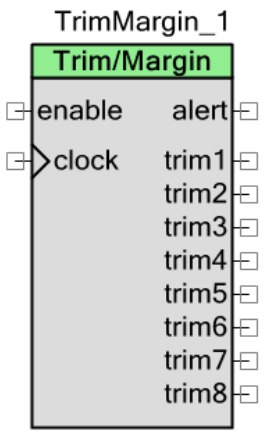
Trim and Margin
1.20

Features



- Works with most adjustable DC-DC converters or regulators including LDOs, switchers and modules
- Supports up to 24 DC-DC converters
- 8 to 10-bit resolution PWM pseudo-DAC outputs
- Supports real-time, closed-loop active trimming when used in conjunction with the Power Monitor component
- Built-in support for margining

General Description

The Trim and Margin component provides a simple way to adjust and control the output voltage of up to 24 DC-DC converters to meet system power supply requirements.

Users of this component simply enter the power converter nominal output voltages, voltage trimming range, margin high and margin low settings into the intuitive, easy-to-use graphical configuration GUI and the component takes care of the rest. The component will also assist the user to select appropriate external passive component values based on performance requirements.

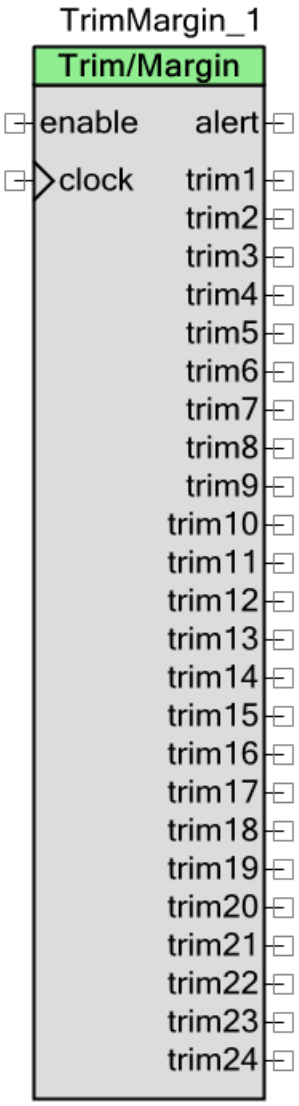
The provided firmware APIs enable users to manually trim the power converter output voltages to any desired level within the operational limits of the power converter. Real-time active trimming or margining is supported via as a continuously running background task with an update frequency controlled by the user.

When to Use a Trim and Margin

The Trim and Margin component should be used in any application that requires PSoC to adjust and control the output voltage of multiple DC-DC power converters. Use the Trim and Margin component along with other Power Supervision components to build your own custom power supervisor solution.

Input/Output Connections

This section describes the various input and output connections for the Trim and Margin component. An asterisk (*) in the list of I/Os means that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.



clock – Input

Clock signal used to drive the PWM pseudo-DAC outputs

enable – Input

Active high clock enable synchronous with the clock input. Asserting this signal enables the PWMs. This synchronous active high signal is used as a clock enable to the PWMs

alert – Output

Active high signal is asserted when closed loop trimming/margining is not achievable because PWM is at min or max duty cycle, but desired power converter output voltage has not been achieved. Remains asserted as long as the alert condition exists on any output

trim[1..24] – Output *

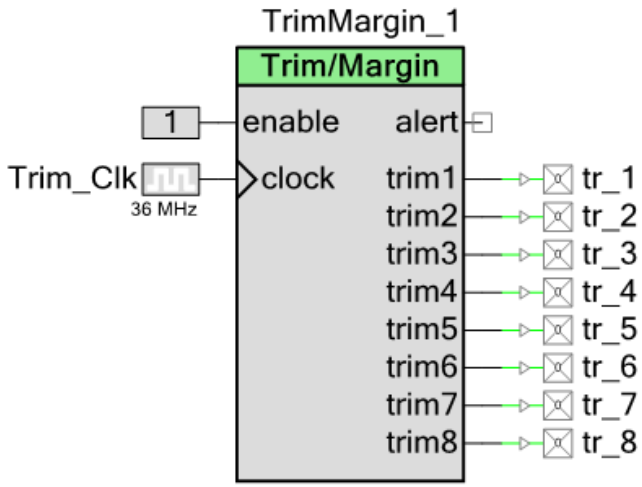
These terminals are the PWM outputs that pass through an external RC filter to produce an analog control voltage that adjusts the output voltage of the associated power converter. The number of these terminals depends on **Number of converters** parameter.

Schematic Macro Information

This section contains pertinent information regarding the Trim and Margin component schematic macros.

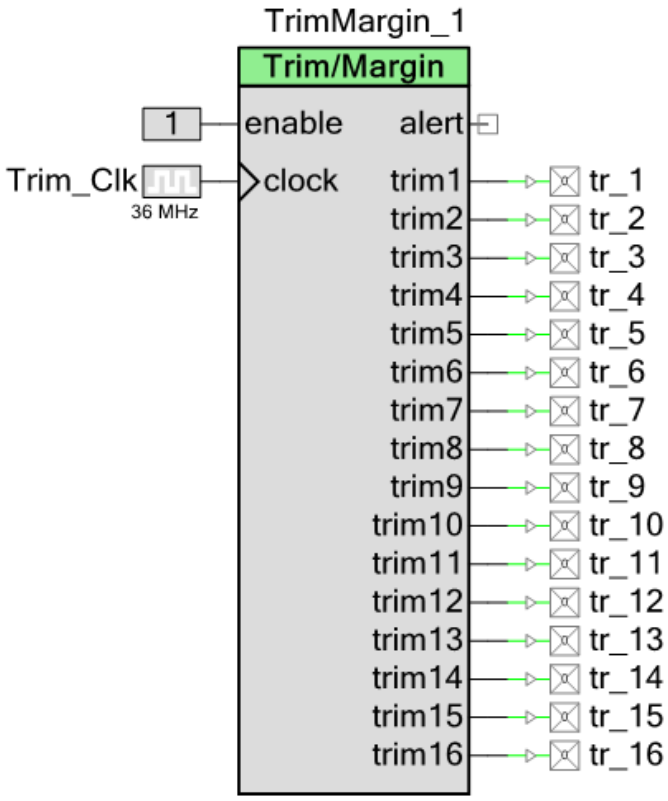
Trim and Margin - 8 Rails

The macro supports 8 PWM outputs at 8-bit resolution. Clock input is set to 36 MHz.



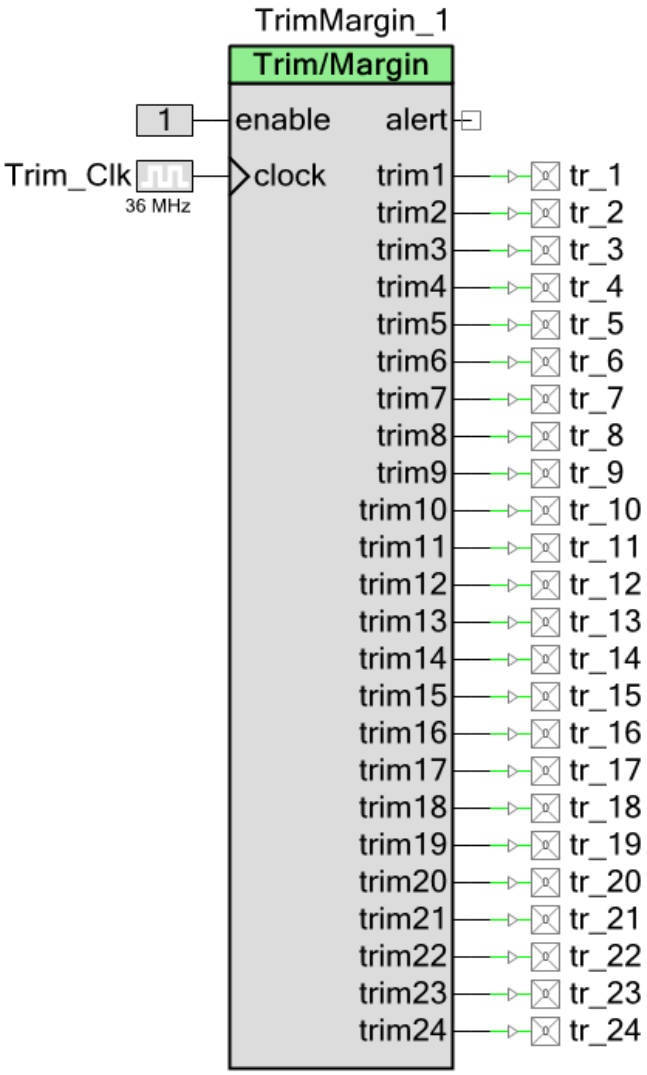
Trim and Margin - 16 Rails

The macro supports 16 PWM outputs at 8-bit resolution. Clock input is set to 36 MHz.



Trim and Margin - 24 Rails

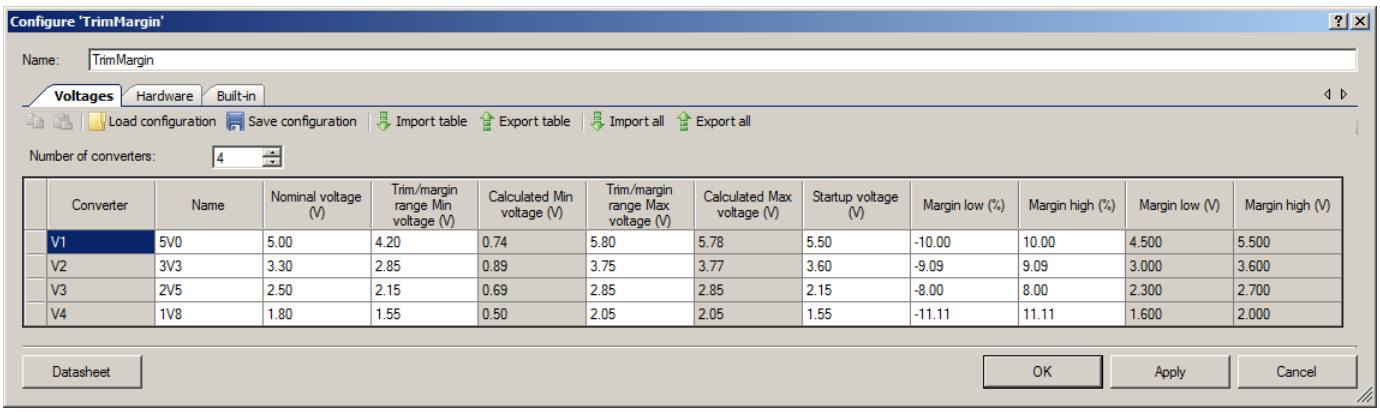
The macro supports 24 PWM outputs at 8-bit resolution. Clock input is set to 36 MHz.



Component Parameters

Drag a Trim and Margin component onto your design and double click it to open the Configure dialog. Figure 1 shows the Configure dialog.

Figure 1. Trim and Margin Configure Dialog



Common Commands

Load configuration

Restores all customizer settings, including tables, from an external file. Keyboard shortcut – [Ctrl] [L]

Save configuration

Stores all customizer settings, including tables, in an external file. Keyboard shortcut – [Ctrl] [S]

Import table

Imports data from file to table cells on active tab. Supports .csv file format. Keyboard shortcut – [Ctrl] [M]

Export table

Exports data from table cells active tab to file. Supports .csv file format. Keyboard shortcut – [Ctrl] [R].

Import all

Executes import functionality for **Voltages** and **Hardware** tables. Keyboard shortcut – [Ctrl] [Alt] [M]

Export all

Executes export functionality for **Voltages** and **Hardware** tables. Keyboard shortcut – [Ctrl] [Alt] [R].

Voltages Options

The voltages tab enables the user to describe the voltage characteristics of each power converter: nominal voltage, desired trim/margin dynamic range and margin low/high settings. The number of rows presented depends on the **Number of converters** parameter.

Number of converters

Number of power converters to trim or margin.

Name

Text field, 16 characters. For annotation purposes only to assist users to relate the PWM outputs of this component to the functions of the power converters they control.

Nominal voltage

Nominal converter output voltage. For annotation purposes only.

Trim/margin range Min voltage

Minimum converter output voltage. Impacts external components to achieve the low side of the desired dynamic range.

Calculated Min voltage

Calculated minimum voltage based on actual component values used as entered in the Hardware tab.

$$V_{\min calc[x]} = V_{nom[x]} - V_{resolution[x]} * \left(\left(2^{\wedge PWMresolution} - 1\right) - PWM_{dutycycle}\right)$$

Trim/margin range Max voltage

Maximum converter output voltage. Impacts external components to achieve the high side of the desired dynamic range.

Calculated Max voltage

Calculated maximum voltage based on actual component values used as entered in the Hardware tab.

$$V_{\max calc[x]} = V_{nom[x]} + V_{resolution[x]} * PWM_{dutycycle}$$

Startup voltage

User configurable startup voltage for the converter.

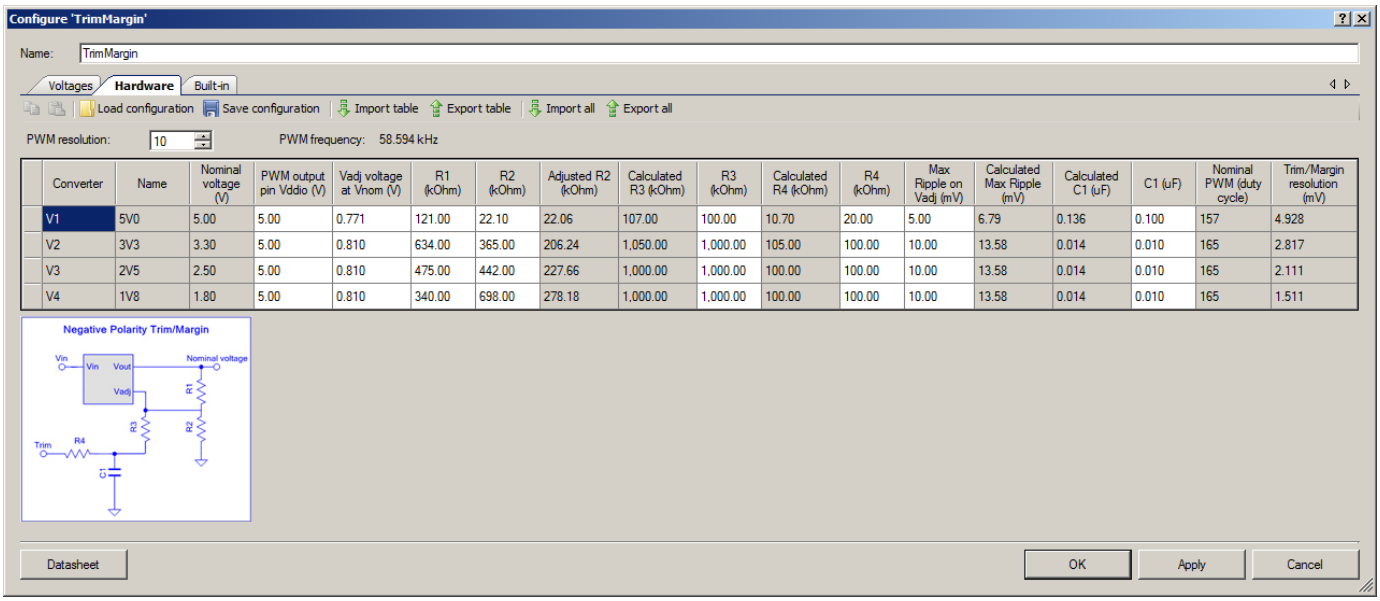
Margin low

Desired converter output voltage in response to a Margin Low request. This value has to be entered in percents of Nominal voltage and it is converted to volts in separate column.

Margin high

Desired converter output voltage in response to a Margin High request. This value has to be entered in percents of Nominal voltage and it is converted to volts in separate column.

Figure 2. Hardware Trim and Margin Dialog



Hardware Options

The hardware tab enables users to set hardware parameters such as the PWM resolution, PWM I/O supply voltage, power converter datasheet parameters and external circuitry.

The circuit diagram graphic at bottom is intended to help you relate the component names in the table to the components required on your schematic.

The number of rows presented depends on the **Number of converters** parameter.

The **Name** and **Nominal voltage** data propagates forward from the **Voltages** tab for reference. These are displays only that cannot be edited.

PWM resolution

Resolution of the PWM pseudo-DAC outputs. Selectable between 8 through 10 bits to enable users to select an optimal tradeoff between granularity of control voltage and PWM output frequency.

PWM frequency

Calculated PWM output frequency based on the input clock.

$$F_{pwm} = \frac{F_{Clock}}{2^{\wedge PWMresolution}}$$

PWM output pin Vddio

The Vddio voltage that will be used with the associated PWM output pin.

Vadj voltage at Vnom

Control voltage at the Adjust/Feedback control pin to achieve nominal output voltage.

R1

External scaling resistor value (in kΩ) required to achieve correct voltage on Adjust/Feedback control pin to achieve nominal output voltage when the PWM output is disabled. This value comes from user's PCB based on power converter datasheet specifications.

R2

External scaling resistor value (in kΩ) required to achieve correct voltage on Adjust/Feedback control pin to achieve nominal output voltage when the PWM output is disabled. This value comes from user's PCB based on power converter datasheet specifications.

Adjusted R2

The real value of the R2 could be different from the value entered by the user. It could be affected by internal power converter impedance which is parallel to R2. For example ADP3331 regulator has the $R3_{int}$ + $R4_{int}$ resistances. Therefore **$R2_{adj}$** is recalculated using following formula.

$$R_{2adj[x]} = \frac{V_{adj[x]} \times R_{1[x]}}{V_{nom[x]} - V_{adj[x]}}$$

Calculated R3

External control voltage summing resistor value (in kΩ). This value primarily controls the dynamic range capability of the PWM output. The value is calculated from the desired trim/margin range

min/max values specified in the voltages tab, control voltage (V_{adj}) and from the values of **R1** and **R2adj**.

$$(R_3 + R_4)_{calc[x]} = \frac{V_{adj[x]} \times R_{1[x]} \times R_{2adj[x]}}{\left(V_{\max[x]} \times R_{2adj[x]}\right) - V_{adj[x]} \times \left(R_{1[x]} + R_{2adj[x]}\right)}$$

This total resistance should be split between the **R3** and filtering resistor **R4**.

$$R_{3calc[x]} = (R_3 + R_4)_{calc[x]} - R_{4calc[x]}$$

The displayed value is closest 1% standard resistor value.

R3

User needs to enter the actual resistor value they will use on the PCB in order to calculate the actual dynamic range.

Calculated R4

External control voltage filtering resistor value (in kΩ). This value controls the cutoff frequency of the RC filter on the PWM output. The displayed value is closest 1% standard resistor value.

$$R_{4calc[x]} = \frac{(R_3 + R_4)_{calc[x]}}{11}$$

R4

User needs to enter the actual resistor value they will use on the PCB in order to calculate the actual ripple.

Max Ripple on Vadj

User can specify the maximum allowable ripple that appears on the output of the RC filter formed by **R4** and **C1**. Setting very low values for this parameter will result in large component values for **C1**.

Calculated Max Ripple

Shows expected ripple based on actual values entered for **R4** and **C1**.

$$V_{MaxRipple\_calc[x]} = \frac{V_{ddio[x]}}{2 \times \pi \times R_{4[x]} \times F_{pwm} \times C_{1[x]}}$$

Calculated C1

External control voltage filtering capacitor value (in µF). This value controls the cutoff frequency of the RC filter on the PWM output. The value is calculated from the R4, PWM resolution entered in the Voltages tab and from the source clock frequency input to this component.

$$C_{1calc[x]} = \frac{V_{ddio[x]}}{2 \times \pi \times R_{4[x]} \times F_{pwm} \times V_{MaxRipple[x]}}$$

C1

User needs to enter the actual capacitor value they will use on the PCB in order to calculate the actual ripple.

Nominal PWM

Shows the calculated PWM duty cycle to achieve nominal voltage.

$$PWM_{dutycycle} = \frac{V_{adj[x]}}{V_{ddio[x]} \times 2^{\wedge PWMresolution}}$$

Trim/Margin resolution

This parameter means how much the power converter output voltage will change as a result of a change in duty cycle of one step. It is calculated by the following formulas.

$$V_{resolution[x]} = \frac{V_{\max real[x]} - V_{nom[x]}}{PWM_{dutycycle}}$$

Where the real maximum output voltage (**$V_{maxreal}$**) is recalculated based on actual **R3** and **R4** entered by user and recalculated **$R2_{adj}$**.

$$V_{\max real[x]} \frac{V_{adj[x]} \times R_{1[x]}}{R_{2adj_{[x]}} \,||\, (R_{3[x]} + R_{4[x]})} + V_{adj[x]}$$

Clock Selection

There is no internal clock in this component. You must attach a clock source. This component operates from a single clock connected to the component.

Application Programming Interface

Application Programming Interface (API) routines allow you to configure and control the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name "TrimMargin_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. For readability, the instance name used in the following table is "TrimMargin".

Functions

Function	Description
TrimMargin_Start()	Enables the component
TrimMargin_Stop()	Disables the component
TrimMargin_Init()	Initializes component's parameters
TrimMargin_Enable()	Enables outputs and starts the PWMs
TrimMargin_SetMarginHighVoltage()	Sets the margin high output voltage parameter
TrimMargin_GetMarginHighVoltage()	Returns the margin high output voltage parameter
TrimMargin_SetMarginLowVoltage()	Sets the margin low output voltage parameter
TrimMargin_GetMarginLowVoltage()	Returns the margin low output voltage parameter
TrimMargin_ActiveTrim()	Adjusts the PWM duty cycle of the specified power converter to get the power converter actual voltage output closer to the desired voltage output
TrimMargin_SetDutyCycle()	Sets PWM duty cycle of the PWM associated with the specified power converter
TrimMargin_GetDutyCycle()	Gets the current PWM duty cycle of the PWM associated with the specified power converter
TrimMargin_GetAlertSource()	Returns a bit mask indicating which PWMs are generating an alert
TrimMargin_MarginLow()	Sets power converter output voltage to the Margin low voltage
TrimMargin_MarginHigh()	Sets power converter output voltage to the Margin high voltage
TrimMargin_SetNominal()	Sets power converter output voltage to the Nominal voltage
TrimMargin_SetPreRun()	Sets the pre-charge PWM duty cycle required to achieve nominal voltage before the power converter is enabled
TrimMargin_SetStartup()	Sets power converter output voltage to the Startup voltage
TrimMargin_SetStartupPreRun()	Sets the pre-charge PWM duty cycle to achieve the Startup voltage before power converter is enabled
TrimMargin_ConvertVoltageToDutyCycle()	Returns the PWM duty cycle required to achieve the desired voltage on the selected power converter
TrimMargin_ConvertVoltageToPreRunDutyCycle()	Returns the pre-charge PWM duty cycle required to achieve the desired voltage on the selected power converter

Global Variables

Function	Description
TrimMargin_initVar	The initVar variable is used to indicate initial configuration of this component. This variable is prepended with the component name. The variable is initialized to zero and set to 1 the first time TrimMargin_Start() is called. This allows for component initialization without reinitialization in all subsequent calls to the TrimMargin_Start() routine.
TrimMargin_vMarginLow[]	Margin low output voltage parameter. It is initialized by Init() function to the Margin low value entered in the Voltages Tab of the customizer and could be changed by TrimMargin_SetMarginLowVoltage() function.
TrimMargin_vMarginHigh[]	Margin high output voltage parameter. It is initialized by Init() function to the Margin high value entered in the Voltages Tab of the customizer and could be changed by TrimMargin_SetMarginHighVoltage() function.
TrimMargin_vMarginLowDutyCycle[]	Precalculated PWM duty cycle for Margin low voltage copied from ROM in the Init() function. These values are recalculated by TrimMargin_SetMarginLowVoltage() function when new margin value is set. Used by MarginLow() to set PWM for open loop margin.
TrimMargin_vMarginHighDutyCycle[]	Precalculated PWM duty cycle for Margin high voltage copied from ROM in the Init() function. These values are recalculated by TrimMargin_SetMarginHighVoltage() function when new margin value is set. Used by MarginHigh() to set PWM for open loop margin.

void TrimMargin_Start(void)

Description: Enables the component. Calls the Init() API if the component has not been initialized before. Calls Enable() API.

Parameters: None

Return Value: None

Side Effects: None

void TrimMargin_Stop(void)

Description: Disables the component. Stops the PWMs.

Parameters: None

Return Value: None

Side Effects: trim[x] outputs halted in undefined state. Use the pin-specific API PinName_SetDriveMode(PIN_DM_DIG_HIZ) to change the drive mode of the connected to these outputs pins to High Impedance Digital.

void TrimMargin_Init(void)

Description: Initialize component's parameters to the parameters set by user in the customizer of the component placed onto schematic. Usually called in TrimMargin_Start() API.

PWM duty cycles are set to a “pre-run” target that assumes the power converters are not yet turned on (disabled).

Parameters: None

Return Value: None

Side Effects: None

void TrimMargin_Enable(void)

Description: Enables and starts the PWMs.

Parameters: None

Return Value: None

Side Effects: None

void TrimMargin_SetMarginHighVoltage(uint8 converterNum, uint16 marginHiVoltage)

Description: Sets the margin high output voltage parameter of the specified power converter. This function overrides the TrimMargin_vMarginHigh[x] setting made in the customizer on the Voltages Tab and recalculate TrimMargin_vMarginHighDutyCycle[x] to be ready for using by TrimMargin_MarginHigh() macro. Note that calling this API does not cause any change in the PWM output duty cycle.

Parameters: uint8 converterNum. Specifies the power converter number

Valid range: 1..24

uint16 marginHiVoltage. Specifies the desired power converter output margin high voltage in mV

Valid range: 1..12,000

Return Value: None

Side Effects: None

uint16 TrimMargin_GetMarginHighVoltage(uint8 converterNum)

Description: Returns the margin high output parameter of the specified power converter

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: uint16: Specifies the desired power converter margin high output voltage in mV

Side Effects: None

void TrimMargin_SetMarginLowVoltage(uint8 converterNum, uint16 marginLoVoltage)

Description: Sets the margin low output voltage parameter of the specified power converter. This function overrides the TrimMargin_vMarginLow[x] setting made in the customizer on the Voltages Tab and recalculate TrimMargin_vMarginLowDutyCycle[x] to be ready for using by TrimMargin_MarginLow() macro. Note that calling this API does not cause any change in the PWM output duty cycle.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

uint16 marginLoVoltage. Specifies the desired power converter output margin low voltage in mV
Valid range: 1..11,999

Return Value: None

Side Effects: None

uint16 TrimMargin_GetMarginLowVoltage(uint8 converterNum)

Description: Returns the margin low output parameter of the specified power converter

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: uint16: Specifies the desired power converter margin low output voltage in mV

Side Effects: None

void TrimMargin_ActiveTrim(uint8 converterNum, uint16 actualVoltage, uint16 desiredVoltage)

Description: This API adjusts the PWM duty cycle of the specified power converter to get the power converter actual voltage output closer to the desired voltage output. It needs to be called on a regular basis to ensure proper closed-loop regulation is achieved

Parameters: uint8 converterNum. Specifies the power converter number

Valid range: 1..24

uint16 actualVoltage

Specifies the current actual power converter output voltage reading in mV. This value can be obtained using the Power Monitor component connected to the power converter output voltage

Valid range: 1..12,000

uint16 desiredVoltage: Specifies the desired power converter output voltage in mV

Valid range: 1..12,000

Return Value: None

Side Effects: Calling this API may result in a change of PWM duty cycle driving the control voltage of the selected power converter causing a change in power converter output voltage.

If the desiredVoltage cannot be achieved because the PWM duty cycle is at min or max level, the alert signal will be asserted until the alert condition is removed, only possible by calling this API with an achievable desiredVoltage.

void TrimMargin_SetDutyCycle(uint8 converterNum, uint8/uint16 dutyCycle)

Description: Sets PWM duty cycle of the PWM associated with the specified power converter. This API can be used for open-loop trimming or margining purposes. The PWM period is always fixed at the maximum value depending on the resolution set in the customizer.

Parameters: uint8 converterNum. Specifies the power converter number

Valid range: 1..24

uint8/uint16 dutyCycle. Specifies the PWM duty cycle in PWM clock counts

Valid range: 0..255 up to 0..1023 depending on the resolution set in the customizer

Return Value: None

Side Effects: None

uint8/uint16 TrimMargin_GetDutyCycle(uint8 converterNum)

Description: Gets the current PWM duty cycle of the PWM associated with the specified power converter. Note that if the TrimMargin_ActiveTrim() API is being called regularly, the value returned should be expected to change over time.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: uint8/uint16. Specifies the PWM duty cycle in PWM clock counts
Valid range: 0..255 up to 0..1023 depending on the resolution set in the customizer

Side Effects: None

uint8/uint16/uint32 TrimMargin_GetAlertSource(void)

Description: Returns a bit mask indicating which PWMs are generating an alert

Parameters: None

Return Value: uint8/uint16/uint32.

Bit Field	Alert Source
bit0	1 = Failure to achieve power converter regulation on trim1 output
bit1	1 = Failure to achieve power converter regulation on trim2 output
…	…
bit23	1 = Failure to achieve power converter regulation on trim24 output

Side Effects: None

void TrimMargin_MarginLow(uint8 converterNum)

Description: Sets the selected power converter output voltage to the desired Margin low setting as specified in the Voltages Tab of the customizer or as per the SetMarginLowVoltage() API.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: None

Side Effects: None

void TrimMargin_MarginHigh(uint8 converterNum)

Description: Sets the selected power converter output voltage to the desired Margin high setting as specified in the Voltages Tab of the customizer or as per the SetMarginHighVoltage() API.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: None

Side Effects: None

void TrimMargin_SetNominal(uint8 converterNum)

Description: Sets the selected power converter output voltage to the Nominal Voltage setting as specified in the Voltages Tab of the customizer.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: None

Side Effects: None

void TrimMargin_SetPreRun(uint8 converterNum)

Description: Sets the pre-charge PWM duty cycle required to achieve nominal voltage before power converter is enabled with the assumption that the R1 is grounded in parallel with R2.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: None

Side Effects: None

void TrimMargin_SetStartup(uint8 converterNum)

Description: Sets the selected power converter output voltage to the Startup Voltage setting as specified in the Voltages Tab of the customizer.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: None

Side Effects: None

void TrimMargin_SetStartupPreRun(uint8 converterNum)

Description: Sets the pre-charge PWM duty cycle required to achieve the Startup Voltage before the power converter is enabled with the assumption that R1 is grounded in parallel with R2.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24

Return Value: None

Side Effects: None

void TrimMargin_ConvertVoltageToDutyCycle(uint8 converterNum, uint16 desiredVoltage)

Description: Returns the PWM duty cycle required to achieve the desired voltage on the selected power converter.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24
uint16 desiredVoltage. Specifies the desired power converter output voltage in mV

Return Value: None

Side Effects: None

void TrimMargin_ConvertVoltageToPreRunDutyCycle(uint8 converterNum, uint16 desiredVoltage)

Description: Returns the pre-charge PWM duty cycle required to achieve the desired voltage on the selected power converter.

Parameters: uint8 converterNum. Specifies the power converter number
Valid range: 1..24
uint16 desiredVoltage. Specifies the desired power converter output voltage in mV

Return Value: None

Side Effects: None

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. Project deviations are described in the MISRA Compliance section of the *System Reference Guide* along with information on the MISRA compliance verification environment.

The Trim and Margin component has the following specific deviations:

MISRA-C: 2004 Rule	Rule Class (Required/ Advisory)	Rule Description	Description of Deviation(s)
19.7	A	A function should be used in preference to a function-like macro.	Deviated since function-like macros are used to allow more efficient code.

This component has the following embedded component: ControlReg. Refer to the corresponding component datasheet for information on their MISRA compliance and specific deviations.

Sample Firmware Source Code

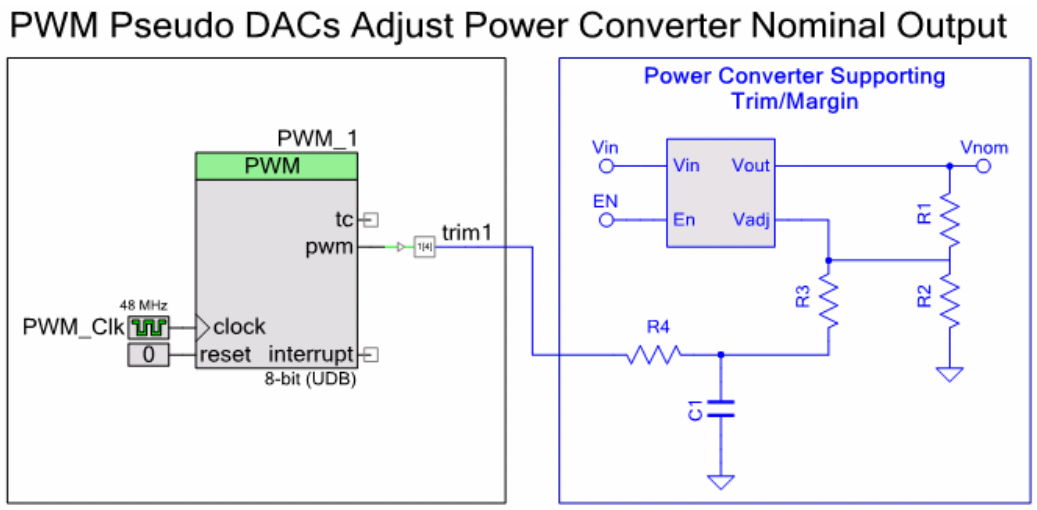
PSoC Creator provides numerous example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start 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 Example Project" topic in the PSoC Creator Help for more information.

Functional Description

The component is built from an array of 8-bit to 10-bit PWMs. The PWM outputs from PSoC are RC filtered to generate analog control voltages that connect to the “adjust”, “sense” or “feedback” point of adjustable power converters through a summing resistor. This is shown in the figure below.

This is a negative feedback control loop whereby increasing the PWM duty cycle increases the analog control voltage which in turn results in a decrease of power converter output voltage. Conversely, a decrease in PWM duty cycles decreases the analog control voltage which in turn results in an increase of power converter output voltage.



This block needs to comprehend the adjust or feedback control voltage level and the values of feedback resistors R1 and R2 required to achieve nominal output voltage in order to power up in a good configuration without adversely affecting the power converter outputs. This information can be found in the power converter datasheet.

The summing resistor R3 and the RC filter values R4 and C1 are recommended to the user based on the parameter settings in the hardware tab of the configuration dialog.

Resources

The Trim and Margin component is placed throughout the UDB array. The component utilizes the following resources.

Configuration	Resource Type					
	Datapath Cells	Macrocells	Status Cells	Control Cells	DMA Channels	Interrupts
4 Output TrimMargin (10-bit)	4	1	–	1	–	–
24 Output TrimMargin (8-bit)	12	1	–	1	–	–

API Memory Usage

The component memory usage varies significantly, depending on the compiler, device, number of APIs used and component configuration. The following table provides the memory usage for all APIs available in the given component configuration.

The measurements have been done with associated compiler configured in Release mode with optimization set for Size. For a specific design the map file generated by the compiler can be analyzed to determine the memory usage.

Configuration	PSoC 3 (Keil_PK51)		PSoC 5LP (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
4 Output TrimMargin	1709	42	1006	45
24 Output TrimMargin	2472	197	1982	197

DC and AC Electrical Characteristics

Specifications are valid for –40 °C ≤ T_A ≤ 85 °C and T_J ≤ 100 °C, except where noted.
Specifications are valid for 1.71 V to 5.5 V, except where noted.

DC Characteristics

Parameter	Description	Min	Typ[1]	Max	Units
$I_{DD(}$	Component current consumption				
	8-bit One trim output	–	2.5	–	µA/MHz
	9-bit or 10-bit One trim output	–	4	–	µA/MHz

AC Characteristics

Parameter	Description	Min	Typ	Max[2]	Units
f_{CLOCK}	Component clock frequency [3]				
	4 Output TrimMargin(10-bit)	–	–	50	MHz
	24 Output TrimMargin(8-bit)	–	–	55	MHz

1. Device IO and clock distribution current not included. The values are at 25 °C.

2. The values provide a maximum safe operating frequency of the component. The component may run at higher clock frequencies, at which point you will need to validate the timing requirements with STA results.

3. The maximum component clock frequency depends on the selected mode and additional features.

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
1.20.a	Edited datasheet to remove references to PSoC 5.	PSoC 5 has been replaced by the PSoC 5LP.
1.20	Added MISRA Compliance section.	The component has specific deviations described.
	Added PWM frequency field in the customizer Hardware tab. Added Load / Save configuration commands.	Usability enhancement.
	Customizer updated with calculated values for Vmax, Vmin, R2, R3, R4, Max Ripple, C1, Nominal PWM.	Usability enhancement.
	Added Startup voltage column in the customizer Voltages tab and following APIs to set up these voltages: TrimMargin_SetStartup(), TrimMargin_SetStartupPreRun().	Ability to start up to the custom voltage.
	Added following APIs to allow converting voltage to PWM duty cycles: TrimMargin_ConvertVoltageToDutyCycle() TrimMargin_ConvertVoltageToPreRunDutyCycle()	Provide support for trimming to an arbitrary voltage.
1.10	Updated Macro names and configuration.	
	Import All and Export All functions import/export all tables to/from one single CSV file, instead of several files. Changed CSV format to use "," as the separator.	This makes it easier for users to manually edit the component configuration.
	Trim/Margin Resolution column added to the Hardware tab.	This parameter indicates how much the power converter output voltage will change as a result of a change in duty cycle of one step.
1.0	Initial version of the component.	

© Cypress Semiconductor Corporation, 2013. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control, or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.