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.

AN205315

FR, MB91460, Up/Down Counter

16-bit Up/Down Counter counts up or down within the range of 0 to 65535. Specifically, Up/Down Counter running in the phase difference count mode is suitable for counting the encoder pulse of motors and other equipment.

Contents

1 Introduction....1
1.1 Key Features....1
2 UP / DOWN Counter....2
2.1 Block Diagrams....2
2.2 Registers....2
2.3 PFM Counter Operation....5
3 Software Example....10
3.1 Basic setting of the up / down counter....10
4 Additional Information....12
Document History....13

1 Introduction

Triggered by an input signal, 16-bit Up/Down Counter counts up or down within the range of 0 to 65535. Specifically, Up/Down Counter running in the phase difference count mode is suitable for counting the encoder pulse of motors and other equipment. When encoder's output signals of phase A, phase B and phase Z are applied, the counter can achieve precise counting of rotation angles or number of revolutions.

1.1 Key Features

- Two 16bit or four 8bit counter
- Four types of count Mode: Timer mode, Up/Down count mode, Phase difference count mode (Multiply by 2) and Phase difference count mode (Multiply by 4).
- Count source can be either Peripheral clock (CLKP) or External trigger
- Counting rage is from 0 to 65535
- Interrupt can be selected out of four type i.e. Counter Match, under flow, over flow and Count direction change.

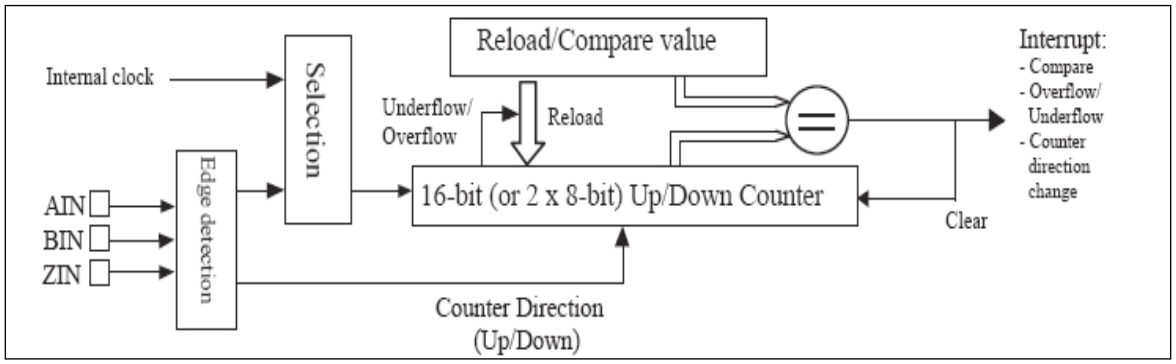
2 UP / DOWN Counter

The basic functionality of up/down counter is explained

2.1 Block Diagrams

Figure 1 shows the internal block diagram of an Up/down counter.

Figure 1. Up/Down counter Block Diagram



2.2 Registers

2.2.1 Counter Control Register (UDCCn)

This register is used to control behaviors of Up/Down Counter.

Table 1. UDCCn

Bit No.	Name	Explanation	Value	Operation
15	M16E	Enable 16 bit mode	0	8 bit × 2 channel operation mode
			1	16 bit × 1 channel operation mode
14	CDCF	Direction change detection	0	Read: No direction change occurred Write: Clear the flag
			1	Read: Direction change occurred Write: No effect
13	CFIE	Direction change interrupt request	0	Disable direction change interrupt requests
			1	Enable direction change interrupt requests
12	CLKS	Internal clock frequency	0	CLKP / 2
			1	CLKP / 8
11-10	CMS1, CMS0	Counter mode	00	Timer mode (Countdown)
			01	Up/down count mode
			10	Phase difference count mode (Multiply by 2)
			11	Phase difference count mode (Multiply by 4)
9-8	CES1,CES0	Edge selection	00	Disable edge detection
			01	Detect a falling edge
			10	Detect a rising edge
			11	Detect both rising and falling edges
7	-	*Reserved*	-	Write “0”. Read value is the value written
6	CTUT	Counter write	0	No impact on operation

Bit No.	Name	Explanation	Value	Operation	
			1	Transfer data from the RCR register to UDCR.	
5	UCRE	Enable compare-match clear	0	Disable counter clear due to compare-match.	
			1	Enable counter clear due to compare-match	
4	RLDE	Enable reload	0	Disable reload function	
			1	Enable reload function	
3	UDCLR	Clear UDCR	0	Set (Clear) Up/Down Counter (UDCR) to “0000H”.	
			1	No impact on operation	
2	CGSC	Select counter clear/gate	0	Counter clear function	
			1	Gate function	
1-0	CGE1,CGE0	Edge detection/level selection		CGSC=“0”	CGSC=“1”
			00	Disable edge detection	Disable level detection
			01	Detect a falling edge	Detect a “L” level
			10	Detect a rising edge	Detect a “H” level
			11	Disable setting	Disable setting

n = 1 to 4

2.2.2 Count Status Register (UDCSn)

This register is used to control Up/Down Counter and to indicate the status of the counter

Table 2. UDCSn

Bit No.	Name	Explanation	Value	Operation
7	CSTR	Enable count operation	0	Disable count operation
			1	Enable count operation
6	CITE	Enable compare-match interrupt requests	0	Disable compare-match interrupt requests
			1	Enable compare-match interrupt requests
5	UDIE	Overflow/underflow interrupt request	0	Disable overflow/underflow interrupt requests.
			1	Enable overflow/underflow interrupt requests
4	CMPF	Compare-match detection flag	0	Read: Comparison results do not match Write: Clear the flag
			1	Read: Comparison results match Write: No effect
3	OVFF	Overflow detection flag	0	Read: No overflow Write: Clear the flag
			1	Read: An overflow has occurred Write: No effect
2	UDFF	Underflow detection flag	0	Read: No overflow Write: Clear the flag
			1	Read: An underflow has occurred Write: No effect
1-0	UDF1-UDF0	Up/down flag	00	No input
			01	Count Down
			10	Count Up
			11	Both of Count up and Count Down

n = 1 to 4

2.2.3 Up/Down Counter Register

This register is used to read the count value of Up/Down Counter.

- 16 Bit Mode (M16E= “1”)

  In the 16 bit mode, UDCR10 and UDCR32 register functions as 16-bit up/down counter register

- 8 Bit Mode (M16E= “0”)

  In the 16 bit mode, UDCR0, UDCR1, UDCR2 and UDCR3 register functions as 8bit up/down counter register

2.2.4 Up/Down Reload/Compare Register

This register is used to reload a value to Up/Down Counter and to compare the register value with the Counter value. This register is also used to write to Up/Down Counter.

- 16 Bit Mode (M16E= “1”)

  In the 16 bit mode, UDRC10 and UDRC32 register functions as 16-bit reload/compare register

- 8 Bit Mode (M16E= “0”)

  In the 16 bit mode, UDRC0, UDRC1, UDRC2 and UDRC3 register functions as 8bit reload/compare register

2.2.5 Port function register (DDRn. PFRn, EPFRn)

To enable AIN, BIN and ZIN pin as a Input one should set corresponding DDRn, PFRn and EPFRn register. For more information please refer data sheet of respective device.

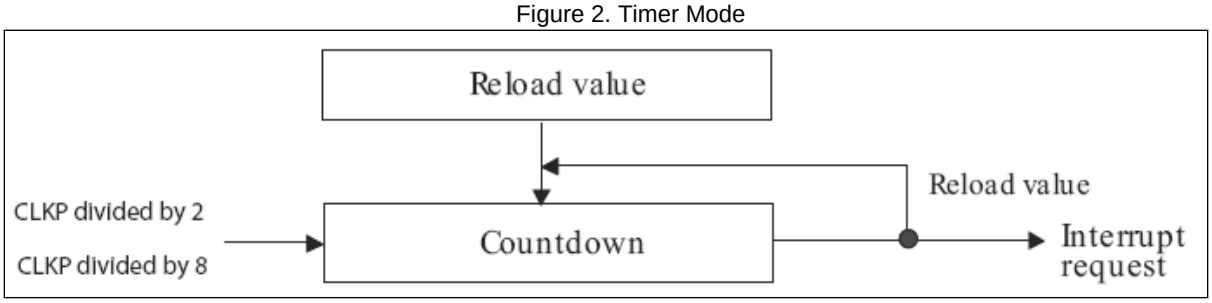
2.3 PFM Counter Operation

This section describes each operation mode for Up/Down Counter

2.3.1 Timer Mode

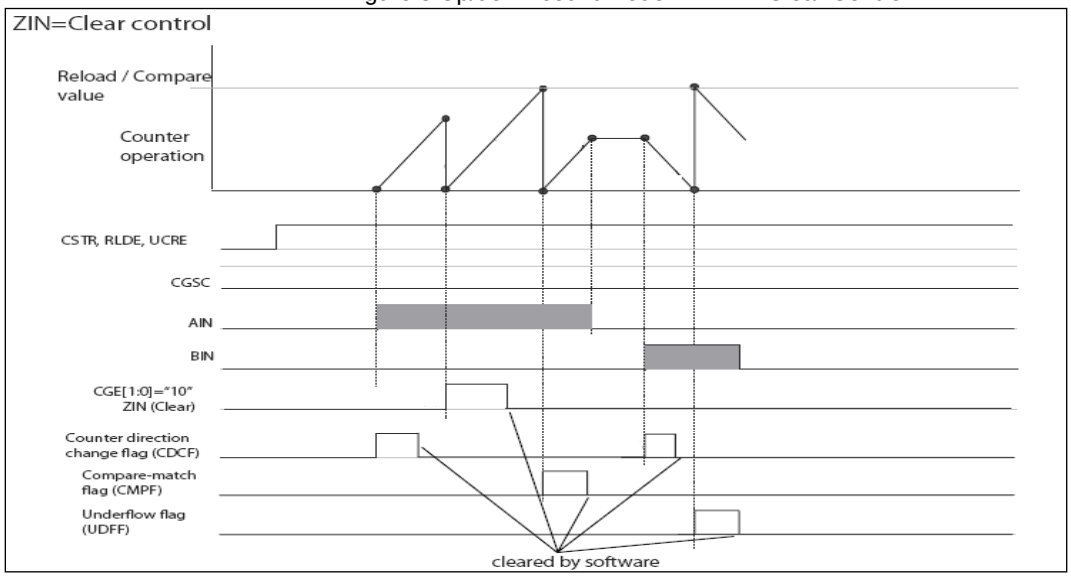
After counter initialization and triggered by software, it start counting down. When under flow occurs, reload value is loaded in Up/down counter and it starts counting down again. Interrupt is generated at underflow if enabled by software.

Figure 2. Timer Mode



2.3.2 Up/Down Count Mode – ZIN -> Clear Control

Figure 3.Up/down count mode – ZIN -> Clear Control



After initialization, when pulse input to the AIN pin is detected, Up/Down Counter counts up. The count direction change flag is set to “1”.

When an edge is applied to the ZIN pin, Up/Down Counter is cleared.

When the Up/Down Counter's count value matches with the compare value (compare-match) than the compare-match flag is set to “1” and counter is cleared

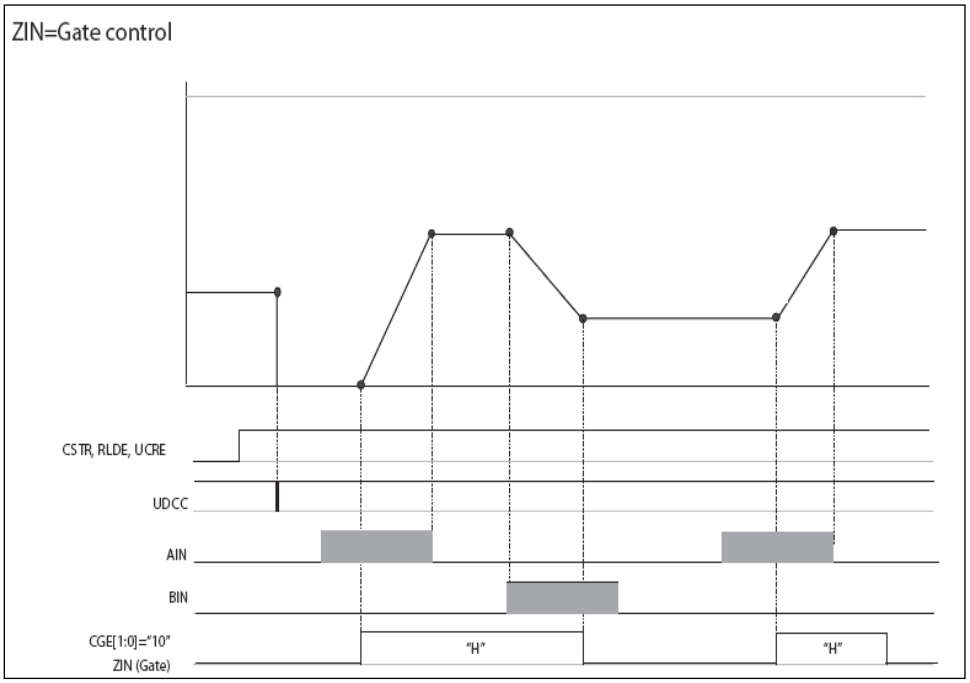
When pulse input to the AIN pin stops, Up/Down Counter stops counting

When pulse input to the BIN pin is detected, Up/Down Counter counts down and count direction change flag is set to “1”.

When Up/Down Counter is under flowed than the underflow flag is set to “1”. The underflow causes the reload value to be reloaded to Up/Down Counter.

2.3.3 Up/Down Count Mode – ZIN -> Gate Control

Figure 4. Up/down count mode – ZIN -> Gate Control



Here counting will not start unless and until ZIN input is enabled.

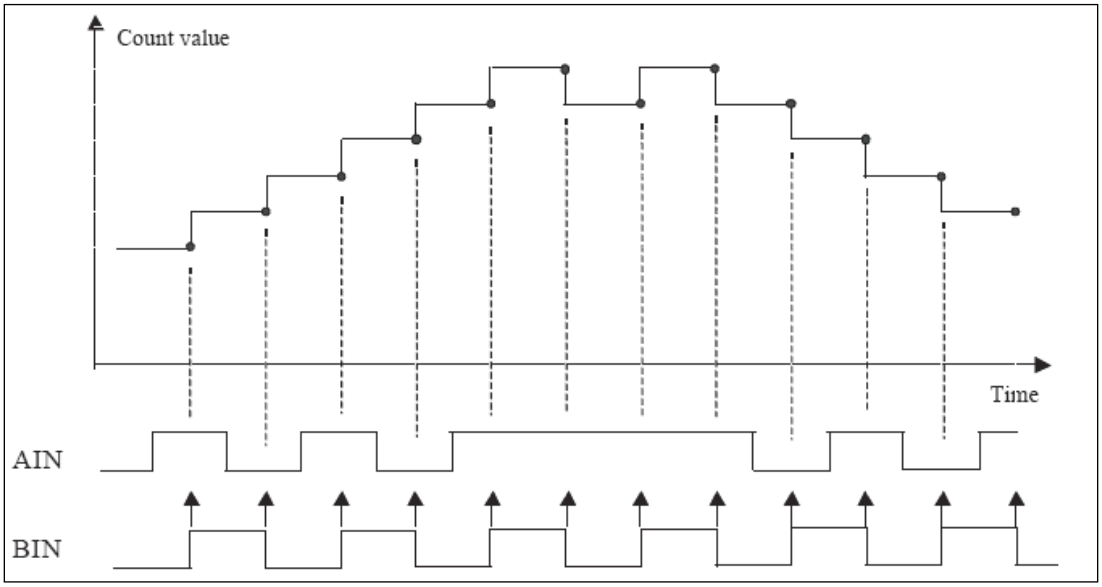
When ZIN is enabled counter starts counting up when pulses are applied at AIN input and Counter counts down when a pulse input to the BIN pin is detected.

Up/Down Counter stops counting when pulse input to the AIN or BIN pin stops and ZIN input is still enabled.

2.3.4 Phase Difference Count Mode (Multiply by 2)

On the rising and falling edges at the BIN count pin, Up/Down Counter counts up or down, depending on the voltage level at the AIN pin.

Figure 5. Phase Difference Count Mode (Multiply by 2)



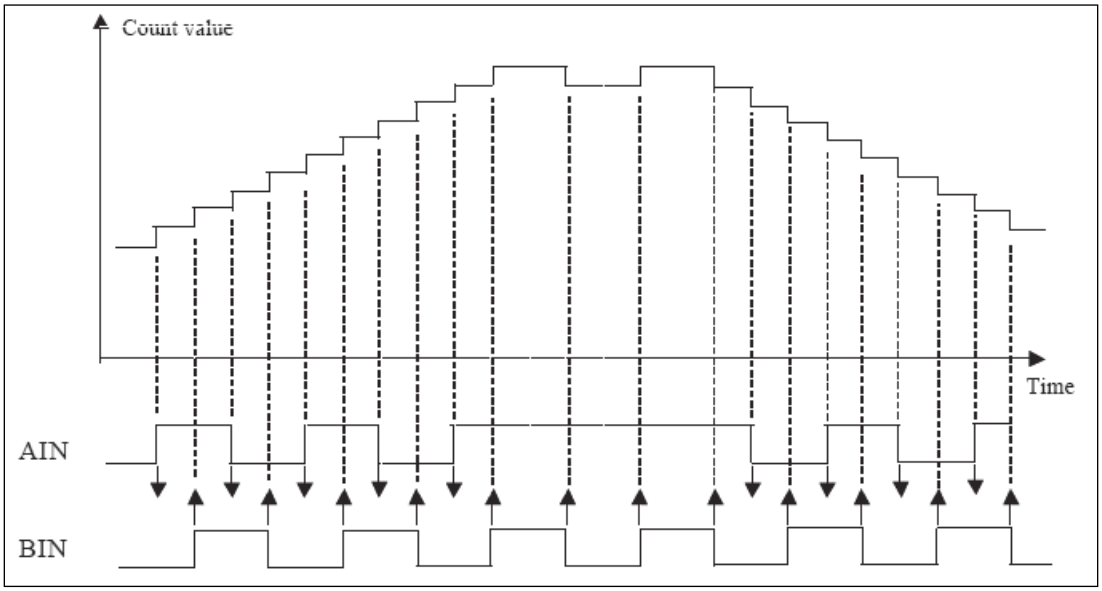
Counter counts up when the ‘H ’voltage level at the AIN pin detected, on the rising edge at the BIN and after that when the ‘L’ voltage level at the AIN pin detected, on the falling edge at the BIN pin.

Counter counts up down when the ‘L’ voltage level at the AIN pin detected, on the rising edge at the BIN pin and after that when the ‘H’ voltage level at the AIN pin detected, on the falling edge at the BIN pin.

When this count mode is selected, selection of the edge to be detected using UDCCn: CES1 or UDCCn: CES0 is disabled.

2.3.5 Phase Difference Count Mode (Multiply by 4)

Figure 6. Phase Difference Count Mode (Multiply by 4)



On the rising and falling edges at the BIN pin, Up/Down Counter counts up or down, depending on the voltage level at the AIN pin, and on the rising and falling edges at the AIN pin, Up/Down Counter counts up or down, depending on the voltage level at the BIN pin.

Counter counts up…

1. When the ‘H’ voltage level at the AIN pin detected, on the rising edge at the BIN pin
2. When the ‘L’ voltage level at the AIN pin detected, on the falling edge at the BIN pin
3. When the ‘L’ voltage level at the BIN pin detected, on the rising edge at the AIN pin
4. When the ‘H’ voltage level at the BIN pin detected on the falling edge at the AIN pin

Counter counts down…

1. When ‘L’ voltage level at the AIN pin detected on the rising edge at the BIN pin
2. When the ‘H’ voltage level at the AIN pin detected on the falling edge at the BIN pin
3. When the ‘H’ voltage level at the BIN pin detected, on the rising edge at the AIN
4. When the ‘L’ voltage level at the BIN pin detected, on the falling edge at the AIN pin

When Up/Down Counter is used to count encoder output, high precise counting of rotation angles and number of revolutions, as well as detecting of rotation directions, can be achieved by applying encoder output signals of phase A and phase B to the AIN and BIN, respectively.

Note that when this count mode is selected, selection of the edge to be detected using UDCCn: CES1 or UDCCn: CES0 is disabled.

2.3.6 Clear Timing

When a clear request (Compare-match, ZIN edge detection and writing “0” to the clear bit UDCCn: UDCLR) is made, clear is performed next time when Up/Down Counter counts up.

Even if a clear request (Compare-match, ZIN edge detection and writing “0” to the clear bit UDCCn: UDCLR) is made, clear is not performed when UP/Down Counter counts neither up nor down.

If Up/Down Counter does not count up after a clear request (Compare-match, ZIN edge detection and writing “0” to the clear bit UDCCn: UDCLR) is made, the counter is cleared when counting is disabled (UDCSn: CSTR=“0”).

When Up/Down Counter exceeds the maximum count, the overflow flag is set to “1” and the counter value is returned to “0000”.

2.3.7 Reload Timing

The next time when Up/Down Counter counts down below “0000”, an underflow occurs (an interrupt request is generated) and then reloading is performed. If clear and reload operations occur at the same time, clear takes precedence.

2.3.8 Writing a Value to Counter

1. Counting of Up/Down Counter is disabled UDCCn: CSTR = “0”
2. A value is written to UDRC
3. “1” is written to the count write bit UDCCn: CTUT
4. A value is transferred from the reload/compare register `UDCR` to Up/Down Counter

3 Software Example

Example for Up / Down Counter

3.1 Basic setting of the up / down counter

The following example shows how to set up the up / down counter.

```
/*                          SAMPLE CODE                                        */
/*                           main.c                                            */
/*-----------------------------------------------------------------------------*/
void Up_Down_Counter0_init(char val_CMS, char val_CGSC)
{
   DDR20_D0 = 0;
   PFR20_D0 = 1;      // AIN input
   DDR20_D1 = 0;
   PFR20_D1 = 1;
   EPFR20_D1 = 1;     // BIN input
   DDR20_D2 = 0;
   PFR20_D2 = 1;
   EPFR20_D2 = 1;     // ZIN input
   UDCC0_CLKS = 1;    // internal clock frequency F_clkp/8
   UDCC0_M16E = 0;    // 8/16 bits
   UDCC0_CMS = val_CMS; // timer mode (countdown)
   UDRC0 = 0xFF;      // reload value
   UDCC0_RLDE = 1;    //Enable reload function
   UDCC0_UCRE = 1;    //Enable counter clear due to compare-match
   UDCC0_CES = 2;     //Detect a rising edge
     UDCC0_CGE = 2;   //Select counter clear/gate edge - Detect a rising
                       //edge
   UDCC0_CTUT = 1;    //Transfer data from the RCR register to UDCR
   UDCS0_UDIE = 1;    // enable over/underflow request
   UDCC0_CFIE = 1;    // Enable direction change interrupt requests
   UDCS0_CITE = 1;    // Enable compare-match interrupt requests.
   UDCC0_CGSC = val_CGSC;  //set CGSC 0: Clear count 1: Gate Function
   UDCC0_CDCF = 0;    // clear the direction flag
   UDCS0_UDFF = 0;    // clear underflow flag
   UDCS0_OVFF = 0;    // clear overflow flag
}
void Up_Down_Counter0_activate(char activate)
{
   if (activate) UDCS0_CSTR = 1; // Start counting
   else UDCS0_CSTR = 0;
}
void main(void)
{
__EI();               //enable interrupts
__set_il(31);         //allow all levels
InitIrqLevels();      //init interrupts

PORTEN = 0x3;         //enable I/O Ports

      Up_Down_Counter0_init (0, 0); // Timer Mode, ZIN - > Clear Control
      // Up_Down_Counter0_init (0, 1); // Timer Mode, ZIN - > Gate Control
      // Up_Down_Counter0_init (1, 0); // UP/Down Count, ZIN - > Clear Control
      // Up_Down_Counter0_init (1, 1); // UP/Down Count, ZIN - > Gate Control
```

```
   // Up_Down_Counter0_init (2, 0); // Phase Difference Count Mode (Multiply
                                    // by 2) ZIN - > Clear Control
  // Up_Down_Counter0_init (2, 1); // Phase Difference Count Mode (Multiply
                                   // by 2) ZIN - > Gate Control
  // Up_Down_Counter0_init (3, 0);  // Phase Difference Count Mode (Multiply
                                    // by 4) ZIN - > Clear Control
  // Up_Down_Counter0_init (3, 1); // Phase Difference Count Mode (Multiply
                                   // by 4) ZIN - > Gate Control
      Up_Down_Counter0_activate (1);
while (1)            // endless loop
      {
      HWWD_CL = 0;
   }
}
__interrupt void UDCounter0IRQHandler (void)
{
   if(UDCS0_UDFF)
   {
      /* DO some thing */
      UDCS0_UDFF = 0;
   }
   if(UDCS0_OVFF)
   {
      /* DO some thing */
      UDCS0_OVFF = 0;
   }
   if(UDCS0_CMPF)
   {
      /* DO some thing */
      UDCS0_CMPF = 0;
   }
   if(UDCC0_CDCF)
   {
      /* DO some thing */
      UDCC0_CDCF = 0;
   }
}
```

```
/*                            SAMPLE CODE                                    */
/*                               vector.c                                    */
/*---------------------------------------------------------------------------*/
void InitIrqLevels(void)
{
   ...
   ICR56 = 30;                       /* Up/Down Counter 0    */
   ...
}
 __interrupt void UDCounter0IRQHandler (void);

 #pragma intvect UDCounter0IRQHandler 128   /* Up/Down Counter 0   */
```

4 Additional Information

Information about Cypress Microcontrollers can be found on the following Internet page:

http://www.cypress.com/cypress-microcontrollers

The software examples related to this application note is:

91460_UpDownCounter

It can be found on the following Internet page:

http://www.cypress.com/cypress-mcu-product-softwareexamples

Document History

Document Title: AN205315 - FR, MB91460, Up/Down Counter

Document Number: 002-05315

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	06/10/2008	V1.0, First draft, HPi
*A	5090479	NOFL	04/12/2016	Converted Spansion Application Note “MCU-AN-300064-E-V10” to Cypress format
*B	5868087	AESATMP9	08/30/2017	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, 2008-2017. 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.