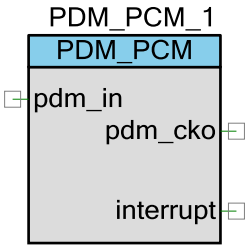
Pulse-Density Modulation to Pulse-Code Modulation Decoder (PDM_PCM_PDL) 2.0

Features



- Supports stereo/mono dual mode PDM to PCM conversion programmable word length
- 1-bit PDM input, 16/18/20/24-bit PCM data output
- Low digital microphone clock rates for low power applications
- PCM sampling rate: from 8 KHz to 48 KHz
- Volume control: programmable gain amplifier (PGA) from -12 dB to +10.5 dB in 1.5 dB steps

General Description

The PDM_PCM_PDL Component converts a bit stream from a PDM source to PCM, which is similar to the output of an ADC. Various sample rates and data formatting options are supported.

The PDM_PCM_PDL Component is a graphical configuration entity built on top of the pdm_pcm driver available in the Peripheral Driver Library (PDL). It allows schematic-based connections and hardware configuration as defined by the Component Configure dialog.

The generated configuration structures are to be passed into the pdm_pcm driver. The Component-specific API can also be used to access higher level functionalities built on top of the base driver.

When to Use a PDM_PCM_PDL Component

The PDM_PCM_PDL Component is typically part of a system to implement digital voice/audio recording and processing such as:

- Wearable Device
- IoT/IoE System
- Virtual Reality Gaming System
- Tablet

- Smart Automobile
- Voice-in-Command Appliance
- Smart Home System
- PMP, MID
- Voice Recorder

Quick Start

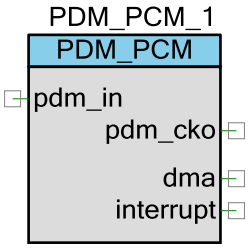
1. Drag a PDM_PCM [v2.0] Component from the Component Catalog Cypress/Communications/PDM_PCM folder onto your schematic (the placed instance takes the name PDM_PCM_1).
2. Double-click to open the Configure dialog.
3. Set up the desired PDM_PCM settings (clock dividers, audio channels, word lengths, interrupts, etc.).
4. Connect all the input/output digital pins (if using the base Component and not a macro).
5. Build the project in order to verify the correctness of your design. This will add the required PDL modules to the Workspace Explorer, and generate the configuration data for the PDM_PCM_1 instance.
6. In *main.c*, initialize the peripheral and start the application (example for transmission only):

```
(void)Cy_PDM_PCM_Init(PDM_PCM_1_HW, &PDM_PCM_1_config);
Cy_PDM_PCM_ClearFifo(PDM_PCM_1_HW);
Cy_PDM_PCM_Enable(PDM_PCM_1_HW);
```

7. Build and program the device.

Input/Output Connections

This section describes the various input and output connections for the PDM_PCM_PDL Component. An asterisk (*) in the following list indicates that it may not be shown on the Component symbol for the conditions listed in the description of that I/O.



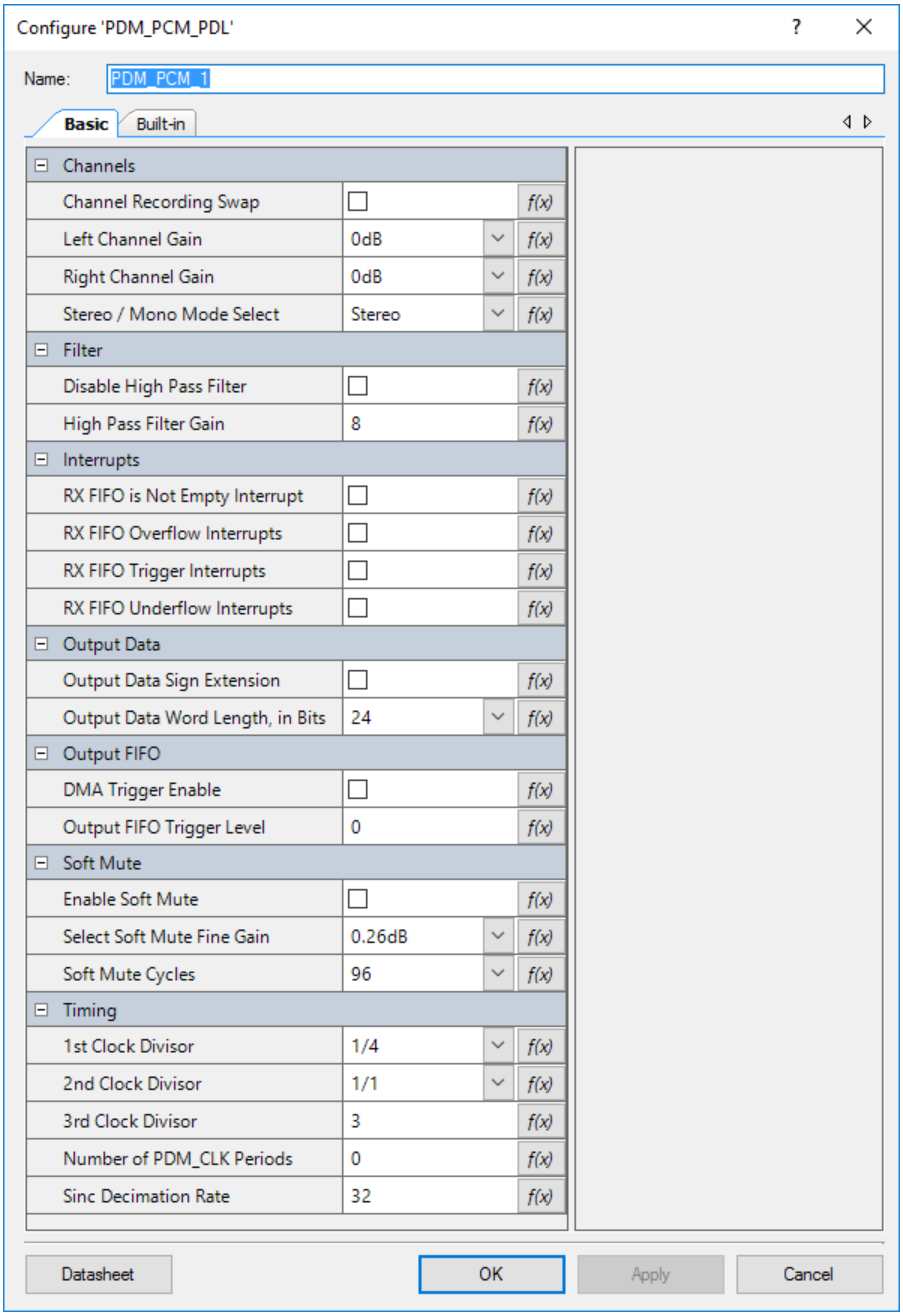
Terminal Name	I/O Type	Description
pdm_in	Digital Input	PDM input signal from PDM device for conversion. Can be connected to digital input pin.
pdm_cko	Digital Output	Clock output signal for PDM sampling. Can be connected to digital output pin.
interrupt	Digital Output	Interrupt signal output. Visible when any interrupt source is activated.
dma*	Digital Output	DMA transfer request signal. Visible when the **DMA Trigger Enable** is selected.

Component Parameters

The PDM_PCM_PDL 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 general peripheral initialization settings.



Parameter Name	Description
Channels	
Channel recording swap	Right-left channel recording swap.
Left channel gain	Left channel gain, in dB. Range is -12…+10.5 with step 1.5. Default is 0.
Right channel gain	Right channel gain, in dB. Range is -12…+10.5 with step 1.5. Default is 0.
Stereo / mono mode select	Stereo / mono Left/Right mode select. Default is Stereo.
Filter	
Disable High Pass Filter	Disables the high pass filter.
High Pass Filter Gain	High pass filter gain. Range: 0 - 15. Default is 8.
Interrupts	
RX FIFO is Not Empty Interrupt	RX FIFO is not empty.
RX FIFO Overflow Interrupts	Attempt to write to a full RX FIFO.
RX FIFO Trigger Interrupts	RX FIFO has more entries than the value specified by “Output FIFO Trigger Level”
RX FIFO Underflow Interrupts	Attempt to read from an empty RX FIFO.
Output Data	
Output Data Sign Extension	When enabled - all the MSB of RX_FIFO_RD beyond the "word length" are filled by the sign bit value. Otherwise (when disabled) - all the MSBa are filled by 0.
Output Data Word Length, in Bits	PCM Word Length in number of bits. Range: 16, 18, 20 and 24. Default is 24.
Output FIFO	
DMA Trigger Enable	Enable Trigger event from output RX FIFO.
Output FIFO Trigger Level	Trigger level. When the RX FIFO has more entries than the number of this field, a trigger event is generated. Range: 0 – 254 (253 for Stereo mode). Default is 0.
Soft Mute	
Enable Soft Mute	Enables/Disables the Soft Mute.
Select Soft Mute Fine Gain	Set the fine gain step for smooth PGA or Soft-Mute attenuation transition. Options are 0.13dB and 0.26 dB. Default is 0.26dB.
Soft Mute Cycles	Set time step for gain change during PGA or soft mute operation in number of 1/a sampling rate. Range is: 64, 96, 128, 160, 192, 256, 384 and 512. Default is 96.

Parameter Name	Description
Timing	
1-st clock divisor	This configures a frequency of PDM CLK. Range is: 1/1, 1/2, 1/3 and 1/4, Default is 1/4.
2-st clock divisor	MCLKQ divider (2nd divider). Range is: 1/1, 1/2, 1/3 and 1/4, Default is 1/1.
3-st clock divisor	PDM CKO clock divider (3rd divider). Range is: 1 - 15. Default is 3.
Number of PDM_CLK periods	Phase difference from the rising edge of internal sampler clock (CLK_IS) to that of PDM_CKO clock. Range: 0 - 7. Default is 0.
Sinc decimation rate	SINC Decimation Rate. Range: 0 - 127. Default is 32.

Application Programming Interface

The Application Programming Interface (API) is provided by the pdm_pcm driver module from the PDL. The driver is copied into the “pdl\drivers\peripheral\pdm_pcm\” directory of the application project after a successful build.

Refer to the PDL documentation for a detailed description of the complete API. To access this document, right-click on the Component symbol on the schematic and choose the “**Open PDL Documentation…**” option in the drop-down menu.

The Component generates the configuration structures and base address described in the Global Variables and Preprocessor Macros sections. Pass the generated data structure and the base address to the associated pdm_pcm driver function in the application initialization code to configure the peripheral. Once the peripheral is initialized, the application code can perform run-time changes by referencing the provided base address in the driver API functions.

In addition to the PDL API, the PDM_PCM component provides an instance-based component API that provide additional functionality available through PSoC Creator.

By default, PSoC Creator assigns the instance name PDM_PCM_1 to the first instance of a Component in a given design. You can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol.

Global Variables

The PDM_PCM_PDL 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. PDM_PCM_1.c). Each variable is also prefixed with the instance name of the Component.

cy_stc_pdm_pcm_config_t PDM_PCM_1_config

The Component generates an instance-specific configuration structure called PDM_PCM_1_config. This should be passed to the PDM_PCM_1_Init function to initialize the Component settings selected in the GUI.

Preprocessor Macros

The PDM_PCM_PDL Component generates the following preprocessor macro(s). Note that each macro is prefixed with the instance name of the Component (e.g. “PDM_PCM_1”).

#define PDM_PCM_1_HW PDM_PCM_1_cy_mxs40_pdm__HW

The pointer to the base address of the HW PDM_PCM instance.

#define PDM_PCM_1_RX_FIFO_DMA_PTR (void *) &(PDM_PCM_1_HW->RX_FIFO_RD)

Rx FIFO Read register pointer for DMA initialization.

#define PDM_PCM_1_INT_MASK

The default (configured in GUI) interrupt mask.

Component Functions

This Component also includes a set of Component-specific wrapper functions that provide simplified access to the basic PDM_PCM operation. These functions are generated during the build process and are all prefixed with the name of the Component instance.

Data in RAM

The generated configuration 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 CY_CONST_CONFIG to make your selection. The default option is to place the data in flash.

Code Examples and Application Notes

Code Examples

PSoC Creator provides access to code examples in the Find Code Example dialog. For Component-specific examples, open the dialog from the Component Catalog or an instance of the Component in a schematic. For general examples, open the dialog from the Start Page or **File** menu. As needed, use the **Filter Options** in the dialog to narrow the list of projects available to select.

Refer to the "Find Code Example" topic in the PSoC Creator Help for more information.

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

Application Notes

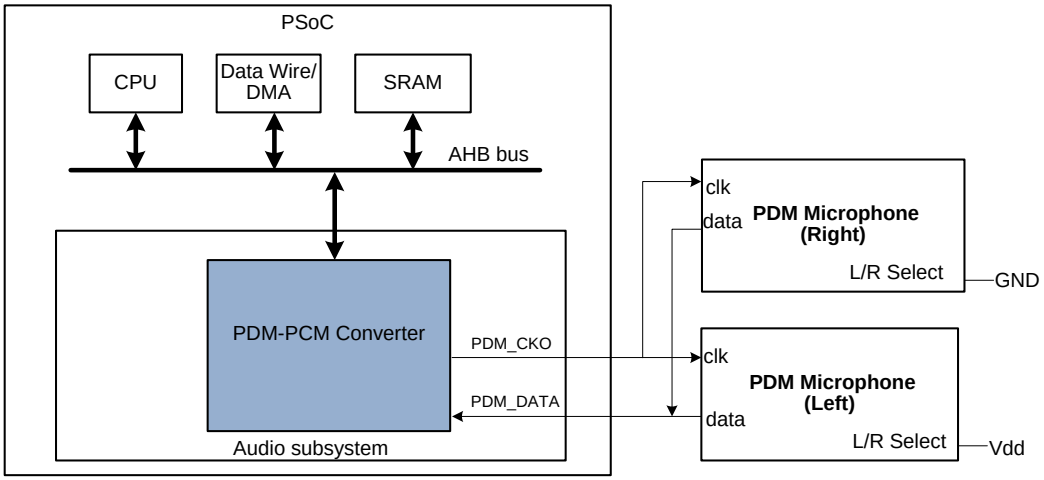
Cypress provides a number of application notes describing how PSoC can be integrated into your design. You can access the Cypress Application Notes search web page at www.cypress.com/appnotes.

Functional Description

The PDM_PCM unit takes in a stereo or mono serial stream coming from e.g. an external digital PDM microphone.

Block Diagram and Configuration

The following is a simplified diagram of the PDM_PCM hardware:



Clock Selection

The PDM interface is a de-facto industry standard used widely for digital microphones. In the PDM interface the left/right channels are sampled at the rising/falling edges of the PDM clock (PDM_CKO).

The internal clock source for PDM_PCM is HFCLK1.

The main clock input is divided by the **1st Clock divider** (PDM_PCM_1_CLK_DIV) and **2nd Clock divider** (PDM_PCM_1_MCLK_DIV) settings. This is again divided by the **3rd Clock divider** (PDM_PCM_1_CKO_DIV) setting to generate the PDM_CKO output.

In order to compensate for PDM_IN delay, the PDM_PCM_1_CKO_DELAY field is used to delay the internal PDM_IN sampling clock.

DMA Support

The PDM_PCM_PDL Component supports Direct Memory Access (DMA) transfers. The Component may transfer from the following source.

Name of DMA Source / Destination	Length	Direction	DMA Req Signal	DMA Req Type	Description
PDM_PCM_1_RX_FIFO_DMA_PTR	32	Source	dma	N/A	RX Fifo data register

API Memory Usage

The Component is designed to use API from the pdm_pcm PDL module. That is why the Component itself only consumes resources necessary to allocate structures for driver operation and start the Component.

Industry Standards

MISRA Compliance

This section describes the MISRA-C:2004 compliance and deviations for the Component. There are two types of deviations defined:

- project deviations – deviations that are applicable for all PSoC Creator Components
- specific deviations – deviations that are applicable only for this Component

This section provides information on Component-specific deviations. Refer to PSoC Creator Help > Building a PSoC Creator Project > Generated Files (PSoC 6) for information on MISRA compliance and deviations for files generated by PSoC Creator.

The PDM_PCM_PDL Component does not have any specific deviations.

This Component uses firmware drivers from the pdm_pcm PDL module. Refer to the PDL documentation for information on their MISRA compliance and specific deviations.

Registers

See the PDM to PCM decoder (PDM_PCM) Registers section in the chip *Technical Reference Manual (TRM)* for more information about the registers.

Resources

The PDM_PCM_PDL Component uses the PDM_PCM part of the AUDIOSS peripheral block.

DC and AC Electrical Characteristics

Note Final characterization data for PSoC 6 devices is not available at this time. Once the data is available, the Component datasheet will be updated on the Cypress web site.

References

List of references related to PDM_PCM_PDL Component:

- PDM – “pulse density modulation” data format
- PCM – “pulse code modulation” data format

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
2.0.c	Minor datasheet edits.	
2.0.b	Minor datasheet edits.	
2.0.a	Updated the datasheet, minor changes: • I/O section • Parameters section • API section • MISRA section	
2.0	Updated the interface (GUI and API)	PSoC 6 public release.

Version	Description of Changes	Reason for Changes / Impact
1.0.b	Additional datasheet edits.	
1.0.a	Updated the datasheet.	
1.0	Initial Version	

© Cypress Semiconductor Corporation, 2017. 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.