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.

AN205313

FR, MB91460, Free Running Timer

The Free-Run timer consists of a 16-bit timer (up counter) and control circuits. Output Compare unit and Input Capture unit is also discussed here since Free-Run timer is an integral part of their operation.

Contents

1 Introduction

The Free-Run timer consists of a 16-bit timer (up counter) and control circuits. Output Compare unit and Input Capture unit is also discussed here since Free-Run timer is an integral part of their operation.

The Input Capture unit (ICU) stores the free-run timer value if an edge is detected at an ICU input pin. Using this stored value it is possible to calculate the time between edges from the external signal.

The Output compare unit reverses the level of the output pin (OCUn), if the compare register value matches the timer value of the Free-Run timer.

1.1 Key Features

1.1.1 16-Bit Free Running Timer

- 8 Timers available (Free-Run timer 0 up to Free-Run timer 7)
- Clock selectable for Free Running Timer are 1/4, 1/16, 1/32, and 1/64 of Peripheral Clock
- External clocking possible
- Interrupt Generation at counter Overflow or Counter Match with Compare Clear register.
- Up-counter

1.1.2 Output Compare Unit

- Per channel 16-bit compare register available, total four groups, eight channels available (output compare channels 0/1, 2/3, 4/5, 6/7).
- Output compare channels 0/1 use Free-Run timer 2, Output compare channels 2/3 use Free-Run timer 3, Output compare channels 4/5 use Free-Run timer 6, Output compare channels 6/7 use Free-Run timer 7.
- Setting of initial output level value is possible
- Interrupt Generation on Compare Match

1.1.3 Input Capture Unit

- Per channel 16-bit buffer available, total four groups, eight channels available (input capture channels 0/1, 2/3, 4/5, 6/7).
- Input Capture channels 0/1 use Free-Run timer 0, Input Capture channels 2/3 use Free-Run timer 1, Input Capture channels 4/5 use Free-Run timer 4, Input Capture channels 6/7 use Free-Run timer 5.
- Rising, falling or both Edge selectable for Input Event
- Interrupt Generation on Input Event
- Detection of start/stop edges in LIN-Sync-field connected to ICU 0 and 2

2 Free-Run Timer, Input Capture unit & Output Compare Unit

The Basic Functionality of the Free Run Timer, Input Capture Unit and Output Compare Unit Module

2.1 Block Diagrams

Figure 1 shows the internal block diagram of an FREE-RUN Timer.

Figure 1. Free-RUN Timer Block Diagram

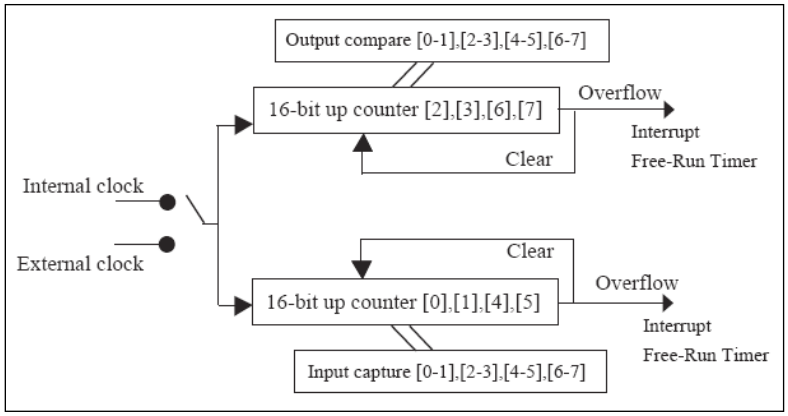


Figure 2. Input Capture Block Diagram

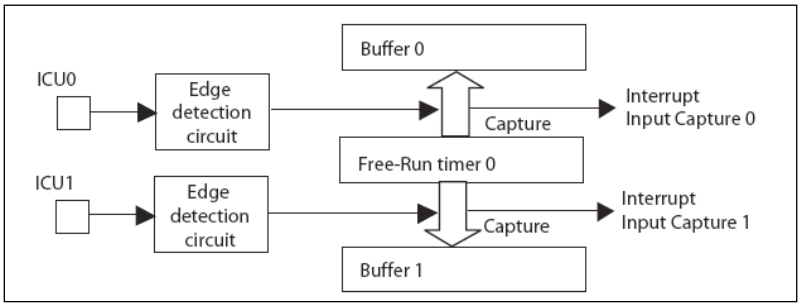
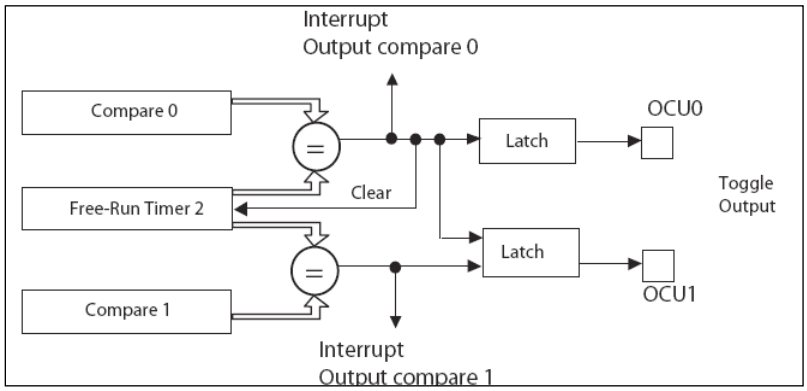


Figure 3. Output Compare Block Diagram



2.2 Registers

2.2.1 16-Bit Free Running Timer

The 16-Bit Free Running Timer is an up-counter.

2.2.1.1 Timer Control Register (TCCSn)

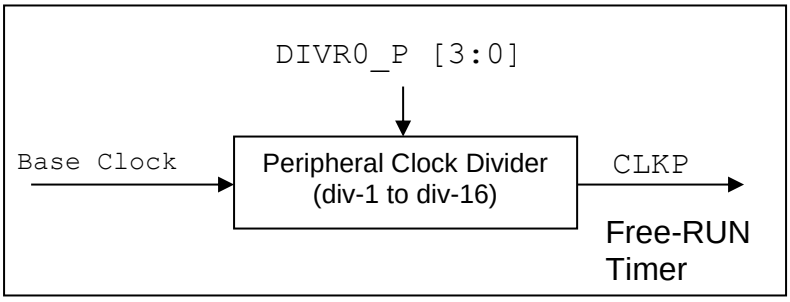
This Byte contains several status and control bits.

Table 1. TCCSL

Bit No.	Name	Explanation	Value	Operation
7	ECLK	Select the count clock	0	Internal clock (the peripheral clock divided by n)
			1	External clock (CK pin)
6	IVF	Interrupt Request	0	Read: No interrupt, Write: Clear Request
			1	Read: Interrupt request; Write: No effect
5	IVFE	Interrupt Request Enable	0	No Interrupts
			1	Interrupt Enabled
4	STOP	Stop the Counter	0	Counter Enabled
			1	Counter Disabled
3	MODE	Set Reset Condition of Timer	0	Initialization by Reset or Clear Bit
			1	Initialization by Reset, Clear Bit, or Compare Register (OCCP)
2	CLR	Clear Timer	0	Write: no effect
			1	Write: Clear to “0000”
1,0	CLK1,CLK0	Count clock division ratio selection	0, 0	CLKP / 4 CLKP / 16 CLKP / 32 CLKP / 64
			0, 1	
			1, 0	
			1, 1	

Please consider that the source clock frequency of the Reload Timer (CLKP) depends on the settings of the Clock Division setting register (DIVR0).

Figure 4. Free-RUN Timer Clock



2.2.1.2 Data Register of the Free-Running Timer (TCDT)

This 16-Bit register reads out the current counter value. Writing sets a new value to it. This register must be accessed by word access instructions.

2.2.2 Output Compare Unit

2.2.2.1 Output Control Register (OCSmn)

This 16-Bit register contains several control and status bits.

Table 2. OCS

Bit No.	Name	Explanation	Value	Operation for Channel *(2n+0)* or *(2n+1)*
15, 13	–	*Undefined*	-	*Writing does not affect the operation. The read out value is “1”.*
12	CMOD	Reverse Mode	0	Independent operation
			1	Combined operation
11,10	–	*Undefined*	-	*Writing does not affect the operation. The read out value is “1”.*
9	OTDn	Pin-level settings	0	Set the output level of pin OCUn to “L”
			1	Set the output level of pin OCUn to “L”
8	OTDm	Pin-level settings	0	Set the output level of pin OCUn to “H”
			1	Set the output level of pin OCUn to “H”
7	ICPn	Interrupt flag	0	Read: No Interrupt present Write: Clear flag
			1	Read: Interrupt present Write: No effect on operation
6	ICPm	Interrupt flag	0	Read: No Interrupt present Write: Clear flag
			1	Read: Interrupt present Write: No effect on operation
5	ICEn	Interrupt request enable	0	Disable output compare n interrupt requests
			1	Enable output compare n interrupt requests
4	ICEm	Interrupt request enable	0	Disable output compare m interrupt requests
			1	Enable output compare m interrupt requests
3,2	–	*Undefined*	-	*Writing does not affect the operation. The read out value is “1”.*
1	CSTn	Enable operation requests	0	Stop operation of output compare n
			1	Enable operation of output compare n
0	CSTm	Enable operation requests	0	Stop operation of output compare m
			1	Enable operation of output compare m

mn = 01,23,45,67

2.2.2.2 Output Compare Register (OCCPn)

For each channel, there are two 16-Bit wide Output Compare Registers. The 16-bit Output Compare registers are compared with the 16-bit Free-Running Timer. When the value of the register matches that of the 16-bit Free-Running Timer, a compare signal is generated and the output compare interrupt flag is set. If output is enabled, the output level corresponding to the compare register is reversed.

Note: n = 0 to 7

2.2.3 Input Capture Unit

2.2.4 Input Capture Control Register (ICSmn)

This 8-Bit register contains several control and status bits.

Table 3. ICS

Bit No.	Name	Explanation	Value	Operation for Channel *(2n+0)* or *(2n+1)*
7	ICPn	Input capture n interrupt flag	0	Read: No edge detected, Write: Clear Flag
			1	Edge detected
6	ICPm	Input capture m interrupt flag	0	Read: No edge detected, Write: Clear Flag
			1	Read: Edge detected Write: No effect on operation
5	ICEn	Interrupt Enable Bit	0	Interrupt disabled
			1	Interrupt enabled
4	ICEm	Interrupt Enable Bit	0	Interrupt disabled
			1	Interrupt enabled
3, 2	Egn1, Egn0	Edge Selection Bit	0, 0	Input capture unit is stopped
			0, 1	Rising Edge Detection
			1, 0	Falling Edge Detection
			1, 1	Both Edge Detection
1, 0	Egm1, EGm0	Edge Selection Bit	0, 0	Input capture unit is stopped
			0, 1	Rising Edge Detection
			1, 0	Falling Edge Detection
			1, 1	Both Edge Detection

mn = 01,23,45,67

2.2.4.1 Input Capture Data Register (IPCP)

This 16-Bit register stores the timer value if an edge event occurs.

2.2.4.2 Input Capture Unit and LIN-USART

Some ICUs can be connected internally to LIN-USARTs. This feature is for measurement of baud rates for LIN-USART as LIN slave.

Hence while designing a LIN-slave application, ICUs for LIN-USART and ICUs for external events has to be identified exclusively.

3 Software Example

Example For I/O Timer

3.2 Basic Functionality of the Free Running Timer

The following example shows how to set up the Free Running Timer.

```
/*                              SAMPLE CODE                                    */
/*-----------------------------------------------------------------------------*/

void InitFRTimer0(void)
{
   TCCS0 = 0x20;           // CLKP clock / 4, interrupt enabled, clear
                           // timer
}

   . . .

__interrupt void FRTimer0(void)
{
   TCCS0_IVF = 0;          // clear interrupt request

   . . .
  }
```

At 16 MHz Peripheral Clock, the above example generates interrupts at an interval of 16.38 ms.

Please note, that the corresponding interrupt vector and level has to be defined in the vectors.c module of our standard template project.

```
/*                          SAMPLE CODE                                        */
/*-----------------------------------------------------------------------------*/
void InitIrqLevels(void)
{
. . .

ICR12 = 30;                              // Free Run Timer 0
. . .
}

__interrupt void FRTimer0 (void);        // prototype

   . . .

  #pragma intvect FRTimer0          40  // FRT0

   . . .
```

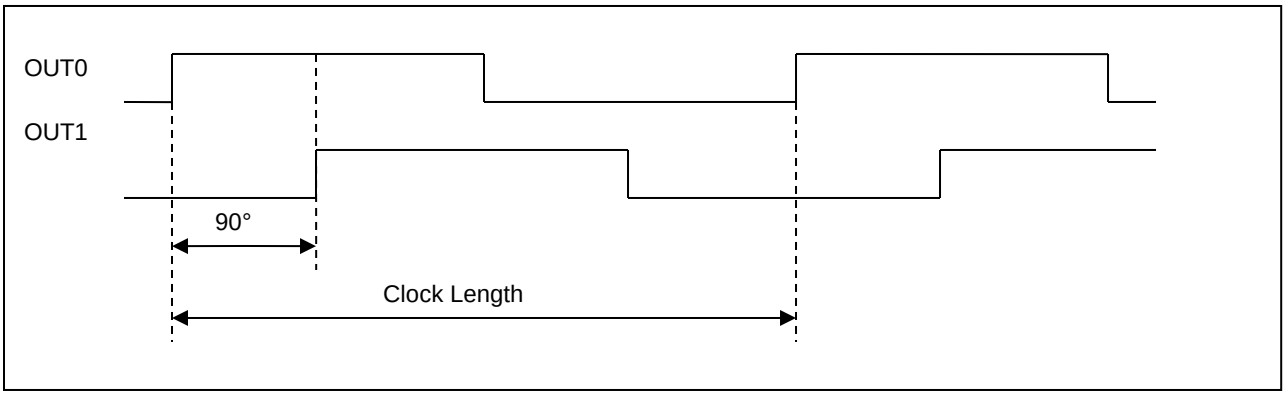
The following example shows how to set up the Output Compare. OCU0 and OCU1 are used with the Free Running 0. OCU1 is phase-shifted by 90° (Counter Value = 0x8000 = ½ · 0xFFFF + 1).

```
/*                              SAMPLE CODE                                    */
/*-----------------------------------------------------------------------------*/

void InitFRTimer0(void)
{
    TCCS0 = 0x00;    // CLKP/4, interrupt disabled, clear timer
}

void InitOCU01(void)
{
    EPFR15_D0 = 0;   // OCU 0 & 1 output enable
    PFR15_D0 = 1;
    EPFR15_D1 = 0;
    PFR15_D1 = 1;
    OCS01 = 0x0003;  // Mode 0, Pin-level settings 'L', Interrupt Diasble
    OCCP0 = 0x0000;  // Frequency = Free Running Timer
    OCCP1 = 0x8000;  // Phase = 90 deg.
 }
```

The above configuration generates a PWM signal with 50% duty cycle. The cycle length depends solely on the CLKP frequency and the selected pre-scalar setting of the Free-RUN Timer

The Figure 5 shows the output waveform:

Figure 5. Output Compare - Waveform



3.3 Basic Functionality of the Input Capture Unit

Assume you want to use Input Capture Unit 0 with both edge detection. The following code shows a solution. Please note, that for all Inputs the corresponding Port Input Enable Register has to be set first.

The main difference to External Interrupts is that the actual counter value of the Free Running Timer is stored in the Input Capture Data Register (IPCPn) at interrupt occurrence.

```
/*                                SAMPLE CODE                                  */
/*-----------------------------------------------------------------------------*/

void InitFRTimer0(void)
{
TCCS0 = 0x00;                // CLKP/4, interrupt disabled, clear timer
}

void InitICU0(void)
{
    EPFR14_D0 = 0;           // Enable ICU port input
    PFR14_D0 = 1;
ICS01 = 0x13;               // Interrupt enable, both edge detection
}
    . . .

__interrupt void InterruptICU0(void)
{
    ICS01_ICP0 = 0;          // clear request
    . . .                   // IPCP0 can be read here for external
                            // time/frequency measurement
 }
```

Please also note, that the corresponding interrupt vector and level has to be defined in the *vectors.c* module of our standard template project.

```
/*                                SAMPLE CODE                                  */
/*-----------------------------------------------------------------------------*/
void InitIrqLevels(void)
{
. . .
ICR38 = 30;                                 // Input Capture Unit 0

. . .
}
__interrupt void InterruptICU0 (void);  // prototype
    . . .
 #pragma intvect InterruptICU0    92    // ICU0

    . . .
```

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:

96340_frt_icu_frt_ocu

91460_icu-v10

It can be found on the following Internet page:

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

Document History

Document Title: AN205313 - FR, MB91460, Free Running Timer

Document Number:002-05313

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	04/11/2008	V1.0, First draft, HPi
*A	5089988	NOFL	04/06/2016	Converted Spansion Application Note “MCU-AN-300062-E-V10” to Cypress format
*B	5846500	MALI	08/08/2017	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

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