Customer training workshop: Device Configurator_Clock configuration

Please read the Important notice and warnings at the end of this document
Scope of work

- This document helps application developers understand how to use the clock configuration of the Device Configurator as part of creating a ModusToolbox™ application
  - The Device Configurator is part of a collection of tools included with the ModusToolbox™ software. It provides a GUI to configure the target device.

- ModusToolbox™ tools package version
  - 3.0.0

- Device Configurator version
  - 4.0

- Device
  - The TRAVEO™ T2G CYT4BFCH device is used in this code example.

- Board
  - The TRAVEO™ T2G KIT_T2G-B-H_EVK board is used for testing.
Introduction

› Clock system for TRAVEO™ T2G body high has the following features:
  – Supports high and low-speed clocks, using both internal and external clock sources.
  – Internal real time clock (RTC) for the clock input.
  – Supports phase-locked loop (PLL) and frequency-locked loop (FLL) to generate clocks that operate the internal circuit at a high speed.
  – Supports a function to monitor clock operation and to measure the clock difference of each clock with reference to a known clock.
Clock configuration in Device Configurator:

- The device configurator allows to configure the following clocks:
  - System clocks
    - FLL
    - PLL0/1 (200MHz PLL), PLL400M0/1 (400MHz PLL)
  - High-frequency clocks
    - CLK_HF0 to CLK_HF7
  - Input clocks
    - ECO, EXTCLK, ILO0, ILO1
    - IMO (locked enabled by default because it is not supported)
    - WCO (enabled on startup timing)
  - Miscellaneous clocks
    - CLK_ALT_SYS_TICK, CLK_BAK, CLK_LF, ECO pre-scaler etc
  - Peripheral clock dividers
    - Peripheral clock dividers (8-bit, 16-bit, and 24.5-bit)
Launch the Device Configurator

› **From Eclipse IDE**

You can launch the Device Configurator by following either of these methods:

a) Right-click on the project in the Project Explorer and select ModusToolbox™ > Device Configurator <version>

b) Click the Device Configurator link in the Quick Panel
Device Configurator for clock configuration

Device Configurator view – System tab

- Configure each of the clocks (system clocks, high-frequency clocks, input clocks, miscellaneous clocks)

Select the used clock, such as system clocks, high-frequency clocks, and input clocks.

Show the output frequency by current setting. Also, you can select a clock to configure by clicking the box in here.

Configure the parameters of selected clock

If you select the check box of the resource clock to be used, it reflects in the clock supply system diagram in the middle window, and the details are displayed in the parameter window.
Device Configurator for clock configuration (contd.)

› **Device Configurator view – Peripheral-Clocks tab**
  - Configure peripheral clock dividers

Select the used clock divider

Configure the parameters of the selected divider, and display setting in the Code Preview tab
What the Device Configurator can do for Clock Settings

› **Follow these steps to use the Device Configurator for clock setting:**
  
  – Save the file to generate source code.
  
  – The Device Configurator generates code into a "GeneratedSource" directory in your Eclipse IDE application, or in the same location you saved the *.modus file for non-IDE applications. That directory contains the necessary source (.c) and header (.h) files for the generated firmware, which uses the relevant driver APIs to configure the hardware.
  
  – No additional settings are required as the clock settings use a preconfigured structure in Device Configurator. It is set in the `cybsp_init()` function in main.c.
Clock configuration

Create project

1) Click “New Application” in the Quick Panel and open the Choose Board Support Package (BSP) window

2) Select “TRAVEO™ BSPs” and “KIT_T2G-B-H_EVK”

3) Click the “Next” button and open the Application window

4) Check the “Empty App”
   In this use case, it changes to “Clock_Configuration_training”. You can change the application name.

5) Click the Create button to start application creation
Clock configuration (contd.)

› Launch the Device Configurator

1) Select the “Clock_configuration_training” project.
2) Click the Device configurator in the Quick Panel
3) Then, open the Device Configurator window
Setting example: CPU clocks

› Use case

- Input clocks
  - IMO: 8.0000 MHz
- System clocks
  - PLL400M0: Input ECO, Output 100.000 MHz
  - PLL400M1: Input ECO, Output 350.000 MHz
- High-frequency clocks
  - CLK_HF0: Input PLL400M0 (CLK_PATH1), Output 100.000 MHz (Divide by 1)
  - CLK_HF1: Input PLL400M1 (CLK_PATH2), Output 350.000 MHz (Divide by 1)
- Output
  - CLK_MEM: Output 100.000 MHz (Divide CLK_HF0 by 1)
  - CLK_SLOW: Output 100.000 MHz (Divide CKK_MEM by 1), connected to CM0+ clock input
  - CLK_FAST0: Output 350.000 MHz (Divide CLK_HF1 by 1), connected to CM7 instance 0 clock input.
  - CLK_FAST1: Output 350.000 MHz (Divide CLK_HF1 by 1), connected to CM7 instance 1 clock input.
Setting example: CPU clocks (contd.)

› Input clocks

Select IMO

Confirm Input Clock is 8.000 MHz
Setting example: CPU clocks (contd.)

System clocks (PATH_MUX1/2, PLL400M0/1)

- Select PATH_MUX1 for PLL#0 input
- Select PATH_MUX2 for PLL#1 input
- Select PLL400M0
- Select PLL400M1
- Set 100.000 MHz
- Set 350.000 MHz
- Select the IMO
Setting example: CPU clocks (contd.)

- High-frequency clocks (CLK_HF1, CLK_HF2)

Select CLK_HF0
Select CLK_HF1
Select CLK_PATH1 as Source Clock
Divide by 1

Select CLK_PATH2 as Source Clock
Divide by 1

You can see output clock after divider setting
Setting example: CPU clocks (contd.)

› Output (CLK_MEM, CLK_SLOW, CLK_FAST0/1)

Select each clocks

- Divide by 1
- Divide by 1
- Divide by 1
- Divide by 1
Setting example: I2S clock

› **Use case**
  - Input clocks
    - ECO: 16.0000 MHz
  - System clocks
    - PLL400M1: Input ECO, Output 196.608 MHz
  - High-frequency clocks
    - CLK_HF5: Input PLL400M1 (CLK_PATH2), Output 196.608 MHz (Divide by 1)
  - Output
    - Audio (I2S): Output 24.576 MHz (Divide CLK_HF5 by 8)
Setting example: I2S clock (contd.)

› Input clocks

Select ECO

Set to 16.000
Setting example: I2S clock (contd.)

› System clocks (PATH_MUX2, PLL400M1)

Select PATH_MUX2 for PLL#0 input
Select PLL400M1
Select the ECO
Set 196.608 MHz
Setting example: I2S clock (contd.)

› High-frequency clocks (CLK_HF5) and audio (I2S) output

Select CLK_HF5

Select CLK_PATH2 as Source Clock

Divide by 8

You can see output clock after divider setting
Setting example: ADC clock

› **Use case**
  
  - **Input clocks**
    - IMO: 8.0000 MHz
  
  - **System clocks**
    - PLL400M1: Input IMO, Output 196 MHz
  
  - **High-frequency clocks**
    - CLK_HF2: Input PLL400M1 (CLK_PATH2), Output 196 MHz (Divide by 1)
  
  - **Output (Peripheral Clock Group 1)**
    - 16 bit Divider 0: Output 13.07 MHz (Divide CLK_HF2 by 15)
  
  - See the EVTGEN_trigger_ADC application for operation
Setting example: ADC clock (contd.)

- Input clocks

Select IMO

Confirm Input Clock is 8.000 MHz
Setting example: ADC clock (contd.)

› System clocks (PATH_MUX2, PLL400M1)

- Select PATH_MUX2 for PLL#0 input
- Select PLL400M1
- Select the IMO
- Set 196 MHz
Setting example: ADC clock (contd.)

› High-frequency clocks (CLK_HF2)

- Select CLK_HF2
- Select CLK_PATH2 as Source Clock
- Divide by 1
- You can see output clock after divider setting
Setting example: ADC clock (contd.)

› Output (Peripheral Clock Group 1)

- Select 16 bit Divider 0
- Divide by 15
- Select connection peripheral (SAR ADC0)
- You can see output clock after divider setting
Setting example: CAN FD clock

› **Use case**

- Input clocks
  - ECO: 16.0000 MHz
- System clocks
  - PLL400M1: Input ECO, Output 200 MHz
- High-frequency clocks
  - CLK_HF2: Input PLL400M1 (CLK_PATH2), Output 200 MHz (Divide by 1)
- Output (Peripheral Clock Group 1)
  - 16 bit Divider 0: Output 40 MHz (Divide CLK_HF2 by 5)
- See the CAN_FD application for operation
Setting example: CAN FD clock (contd.)

- Input clocks

Select ECO

Confirm Input Clock is 16.000 MHz
Setting example: CAN FD clock (contd.)

- System clocks (PATH_MUX1, PLL400M1)
  - Select PATH_MUX2 for PLL#0 input
  - Select PLL400M1
  - Select the ECO
  - Set 200 MHz
Setting example: CAN FD clock (contd.)

› High-frequency clocks (CLK_HF2) and Peripheral group 1 Output

- Select CLK_HF2
- Select CLK_PATH2 as Source Clock
- Divide by 1

You can see output clock after divider setting
Setting example: CAN FD clock (contd.)

› Output (Peripheral Clock Group 1)

- Select 16 bit Divider 0
- Select connection peripheral (CANFD)
- Divide by 5
- You can see output clock after divider setting
Setting example: UART clock

› **Use case**

- Input clocks
  - IMO: 8.0000 MHz
- System clocks
  - PLL400M1: Input IMO, Output 196 MHz
- High-frequency clocks
  - CLK_HF2: Input PLL400M1 (CLK_PATH2), Output 196 MHz (Divide by 1)
- Output
  - Peripheral group 1: Output 920.2 kHz (Divide CLK_HF2 by 213)
  - See the SCB_UART_Transmit_and_Receive_using_DMA application for operation
Setting example: UART clock (contd.)

› Input clocks

Select IMO

Confirm Input Clock is 8.000 MHz
Setting example: UART clock (contd.)

› System clocks (PATH_MUX2, PLL400M1)

- Select PATH_MUX2 for PLL#0 input
- Select PLL400M1
- Select the IMO
- Set 196 MHz
Setting example: UART clock (contd.)

- High-frequency clocks (CLK_HF2) and Peripheral group 1 Output

Select CLK_HF2

Select CLK_PATH2 as Source Clock

Divide by 1

You can see output clock after divider setting
Setting example: UART clock (contd.)

› Output (Peripheral Clock Group 1)

Select 8 bit Divider 0

Divide by 213

Select connection peripheral (SCB3)

You can see output clock after divider setting
References

Datasheet
› CYT4BF datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family

Architecture Technical reference manual
› TRAVEO™ T2G automotive body controller high family architecture technical reference manual

Registers Technical reference manual
› TRAVEO™ T2G Automotive body controller high registers technical reference manual

PDL/HAL
› PDL
› HAL

Training
› TRAVEO™ T2G Training

Application note
› Clock configuration setup for TRAVEO™ T2G family MCUs in ModusToolbox™ (Doc No. 002-35303)
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>ECN</th>
<th>Submission Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>7845000</td>
<td>2022/12/07</td>
<td>Initial release</td>
</tr>
<tr>
<td>*A</td>
<td>7897805</td>
<td>2023/04/05</td>
<td>Updated to change IMO to ECO for CAN FD use case in “Setting example: CAN FD clock”</td>
</tr>
</tbody>
</table>
Important notice and warnings

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-04
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2023 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about this document?
Go to: www.infineon.com/support

Document reference
002-36685 Rev. *A

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.

Important notice and warnings