The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

FM3 MB9B100A/300A/400A/500A Series Microcontroller Space Vector Pulse Width Modulation

This application note describes SVPWM function Theory, Block, Function, Flow, Sample, Parameter in FM3 family Inverter Platform.

Contents

1 Introduction

This application note describes SVPWM function Theory, Block, Function, Flow, Sample, Parameter in FM3 family Inverter Platform.

2 SVPWM Theory

Technical Background

2.1 Overview

In motor theory, motor speed is defined by the formula:

$$\boldsymbol{n = \frac{60f}{p}} \qquad (1)$$

Where f-Power frequency,p-pole pairs

Through the formula ,speed regulate need change the power frequency or the motor pole pairs, and change power frequency easier than change motor pole pairs, so, at present, the great mass of motor speed regulate use frequency convert.

How to change power frequency? Currently, use the type of: AC-DC-AC. It's means, convert 3-phase power AC into DC, then, invert the DC into 3-phase AC and change frequency. First, define the 3-phase power AC interphase voltage is:

$$\begin{cases} u_a = \sqrt{2}U\cos(\omega t) \\ u_b = \sqrt{2}U\cos(\omega t - 2\pi/3) \\ u_c = \sqrt{2}U\cos(\omega t - 4\pi/3) \end{cases} \qquad (2)$$

Take this 3-phase AC voltage into stationary system (α-β):

(The formulae deduce please reference the [AN] coordinate transform Clarke Transform formulae (12))

$$\begin{cases} u_\alpha = \left(\frac{2}{3}\right)(\mathrm{u_a} - 0.5\mathrm{u_b} - 0.5\mathrm{u_c}) = \sqrt{2}U\cos(\omega t) \\ u_\beta = \left(-\frac{2}{3}\right)\left(\frac{\sqrt{3}}{2}\mathrm{u_b} - \frac{\sqrt{3}}{2}\mathrm{u_c}\right) = \sqrt{2}U\sin(\omega t) \end{cases} \quad (3)$$

Take this into voltage space vector:

$$\mathrm{u_s} = u_\alpha + ju_\beta \quad (4)$$

In sine power:

$$\mathrm{u_s} = \sqrt{2}U(\cos\omega t + j\sin\omega t) = \sqrt{2}Ue^{j\omega t} \quad (5)$$

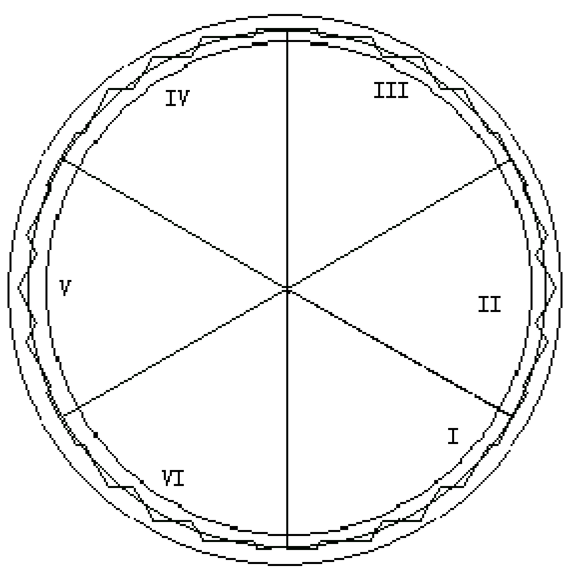
This equation is a circularity in plural plane. So, in motor theory, invariableness' voltage(U) and frequency, stator voltage space vector in plural plane move along with a circularity , and space vector move a cycle in a power frequency cycle.

Figure 1, the line voltage space vector will move other circularity , it's radius more than phase voltage circularity $\sqrt{3}$,and they have 30° phase angle.

In stator 3-phase voltage space vector will move along with a circularity (start) into another circularity (stabile).

The SVPWM use to regulate power frequency and draw a nice circularity through DC–AC part.

Figure 1



Theory

SVPWM: Space Vector Pulse Width Modulation.

Purpose: SVPWM can directly transform the stator voltage vectors from the two-phase α,β-coordinate system into pulse-width modulation (PWM) signals (duty cycle values). proverbial motor formulae is :

$$U_1 = \frac{d\Psi_1}{dt} + R_1 i_1 \qquad (6)$$

U_1 is stator voltage vector, Ψ_1 is stator flux vector, i_1 is stator current vector.

When U_1 is a standard sine wave, that is the same size and direction continuous change, the trajectory is an ideal circle. So,

$$\Psi_1 = \oint (U_1 - R_1 i_1)\, dt \qquad (7)$$

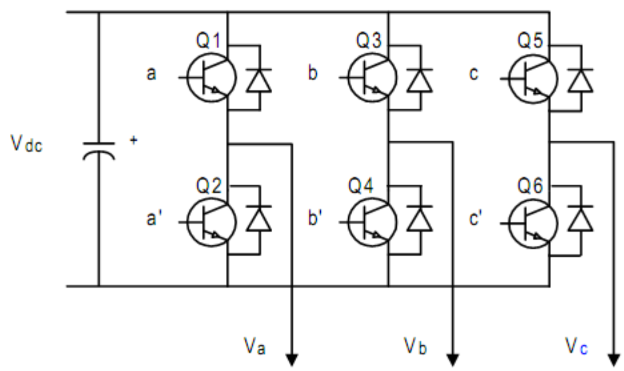
When $R_1 * i_1 << U_1$:

$$\Psi_1 = \oint U_1\, dt \qquad (8)$$

Flux trajectory depends on voltage vector.

So we can design the circuit of DC to 3-phase AC

Figure 2



As shown in figure 2, in the circuit diagram, have 6 power switch a, b, c, a', b', c' plot out they into 2 groups, S[a,b,c] , S'[a', b', c'], they control the switch ON/OFF, and they mutex,

$$S_x = \begin{cases} S = 1 \\ S' = 0 \end{cases}$$

It's means each voltage vector is coded by the three-digit number and have eight possible switching states (vectors) are feasible.

For example: S_x = [1, 0, 0].

The point [a], [b'], [c'] switch ON, and [a'], [b], [c] switch OFF.

$$U_{aN} = U_{dc}, \qquad U_{bN} = U_{cN} = 0$$

In motor theory,

$$\begin{cases} U_a - U_b = U_{aN} - U_{bN} = U_{dc} \\ U_a - U_c = U_{dc} \\ U_a + U_b + U_c = 0 \end{cases}$$

Take $U_{aN} = U_{dc},\ U_{bN} = U_{cN} = 0$ into the formula,

$$U_a = \frac{2U_{dc}}{3},\ \ U_b = -\frac{U_{dc}}{3},\ \ U_c = -\frac{U_{dc}}{3}$$

Similarly, can calculate other switching states, below in Table 2-1.

Table 1

Sa	Sa	Sa	Vector	UaN	UbN	UcN	Ua	Ub	Uc
0	0	0	U0	0	0	0	0	0	0
1	0	0	U4	Udc	0	0	2Udc/3	- Udc/3	- Udc/3
1	1	0	U6	Udc	Udc	0	Udc/3	Udc/3	- 2Udc/3
0	1	0	U2	0	Udc	0	- Udc/3	2Udc/3	- Udc/3
0	1	1	U3	0	Udc	Udc	- 2Udc/3	Udc/3	Udc/3
0	0	1	U1	0	0	Udc	- Udc/3	- Udc/3	2Udc/3
1	0	1	U5	Udc	0	Udc	Udc/3	- 2Udc/3	Udc/3
1	1	1	U7	0	0	0	0	0	0

Form formula (3) and (4), the voltage space vector formula:

$$u_s = \left(\frac{2}{3}\right)(u_a + a u_b + a^2 u_c) \qquad (9)$$

Where

$$a = \exp\left(j\frac{2\pi}{3}\right) \qquad a^2 = \exp\left(j\frac{4\pi}{3}\right)$$

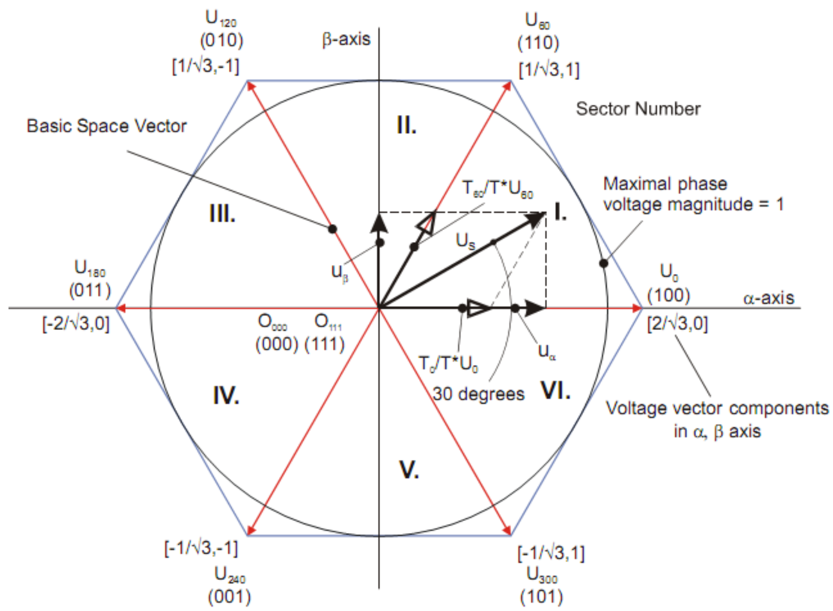
Take table 2-1 vector:

$$U_4 = \frac{2}{3}U_{dc} \qquad U_6 = \frac{2}{3}U_{dc}\ \exp\left(j\frac{\pi}{3}\right)$$
$$U_2 = \frac{2}{3}U_{dc}\ \exp\left(j\frac{2\pi}{3}\right) \qquad U_3 = \frac{2}{3}U_{dc}\ \exp(j\pi) \qquad (10)$$
$$U_1 = \frac{2}{3}U_{dc}\ \exp\left(j\frac{4\pi}{3}\right) \qquad U_5 = \frac{2}{3}U_{dc}\ \exp\left(j\frac{5\pi}{3}\right)$$

All of non-zero vectors have the same amplitude $\frac{2}{3}U_{dc}$, and between border two vectors angle is , $\frac{\pi}{3} = 60°$ for example, f=50Hz, the vector change time: $t = \frac{60}{360} * \frac{1}{50} = 3.33ms$.

Graphical representation of all combinations is the hexagon shown in Figure 3. There are six non-zero vectors, U0, U60, U120, U180, U240, U300, and two zero vectors, O000 and O111, defined in α, β coordinates.

Figure 3



In figure 4, the red line plot out the hexagon into six sectors, every between border two vectors angle is 60°, the border two vectors and zero vector could compose random voltage vector.

$$\int_0^T U_{ref}\,dt = \int_0^{T_x} U_x\,dt + \int_{T_x}^{T_x+T_y} U_y\,dt + \int_{T_x+T_y}^{T} U_0\,dt \qquad (11)$$

Or

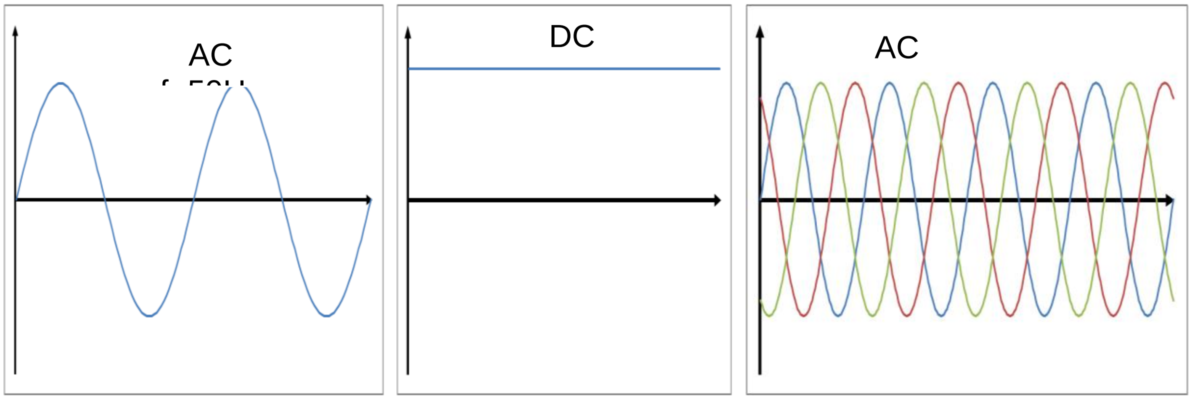
$$U_{ref}\,T = U_x\,T_x + U_y\,T_y + U_0\,T_0 \qquad (12)$$

This formula means, in sector the voltage vector rotating course could plot out many small part , it's have the same effect.

So, in the interest of bring 3-phase sine AC voltage, use the voltage vector compose, begin from U4(100),every time increase a little increment, every increment could compose from between border two vectors and zero vector. Then the enactment voltage vector equivalent with the rotating voltage vector, and that all is SVPWM.

The figure2-4 show the SVPWM transform course, a f = 50Hz single phase voltage transform into DC voltage, then transform into a f=100Hz 3-phase voltage.

Figure 4

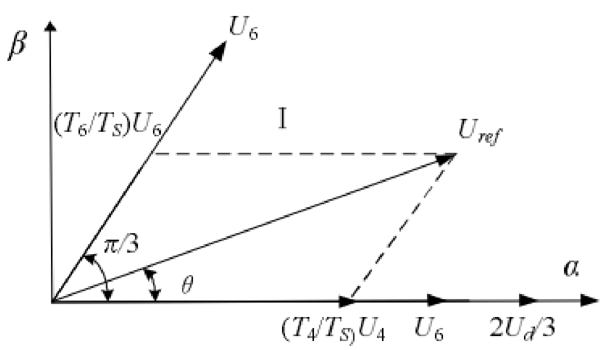


SVPWM Actualize

3-phase AC voltage rotating speed : $\omega = 2\pi f$, rotating a cycle need time T=1/f, if carrier wave frequency is f_s , the frequency ratio R =f_s/f, it's means voltage rotating plane plot out R unit increment , so the voltage vector increment angle is $r = \frac{2\pi}{R} = \frac{2\pi f}{f_s} = 2\pi T_s/T$

For example, need compose a voltage vector U_{ref} , it's station at I at sector first increment, then will use two non-zero vector(U4,U6) and two zero vector(U0,U7), shown the Figure 5.

Figure 5



In stationary system 2-phase (α-β), the U_{ref} with U4 angle is θ, then

$$\frac{|U_{ref}|}{\sin 2\pi/3} = \frac{|T_6 U_6/T_z|}{\sin\theta} = \frac{|T_4 U_4/T_z|}{\sin(\frac{\pi}{3} - \theta)} \tag{13}$$

Because: $U_4 = U_6 = \frac{2}{3} U_{dc}$, the vector hold time is

$$\begin{cases} T_4 = mT_s \sin(\frac{\pi}{3} - \theta) \\ T_6 = mT_s \sin\theta \end{cases} \tag{14}$$

$$m = \frac{\sqrt{3}|U_{ref}|}{U_{dc}}$$

Where m is SVPWM regulate ratio:

Then zero vector hold time: $T_7 = T_0 = (T_s - T_4 - T_6)/2$

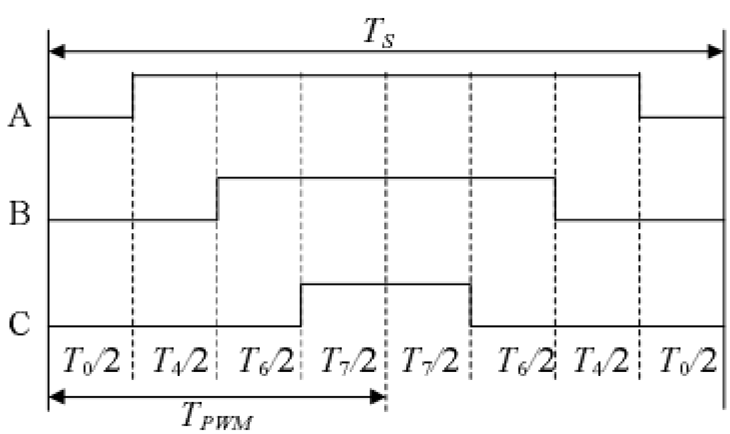
For reduce the switch waste, every time only changes one phase switch state, and the best sequence is below the Table 2-2.

Table 2

Uref	**switch sequence**
I sector (0°≤θ≤60°)	0-4-6-7-7-6-4-0
II sector (60°≤θ≤120°)	0-2-6-7-7-6-2-0
III sector (120°≤θ≤180°)	0-2-3-7-7-3-2-0
IV sector (180°≤θ≤240°)	0-1-3-7-7-3-1-0
V sector (240°≤θ≤300°)	0-1-5-7-7-5-1-0
VI sector (300°≤θ≤360°)	0-4-5-7-7-5-4-0

Then use the table 2-2 ,after voltage rotating a cycle, will have R unit compose vector, and in I sector have 3-phase voltage wave shown in figure 6

Figure 6



2.2 SVPWM Arithmetic in FM3 family

In FM3 family Inverter platform SVPWM compose sine wave need have 4 steps:

1. Make sure the U_{ref} belong to which sector.
2. To calculate X,Y,Z which are used to determine open time of 2 adjacent basic vector.
3. To determine each conduction time for 2 adjacent basic vector.
4. To determine To write OCCP register value to CH1,CH3,CH5 which used to create SVPWM waveform

2.2.1 Make sure the U_{ref} belong to which sector

Function Application

Function: To calculate sector no. of SVPWM voltage vector.

Function Name: CalcSectorNo

C file name: SVPWM.C, SVPWM.H

Function interface:

```
INT8U CalcSectorNo(_stDataInFixAxis *pstDataInFixAxis //fixed 2-axis system)
typedef struct
{
   Q15_VAL32 alpha_Q15;//
   Q15_VAL32 beta_Q15;//
}_stDataInFixAxis;
```

Table 3

Item	Name	Description	Format
Inputs	alpha	phase- alpha of fixed 2- phase	Q15_VAL32
	beta	phase- beta of fixed 2- phase	Q15_VAL32
Outputs	SectorNo	Sector no. of SVPWM voltage vector	Q15_VAL32

The following code is example for this module.

```
void example_ CalcSectorNo ()
{
    CurrentInFixAxis. alpha_Q15= INa;//Input alpha
    CurrentInFixAxis. beta_Q15= INb;//Input beta
    CalcSectorNo (&pstDataInFixAxis);
    OUTa= SectorNo;
}
```

Determine open time of 2 adjacent basic vector

Function Application

Function: To calculate X,Y,Z which is used to determine open time of 2 adjacent basic vectors.

Function Name: CalcDutyTime

C file name: SVPWM.C, SVPWM.H

```
void CalcDutyTime(_stDataInFixAxis *pstDataInFixAxis, _stXYZ *pstXYZ,
  INT8U WhichMFT, INT8U WhichFRTCh)
typedef struct
{
   Q15_VAL32 alpha_Q15;
   Q15_VAL32 beta_Q15;
}_stDataInFixAxis;
typedef struct
{
   INT32S X;
   INT32S Y;
   INT32S Z;
}_stXYZ;
```

Table 4

Item	Name	Description	Format
Inputs	alpha	phase- alpha of fixed 2- phase	Q15_VAL32
	beta	phase- beta of fixed 2- phase	Q15_VAL32
	WhichMFT	Which MFT, valid value is MFT0, MFT1	INT8U
	WhichFRTCh	Which FRT channel to configure	INT8U
Outputs	X	X is conduction time	INT32S
	Y	Y is conduction time	INT32S
	Z	Z is conduction time	INT32S

The following code is example for this module.

```
void  example_ CalcDutyTime ()
    pstDataInFixAxis. alpha_Q15= INa;//Input alpha
    pstDataInFixAxis. beta_Q15= INb;//Input beta
    WhichMFT=0;
    WhichFRTCh=0;
    CalcDutyTime (&pstDataInFixAxis,& pstXYZ, WhichMFT, WhichFRTCh);
    OUTa= pstXYZ->X;
    OUTb= pstXYZ->Y;
    OUTc= pstXYZ->Z;
```

2.2.2 Calculate conduction time

Function Application

Function: To determine each conduction time for 2 adjacent basic vectors.

Function Name: CalcTon

C file name: SVPWM.C, SVPWM.H

```
void CalcTon(INT8U SectorNo, _stXYZ *pstXYZ, _stOnTimingOfUVW *pstOnTimingOfUVW,
             INT8U WhichMFT, INT8U WhichFRTCh)
typedef struct {
   INT32S X;
   INT32S Y;
   INT32S Z;
}_stXYZ;
typedef struct
{
   INT32S TU;
   INT32S TV;
   INT32S TW;
}_stOnTimingOfUVW;
```

Table 5

Item	Name	Description	Format
Inputs	X	X is conduction time	INT32S
	Y	Y is conduction time	INT32S
	Z	Z is conduction time	INT32S
	SectorNo	Sector no. of SVPWM voltage vector	INT8U
	WhichMFT	Which MFT, valid value is MFT0, MFT1	INT8U
	WhichFRTCh	Which FRT channel to configure	INT8U
Outputs	TU	Conduction time of A phase connecting to DC power supply	INT32S
	TV	Conduction time of B phase connecting to DC power supply	INT32S
	TW	Conduction time of C phase connecting to DC power supply	INT32S

The following code is example for this module.

```
void  example_ CalcTon ()
{
    pstXYZ .X= INa;
    pstXYZ .Y = INb;
    pstXYZ .Z=INc;
    SectorNo=1;
    WhichMFT=0;
```

```
    WhichFRTCh=0;
    CalcTon (SectorNo ,& pstXYZ,& pstOnTimingOfUVW, WhichMFT, WhichFRTCh);
    OUTa= pstOnTimingOfUVW ->TU;
    OUTb= pstOnTimingOfUVW ->TV;
    OUTc= pstOnTimingOfUVW ->TW;
}
```

2.2.3 Create SVPWM waveform

Function Application

Function: To determine To write OCCP register value to CH1,CH3,CH5 which used to create SVPWM waveform

Function Name: WriteCmpValToReg

C file name: SVPWM.C, SVPWM.H

```
void WriteCmpValToReg(INT8U WhichMFT, _stOnTimingOfUVW *pstOnTimingOfUVW)
typedef struct
{
   INT32S TU;
   INT32S TV;
   INT32S TW;
}_stOnTimingOfUVW;
```

Table 6

Item	Name	Description	Format
Inputs	TU	Conduction time of A, phase connecting to DC power supply	INT32S
	TV	Conduction time of B, phase connecting to DC power supply	INT32S
	TW	Conduction time of C phase connecting to DC power supply	INT32S
	WhichMFT	Which MFT, valid value is MFT0, MFT1	INT8U

The following code is example for this module.

```
void  example_ WriteCmpValToReg ()
{
    pstOnTimingOfUVW .TU=INa;
    pstOnTimingOfUVW .TV=INb;
    pstOnTimingOfUVW .TW=INc;
    WhichMFT=0;
    WriteCmpValToReg (WhichMFT,& pstOnTimingOfUVW);
```

3 Document History

Document Title: AN204347 - FM3 MB9B100A/300A/400A/500A Series Microcontroller Space Vector Pulse Width Modulation

Document Number: 002-04347

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	–	CBZH	07/04/2011	Initial Release
*A	5029359	CBZH	12/04/2015	Migrated Spansion Application Note AN706-00036-1v0-E to Cypress format

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

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless
Spansion Products	spansion.com/products

PSoC® Solutions

psoc.cypress.com/solutions

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

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2011-2015. 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.

This 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.