AN2246

PSoC® 1 – PWM Source – High Frequency, High Resolution

Author: Victor Kremin

Associated Project: Yes

Associated Part Family: CY8C21xxx, CY8C24xxxA, CY8C27xxx, CY8C28xxx, CY8C29xxx

Software Version: PSoC® Designer™ 5.4

Related Application Notes: AN2041

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN2246.

AN2246 describes PSoC® 1, a high speed, high-resolution PWM source that overcomes traditional PWM source limitations in output signal frequency and resolution.

Contents

Introduction 1
A Solution.....3
A Practical Example 4
Firmware 5
Experimental Testing.....6
Special Applications 7
References.....8
Summary.....8

Introduction

Pulse width modulation (PWM) control is widely used in modern switch mode power supplies and regulators, battery chargers, motor drivers, various digital-to-analog converters, automatic control systems and high-efficiency class-D amplifiers. Nowadays, most electronic systems contain at least one PWM source.

Progress in semiconductor technology reduces the load control transistors' (MOSFET, IGBT, or even classic bipolar) switch time and dynamic losses, allowing converter switches to have higher control frequencies. There are various benefits from this for inductors and filter capacitors including value, dimension, and weight and cost reduction. Using smaller inductors and capacitors reduces energy loss, increases regulator effectiveness and widens the device operation temperature range by using high Q-factor ceramic capacitors instead of bulky electrolytic ones.

The PSoC® device is able to create the PWM signals for these applications with a resolution of up to 32 bits and a clock source up to 48 MHz. There are two main ways to build a **digital** PWM generator with PSoC:

- Use a counter followed by a programmable threshold comparator
- Use a pseudo-random generator and threshold comparator

The first approach is a classic approach. The fixed-period down counter is stimulated by a clock source. The output comparator compares the counter content with a preset threshold value and sets the high level on the output when the counter is less (or less than and equal, depending upon the selected comparator mode) than the threshold. Using this approach, the PWM source duty cycle can be evaluated by the following formula:

$$D_c = \frac{T_{th} + \partial}{T_p}, \partial = \overline{0..1} \qquad \text{Equation 1}$$

- T_p - Counter period, which is equal to the increase by one counter period register value
- T_{th} - Comparator threshold value
- ∂ - Constant, which is determined by selected comparator mode (*Compare* less or *Compare less or equal*)

Output signal frequency F_{out} is determined by the PWM clock frequency F_{clk} and the counter Period register content:

$$F_{out} = \frac{F_{clk}}{T_p} \qquad \text{Equation 2}$$

The duty cycle step ΔD_c, or PWM resolution, is determined by the distinct nature of counter/threshold comparator operation and is defined by the counter period:

$$\Delta D_c = \frac{1}{T_p} \qquad \text{Equation 3}$$

By combining equations (2) and (3), the following equation between ΔD_c, F_{clk} and F_{out} is obtained:

$$F_{out} = \Delta D_c \cdot F_{clk} \qquad \text{Equation 4}$$

Therefore, getting a smaller duty cycle step proportionally reduces the PWM output frequency. At a maximum clock rate of 48 MHz and a duty cycle step of 0.5%, we only get 240 kHz of output frequency, which is insufficient for most end applications.

The second approach for PWM signal generation uses a pseudo random generator (PRS), which produces the periodic pseudo random bit sequence with maximum period of $T_p = 2^N\text{-}1$ of clock periods, where N is the length of the Shift register. (In PSoC, N is a multiple of 8.) When this bit sequence is separated by bytes, the bytes are uniformly distributed on a $1..2^{N-1}$ interval. While these values are compared with a preset threshold, the probability that the generated value is less than this compare value is directly proportional to the threshold. In other words, the number of Shift register clock periods, where the Shift register value is smaller than the compare threshold, is directly proportional to this value at the pseudo random sequence repeat period. Therefore, the equivalent duty cycle ratio is directly proportional to the compare value.

Note The PRS PWM output signal consists of variable width pulses with spaces in between. The maximum length of a one's and zero's sequence can be modeled for a particular PRS pole and compare values. The PRS signal can be considered as a periodic signal with a wide frequency range, including frequencies starting at $F_{clk} / \left(2^N - 1\right)$. The effective duty cycle resolution of a PRS PWM can be evaluated by the same formula as for the counter-based PWM source, Equation (3), due to the distinct nature of operation and compare value setting.

One of the well-designed ways for forming the variable duty cycle signals lies in using the PSoC's analog sigma delta modulator.

The modulator provides a synchronous output bit stream where the relation between numbers of output cycles when the modulator output is positive to the total number of produced bits is directly proportional to the input signal level in steady mode. Therefore, the effective duty cycle is clearly relative to the input signal level. Pulses from a sigma delta modulator consist of one or more one bits separated by one or more zero bits. In other words, the sigma delta modulator can be considered a synchronous pulse-frequency-width modulator. The output stream uniformly receives ones and zeros, giving a low ripple analog signal of the modulator bit stream in the analog filter.

A sigma delta modulator duty cycle resolution is limited by noise and input voltage source resolution. You can reduce unwanted noise by using a higher resolution. However, higher resolution can only be obtained on longer integration intervals. PSoC sigma delta modulators are able to work up to a 2-MHz internal clock frequency. For example, you can form a modulator input signal by using the switched capacitor DAC.

Therefore, none of the previously described methods are directly suitable for forming the high frequency, high-resolution PWM signal.

A Solution

One simple way to build a high resolution PWM is by **interlacing** two PWM signals with different duty cycles to produce the PWM signal with an intermediately effective duty cycle. Suppose, we have two PWM sources with the same period register T_{PWM} but different compare values, T_{th1} and T_{th2}. The $T_{th2} = T_{th1} + 1$ and $T_{PWM} + 1 = T_p$. The duty cycle for these PWM sources is:

$$D_{c1} = \frac{T_{th1}}{T_p}; \quad D_{c2} = \frac{T_{th1} + 1}{T_p} \qquad \text{Equation 5}$$

The signals from these PWM are multiplexed by a third control signal with duty cycle D_{c3} in such a way that the first PWM source drives output when the control signal is low and the second PWM passes to the output in the opposite case. The effective duty cycle of this interlaced PWM signal can be evaluated in the following way:

$$D_{eff} = \left(1 - D_{c3}\right) D_{c1} + D_{c3} D_{c2} \qquad \text{Equation 6}$$

By varying D_{c3} between $\overline{0..1}$, we can scan between D_{c1} and D_{c2}, or separate the $D_{c2} - D_{c1}$ duty cycle for multiple pieces according to the duty cycle resolution of D_{c3}. D_{c3} can accept T_m as a possible "quantum," or be expressed as:

$$D_c = \frac{T_{th3}}{T_m} \qquad \text{Equation 7}$$

T_{th3} is the variable that determines the duty cycle control signal. For example, if a conventional PWM is used, T_{th3} is the compare value and T_m is the PWM period.

By substituting Equations (5) and (7) into Equation (6), we get:

$$D_{eff} = \frac{1}{T_p}\left(T_{th1} + \frac{T_{th3}}{T_m}\right) \qquad \text{Equation 8}$$

Therefore, one T_{th1} "quantum" is cut into multiple pieces according to the value of T_m. To apply Equation (8), we need to develop a simple rule on how to calculate T_{th1} and T_{th3} for any given D_{eff}. The following rule is recommended:

1. Select T_p and T_m according to the demanded output frequency and resolution. The effective duty cycle minimum step is from Equation (8) as $\left(T_p \cdot T_m\right)^{-1}$.
2. Scale D_{eff} by multiplying D_{eff} by T_p. Take the integer from this product. It forms T_{th1}.
3. Take the fractional part obtained in the product of step 2 and multiply by T_m. Leave the integer part of the result. It gives T_{th3}.

It is advised to select T_p and T_m as powers of two. In this case, the values for T_{th1} and T_{th3} can be obtained by using the binary shift operation when the duty cycle is represented as two integer type variables and 100% duty cycle corresponds to the maximum value, which is the power of two as well. Suppose we select T_p = 64 and T_m = 256 to produce a 14-bit PWM source resolution.

The 8 least significant bits are used to set the source duty cycle and the 6 most significant bits determine the counter's duty cycle.

Note To eliminate any glitches, the control signal should be synchronized to the PWM source signal. This can be achieved by using PSoC's counter *Terminal Count* output as the control generator clock source. The following sources can be used in the control signal generator:

- Conventional PWM generator
- PRS PWM source
- Sigma delta modulator with reference DAC

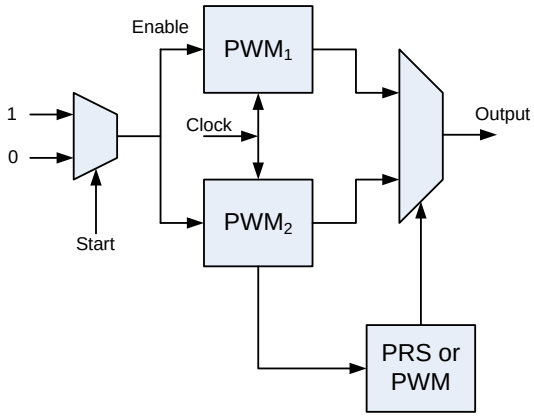
The conventional PWM source, built around the timer/counter, continually produces a one's sequence followed by a zero's sequence within the PWM generator period interval. When this signal is used to alternate between two PWM sources, the higher level of filtered-signal ripples can be achieved because one generator can be turned on for a relatively long time. The PRS PWM and sigma delta modulators better distribute zeros and ones in the output bit stream and reduce the level of ripples when the signal that is produced is filtered. Therefore, the last two signal source types are recommended.

A Practical Example

We now demonstrate how to build the high-resolution, high-speed PWM generator using PRS PWM or conventional PWM source as the control signal to alternate between the two duty cycle PWM sources. This example demonstrates how to control the 14-bit, 375-kHz PWM source. The single counter-based PWM source at this resolution is able to operate only at 3 kHz or 120 times less! This source can be used in the low-cost audio DAC or switched-regulator controller. You can easily increase the output frequency to several MHz with a corresponding resolution reduction. For example, you can create a 12-bit PWM with an output frequency of 1.5 or 3 MHz, depending on which clock rate (24 or 48 MHz) is used for the PWM source. Simply reduce the PWM counter periods.

The generator flowchart is illustrated in Figure 1.

Figure 1. The Generator Flowchart

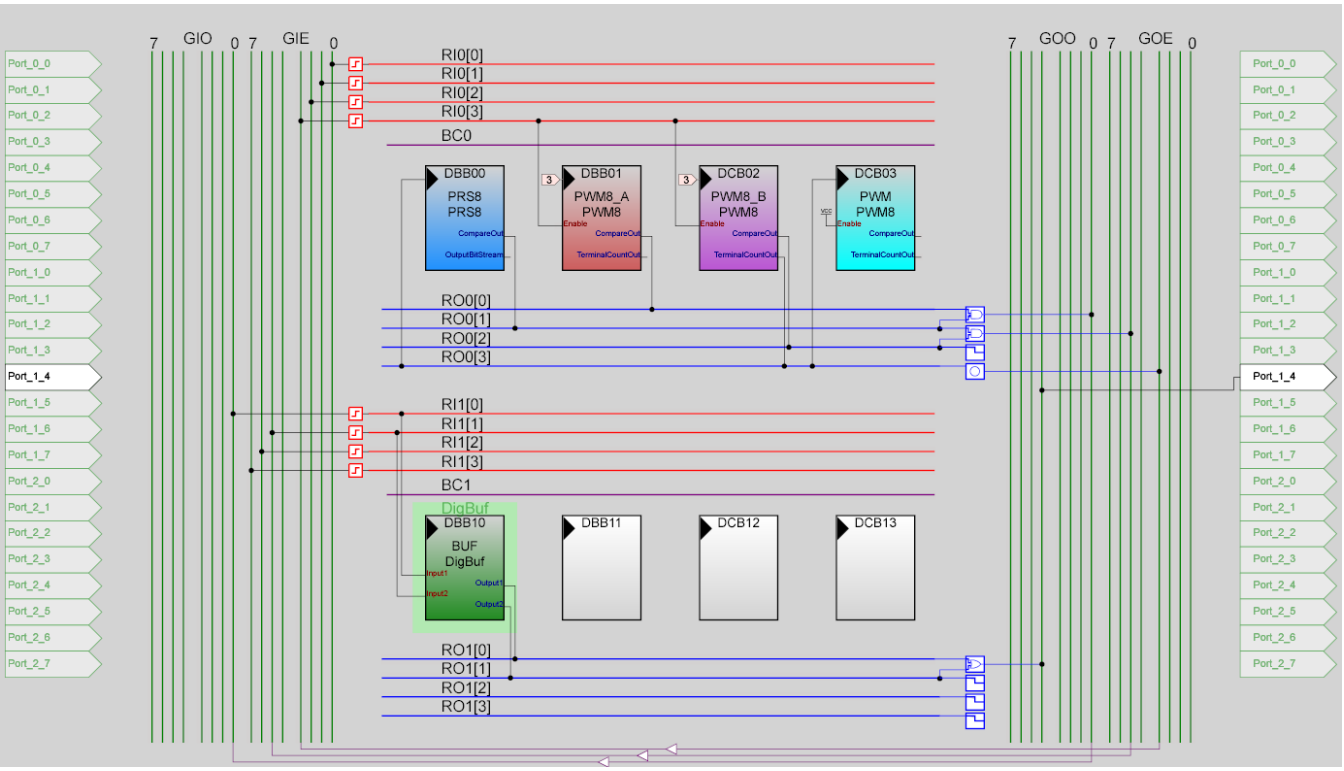


There are two counters with the same period and compare values. To shift by one clock period pulse width, the compare mode is set to "*Less than*" for PWM1 and "*Less than or Equal*" for PWM2. This combination gets the duty cycle from PWM1 to start from zero and the duty cycle for PMW2 to reach 100%, as required for the application. The counters need to be clocked by the same source; the *SysCLK* clock is sent directly by setting the counter synchronization settings to "*Use SysCLK direct*." To eliminate any glitches in the output signal, the counters should start together. To achieve this, they are enabled at the same time by asserting "*Enable*" input using one look-up-table (LUT) in the user firmware.

The PWM terminal count signal is used to clock the PRS PWM that produces the synchronous operation of all three digital modules. PSR PWM compare mode is set to "*Less than or Equal*." If the conventional PWM is used as control source, the compare mode is set to "*Less than*." You can evaluate two different methods of the controlling multiplexer by rerouting PRS PWM or the counter PWM output signal to *Row_0_Output_1* bus.

The 2-to-1 multiplexer is created using three LUTs, two with the AND function and one OR. The multiplexer is controlled by a signal provided by the PRS PWM or the counter PWM source from *Row_0_Output_1* bus. The digital buffer helps route signals inside the PSoC device without using the external connections. If you want to save one digital block, you can make a wired-OR using open drain high drive mode and one external pull-down, low-resistance resistor. Using the internal pull-down resistors is not recommended due to a long rising time. Note that if you want to modify the signal routing, it is imperative to balance the propagation delay in the two PWM channels by properly routing and synchronizing output signals. I made the mistake of passing one signal via a digital buffer and the other directly to the output LUT, which produced duty cycle transfer characteristic drops as large as 3 LSB! Please do not repeat this mistake! Figure 2 shows the placement of the digital modules inside PSoC.

Figure 2. Digital Module Placement



Firmware

The firmware is very straightforward. The initialization routine reads the PWM configuration and starts the user modules. There is a common code to control the PSR PWM or the counter-based PWM; you need to leave only one control source in the end application:

Code 1. Common Code

```
void PRSPWM_Start(void)
{
    LoadConfig_PRSPWM();

    BUF_Start();

    PRS8_WriteSeed(0x01);
    PRS8_WritePolynomial(0xB8);
    PRS8_Start();
    PRS8_SEED_REG = 0x00;

    PWM_WritePeriod(0xff);
    PWM_WritePulseWidth(0);
    PWM_Start();


    PWM8_A_WritePeriod(MAIN_PWM_PERIOD);
    PWM8_A_WritePulseWidth(0);
    PWM8_A_Start();

    PWM8_B_WritePeriod(MAIN_PWM_PERIOD);
    PWM8_B_WritePulseWidth(0);
    PWM8_B_Start();

    RDI0LT1 |= 0xF0;
}
```

You can adapt the proposed source for a different resolution and output frequency by changing the define directives:

Code 2. Define Directives

```
#define MAIN_PWM_RES_BITS 6
#define PRS_PWM_RES_BITS 8
```

First, determine the resolution of the counter-based PWM. For example, if you want to build a 12-bit PWM, you can decrease the resolution of the counter-based PWM to 4. Second, determine the interlacing source resolution.

The routine that calculates and sets the duty cycle values is straightforward, as well:

Code 3. Calculate and Set Duty Cycle Values

```
void PRSPWM_WritePulseWidth(WORD dur)
{
  WORD tmp = (dur >> PRS_PWM_RES_BITS);
  BYTE pw = tmp;

  PWM8_A_WritePulseWidth(pw);
  PWM8_B_WritePulseWidth(pw);

  PRS8_SEED_REG = dur;

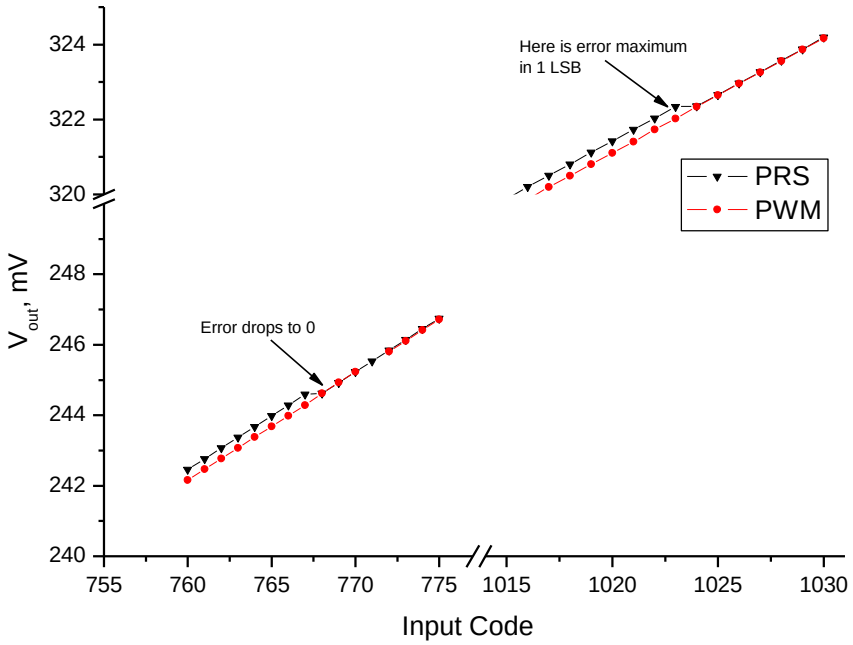
  PWM_WritePulseWidth(dur);
}
```

Experimental Testing

The example project is built and tested. The transfer function linearity was analyzed by measuring the filtered voltage output for different input codes. Figure 3 shows the fragments of the PWM generator transfer function, especially in the interesting “glue regions” when different duty cycle values are loaded into the two high-frequency PWM generators. As can be easily seen in the graph, when the counter-based PWM is used as an interlacing source signal, the characteristic is linear the entire range of code.

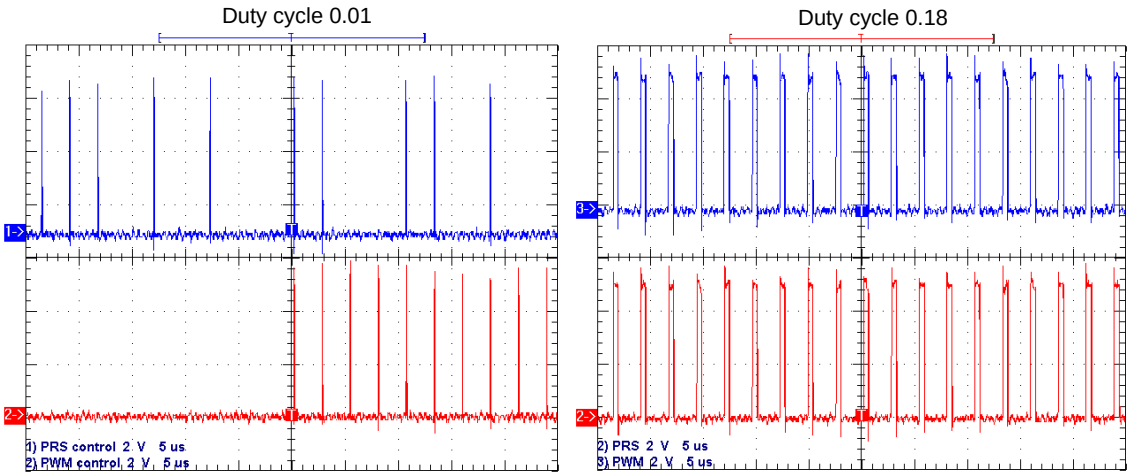
A different situation occurs when the PRS PWM is used as an interlace source signal. When the sequence-repeat period is $2^8 – 1 = 255$, not 256, there is a duty cycle error, which is linearly proportional to the PRS PWM compare value. This error reaches the maximum in 1 LSB at the maximum PRS PWM compare value, and jumps to zero when the PRS compare value is zeroed. The maximum error in 1 LSB is no problem for most applications and provides much more uniform mixing of PWM sources by the PRS generator.

Figure 3. The Generator Transfer Characteristic Fragments



Here are several scope screenshots to demonstrate the source output waveforms for different duty cycles for both the PRS PWM and the conventional PWM control signals. As can be easily seen in Figure 4, the PRS PWM provides uniform pulse distribution at low duty cycles.

Figure 4. Scope Images for Output Signal when PRS PWM or Counter-PWM Signals were used for Interlacing

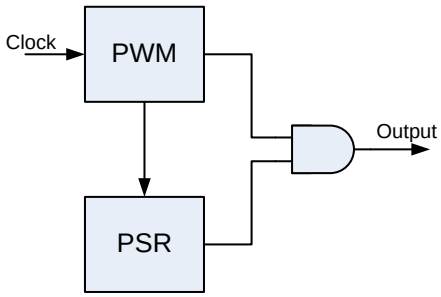


Special Applications

The interpolating PWM source can be adapted for special applications. Let's describe generator operation at very low duty cycles. In this mode, the first PWM source does not produce any pulses and the second PWM produces single clock pulses. Duty cycle modulation is done by varying the number of second generator pulses passed within the control signal generator-repeat period. Therefore, the source operates as a pulse frequency modulator (PFM), which generates fixed width pulses with variable frequency. With increasing duty cycle, the generator automatically switches to a pulse width modulation scheme. This mode is useful for high-efficiency DC-DC converters, such as, battery-powered devices that are in idle mode. These DC-DC converters should be kept high at low load currents.

The proposed PWM source can be simplified to operate only as the spread-spectrum PFM source. This can be used with step-down DC-DC converters when the input voltage is many times higher than the output. A typical application is a truck alarm system that operates with a 24 V battery and uses 3.3 V for a microprocessor. This can be used in modern vehicles with a supply voltage of 42 V. The regulator duty cycle is very low and the PFM is more efficient than the conventional PWM, therefore, most of the energy dissipation is carried by dynamic losses during transistor switching. Only one PWM counter source can be used and one PRS PWM or sigma delta modulator gates the PWM counter output. Figure 5 shows the flowchart of this modulator.

Figure 5. Spread Spectrum Pulse-Frequency Modulator Flowchart



Note By applying external voltage to the sigma delta modulator input, you can completely build a hardware-switching power supply that does not need any CPU influence upon runtime. The PSoC device possibilities are unlimited!

References

1. “Principles of Sigma-Delta Modulation for Analog-to-Digital Converters”, http://www.web-ee.com/primers/files/Sigma_Delta_Principles.pdf.

Summary

This application note describes PSoC® 1, a high speed, high-resolution PWM source that overcomes traditional PWM source limitations in output signal frequency and resolution.

About the Author

Name: Victor Kremin.

Document History

Document Title: PSoC® 1 – PWM Source – High Frequency, High Resolution – AN2246

Document Number: 001-33721

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1499983	YJI	10/04/2007	New application note.
*A	3178388	YJI	03/07/2011	Updated Document Title. Updated Introduction.
*B	4325065	GRAA	03/28/2014	Added Summary. Updated to new template. Completing Sunset Review.
*C	4622272	ASRI	01/13/2015	Updated Software Version as “PSoC® Designer™ 5.4” in page 1. Updated attached associated project to PSoC Designer 5.4. Completing Sunset Review.
*D	5700210	AESATP12	04/26/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2007-2017. 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.