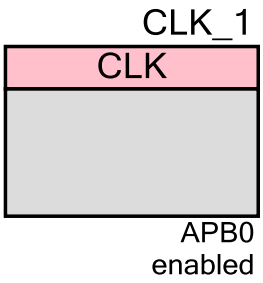
Clock Function (PDL_CLK)
1.0



Features

- Selecting Clock mode
- Internal Bus Clock Frequency Division Control
- PLL Clock Control
- Setting Oscillation Stabilization Wait Time of the main clock (CLKMO) and sub clock (CLKSO)
- Interrupts Factors

General Description

The Peripheral Driver Library (PDL) Clock functions (CLK) unit generates various types of clocks used to operate the MCU. Source clock is the generic name for external and internal oscillation clocks of this MCU. The source clocks can be: Main clock (CLKMO), Sub clock (CLKSO), High-speed CR clock (CLKHC), Low-speed CR clock (CLKLC), Main PLL clock (CLKPLL).

Dividing the master clock frequency can generate a base clock. In addition, dividing the base clock can generate each bus clock.

This component uses firmware drivers from the PDL_CLK module, which is automatically added to your project after a successful build.

When to Use a PDL_CLK Component

Use the PDL_CLK component for configuration of the settings of timing.

Quick Start

1. Drag a PDL_CLK component from the Component Catalog FMx/System/ folder onto your schematic. The placed instance takes the name CLK_1.
2. Double-click to open the component's Configure dialog.
3. On the Basic tab set the following parameters:
    - enable needed peripheral buses
    - set clock divider value

4. Build the project to verify the correctness of your design. This will add the required PDL modules to the Workspace Explorer and generate configuration data for the CLK_1 instance.
5. In the *main.c* file, initialize the peripheral and start the application.

```
Clk_Init(&CLK_1_Config);
Clk_EnableMainClock(TRUE);/* enable Main oscillator */
Clk_EnablePllClock(TRUE);/* enable PLL oscillator */
Clk_SetSource(ClkPll);/* transit to main clock */
```

6. Build and program the device.

Component Parameters

The PDL_CLK component Configure dialog allows you to edit the configuration parameters for the component instance.

Basic Tab

This tab contains the component parameters used in the basic peripheral initialization settings.

Parameter Name	Description
enAPB0Div	APB0 clock divider.
bAPB1Disable	Disable APB1 – over-rides the divider setting.
enAPB1Div	APB1 clock divider.
bAPB2Disable	Disable APB2 – over-rides the divider setting.
enAPB2Div	APB2 clock divider.
enBaseClkDiv	Base clock divider.

Oscilators Tab

This tab contains the Oscilators configuration settings.

Parameter Name	Description
bMcoIrq	Enable main clock oscillation stabilization completion interrupt.
enMCOWaitTime	Main clock oscillator startup wait time (2^ enMCOWaitTime/CLK).
pfnMcoStabCb	Main clock stabilization call-back function. Note: this generates a declaration only - USER must implement the function.
bPllIrq	Enable PLL oscillation stabilization completion interrupt.
enPLLOWaitTime	PLL startup wait time (2^ enPLLOWaitTime/CLK).

Parameter Name	Description
pfnPllStabCb	PLL stabilization call-back function – enter function name or address, or modify in firmware.
u8PllK	PLL input clock frequency division ratio, PLLK.
u8PllM	PLL VCO clock frequency division ratio, PLLM.
u8PllN	PLL feedback frequency division ratio, PLLN.
bScoIrq	Enable sub-clock oscillation stabilization completion interrupt.
enSCOWaitTime	Sub-clock oscillator startup wait time (2^ enSCOWaitTime/CLK).
pfnScoStabCb	Sub-clock stabilization call-back function. Note: this generates a declaration only - USER must implement the function.

Component Usage

After a successful build firmware drivers from the PDL_CLK module, are added to your project in the pdl/drivers/clk folder. Pass the generated data structures to the associated PDL functions in your application initialization code to configure the peripheral.

Generated Data

The PDL_CLK component populates the following peripheral initialization data structure(s). The generated code is placed in C source and header files that are named after the instance of the component (e.g. *CLK_1_config.c*). Each variable is also prefixed with the instance name of the component.

Data Structure Type	Name	Description
stc_clk_config_t	CLK_1_Config	Configuration structure. Pass this to Clk_Init() in order to initialize the peripheral.

Once the component is initialized, the application code should use the peripheral functions provided in the referenced PDL files. Refer to the PDL for the list of provided API functions. To access this document, right-click on the component symbol on the schematic and choose "**Open API Documentation…**" option in the drop-down menu.

Data in RAM

The generated data may be placed in flash memory (const) or RAM. The former is the more memory-efficient choice if you do not wish to modify the configuration data at run-time. Under the **Built-In** tab of the Configure dialog set the parameter CONST_CONFIG to make your selection. The default option is to place the data in flash.

Interrupt Support

If the PDL_CLK component is specified to trigger interrupts, it will generate the callback function declaration that will be called from the CLK ISR. The user is then required to provide the actual callback code. If a null string is provided the struct is populated with zeroes and the callback declaration is not generated. Thus, it is the user's responsibility to modify the struct in firmware.

The component generates the following function declarations.

Function Callback	Description
CLK_1_PllStabIrqCb	PLL stabilization call-back function – enter function name or address, or modify in firmware. Note: this generates a declaration only - USER must implement the function.
CLK_1_McoStabIrqCb	Main clock stabilization call-back function. Note: this generates a declaration only - USER must implement the function.
CLK_1_ScoStabIrqCb	Sub-clock stabilization call-back function. Note: this generates a declaration only - USER must implement the function.

Code Examples and Application Notes

There are numerous code examples that include schematics and example code available online at the Cypress Code Examples web page.

Cypress also provides a number of application notes describing how FMx devices can be integrated into your design. You can access the Cypress Application Notes search web page at www.cypress.com/appnotes.

Resources

The PDL_CLK component uses the Clock Functions (CLK) peripheral block.

References

- FM0+ Family of 32-bit ARM® Cortex®-M0+ Microcontrollers Peripheral Manuals
- Cypress FM0+ Family of 32-bit ARM® Cortex®-M0+ Microcontrollers

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
1.0	Initial Version	

© Cypress Semiconductor Corporation, 2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

