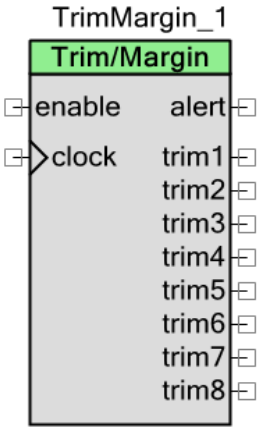
Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

Trim and Margin

3.0



Features

- Works with most adjustable DC-DC converters or regulators, including low-dropouts (LDOs), switchers, and modules
- Supports positive and negative feedback control loop
- 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 or ADC 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 enter the power converter nominal output voltages, voltage trimming range, margin high and margin low settings into the intuitive, graphical configuration GUI and the component calculates all the required parameters for injecting a pulse width modulated signal into the feedback network of a power converter. 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 Component

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 MarginTrim and Margin component along with other Power Supervision components to build your own custom power supervision 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 the minimum or maximum duty cycle, but desired power converter output voltage has not been achieved. Remains asserted as long as the alert condition exists on any output.

Note This terminal is only available on devices that contain the UDB array. You still can monitor the alert status in firmware by calling TrimMargin_GetAlertSource() API function.

trim[1..24] – Output *

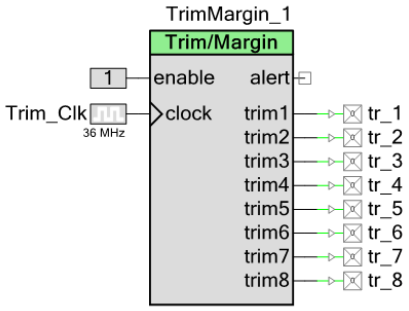
These terminals are the PWM outputs that pass through an external RC filter to produce an analog control voltage that adjusts the output voltage of the associated power converter. The number of these terminals depends on Number of converters parameter.

Schematic Macro Information

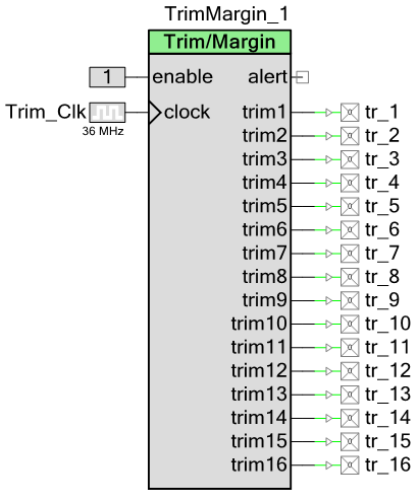
This section contains pertinent information regarding the Trim and Margin component schematic macros.

Schematic Macro Name	Purpose	Description
Trim and Margin – 8 Rails	Intended to be competitive with most off-the-shelf, 8 rail Power Supervisor ASSPs.	Supports 8 outputs and the component is configured for 8-bit resolution. Clock input is set to 24 MHz.
Trim and Margin – 16 Rails	Intended to be competitive with newer off-the-shelf, 16 rail Power Supervisor ASSPs	Supports 16 outputs and the component is configured for 8-bit resolution. Clock input is set to 24 MHz.
Trim and Margin – 24 Rails	Intended to demonstrate PSoC's ability to implement a full-featured Power Supervisor with support for more power converters than any other competitor.	Supports 24 outputs and the component is configured for 8-bit resolution. Clock input is set to 24 MHz

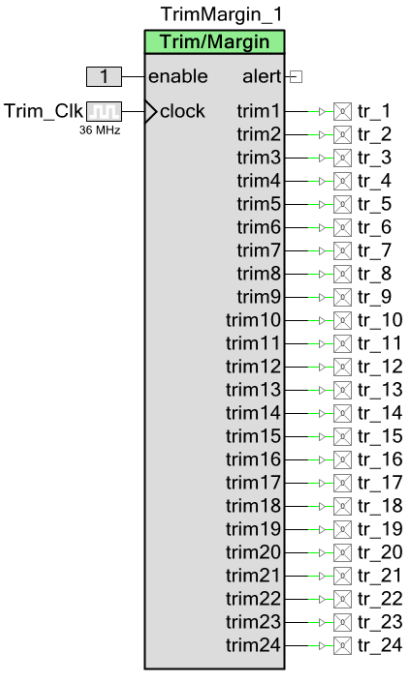
Trim and Margin - 8 Rails



Trim and Margin - 16 Rails



Trim and Margin - 24 Rails

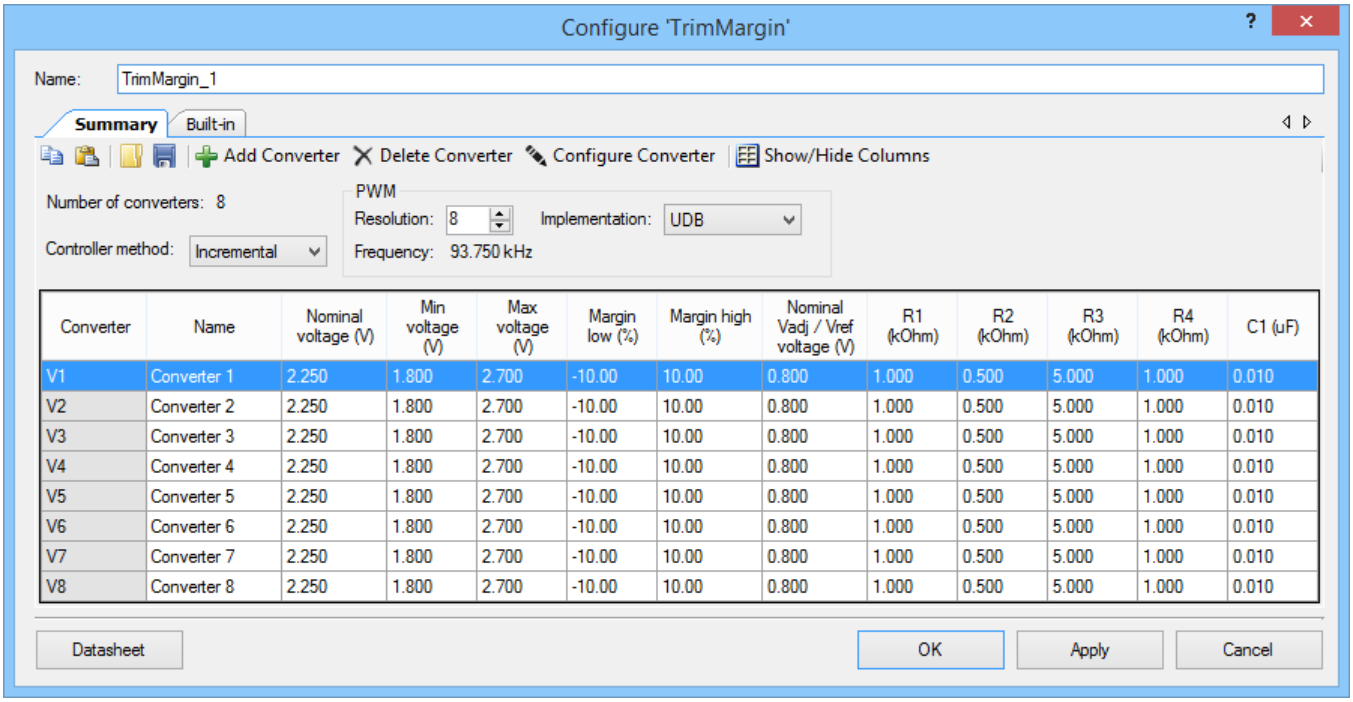


Component Parameters

Drag a Trim and Margin component onto your design and double click it to open the Configure dialog. This dialog has the following tabs with different parameters.

Summary Tab

The summary tab shows a summary of all the rails that requires trim/margin. The table is read only. Double click the table or press the “Add Converter”, “Delete Converter” or “Configure” to change it.



Converter	Name	Nominal voltage (V)	Min voltage (V)	Max voltage (V)	Margin low (%)	Margin high (%)	Nominal Vadj / Vref voltage (V)	R1 (kOhm)	R2 (kOhm)	R3 (kOhm)	R4 (kOhm)	C1 (uF)
V1	Converter 1	2.250	1.800	2.700	-10.00	10.00	0.800	1.000	0.500	5.000	1.000	0.010
V2	Converter 2	2.250	1.800	2.700	-10.00	10.00	0.800	1.000	0.500	5.000	1.000	0.010
V3	Converter 3	2.250	1.800	2.700	-10.00	10.00	0.800	1.000	0.500	5.000	1.000	0.010
V4	Converter 4	2.250	1.800	2.700	-10.00	10.00	0.800	1.000	0.500	5.000	1.000	0.010
V5	Converter 5	2.250	1.800	2.700	-10.00	10.00	0.800	1.000	0.500	5.000	1.000	0.010
V6	Converter 6	2.250	1.800	2.700	-10.00	10.00	0.800	1.000	0.500	5.000	1.000	0.010
V7	Converter 7	2.250	1.800	2.700	-10.00	10.00	0.800	1.000	0.500	5.000	1.000	0.010
V8	Converter 8	2.250	1.800	2.700	-10.00	10.00	0.800	1.000	0.500	5.000	1.000	0.010

Toolbar

- Copy/Paste – Copies the selected row in the table and pastes the same data on the table. Keyboard shortcut (copy) – [Ctrl] [C] Keyboard shortcut (paste) – [Ctrl] [V]
- Load configuration – Restores all customizer settings, including table, from an external file. Supports XML file format. Keyboard shortcut – [Ctrl] [L]
- Store configuration – Stores all customizer settings, including tables, in an external XML file. Keyboard shortcut – [Ctrl] [S]
- Add converter – Adds a converter in the summary table. A dialog window opens to input the converter configuration.
- Delete converter – Deletes a converter in the summary table. You need to select the converter to be deleted. If no converter is selected, this button is grayed out.

- Configure converter – Opens a configuration window for the selected power converter. If no converter is selected, this button is grayed out.
- Show/Hide columns – Opens a list of all the columns shown in the table. You can choose the columns to display.

Number of converters

Number of converters to trim and margin. The number increases or decreases when adding and deleting rails from the table.

Controller method

Use this parameter to choose between the following types of controllers when actively trimming: **Incremental** or **Integral**.

- The **Incremental** controller is limited to increment or decrement based on the difference between actual and desired voltage and the converter feedback. The response of this controller is very slow when the desired voltage is significantly different from an actual voltage.
- The integral controller (I Control) is a type of a PID Controller with the P (proportional) and D (derivative) gain terms set to zero. This type of controller has better response time compared to the **Incremental** option. Refer to Functional Description section for more information about the **Integral** controller method.

The default setting is **Incremental**.

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. This configuration applies to all rails. Range = 8, 9, 10 bits (default 8).

PWM Frequency

Calculated PWM output frequency based on the input clock and PWM resolution.

$f_{PWM} = \frac{f_{CLOCK}}{2^n}$, n is PWM resolution.

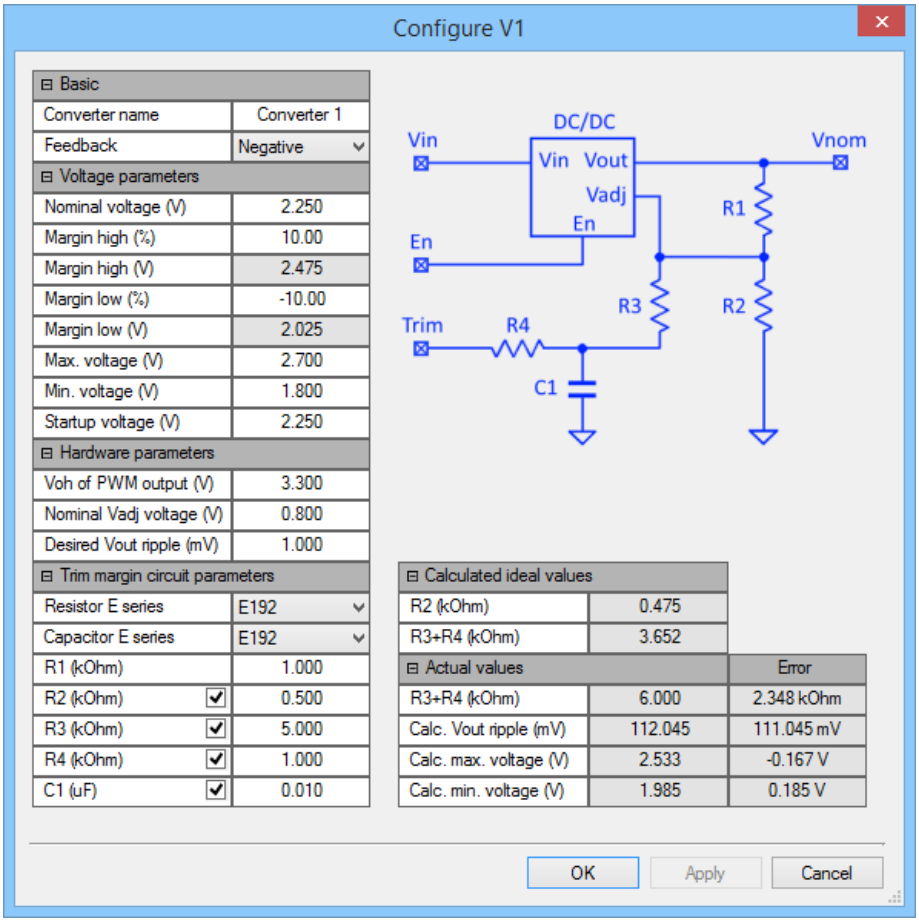
PWM Implementation

Use this parameter to choose between a fixed-function timer/counter/PWM (TCPWM) implementation and a UDB implementation of the internal PWMs. If the target device does not have a UDB array, the **UDB** option cannot be selected. If the device does not have any TCPWM

blocks, the **Fixed Function** option cannot be selected. This parameter is only applicable for the PSoC 4 device family.

Configure Window

This window is opened by pressing the **Add Converter** or **Configure** button, or by double-clicking the desired converter row in the summary table.



Converter name

This is a text field, 16 characters. For annotation purposes only to relate the PWM outputs of this component to the functions of the power converters they control.

Feedback

This parameter determines the voltage adjust feedback input polarity from the power converter. This will decide if increasing the PWM duty cycle will increase (Positive) or decrease (Negative) the converter output voltage. Options = Positive, Negative (default).

Control sample time

This is the period time in milliseconds for how often this power converter is trimmed, that is, the TrimMargin_ActiveTrim() API is called for this converter. This time has to be on average three times longer than the time it takes to scan the voltage output of the converter (filters should also be taken in consideration).

For example, the Power Monitor takes in average 150 µs per rail to measure the voltage. If the Power Monitor is configured to scan 16 rails and use an average filter of 8 samples, the recommended control sample time is:

$$ct = 150\,\mu s \times 8 \times 16 \times 3 = 57.6\,ms$$

This parameter is only available when the **Integral** option is selected for the Controller method parameter. Range = 1..65,535 ms. Default value is 100 ms.

Nominal voltage (V)

Nominal converter output voltage. Range = 0.001..65.535 V.

Margin low (%)

Desired converter output voltage in response to a margin low request. This value is entered in percent of nominal voltage. The calculated value is displayed below in the **Margin low (V)** field. Range = -100.00..0 %.

Margin high (%)

Desired converter output voltage in response to a margin high request. This value is entered in percent of nominal voltage. The calculated value is displayed below in the **Margin high (V)** field. Range = 0..100.00 %.

Max. voltage (V)

Maximum converter output voltage. Impacts external components to achieve the high side of the desired dynamic range. Range = 0.001..65.535 V.

Min. voltage (V)

Minimum converter output voltage. Impacts external components to achieve the low side of the desired dynamic range. Range = 0.001..65.535 V.

Startup voltage (V)

User configurable startup voltage for the converter. The PWM duty cycle is set to pre-run value to achieve the startup voltage when the component is initialized by calling the TrimMargin_Init() API. Range = 0.001..65.535 V.

Voh of PWM output (V)

The output voltage that will be used with the associated PWM pin. It can be the Vddio for a GPIO pin or an internal Vref for SIO pin. Range = 0.001..5.5 V.

Nominal Vadj voltage

Control voltage at the adjust/feedback control pin to achieve nominal output voltage based on power converter datasheet specification. Range = 0.001..65.535 V.

Desired Vout ripple

Specifies the maximum ripple that appears on the output of the voltage regulator. The ripple effect comes from the RC circuit designed to trim/margin the converter.
Range = 0.001..100.000 mV.

Resistor E Series

Specifies preferred E series for the external resistors on your PCB. The calculated resistor values will be rounded up to the closest nominal values from the specified E series.
Range = E24, E48, E96 or E192 (default).

Capacitor E Series

Specifies preferred E series for the external capacitor on your PCB. The calculated capacitor value will be rounded up to the closest nominal value from the specified E series. Range = E24, E48, E96 or E192 (default).

R1 (kOhm)

External scaling resistor value (in kΩ) to achieve the correct voltage on adjust/feedback control pin required for the nominal output voltage when the PWM output is disabled. This value comes from user's PCB based on power converter datasheet specifications.
Range = 0.001..100,000 kΩ.

R2 (kOhm) / Calculated R2 (kOhm)

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

If the checkbox on the right side is unchecked, the value of R2 is calculated automatically and overrides the current R2 value. The value belongs to the specified Resistor E series.

Range = 0.001..65,535 kΩ.

R3 (kOhm)/ Calculated R3 (kOhm)

The actual resistor value used on the PCB in order to calculate the actual dynamic range.

If the checkbox on the right side is unchecked, the value of R3 is calculated automatically and overrides the current R3 value. The value belongs to the specified Resistor E series. Range = 0.001..65,535 kΩ.

It is recommended to have $R3 > 10 \times R4$ to avoid instability on the converter response. Combining this recommendation with $R_{serial} = R3 + R4$, R3 is calculated as $R3 = R_{serial} \times 10/11$.

R4 (kOhm) / Calculated R4 (kOhm)

The actual resistor value used on the PCB in order to calculate the actual dynamic range.

If the checkbox on the right side is unchecked, the value of R4 is calculated automatically and overrides the current R4 value. The value belongs to the specified Resistor E series.

Range = 0.001..65,535 kΩ.

The resistor R4 is calculated as $R4 = R_{serial} - R3$.

C1 (µF) / Calculated C1 (µF)

The actual capacitor value use on the PCB in order to calculate the actual ripple.

If the checkbox on the right side is unchecked, the value of C1 is calculated automatically and overrides the current C1 value. The value belongs to the specified Capacitor E series. Range = 0.001..65.535 µF. The capacitor C1 is calculated as:

$$C1 = \frac{V_{OH}}{2\pi \times R4 \times f_{PWM} \times V_{ripple}} \times \frac{R1}{R3}$$

Calculated ideal value for R2 (kOhm)

The ideal value for R2 depends on the Feedback parameter and is calculated as:

Negative feedback: $$R2 = \frac{R1 \times V_{ADJ} \times R_{serial}}{R1 \times (V_{OH} - V_{ADJ}) + R_{serial} \times (V_{MIN} - V_{ADJ})}$$

Positive feedback: $$R2 = \frac{R1 \times V_{OH} \times V_{MIN}}{V_{OH} \times (V_{REF} - V_{MIN}) - V_{REF} \times (V_{MAX} - V_{MIN})}$$

Calculated ideal value for R3 + R4 (kOhm)

The value is dependent on the Feedback parameter and is calculated as:

Negative feedback: $$R3 + R4 = R_{serial} = R1 \times \left(\frac{V_{OH}}{V_{MAX} - V_{MIN}} \right)$$

Positive feedback: $R3 + R4 = R_{serial} = \frac{V_{OH} \times R1}{\left(\frac{V_{MAX}}{V_{MIN}} - 1\right) \times V_{REF}}$

Actual R3+R4 (kOhm)

Sum of the actual resistance values selected in the customizer for R3 and R4.

Actual Vout ripple (mV)

The actual value for Vout ripple is calculated as:

$$V_{ripple} = \frac{V_{DDIO}}{2\pi \times R4 \times f_{PWM} \times C1} \times \frac{R1}{R3}$$

Calculated Max. voltage (V)

Depending on the Feedback parameter, the actual value of the maximum output voltage for the given circuit is calculated as:

Negative feedback: $V_{MAX} = V_{ADJ} \times \left(1 + \frac{R1}{R2} + \frac{R1}{R_{serial}}\right)$

Positive feedback: $V_{MAX} = \frac{\frac{V_{REF}}{R1} + \left(\frac{V_{OH}}{R_{serial}} \times \frac{2^n - 1}{2^n}\right)}{\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R_{serial}}}$

Calculated Min. voltage (V)

The actual value of the minimum output voltage for the given circuit depends on the Feedback parameter, and is calculated as:

Negative feedback: $V_{MIN} = V_{ADJ} \times \left(1 + \frac{R1}{R2} + \frac{R1}{R_{serial}}\right) - V_{OH} \times \left(\frac{R1}{R_{serial}} \times \frac{2^n - 1}{2^n}\right)$

where *n* is PWM resolution.

Positive feedback: $V_{MIN} = \frac{\frac{V_{REF}}{R1}}{\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R_{serial}}}$

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 tables is "TrimMargin".

Functions

Function	Description
TrimMargin_Start()	Starts the component operation.
TrimMargin_Stop()	Disables the component.
TrimMargin_Init()	Initializes component's parameters.
TrimMargin_Enable()	Enables the generation of PWMs outputs.
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_SetNominalVoltage()	Sets the nominal output voltage parameter.
TrimMargin_GetNominalVoltage()	Returns the nominal 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_MarginHigh()	Sets power converter output voltage to the Margin high voltage.
TrimMargin_MarginLow()	Sets power converter output voltage to the Margin low voltage.
TrimMargin_Nominal()	Sets power converter output voltage to the Nominal voltage.

Function	Description
TrimMargin_PreRun()	Sets the pre-charge PWM duty cycle required to achieve nominal voltage before the power converter is enabled.
TrimMargin_Startup()	Sets power converter output voltage to the Startup voltage.
TrimMargin_StartupPreRun()	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.
TrimMargin_SetTrimCycleCount()	Set the internal trim cycle counter that affects how often the PWM duty cycle is updated when calling TrimMargin_ActiveTrim() API. Applicable for Incremental controller type only.

void TrimMargin_Start(void)

Description: Starts the component operation. Calls the TrimMargin_Init() API if the component has not been initialized before. Calls TrimMargin_Enable() API.

Parameters: None

Return Value: None

Side Effects: None

void TrimMargin_Stop(void)

Description: Disables the component. Stops the PWMs.

trim[x] outputs halted in an 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.

Parameters: None

Return Value: None

Side Effects: None

void TrimMargin_Init(void)

Description: Initializes component's parameters to those set in the customizer.

It is not necessary to call TrimMargin_Init() because the TrimMargin_Start() routine calls this function, which is the preferred method to begin the component operation. PWM duty cycles are set to pre-run values to achieve the startup voltage target assuming that the power converters are not yet turned on (disabled).

Parameters: None

Return Value: None

Side Effects: None

void TrimMargin_Enable(void)

Description: Enables PWMs outputs generation.

Parameters: None

Return Value: None

Side Effects: None

void TrimMargin_SetMarginHighVoltage(uint8 converterNum, uint16 marginHiVoltage)

Description: Sets the margin high output voltage of the specified power converter.

This overrides the present vMarginHigh[x] setting and recalculates vMarginHighDutyCycle[x] to be ready for use by TrimMargin_MarginHigh(). Note: calling this API does NOT cause any change in the PWM output duty cycle.

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

marginHiVoltage: Specifies the desired power converter output margin high voltage in mV. Valid range: 1..65,535

Return Value: None

Side Effects: None

uint16 TrimMargin_GetMarginHighVoltage(uint8 converterNum)

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

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

Return Value: 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 of the specified power converter.

This overrides the present vMarginLow[x] setting and recalculates vMarginLowDutyCycle[x] to be ready for use by TrimMargin_MarginLow(). **Note** Calling this API does NOT cause any change in the PWM output duty cycle.

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

marginLoVoltage: Specifies the desired power converter output margin low voltage in mV. Valid range: 1..65,535

Return Value: None

Side Effects: None

uint16 TrimMargin_GetMarginLowVoltage(uint8 converterNum)

Description: Returns the margin low output voltage of the specified power converter.

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

Return Value: Power converter margin low output voltage in mV.

Side Effects: None

void TrimMargin_SetNominalVoltage(uint8 converterNum, uint16 nominalVoltage)

Description: Sets the nominal output voltage of the specified power converter.

This overrides the present vNom[x] setting and recalculates vNominalDutyCycle to be ready for use by TrimMargin_Nominal(). **Note** Calling this API does NOT cause any change in the PWM output duty cycle.

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

nominalVoltage: Specifies the desired power converter output margin low voltage in mV. Valid range: 1..65,535

Return Value: None

Side Effects: None

uint16 TrimMargin_GetNominalVoltage(uint8 converterNum)

Description: Returns the margin low output voltage of the specified power converter.

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

Return Value: Power converter nominal output voltage in mV.

Side Effects: None

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

Description: 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. If the integral controller is used, the time between two consecutive calls is based on the Control sample time parameter provided in the GUI. If the incremental controller is used, the adjustments rely on an internal counter set by the SetTrimCycleCount() API.

The implementation is dependent on the Controller method parameter.

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

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..65,535

desiredVoltage: Specifies the desired power converter output voltage in mV. Valid range: 1..65,535

Return Value: None

Side Effects: Calling this API may change the PWM duty cycle driving the control voltage of the selected power converter causing the change in the power converter output voltage. If the desiredVoltage cannot be achieved because the PWM duty cycle is at the min or max level, the alert signal will be asserted until the alert condition is removed, only possible by calling this API with achievable desiredVoltage.

void TrimMargin_SetDutyCycle(uint8 converterNum, uint8/uint16 dutyCycle)

Description: Sets the PWM duty cycle of the PWM associated with the specified power converter.

The PWM period is always fixed at the maximum value depending on the resolutions set in the customizer.

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

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

Return Value: None

Side Effects: None

uint8/uint16 TrimMargin_GetDutyCycle(uint8 converterNum)

Description: Gets the current 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: converterNum. Specifies the power converter number. Valid range: 1..24

Return Value: PWM duty cycle in PWM clock counts. Valid range: 0..255 (uint8) up to 0..1023 (uint16) 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_MarginHigh(uint8 converterNum)

Description: Sets the selected power converter output voltage to the desired margin high setting as specified in the customizer or be calling TrimMargin_SetMarginHighVoltage() API.

Parameters: uint8 converterNum. Specifies the power converter number

Valid range: 1..24

Return Value: None

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 customizer or be calling TrimMargin_SetMarginLowVoltage() API.

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

Return Value: None

Side Effects: None

void TrimMargin_Nominal(uint8 converterNum)

Description: Sets the selected power converter output voltage to the desired margin low setting as specified in the customizer or be calling TrimMargin_SetNominalVoltage() API.

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

Return Value: None

Side Effects: None

void TrimMargin_PreRun(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: converterNum: Specifies the power converter number. Valid range: 1..24

Return Value: None

Side Effects: None

void TrimMargin_Startup(uint8 converterNum)

Description: Sets the selected power converter output voltage to the startup voltage setting as specified in the customizer.

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

Return Value: None

Side Effects: None

void TrimMargin_StartupPreRun(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: converterNum: Specifies the power converter number. Valid range: 1..24

Return Value: None

Side Effects: None

uint8/uint16 TrimMargin_ConvertVoltageToDutyCycle(uint8 converterNum, uint16 desiredVoltage)

Description: Returns the PWM duty cycle required to achieve the desired voltage on the selected power converter. The PWM duty cycle for the specified output voltage is dependent on the corresponding power converter and is calculated as:

Negative feedback:

$$D = \frac{V_{MAX} - V_{out}}{V_{MAX} - V_{MIN}} \times (2^n - 1)$$

Positive feedback:

$$D = \frac{V_{out} - V_{MIN}}{V_{MAX} - V_{MIN}} \times (2^n - 1)$$

Where V_{MAX} and V_{MIN} are calculated maximum and minimum output voltages for the given power converter, and n is the PWM resolution.

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

desiredVoltage. Specifies the desired power converter output voltage in mV.

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

Side Effects: None

uint8/uint16 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: converterNum: Specifies the power converter number. Valid range: 1..24

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

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

Side Effects: None

void TrimMargin_SetTrimCycleCount(uint8 count)

Description: Sets the internal adjustment cycle count that affects how often the PWM duty cycle is updated when calling TrimMargin_ActiveTrim() API. The default value is TrimMargin_CYCLE_SLOW. The API is only applicable if **Incremental** option is selected for the Controller method parameter.

Parameters: count: Specifies the cycle count value.

Macro name	Value
TrimMargin_CYCLE_SLOW	20
TriMargin_CYCLE_MEDIUM	15
TrimMargin_CYCLE_FAST	10

Return Value: None

Side Effects: None

Global Variables

Function	Description
TrimMargin_initVar	Indicates whether the component has been initialized. It is initialized to zero and set to one the first time TrimMargin_Start() is called. This allows a component restart without re-initialization in all subsequent calls to the TrimMargin_Start() routine.
TrimMargin_vMarginLow[]	Margin low output voltage for all power converters. Initialized by Init() function with the customizer settings. Can be changed at run time by calling SetMarginLowVoltage() function.
TrimMargin_vMarginHigh[]	Margin high output voltage for all power converters. Initialized by Init() function with the customizer settings. Can be changed at run time by calling SetMarginHighVoltage() function.
TrimMargin_vNom[]	Nominal output voltage for all power converters. Initialized by Init() function with the customizer settings. Can be changed at run time by calling SetNominalVoltage() function.
TrimMargin_vMarginLowDutyCycle[]	The PWM duty cycle for the margin low output voltage. Initialized with customizer settings in Init(). Recalculated in SetMarginLowVoltage() when a new value is set. Used by MarginLow() to set PWM for the open loop margin.
TrimMargin_vMarginHighDutyCycle[]	The PWM duty cycle for the margin low output voltage. Initialized with customizer settings in Init(). Recalculated in SetMarginHighVoltage() when a new value is set. Used by MarginHigh() to set PWM for the open loop margin.
TrimMargin_vNominalDutyCycle[]	The PWM duty cycle for the margin low output voltage. Initialized with customizer settings in Init(). Recalculated in SetNominalVoltage() when a new value is set. Used by Nominal() to set PWM for the open loop margin.

Example Projects, Code Examples, and Application Notes

Example Projects

PSoC Creator provides access to example projects 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.

Code Examples

There are numerous code example projects that include schematics and example code available online at the Cypress Code Examples web page. Code Example CE95387 - Trim voltage rails with PSoC 3/5LP demonstrates the basic operation of the Trim Margin component.

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 (www.cypress.com/appnotes). AN93529 - Introduction to Power Supervision with PSoC 5LP demonstrates how to configure a fully-featured Power Supervision solution using the PSoC Power Supervision Tool. Note this application note is not posted publically. Please contact Cypress technical support to obtain this document.

Deprecated code

The Trim and Margin component contains deprecated code that is not recommended for use but is kept to preserve backward compatibility with the existing designs. The deprecated code is located under the next comment header in the component source files:

```
/***************************************
* The following code is DEPRECATED and
* should not be used in new projects.
***************************************/
```

Follow instructions in the following table on how to update your design.

What is deprecated	Reason for deprecation	How to handle it
TrimMargin_SetNominal(n)	Updated naming convention for consistency and addition of new APIs.	Use TrimMargin_Nominal(n)
TrimMargin_SetPreRun(n)		Use TrimMargin_PreRun(n)
TrimMargin_SetStartup(n)		Use TrimMargin_Startup(n)
TrimMargin_SetStartupPreRun(n)		Use TrimMargin_StartupPreRun(n)
TrimMargin_DUTYCYCLE		Use TrimMargin_VNOMINAL_DUTYCYCLE

What is deprecated	Reason for deprecation	How to handle it
TrimMargin_TRIM_SLOW_PERIOD		Use TrimMargin_TRIM_CYCLE_SLOW

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 the 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 4 (GCC)		PSoC 5LP (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
8 outputs; 8-bit UDB PWM	1374	91	940	93	968	93
8 outputs; 9- or 10-bit UDB PWM	1500	115	1004	117	1056	117
16 outputs; 8-bit UDB PWM	1602	180	1100	181	1176	181
24 outputs; 8-bit UDB PWM	1810	270	N/A	N/A	1388	271
8 outputs; 8-bit fixed-function PWM	N/A	N/A	2356	93	N/A	N/A
8 outputs; 10-bit fixed-function PWM	N/A	N/A	2460	117	N/A	N/A

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.

Figure 1 and Figure 2 show the basic circuit model for trimming a DC / DC regulator with a negative or positive feedback control loop, respectively. An increase in PWM duty cycle increases the analog control voltage. This results in a decrease (negative) or increase (positive) of power converter output voltage. Conversely, a decrease in PWM duty cycle decreases the analog control voltage. This results in an increase (negative) or decrease (positive) of power converter output voltage.

Figure 1. Trimming DC/DC Converter with a Negative Feedback Control Loop

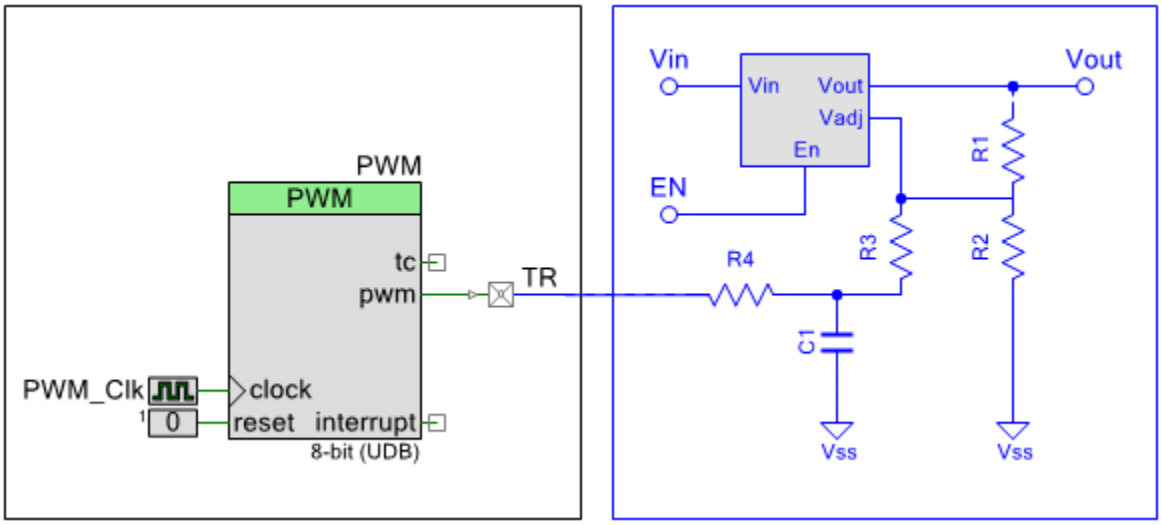
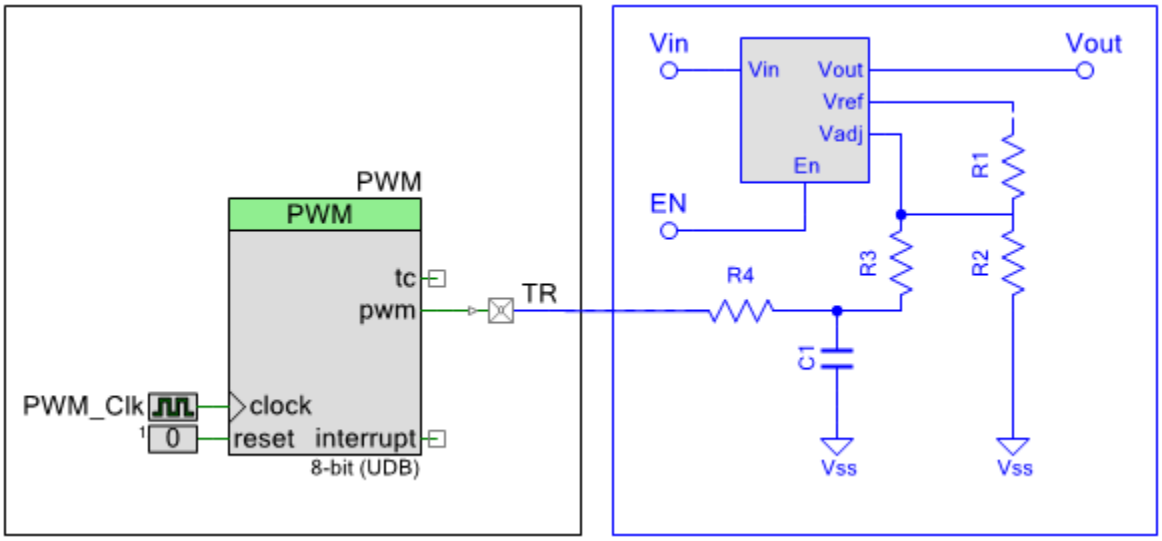


Figure 2. Trimming DC/DC Converter with a Positive Feedback Control Loop



The duty cycle adjustments are based on two different methods depending on the Controller method parameter:

Incremental Controller

The duty cycle is incremented or decremented, depending on whether the desired voltage is higher or lower than the measured voltage, as well as the type of feedback the converter supports.

Integral Controller

The integral controller follows the following equations:

Negative feedback: $Duty\ Cycle[n] = Ki * error * ct - DutyCycle[n-1]$

Positive feedback: $Duty\ Cycle[n] = Ki * error * ct + DutyCycle[n-1]$

Where:

$error = Desired\ Voltage - Actual\ Voltage$

The ct is the Control sample time provided in the GUI. The integral gain Ki is calculated as:

$$Ki = w_{cr}\left(\frac{2^n}{V_{MAX} - V_{MIN}}\right) \times \tau \times \sqrt{{w_{cr}}^2 + \frac{1}{\tau^2}}$$

Where *n* is the PWM resolution and τ is the time constant for the RC circuit, which is calculated as:

$\tau = R4 \times C1$

And w_{cr} is the crossover frequency, which is calculated as:

$$w_{cr} = \frac{2\pi}{ct} \times \frac{1}{10}$$

Or

$$w_{cr} = \frac{1}{\tau} \times \tan(10°) \times \frac{1}{3}$$

The lowest frequency out of the two equations is the one used in the Ki equation.

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 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.

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.

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. 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: Control Register. Refer to the corresponding component datasheet for information on their MISRA compliance and specific deviations.

Resources

On PSoC 4, the Trim and Margin component resource usage is mostly dependent on the implementation type of the internal PWMs. That is, the PWM can either be implemented in the UDB array or use the fixed-function TCPWM block.

On PSoC 3 and PSoC 5LP, there are no options for the implementation type of the PWM. The component is placed throughout the UDB array.

Configuration	Resource Type					
	Datapath Cells	Macrocells	Status Cells	Control Cells	TCPWM	Interrupts
N outputs; 8-bit UDB PWM	CEIL(N/2) [1]	1	–	1	–	–
N outputs; 9- or 10-bit UDB PWM	N	1	–	1	–	–

1 CEIL(x) gives the smallest integer not less than *x*.

Configuration	Resource Type					
	Datapath Cells	Macrocells	Status Cells	Control Cells	TCPWM	Interrupts
N outputs; 8-, 9- or 10-bit fixed-function PWM	–	–	–	1 [2]	N	–

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[3]	Max	Units
I_{DD}	Component current consumption				
	N outputs; 8-bit UDB PWM	–	N x 2.5	–	µA/MHz
	N outputs; 9- or 10-bit UDB PWM	–	N x 4	–	µA/MHz
	N outputs; 8-, 9- or 10-bit fixed-function PWM	–	–	N x I_{TCPWM} [4]	µA

AC Characteristics

Parameter	Description	Min	Typ	Max[5]	Units
f_{CLOCK}	Component clock frequency				
	8-bit UDB PWM	–	–	55	MHz
	9- or 10-bit UDB PWM	–	–	50	MHz
	8-, 9- or 10-bit fixed-function PWM	–	–	F_{CPU} [6]	

2 For fixed-function PWM implementation, a control register is used only if an alert pin is present / used in the design.

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

4 I_{TCPMW} – current consumption of the TCPWM block. Refer to the Timer Counter PWM component datasheet for details on its current consumption.

5 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.

6 Maximum CPU frequency for the target PSoC device.

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
3.0.a	Minor datasheet edits.	
3.0	Added support for PSoC 4 device family.	New device support.
	Implemented an integral control for actively controlled trim operation.	The active trimming supports either incremental or integral control based on the Controller method parameter.
	Added support for power converters with positive feedback control loop.	New circuit model is provided to support the power converters with positive feedback. The Feedback parameter is configured per power converter.
	Datasheet update.	To reflect changes made in version 3.0.
2.0	Corrected erratum ID 152757 (The Trim and Margin component conforms to a specific model when calculating the duty cycles required to achieve desired voltage targets. That model is appropriate for the linear regulator as configured in the Cypress CY8CKIT-035 development kit, but is not appropriate for some other configurations or for switching power supplies.)	Using a converter that deviated from this model could result in voltage overshoot or undershoot when margining.
	A new method was developed to input the trim and margin parameters. Updated the Configure dialog to better reflect an external RC circuit configuration.	**Note** Export/Import of CSV file format is no longer supported with this change. You can import your previously saved configuration to v1.30 and then update to the most recent component version.
	Added SetTrimCycleCount() API to adjust the active trim cycle.	The API allows to speed up or slow down the active trimming.
	Added SetNominalVoltage() and GetNominalVoltage() APIs to adjust the nominal voltage.	Usability enhancement.
	Renamed the following functions: SetNominal() -> Nominal() SetPreRun() -> PreRun() SetStartup() -> Startup() SetStartupPreRun() -> StartupRreRun()	For backward compatibility, all old names are preserved through compatibility macros but not recommended to use in new designs.
	Datasheets edits and corrections. **Note** The errata section was removed from this datasheet.	To reflect changes made in v2.0.
1.30.a	Added Component Errata section.	Document known problems in the component.
1.30	Fixed a defect in the TrimMargin_ConvertVoltageToDutyCycle() API.	The API did not work correctly for PSoC 3.

Version	Description of Changes	Reason for Changes / Impact
	Minor datasheet corrections.	
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, 2016-2017. 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.

