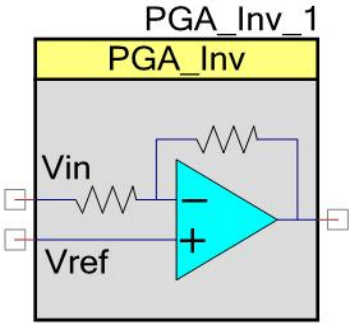
Inverting Programmable Gain Amplifier (PGA_Inv)

1.90



Features

- Gain steps from –1 to –49
- High input impedance
- Adjustable power settings

General Description

The Inverting Programmable Gain Amplifier (PGA_Inv) component implements an opamp-based inverting amplifier with user-programmable gain. It is derived from the switched capacitor/continuous time (SC/CT) block.

The inverting gain can be between –1.0 (0 dB) and –49.0 (+33.8 dB). The gain can be selected using the configuration window or changed at run time using the provided API. The maximum bandwidth is limited by the gain-bandwidth of the opamp and is reduced as the gain is increased. The input of the PGA_Inv operates from rail to rail, but the maximum input swing (difference between Vin and Vref) is limited to V_{DDA}/Gain. The output of the PGA_Inv is class A, and is rail to rail for sufficiently high load resistance.

The PGA_Inv is used when an input signal has insufficient amplitude and the preferred output polarity is the inverse of the input. A PGA_Inv can be placed in front of a comparator, ADC, or mixer to increase the signal amplitude. A unity gain PGA_Inv following another gain stage or buffer can be used to generate differential outputs.

Input/Output Connections

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

Vin – Analog

Vin is the input signal terminal.

Vref – Analog

Vref is the input terminal for a reference signal. The reference input has a high impedance and may be connected to fixed reference (for example, V_{DDA}/2), VDAC8 output, or routed to a pin.

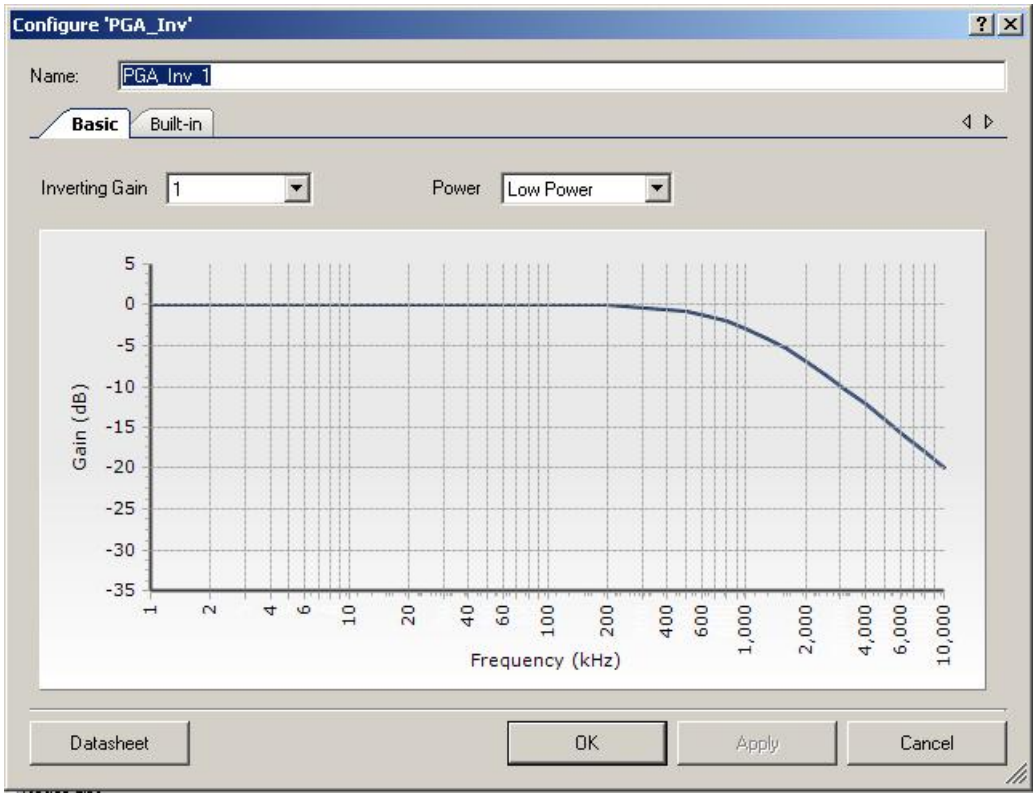
Vout – Analog

Vout is the output voltage signal terminal. Vout is a function of (Vin – Vref) times the Gain:

Vout = Vref + (Vin – Vref) × Gain where Gain is a negative value

Component Parameters

Drag a PGA_Inv component onto your design and double-click it to open the **Configure** dialog.



Inverting_Gain

This parameter is used to set the default gain of the amplifier. The allowable inverting gains provided are: –1 (default), –3, –7, –15, –22, –24, –31, –47, and –49.

Power

This sets the initial drive power of the PGA_Inv. The **Power** setting determines the speed with which the PGA_Inv responds to changes in the input signal. There are four power settings: **Minimum Power**, **Low Power** (default), **Medium Power**, and **High Power**. A **Low Power** setting results in the slowest response time and a **High Power** setting results in the fastest response time. The **Power** setting can be set at run time using the PGA_Inv_SetPower() API.

Application Programming Interface

Application Programming Interface (API) routines allow you to configure 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 “PGA_Inv_1” to the first instance of a component in a given design. You can rename the instance 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 “PGA_Inv.”

Function	Description
PGA_Inv_Start()	Starts the PGA_Inv.
PGA_Inv_Stop()	Powers down the PGA_Inv.
PGA_Inv_SetGain()	Sets gain to predefined constants.
PGA_Inv_SetPower()	Sets drive power to one of four settings.
PGA_Inv_Sleep()	Stops and saves the user configurations.
PGA_Inv_Wakeup()	Restores and enables the user configurations.
PGA_Inv_Init()	Initializes or restores default PGA_Inv configuration.
PGA_Inv_Enable()	Enables the PGA_Inv.
PGA_Inv_SaveConfig()	Empty function. Provided for future use.
PGA_Inv_RestoreConfig()	Empty function. Provided for future use.

Global Variables

Variable	Description
PGA_Inv_initVar	Indicates whether the PGA_Inv has been initialized. The variable is initialized to 0 and set to 1 the first time PGA_Inv_Start() is called. This allows the component to restart without reinitialization after the first call to the PGA_Inv_Start() routine. If reinitialization of the component is required, then the PGA_Inv_Init() function can be called before the PGA_Inv_Start() or PGA_Inv_Enable() function.

void PGA_Inv_Inv_Start(void)

Description: Turns on the PGA_Inv and sets the power level.

Parameters: None

Return Value: None

Side Effects: None

void PGA_Inv_Stop(void)

Description: Turns off PGA_Inv and enables its lowest power state.

Note This API is not recommended for use on PSoC 5 silicon. These devices have a defect that causes connections to several analog resources to be unreliable when not powered. The unreliability manifests itself in silent failures (for example, unpredictably bad results from analog components) when the component using that resource is stopped. When using this silicon, all analog components in a design should be powered up (by calling their respective _Start() APIs, for instance PGA_Inv_Start()) at all times. Do not call the PGA_Inv_Stop() APIs.

Parameters: None

Return Value: None

Side Effects: None. Does not affect power or gain settings.

void PGA_Inv_SetGain(uint8 gain)

Description: Sets gain of amplifier between –1 and –49. The table below shows the valid gain settings.

Parameters: uint8 gain: Sets the gain to a specific value. See table below for valid gain settings.

Gain Setting	Notes
PGA_Inv_GAIN_01	Gain = –1
PGA_Inv_GAIN_03	Gain = –3
PGA_Inv_GAIN_07	Gain = –7
PGA_Inv_GAIN_15	Gain = –15
PGA_Inv_GAIN_22	Gain = –22
PGA_Inv_GAIN_24	Gain = –24
PGA_Inv_GAIN_31	Gain = –31
PGA_Inv_GAIN_47	Gain = –47
PGA_Inv_GAIN_49	Gain = –49

Return Value: None

Side Effects: None

void PGA_Inv_SetPower(uint8 power)

Description: Sets the drive power to one of four settings: minimum, low, medium, or high.

Parameters: uint8 power: Sets the power level to one of four settings: minimum, low, medium, or high.

Power Setting	Notes
PGA_Inv_MINPOWER	Minimum active power and slowest reaction time
PGA_Inv_LOWPOWER	Low power and speed
PGA_Inv_MEDPOWER	Medium power and speed
PGA_Inv_HIGHPOWER	Highest active power and fastest reaction time

Return Value: None

Side Effects: None

void PGA_Inv_Sleep(void)

Description: This is the preferred routine to prepare the component for sleep. The PGA_Inv_Sleep() function saves the current component state. Then it calls the PGA_Inv_Stop() function and calls PGA_Inv_SaveConfig() to save the hardware configuration.

Call the PGA_Inv_Sleep() function before calling the CyPmSleep() or the CyPmHibernate() function. Refer to the PSoC Creator *System Reference Guide* for more information about power management functions.

Parameters: None

Return Value: None

Side Effects: None

void PGA_Inv_Wakeup(void)

Description: This is the preferred routine to restore the component to the state when PGA_Inv_Sleep() was called. The PGA_Inv_Wakeup() function calls the PGA_Inv_RestoreConfig() function to restore the configuration. If the component was enabled before the PGA_Inv_Sleep() function was called, the PGA_Inv_Wakeup() function will also re-enable the component.

Parameters: None

Return Value: None

Side Effects: Calling the PGA_Inv_Wakeup() function without first calling the PGA_Inv_Sleep() or PGA_Inv_SaveConfig() function may produce unexpected behavior.

void PGA_Inv_Init(void)

Description: Initializes or restores the component according to the customizer Configure dialog settings. It is not necessary to call PGA_Inv_Init() because the PGA_Inv_Start() routine calls this function and is the preferred method to begin component operation.

Parameters: None

Return Value: None

Side Effects: All registers will be set to values according to the customizer Configure dialog.

void PGA_Inv_Enable(void)

Description: Activates the hardware and begins component operation. It is not necessary to call PGA_Inv_Enable() because the PGA_Inv_Start() routine calls this function, which is the preferred method to begin component operation.

Parameters: None

Return Value: None

Side Effects: None

void PGA_Inv_SaveConfig(void)

Description: Empty function. Provided for future use.

Parameters: None

Return Value: None

Side Effects: None

void PGA_Inv_RestoreConfig(void)

Description: Empty function. Provided for future use.

Parameters: None

Return Value: None

Side Effects: None

Sample Firmware Source Code

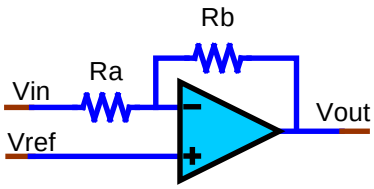
PSoC Creator provides numerous example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start 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 PGA_Inv is constructed from a generic SC/CT block. The gain is selected by adjusting two resistors; Ra and Rb (see Figure 1). Ra may be set to either 20k or 40k ohms, depending on selected gain. Rb may be set between 20k and 1000k ohms, to generate the possible gain values selectable in either the **Configure** dialog or the PGA_Inv_SetGain() function.

Figure 1. PGA_Inv Schematic



The block has a programmable capacitor in parallel with the feedback resistor, Rb. The capacitor value is configured for each gain selection to achieve guaranteed stability. Reassigning Rb values without also selecting the appropriate feedback capacitor value may result in PGA_Inv instability. You are strongly advised to use the provided APIs for gain changes.

The input resistance of the PGA_Inv is finite. The gain accuracy is dependent on the routing resistance between the source and the Vin input. The gain specifications accommodate the nominal variation in the routing resistance.

Registers

The PGA_Inv component configuration is implemented in registers SC[0..3]_CR0, SC[0..3]_CR1 and SC[0..3]_CR2. These can be accessed in user code by reference to the instantiated component name, for example, PGA_Inv_1_CR0_REG. You can review the register contents in the PSoC Creator component debug window. Refer to the applicable TRM, available on the Cypress website, for a detailed description of each register. The following registers are displayed in the PGA component debug window.

Register:	PGA_Inv_1_CR0_REG
Name:	Switched Capacitor Control Register 0
Description:	Register bits 3:1 configure the switch capacitor block operating mode. This field is set to 110b for the PGA component.

Register:	PGA_Inv_1_CR1_REG
Name:	Switched Capacitor Control Register 1
Description:	Register fields configure drive mode, compensation capacitor values, and gain setting of the switch capacitor block.

Register: PGA_Inv_1_CR2_REG

Name: Switched Capacitor Control Register 2

Description: Register fields configure the input impedance, feedback impedance and the reference ground selection for the switch capacitor block.

Register: PGA_Inv_1_PM_ACT_CFG_REG

Name: Active Power Mode Configuration Register 9

Description: Register bits 3:0 enable power to the four switch capacitor blocks.

Resources

The PGA_Inv uses one SC/CT analog block.

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 5 (GCC)		PSoC 5LP (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Default	206	22	360	12	304	5

DC and AC Electrical Characteristics for PSoC 3

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. Typical values are for T_A = 25 °C.

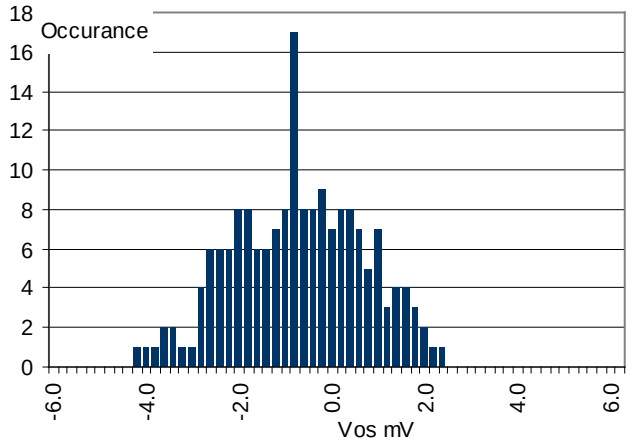
DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{IN}	Input voltage range	Power mode = minimum	V_{SSA}	–	V_{DDA}	V

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{OS}	Input offset voltage	Power mode = high, gain = 1, V_{DDA} = 5 V	–	–	10	mV
TCV_{OS}	Input offset voltage drift with temperature	Power mode = high, gain = 1, V_{DDA} = 5 V	–	–	±30	µV/°C
Ge1	Gain error, gain = 1	V_{DDA} = 5 V	–	–	±0.15	%
Ge15	Gain error, gain = 15	V_{DDA} = 5 V	–	–	±2.5	%
Ge49	Gain error, gain = 49	V_{DDA} = 5 V	–	–	±5	%
V_{ONL}	DC output nonlinearity	Gain = 1	–	–	±0.01	% of FSR
R_{IN}	Input resistance		35	–	–	MΩ
C_{IN}	Input capacitance		–	–	7	pF
V_{OH}, V_{OL}	Output voltage swing	Power mode = high, gain = 1, R_{LOAD} = 100 kΩ to V_{DDA}/2	V_{DDA} – 0.15	–	V_{SSA} + 0.15	V
Vsrc	Output voltage under load	Iload = 250 µA, Vdda ≥ 2.7V, power mode = high	-	-	300	mV
I_{DD}	Operating current	Power mode = high	–	1.5	1.65	mA
PSRR	Power supply rejection ratio		48	–	–	dB

Figures

Histogram Input Offset Voltage

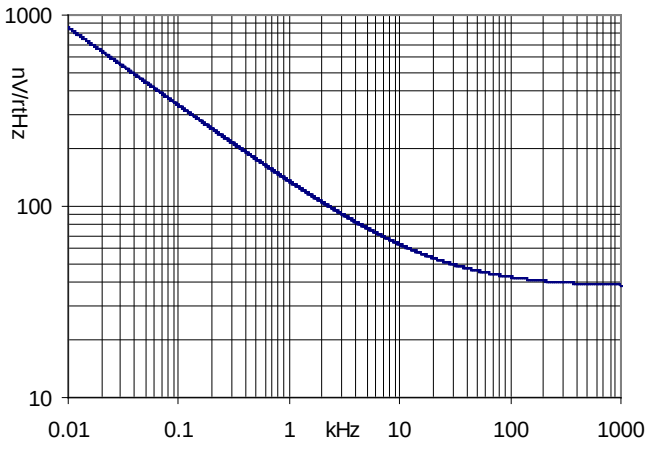


AC Characteristics

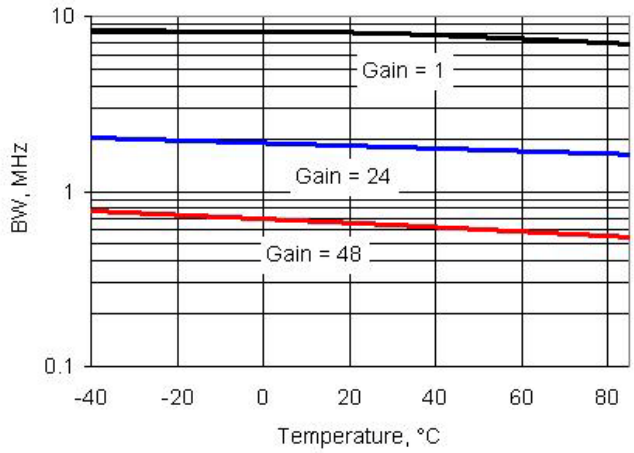
Parameter	Description	Conditions	Min	Typ	Max	Units
BW1	–3 dB bandwidth	Power mode = high, gain = 1, input = 100mV peak-to-peak	6.7	8	–	MHz
SR1	Slew rate	Power mode = high, gain = 1, 20% to 80%	3	–	–	V/µs
e_n	Input noise density	Power mode = high, V_{DDA} = 5 V, at 100 kHz	–	43	–	nV/sqrtHz

Figures

Voltage noise, V_{DDA} = 5.0V, Power = High



Bandwidth vs. Temperature, at Different Gain Settings, Power Mode = High



DC and AC Electrical Characteristics for PSoC 5

Specifications are valid for –40 °C ≤ T_A ≤ 85 °C and T_J ≤ 100 °C, except where noted.
Specifications are valid for 2.7 V to 5.5 V, except where noted. Typical values are for T_A = 25 °C.

DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{IN}	Input voltage range	Power mode = minimum	V_{SSA}	–	V_{DDA}	V
V_{OS}	Input offset voltage	Power mode = high, gain = 1, V_{DDA} = 5 V	–	-	20	mV

Parameter	Description	Conditions	Min	Typ	Max	Units
TCV_{OS}	Input offset voltage drift with temperature	Power mode = high, gain = 1, V_{DDA} = 5 V	–	-	±30	µV/°C
Ge1	Gain error, gain = 1	V_{DDA} = 5 V	–	±2.5	±5.5	%
Ge15	Gain error, gain = 15	V_{DDA} = 5 V	–	±8	±11.5	%
Ge49	Gain error, gain = 49	V_{DDA} = 5 V	–	±13	±19.5	%
Gd1	Gain drift, gain = 1		–	±95	±200	ppm/°C
Gd15	Gain drift, gain = 15		–	±115	±250	ppm/°C
Gd49	Gain drift, gain = 49		–	±350	±850	ppm/°C
V_{ONL}	DC output nonlinearity	Gain = 1	–	–	±0.1	% of FSR
V_{OH}, V_{OL}	Output voltage swing	Power mode = high, gain = 1, R_{LOAD} = 100 kΩ to V_{DDA}/2	V_{DDA} – 0.15	–	V_{SSA} + 0.15	V
Cin	Input capacitance		-	-	7	pF
Vsrc	Output voltage under load	Iload = 250 µA, power mode = high	-	-	300	mV
I_{DD}	Operating current	Power mode = high	–	1.5	1.65	mA
PSRR	Power supply rejection ratio		48	-	–	dB

AC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
BW1	–3 dB bandwidth	Power mode = high, gain = 1, V_{DDA} = 5 V	3.1	–	–	MHz
SR1	Slew rate	Power mode = high, gain = 1, 20% to 80%	3	–	–	V/µs
e_n	Input noise density	Power mode = high, VDDA = 5 V, at 100 kHz	-	43	-	nV/sqrtHz

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
1.90	For low voltage VDDA operation uses a boost clock shared by all the SC/CT based components.	Reduces the number of analog clocks required in the system for boost clocks. With this change a single boost clock is shared instead of using a separate clock for each SC/CT based component.
	Added PSoC 5LP support	
1.80	Added all component APIs with the CYREENTRANT.	
	Minor GUI updates	
	PGA_Inv_Stop() API modified for PSoC 5	Change required to prevent the component from impacting unrelated analog signals when stopped, when using PSoC 5.
1.70	Updated PGA_Inv response graph	Change required to dynamically resize graph to fit window and to add horizontal and vertical grids.
	Added DC and AC electrical characteristics data for PSoC 5	
	Removed VDDA parameter from component customizer	VDDA setting in the component is redundant and unnecessary for multiple components. The parameter was removed and the component queries the global setting for minimum VDDA in the DWR and automatically enables the pump when necessary.
1.60	Configuration window created to include Frequency response graphs a better ease of use GUI.	Previous configuration window did not provide enough information for ease of use.
	SetGain constants corrected in the header file	The constants provided for the SetGain API had incorrect values. These have been corrected.
	Added characterization data to datasheet	
	Minor datasheet edits and updates	
	Added Sleep/Wakeup and Init/Enable APIs.	To support low power modes, as well as to provide common interfaces to separate control of initialization and enabling of most components.
1.50	Removed Gain setting of 23.	The gain of 23 was too close to 22 and 24 and therefore offered no value.
	Updated the symbol and Configure dialog.	These were updated to comply with corporate standards.

© Cypress Semiconductor Corporation, 2012. 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.