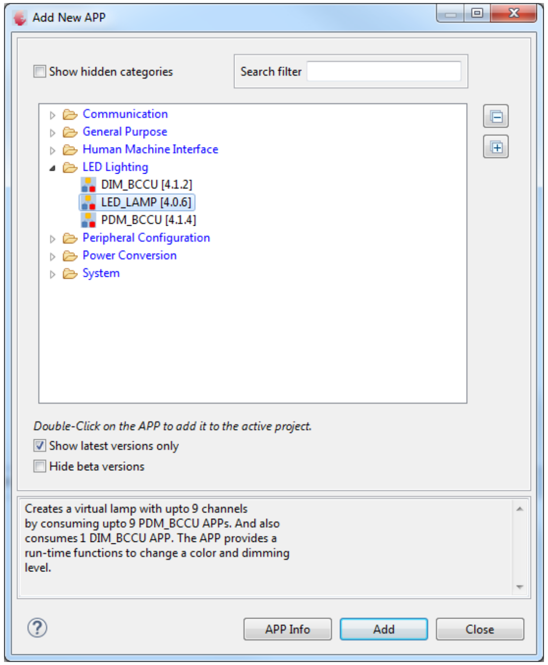
- This example demonstrates White Lamp functionality using LED_LAMP APP
- We will use the System Timer (SysTick) as the time base for the interrupt
  - Time base of 1 s
  - In the interrupt, a new target dimming level is set and the dimming process is started
- Next, we will show you the steps to creating 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

# Getting started – Example – White lamp using LED_LAMP APP (4/19)

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**)



☐ Show hidden categories

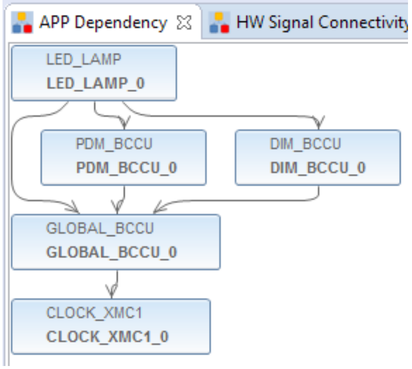
Search filter

*Double-Click on the APP to add it to the active project.*

☑ Show latest versions only

☐ Hide beta versions

Creates a virtual lamp with upto 9 channels by consuming upto 9 PDM_BCCU APPs. And also consumes 1 DIM_BCCU APP. The APP provides a run-time functions to change a color and dimming level.

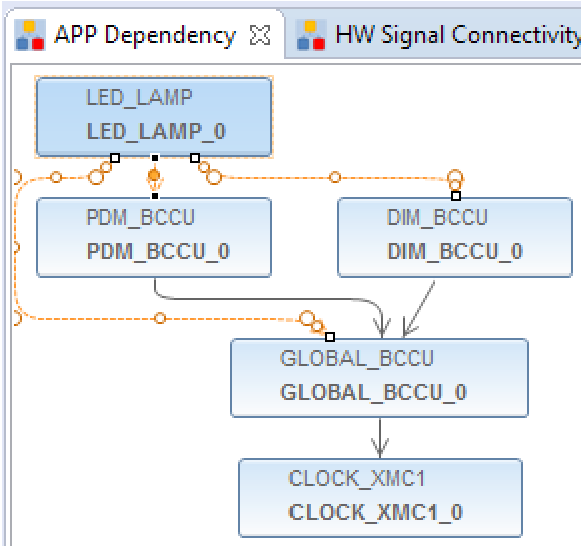


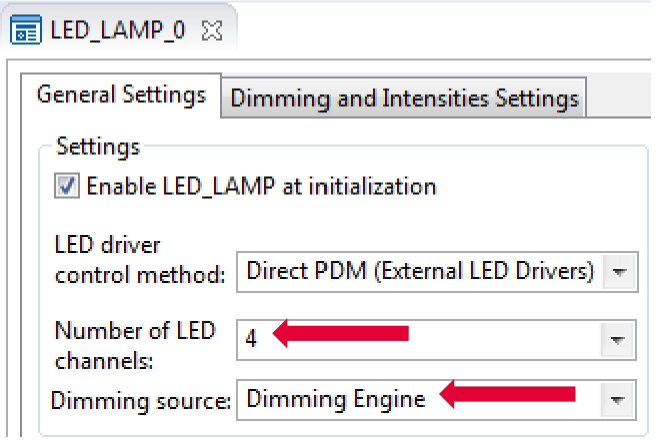
# Getting started – Example 2
# White lamp using LED_LAMP APP (5/19)

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**





☑ Enable LED_LAMP at initialization

LED driver control method: Direct PDM (External LED Drivers)

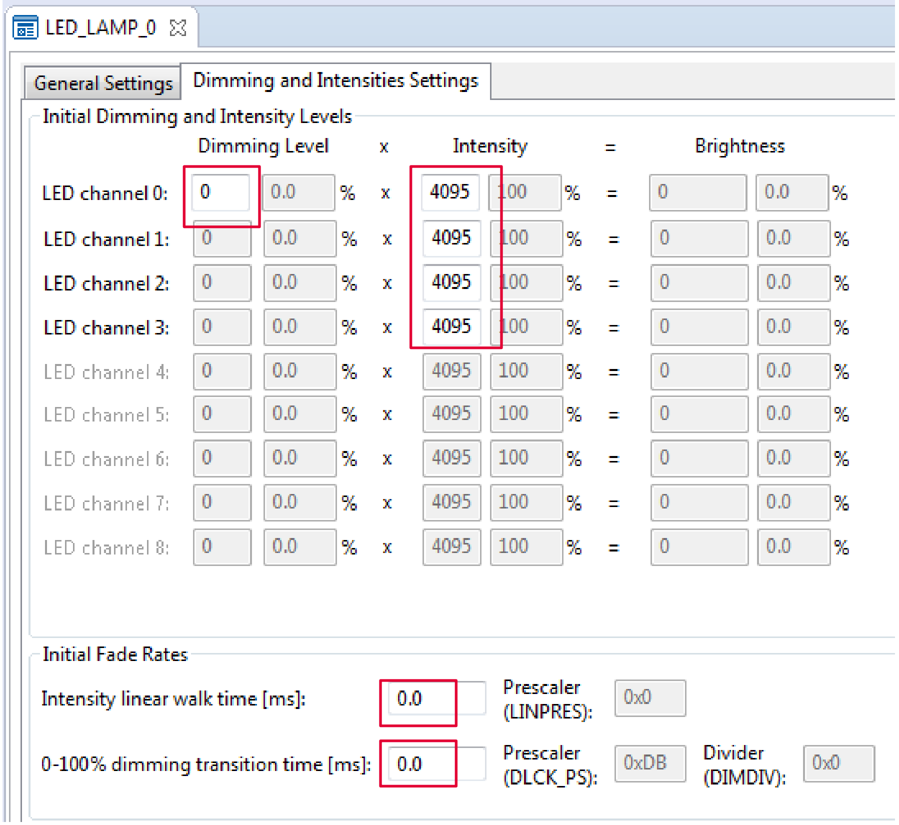
Number of LED channels: 4

Dimming source: Dimming Engine

# Getting started – Example 2
# White lamp using LED_LAMP APP (6/19)

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**



LED_LAMP_0

General Settings | Dimming and Intensities Settings

Initial Dimming and Intensity Levels

| | Dimming Level | | | x | Intensity | | | = | Brightness | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LED channel 0: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED channel 1: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED channel 2: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED channel 3: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED channel 4: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED channel 5: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED channel 6: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED channel 7: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED channel 8: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |

Initial Fade Rates

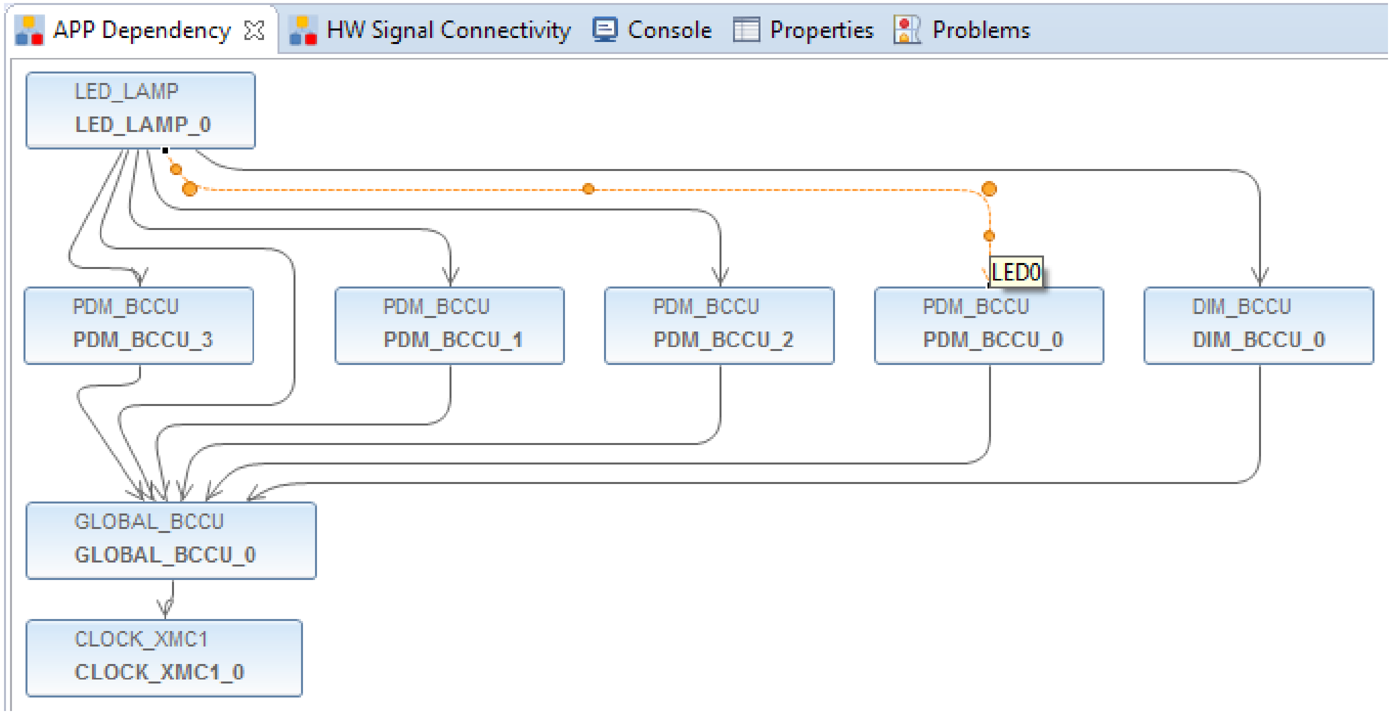
Intensity linear walk time [ms]: 0.0 Prescaler (LINPRES): 0x0

0-100% dimming transition time [ms]: 0.0 Prescaler (DLCK_PS): 0xDB Divider (DIMDIV): 0x0

# Getting started – Example 2
# White lamp using LED_LAMP APP (7/19)

3. Assign PDM_BCCU APPs to the right channels

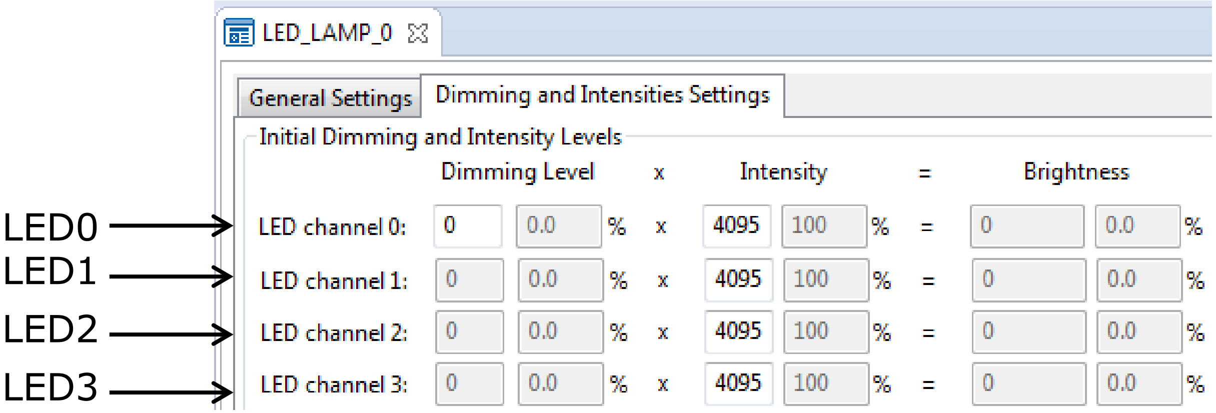
- Hover mouse cursor over the connecting arrow to a PDM_BCCU APP
- A label will appear momentarily e.g. LED0/LED1/LED2/LED3



# Getting started – Example 2
# White lamp using LED_LAMP APP (8/19)

3. Assign PDM_BCCU APPs to the right channels (continued)

- The labels correspond to the LED channels in the UI



Initial Dimming and Intensity Levels

| | Dimming Level | | | x | Intensity | | | = | Brightness | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LED0 → LED channel 0: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED1 → LED channel 1: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED2 → LED channel 2: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |
| LED3 → LED channel 3: | 0 | 0.0 | % | x | 4095 | 100 | % | = | 0 | 0.0 | % |

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

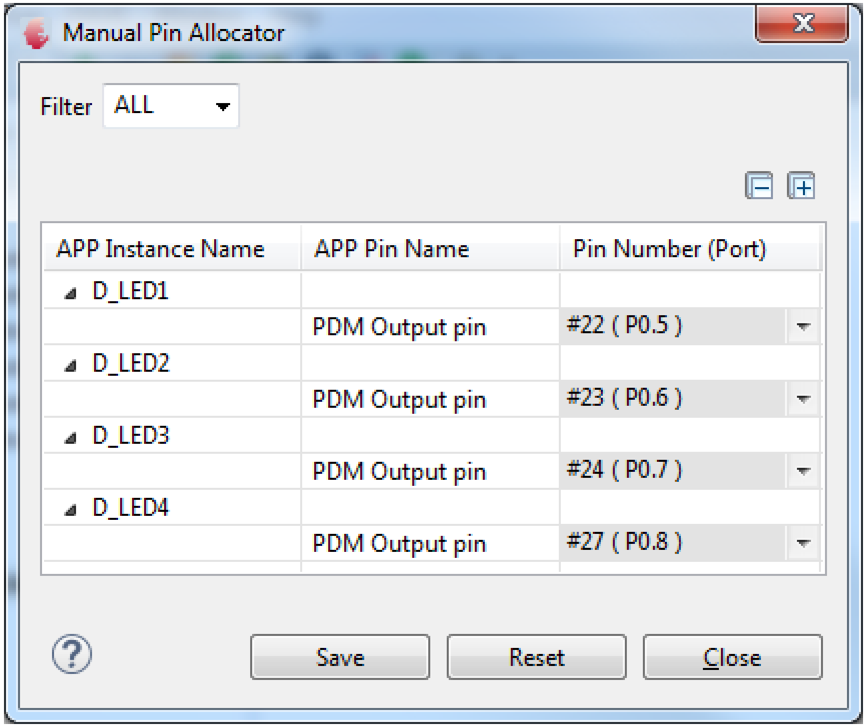
| Label | New Label |
|---|---|
| LED0 | D_LED1 |
| LED1 | D_LED2 |
| LED2 | D_LED3 |
| LED3 | D_LED4 |

  - Repeat the above steps with the other 2 PDM_BCCU APP instances

# Getting started – Example 2
# White Lamp using LED_LAMP APP (9/19)

3. Assign PDM_BCCU APPs to the right channels (continued)

- Click T to assign pins to PDM_BCCU APPs
- Assign pins as shown:



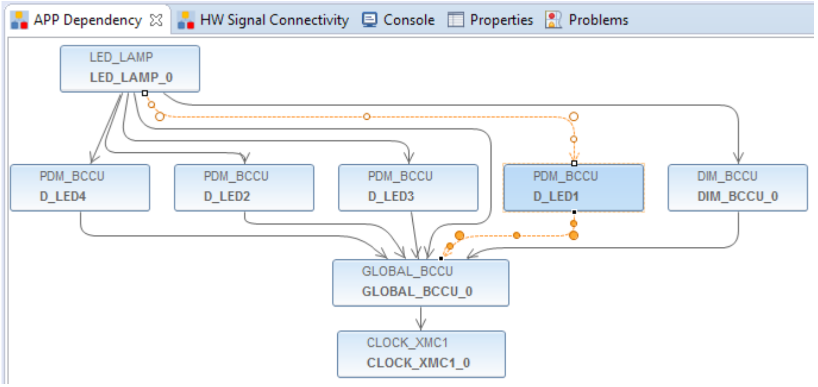
Filter: ALL

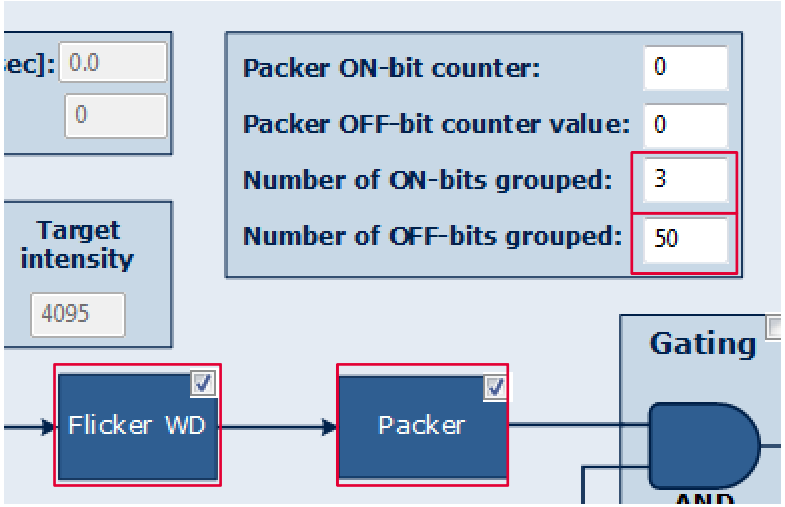
| APP Instance Name | APP Pin Name | Pin Number (Port) |
|---|---|---|
| D_LED1 | | |
| | PDM Output pin | #22 ( P0.5 ) |
| D_LED2 | | |
| | PDM Output pin | #23 ( P0.6 ) |
| D_LED3 | | |
| | PDM Output pin | #24 ( P0.7 ) |
| D_LED4 | | |
| | PDM Output pin | #27 ( P0.8 ) |

# Getting started – Example 2
# White Lamp using LED_LAMP APP (10/19)

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**




ec]: 0.0

0

Target intensity 4095

Packer ON-bit counter: 0

Packer OFF-bit counter value: 0

Number of ON-bits grouped: 3

Number of OFF-bits grouped: 50

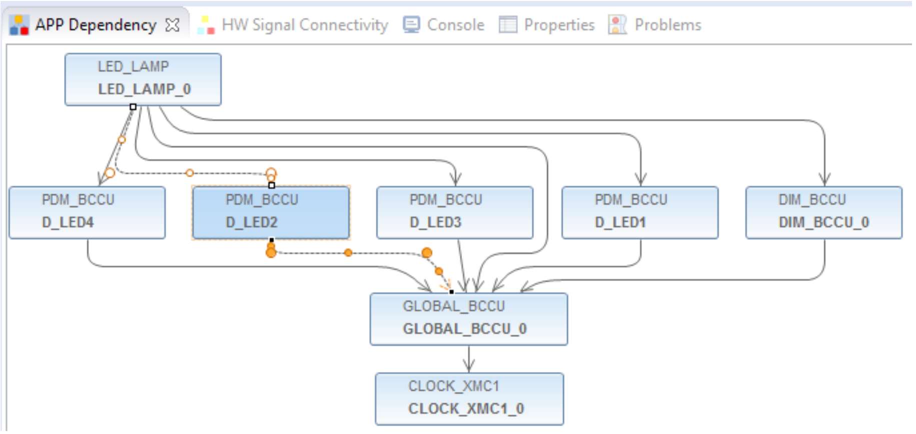
☑ Flicker WD

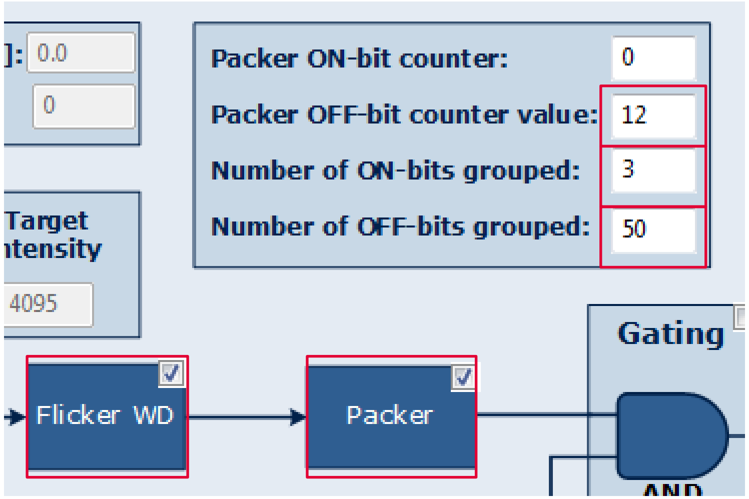
☑ Packer

# 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**





]: 0.0

0

Target ntensity

4095

Packer ON-bit counter: 0

Packer OFF-bit counter value: 12

Number of ON-bits grouped: 3

Number of OFF-bits grouped: 50

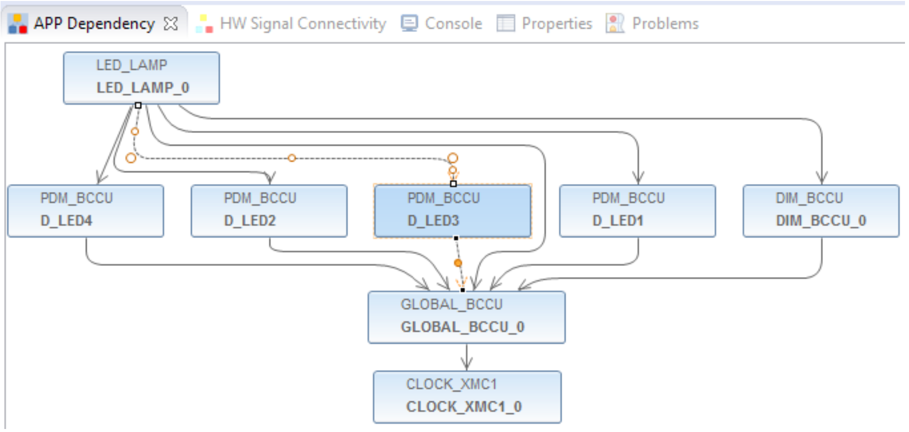
Flicker WD ☑

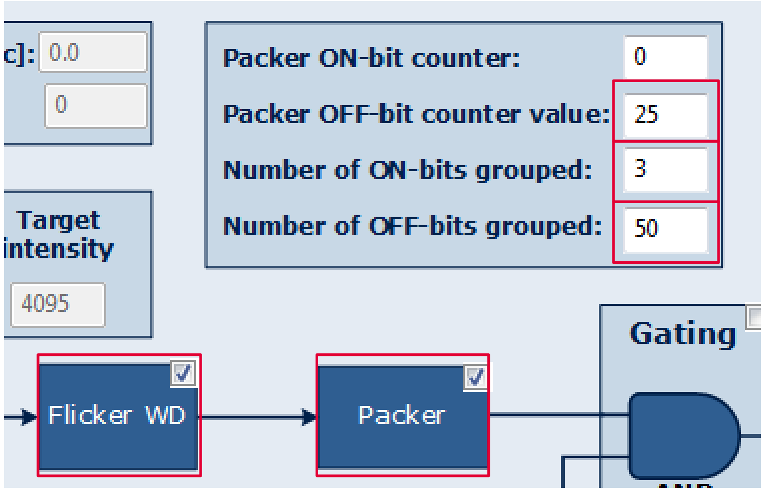
Packer ☑

# 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**





c]: 0.0

0

Packer ON-bit counter: 0

Packer OFF-bit counter value: 25

Number of ON-bits grouped: 3

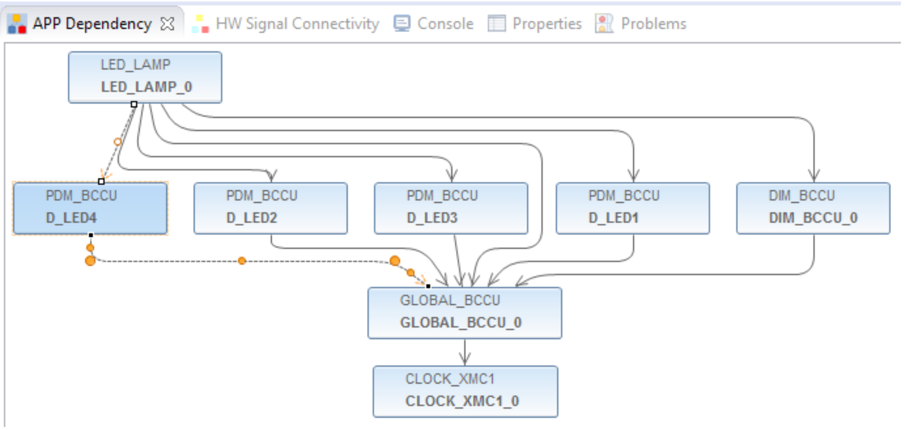
Number of OFF-bits grouped: 50

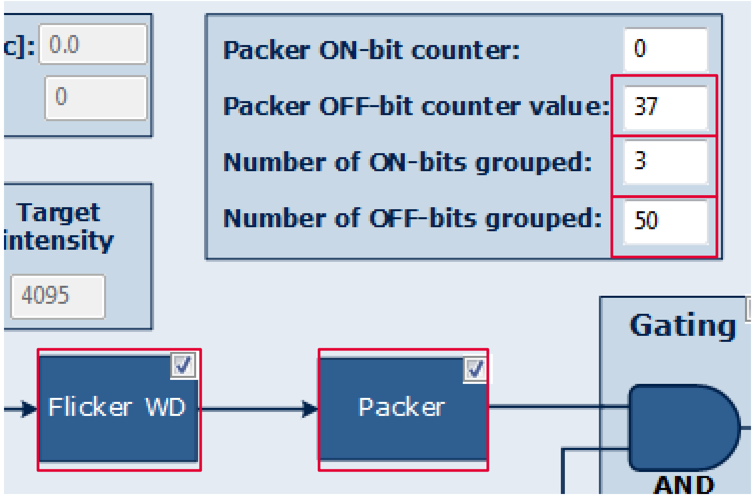
Target intensity 4095

# Getting started – Example 2
# White lamp using LED_LAMP APP (13/19)

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**




c]: 0.0
0

Packer ON-bit counter: 0
Packer OFF-bit counter value: 37
Number of ON-bits grouped: 3
Number of OFF-bits grouped: 50

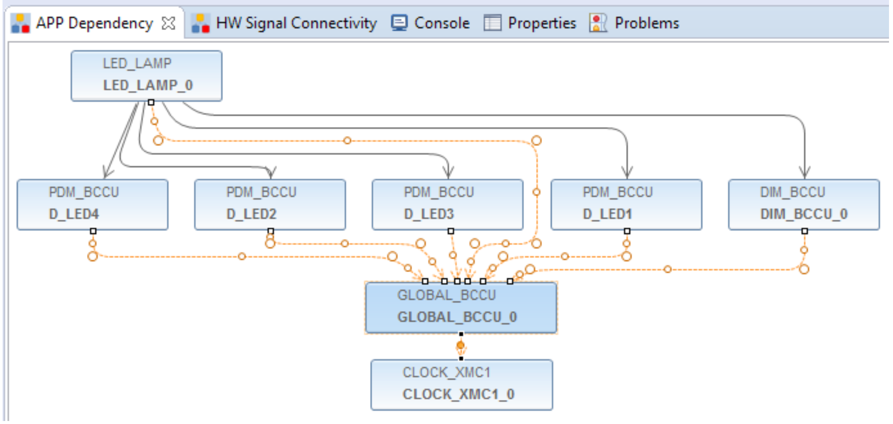
Target intensity
4095

☑ Flicker WD
☑ Packer

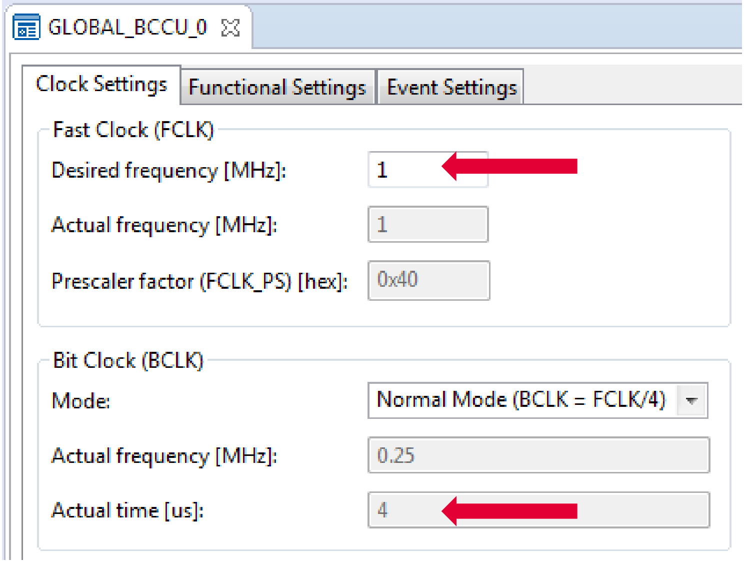
# Getting started – Example 2
# White lamp using LED_LAMP APP (14/19)

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**



Fast Clock (FCLK)

Desired frequency [MHz]: 1

Actual frequency [MHz]: 1

Prescaler factor (FCLK_PS) [hex]: 0x40

Bit Clock (BCLK)

Mode: Normal Mode (BCLK = FCLK/4)

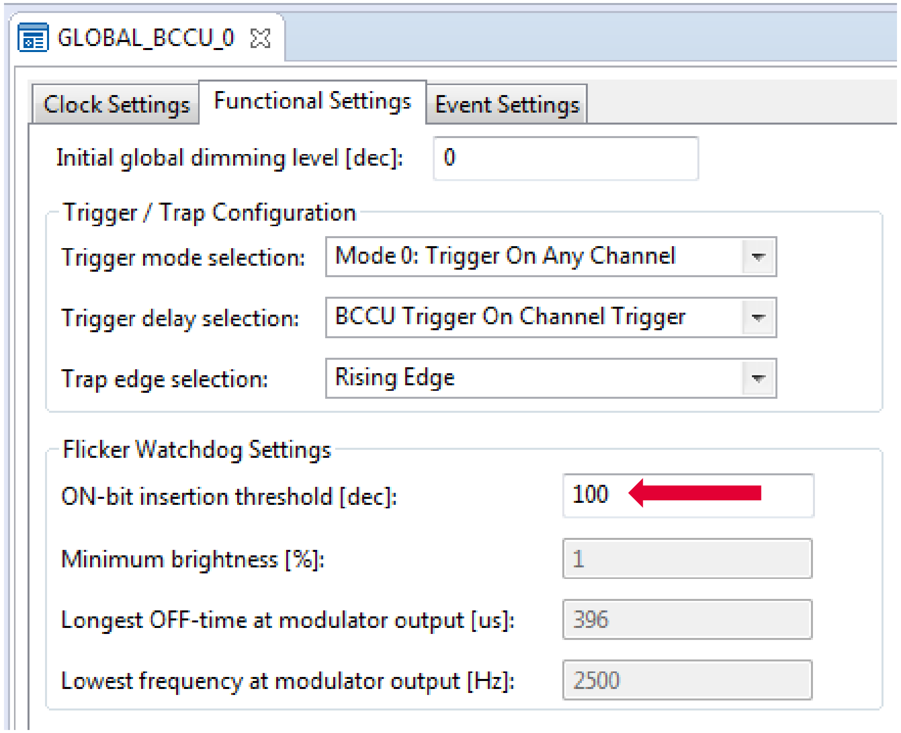
Actual frequency [MHz]: 0.25

Actual time [us]: 4

# Getting started – Example 2
# White lamp using LED_LAMP APP (15/19)

5. Configure BCCU global settings (continued)

- Under **Functional Settings** tab,
  - limit the minimum brightness to 1%
    - change **ON-bit insertion threshold** to **100**



Initial global dimming level [dec]: 0

Trigger / Trap Configuration

Trigger mode selection: Mode 0: Trigger On Any Channel

Trigger delay selection: BCCU Trigger On Channel Trigger

Trap edge selection: Rising Edge

Flicker Watchdog Settings

ON-bit insertion threshold [dec]: 100

Minimum brightness [%]: 1

Longest OFF-time at modulator output [us]: 396

Lowest frequency at modulator output [Hz]: 2500

# Getting started – Example 2
# White lamp using LED_LAMP APP (16/19)

6. Configure SysTick

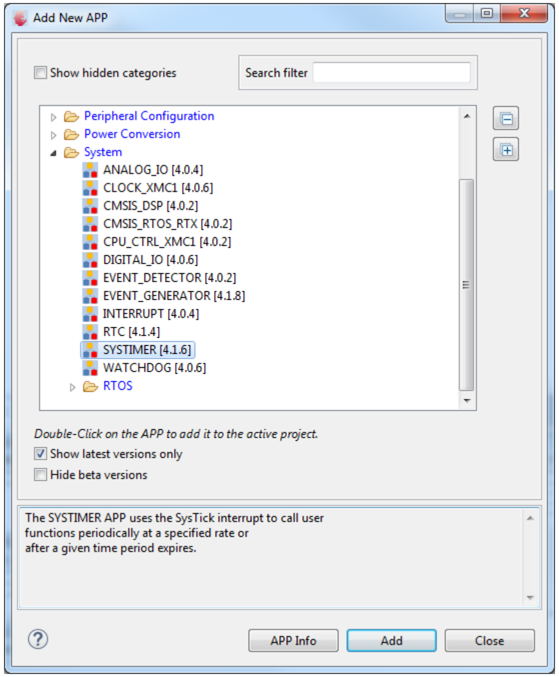
› Add **SYSTIMER** to the project

› Double-click **SYSTIMER** APP to open UI

  – Set **SysTick timer period** to **1000 us**

  – Set **Number of software timers** to **1**

› Click to generate code



Add New APP

□ Show hidden categories

Search filter

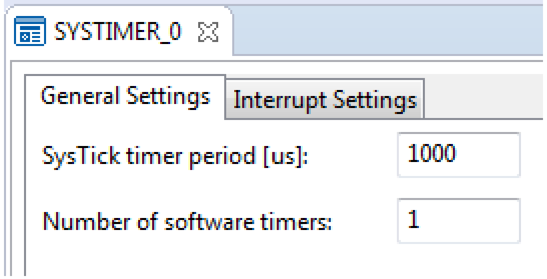
- Peripheral Configuration
- Power Conversion
- System
  - ANALOG_IO [4.0.4]
  - CLOCK_XMC1 [4.0.6]
  - CMSIS_DSP [4.0.2]
  - CMSIS_RTOS_RTX [4.0.2]
  - CPU_CTRL_XMC1 [4.0.2]
  - DIGITAL_IO [4.0.6]
  - EVENT_DETECTOR [4.0.2]
  - EVENT_GENERATOR [4.1.8]
  - INTERRUPT [4.0.4]
  - RTC [4.1.4]
  - SYSTIMER [4.1.6]
  - WATCHDOG [4.0.6]
  - RTOS

*Double-Click on the APP to add it to the active project.*

☑ Show latest versions only

□ Hide beta versions

The SYSTIMER APP uses the SysTick interrupt to call user functions periodically at a specified rate or after a given time period expires.



SYSTIMER_0

General Settings | Interrupt Settings

SysTick timer period [us]: 1000

Number of software timers: 1

# Getting started – Example 2
# White lamp using LED_LAMP APP (17/19)

7. Define SYSTIMER callback function

› Purpose of callback function is to change the brightness of lamp every 10 seconds

› Initialize callback function

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

› Create software timer and start timer

```
uint32_t timer_id;
TimerId = SYSTIMER_CreateTimer(1000000,SYSTIMER_MODE_PERIODIC,OneSecTick,NULL);
SYSTIMER_StartTimer(TimerId);
```

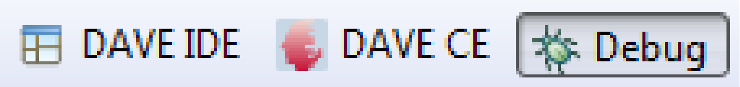
# Getting started – Example 2
# White lamp using LED_LAMP APP (18/19)

## 7. Define SYSTIMER callback function (continued)

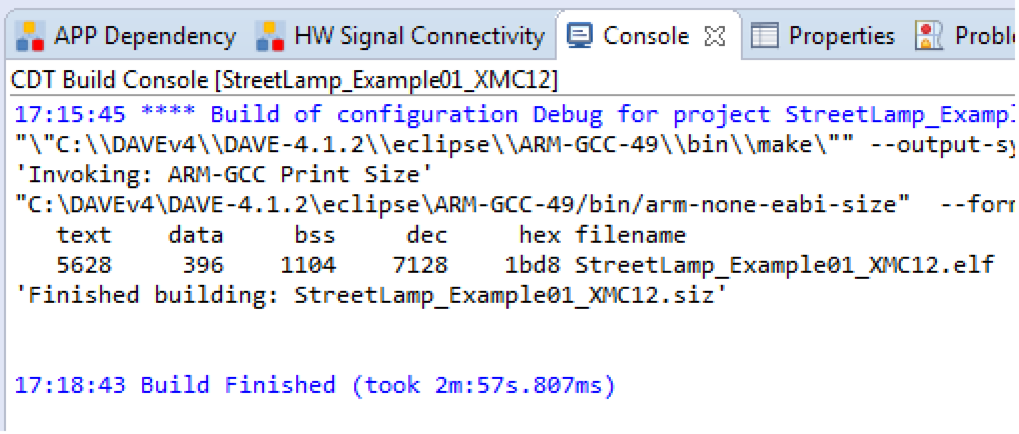
```
void OneSecTick(void)
{
  static uint8_t step = 0;

  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



CDT Build Console [StreetLamp_Example01_XMC12]

```
17:15:45 **** Build of configuration Debug for project StreetLamp_Examp
"\"C:\\DAVEv4\\DAVE-4.1.2\\eclipse\\ARM-GCC-49\\bin\\make\"" --output-s
'Invoking: ARM-GCC Print Size'
"C:\DAVEv4\DAVE-4.1.2\eclipse\ARM-GCC-49/bin/arm-none-eabi-size"  --for
   text    data     bss     dec     hex filename
   5628     396    1104    7128    1bd8 StreetLamp_Example01_XMC12.elf
'Finished building: StreetLamp_Example01_XMC12.siz'


17:18:43 Build Finished (took 2m:57s.807ms)
```

# 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

# General information (1/2)

- Where to buy kit:
  - http://ehitex.com/starter-kits/for-xmc1000
  - Order Number: KIT_XMC1x_AK_LED_001

- Infineon parts utilized on kit:

| Infineon parts | Order number |
|---|---|
| XMC1200 Microcontroller | XMC1200-T038F0200 |
| XMC4200 Microcontroller | XMC4200-Q48F256 |
| 5 V regulator | IFX25001TFV50 |
| 3V3 regulator | IFX25001MEV33 |
| BCR421/SC74 LED Driver | BCR421UE6327HTSA1 |
| BCR450 LED Driver | BCR450E6327HTSA1 |
| TDA7200 RF Receiver | TDA7200XUMA1 |

# General information (2/2)

- Kit documentation:
  - LED Lighting Application Kit

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

# Agenda

1 Kit overview

2 Hardware overview

3 Tooling overview – boot modes

4 Tooling overview – DAVE™

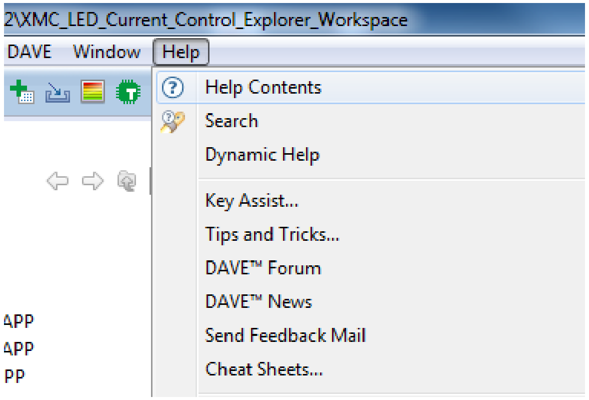
5 Getting started - examples

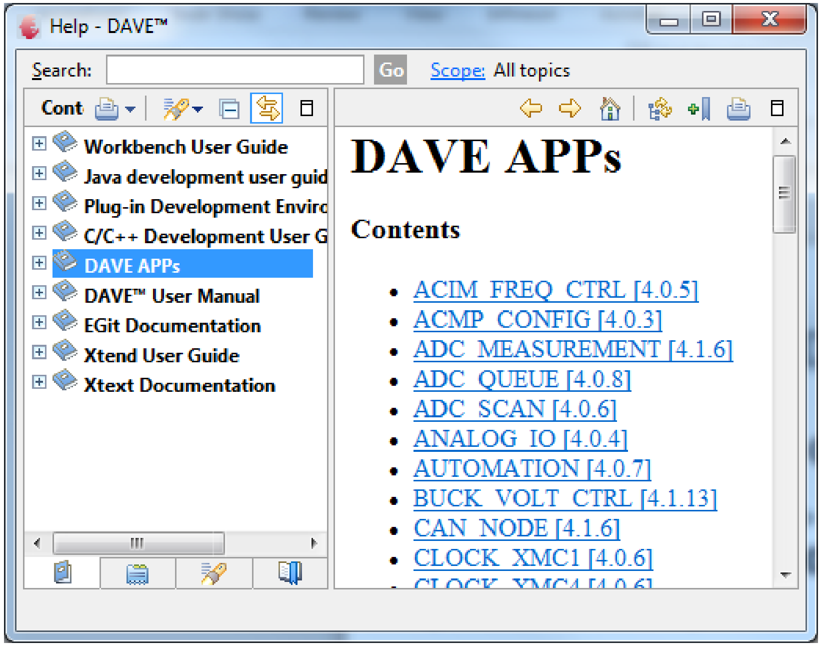
6 General information

7 References

# References
# Where to find APP documentation? (1/2)

- Go to **Help → Help Contents**
- Click **DAVE$^{TM}$ APPs**
- Click APP_Name (e.g. **LED_LAMP**)





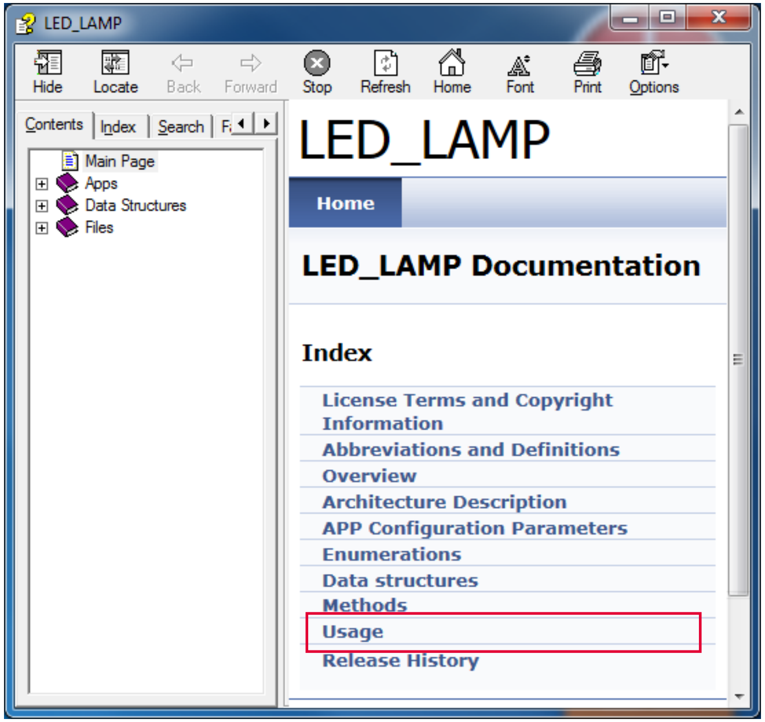
DAVE APPs

Contents

- ACIM_FREQ_CTRL [4.0.5]
- ACMP_CONFIG [4.0.3]
- ADC_MEASUREMENT [4.1.6]
- ADC_QUEUE [4.0.8]
- ADC_SCAN [4.0.6]
- ANALOG_IO [4.0.4]
- AUTOMATION [4.0.7]
- BUCK_VOLT_CTRL [4.1.13]
- CAN_NODE [4.1.6]
- CLOCK_XMC1 [4.0.6]
- CLOCK_XMC4 [4.0.6]

# References
# Where to find APP documentation? (2/2)

- Usage information can be found under **Usage** section

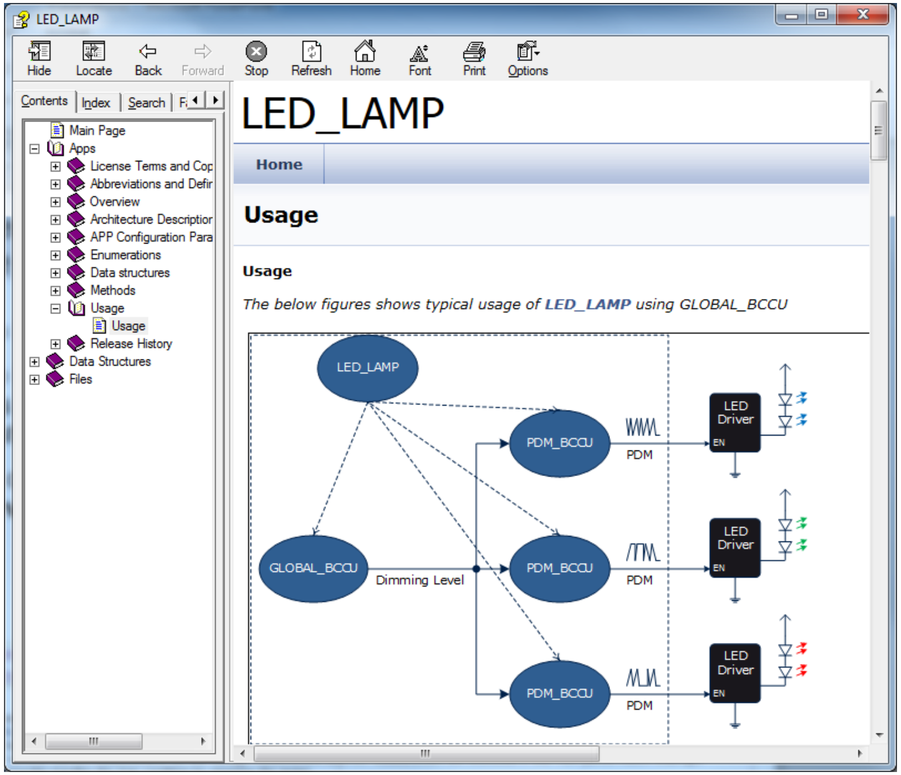


LED_LAMP

**LED_LAMP Documentation**

**Index**

- License Terms and Copyright Information
- Abbreviations and Definitions
- Overview
- Architecture Description
- APP Configuration Parameters
- Enumerations
- Data structures
- Methods
- Usage
- Release History



LED_LAMP

**Usage**

**Usage**

*The below figures shows typical usage of **LED_LAMP** using GLOBAL_BCCU*

# References
# Where to download example projects?

- Two sets of example projects available
  - Additional application examples
    - Can be downloaded directly from the web
  - DAVE$^{TM}$ project library examples
    - Can be downloaded from library in DAVE$^{TM}$
    - Can also be downloaded directly from the web

# References
# Where to download example projects?

- 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$^{TM}$ Lib

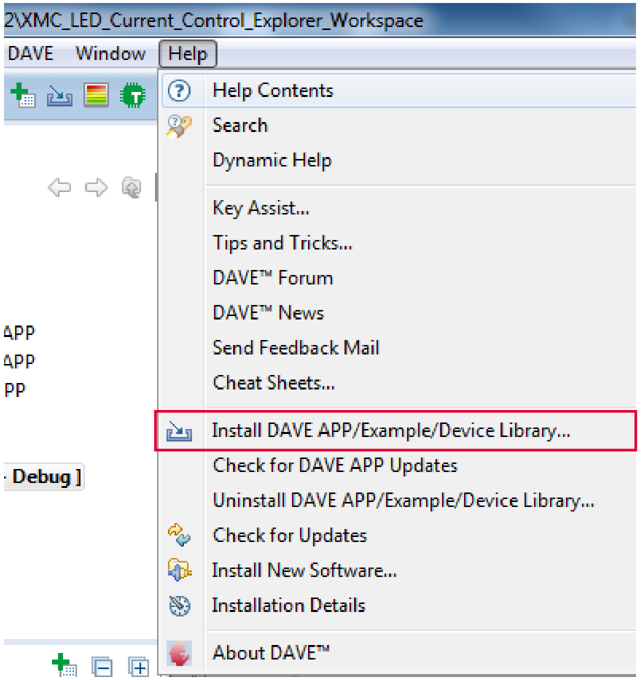
# References
# Where to download example projects?

- Additional application examples available
  - White Lamp Example (BCCU_WHITE_LAMP_EXAMPLE.zip)
    - Demonstrates white lamp functionality using XMC$^{TM}$ Lib
- Can be downloaded from the web HERE

# References
# How to load example project in DAVE$^{TM}$? (1/5)

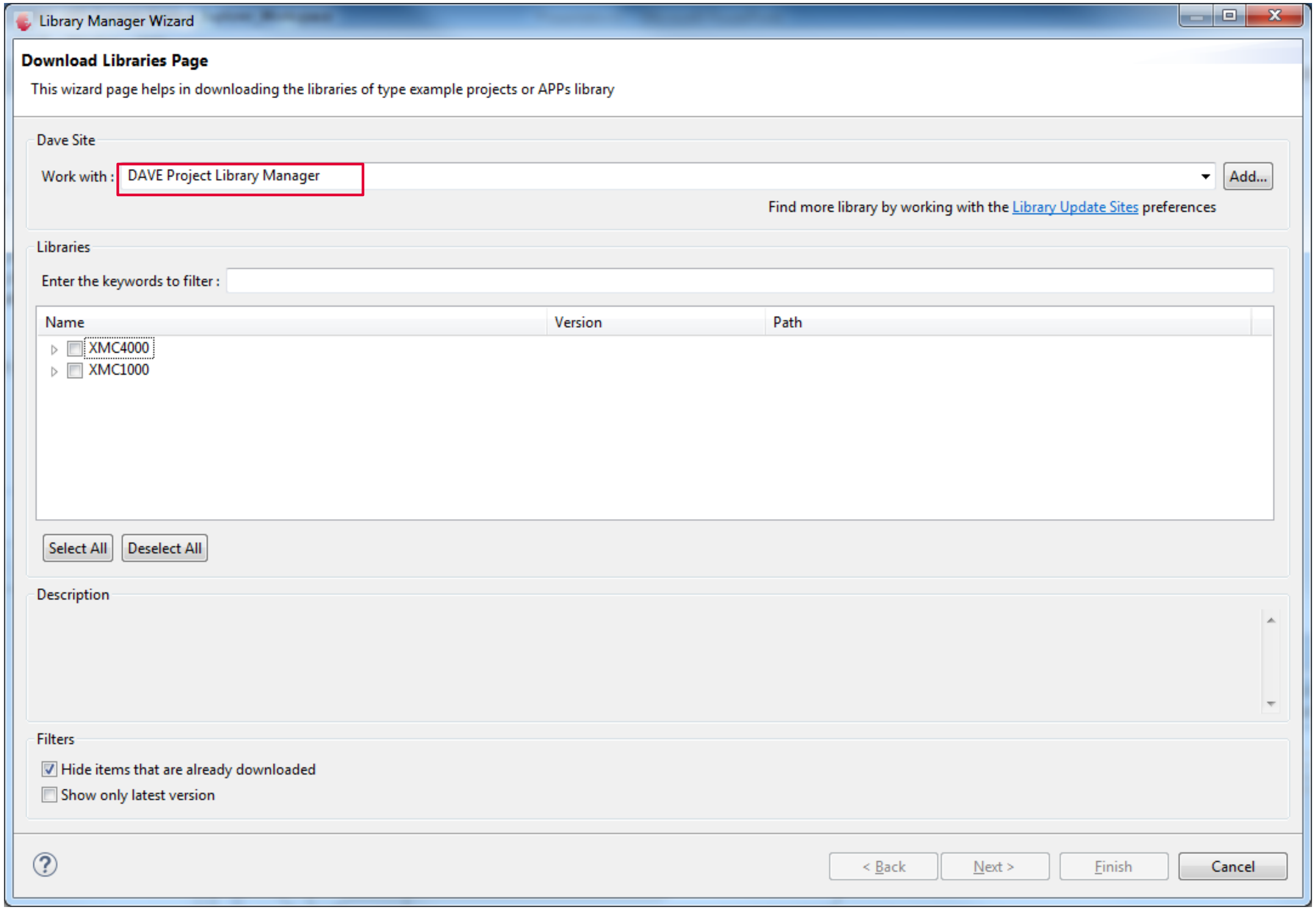
› Download example projects via DAVE$^{TM}$ library store

– **Help → Install DAVE$^{TM}$ APP/Example/Device Library…**



# References
# How to load example project in DAVE$^{TM}$? (2/5)

› Select **DAVE$^{TM}$ Project Library Manager** in the drop-down menu



Library Manager Wizard

**Download Libraries Page**

This wizard page helps in downloading the libraries of type example projects or APPs library

Dave Site

Work with : DAVE Project Library Manager

Find more library by working with the Library Update Sites preferences

Libraries

Enter the keywords to filter :

| Name | Version | Path |
|---|---|---|
| ▷ □ XMC4000 | | |
| ▷ □ XMC1000 | | |

Description

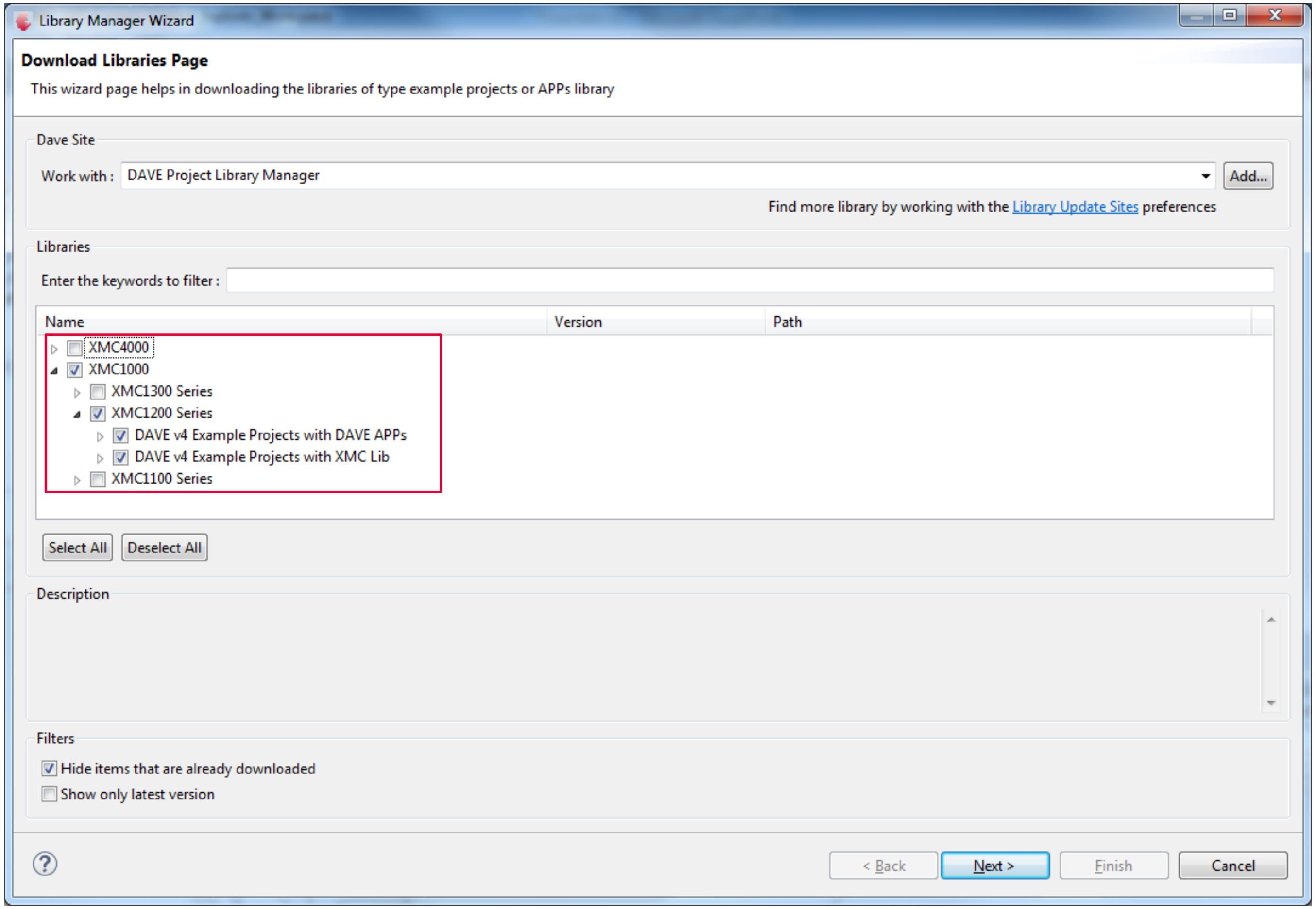
Filters

☑ Hide items that are already downloaded

□ Show only latest version

# References
# How to load example project in DAVE$^{TM}$? (3/5)

- Select examples in the **Libraries** window and click Next



Library Manager Wizard

**Download Libraries Page**

This wizard page helps in downloading the libraries of type example projects or APPs library

Dave Site

Work with : DAVE Project Library Manager

Find more library by working with the Library Update Sites preferences

Libraries

Enter the keywords to filter :

| Name | Version | Path |
|---|---|---|
| ▷ □ XMC4000 | | |
| ◢ ☑ XMC1000 | | |
| ▷ □ XMC1300 Series | | |
| ◢ ☑ XMC1200 Series | | |
| ▷ ☑ DAVE v4 Example Projects with DAVE APPs | | |
| ▷ ☑ DAVE v4 Example Projects with XMC Lib | | |
| ▷ □ XMC1100 Series | | |

Description

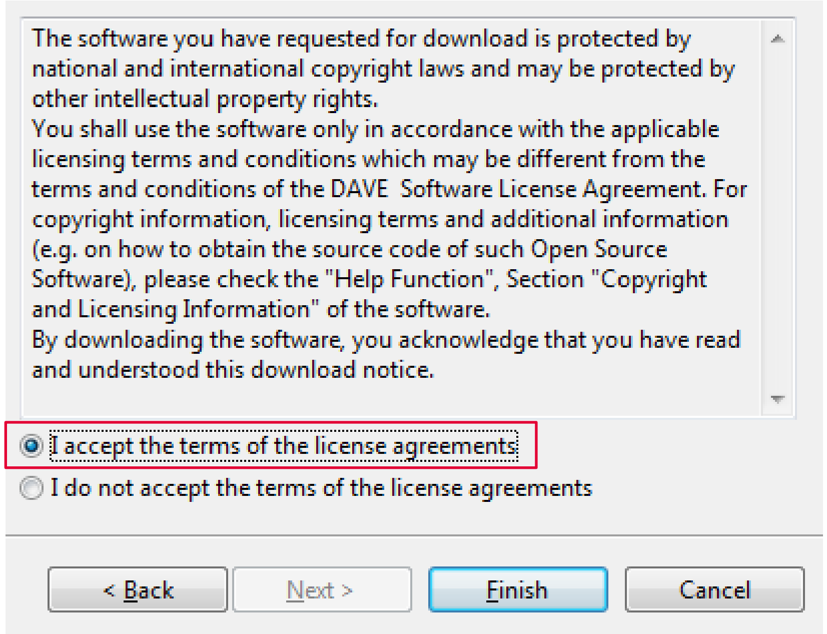
Filters

☑ Hide items that are already downloaded

□ Show only latest version

# References
# How to load example project in DAVE$^{TM}$? (4/5)

- Accept terms of the license agreement and click Finish



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  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

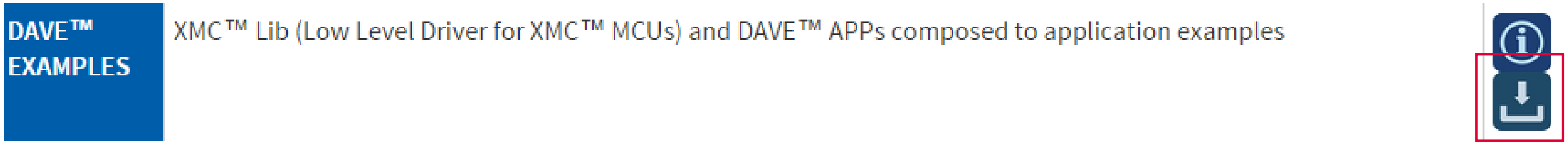
○ I do not accept the terms of the license agreements

- DAVE$^{TM}$ example projects are installed

# References
# How to load example project in DAVE™? (5/5)

- Download Example Projects from the web

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



DAVE™ EXAMPLES

XMC™ Lib (Low Level Driver for XMC™ MCUs) and DAVE™ APPs composed to application examples

  - 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<br>Document<br>Info | › Product Briefs<br>› Selection Guides<br>› Application Brochures<br>› Presentations<br>› Press Releases, Ads | › www.infineon.com/XMC |
| Technical Material<br>Tools<br>Evaluation Board | › Application Notes<br>› Technical Articles<br>› Simulation Models<br>› Datasheets, MCDS Files<br>› PCB Design Data | › www.infineon.com/XMC<br>› Kits and Boards<br>› DAVE$^{TM}$<br>› Software and Tool Ecosystem |
| Videos<br>Play | › Technical Videos<br>› Product Information Videos | › Infineon Media Center<br>› XMC Mediathek |
| Contact<br>Support | › Forums<br>› Product Support | › Infineon Forums<br>› Technical Assistance Center (TAC) |

# Disclaimer

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.