TM4C123 Use of GPTM with GPIO Pins

From EdWiki

Use of Timers with the I/O Pins

If the timers are only used for the timing-base or timing-up indications, one does not need to use these Capture/Compare pins (**CCP**) since these pins are mainly connected to the external signals to detect the number of events that have occurred or the time period events that have been experienced.

In the Lab8 (/edwiki/EmSys:TM4C123_Timer_Programming) session, we showed how to use timers to generate time delay. In this lab session, we will examine the use of timers with the I/O pins. There are five modes for each timer block. They are :

1. One-shot/Periodic mode,
2. Real-time clock mode,
3. Input edge-time mode,
4. Input edge-count mode and
5. Pulse Width Modulation (PWM).

TimerA and TimerB CCP pins

There are TimerA and TimerB for each of the Timer block 0 to 5. There are one or two designated pins for each of the TimerA and TimerB. For example, TimerA of Timer block 4 is connected internally to pin PC0 of PortC and TimerB of Timer block 4 is connected to pin PC1 of PortC. Notice, TimerA pins are also called **CCP0** and TimerB pins are called **CCP1**. Some of the timers have option to be connected to one of the two pins. For example, TimerA of Timer1 (**T1CCP0**) can use **PB0** or **PF2** pins. Table 9.1 shows the pin designations for 16/32-bit Timer block 0 to 5.

	TimerA	Pins	TimerB	Pins
Timer0	T0CCP0	PB6 or PF0	T0CCP1	PB7 or PF1
Timer1	T1CCP0	PB4 or PF2	T1CCP1	PB5 or PF3
Timer2	T2CCP0	PB0 or PF4	T2CCP1	PB1
Timer3	T3CCP0	PB2	T3CCP1	PB3
Timer4	T4CCP0	PC0	T4CCP1	PC1
Timer5	T5CCP0	PC2	T5CCP1	PC3

Table 9.1: The Pin Designation for 16/32 bit Timer Block 0 to 5

General-Purpose Timers CCP pins distributions

Timer Pin	I/O pin	Pin Function
T0CCP0	PB6 PF0	16/32-Bit Timer 0 Capture/Compare/PWM 0
T0CCP1	PB7 PF1	16/32-Bit Timer 0 Capture/Compare/PWM 1
T1CCP0	PB4 PF2	16/32-Bit Timer 1 Capture/Compare/PWM 0
T1CCP1	PB5 PF3	16/32-Bit Timer 1 Capture/Compare/PWM 1
T2CCP0	PB0 PF4	16/32-Bit Timer 2 Capture/Compare/PWM 0

T2CCP1	PB1	16/32-Bit Timer 2 Capture/Compare/PWM 1
T3CCP0	PB2	16/32-Bit Timer 3 Capture/Compare/PWM 0
T3CCP1	PB3	16/32-Bit Timer 3 Capture/Compare/PWM 1
T4CCP0	PC0	16/32-Bit Timer 4 Capture/Compare/PWM 0
T4CCP1	PC1	16/32-Bit Timer 4 Capture/Compare/PWM 1
T5CCP0	PC2	16/32-Bit Timer 5 Capture/Compare/PWM 0
T5CCP1	PC3	16/32-Bit Timer 5 Capture/Compare/PWM 1
WT0CCP0	PC4	32/64-Bit Wide Timer 0 Capture/Compare/PWM 0
WT0CCP1	PC5	32/64-Bit Wide Timer 0 Capture/Compare/PWM 1
WT1CCP0	PC6	32/64-Bit Wide Timer 1 Capture/Compare/PWM 0
WT1CCP1	PC7	32/64-Bit Wide Timer 1 Capture/Compare/PWM 1
WT2CCP0	PD0	32/64-Bit Wide Timer 2 Capture/Compare/PWM 0
WT2CCP1	PD1	32/64-Bit Wide Timer 2 Capture/Compare/PWM 1
WT3CCP0	PD2	32/64-Bit Wide Timer 3 Capture/Compare/PWM 0
WT3CCP1	PD3	32/64-Bit Wide Timer 3 Capture/Compare/PWM 1
WT4CCP0	PD4	32/64-Bit Wide Timer 4 Capture/Compare/PWM 0
WT4CCP1	PD5	32/64-Bit Wide Timer 4 Capture/Compare/PWM 1
WT5CCP0	PD6	32/64-Bit Wide Timer 5 Capture/Compare/PWM 0
WT5CCP1	PD7	32/64-Bit Wide Timer 5 Capture/Compare/PWM 1

Table 9.2: General-Purpose Timers signals and GPIO pins distributions.

Selecting alternate function for Timers pin

Upon Reset, the **GPIOAFSEL** register has all 0s meaning the I/O pins are used as simple I/O. To use an alternate function, we first must set the bit in the AFSEL register to 1 for that pin. For example, for the PB6, we need to write 0x40 to GPIO_PORTB_AFSL_R register. After that, the **GPIOPCTL** register must be configured for the desired function. To do that, we need to use the information in Table 23-5 (Page No.1351) of TI Tiva TM4C123GH6PM data sheet (http://shukra.dese.iisc.ernet.in/emsys/tivac/ek-tm4c123gxl/TM4C123GH6PM_Microcontroller_Data_Sheet.pdf). Table 9.2 provides the summary for the Timers pins.

Timer Pin	**I/O pin**	**How to use peripheral function on the pin**
T0CCP0	PB6	GPIO_PORTB_AFSEL_R = 0x40
	PF0	GPIO_PORTF_AFSEL_R = 0x01
T0CCP1	PB7	GPIO_PORTB_AFSEL_R = 0x80
	PF1	GPIO_PORTF_AFSEL_R = 0x02
T1CCP0	PB4	GPIO_PORTB_AFSEL_R = 0x10
	PF2	GPIO_PORTF_AFSEL_R = 0x04
T1CCP1	PB5	GPIO_PORTB_AFSEL_R = 0x20
	PF3	GPIO_PORTF_AFSEL_R = 0x08
T2CCP0	PB0	GPIO_PORTB_AFSEL_R = 0x40
	PF4	GPIO_PORTF_AFSEL_R = 0x10
T2CCP1	PB1	GPIO_PORTB_AFSEL_R = 0x02
T3CCP0	PB2	GPIO_PORTB_AFSEL_R = 0x04
T3CCP1	PB3	GPIO_PORTB_AFSEL_R = 0x08
T4CCP0	PC0	GPIO_PORTC_AFSEL_R = 0x01

T4CCP1	PC1	GPIO_PORTC_AFSEL_R = 0x02
T5CCP0	PC2	GPIO_PORTC_AFSEL_R = 0x04
T5CCP1	PC3	GPIO_PORTC_AFSEL_R = 0x08
WT0CCP0	PC4	GPIO_PORTC_AFSEL_R = 0x10
WT0CCP1	PC5	GPIO_PORTC_AFSEL_R = 0x20
WT1CCP0	PC6	GPIO_PORTC_AFSEL_R = 0x40
WT1CCP1	PC7	GPIO_PORTC_AFSEL_R = 0x80
WT2CCP0	PD0	GPIO_PORTD_AFSEL_R = 0x01
WT2CCP1	PD1	GPIO_PORTD_AFSEL_R = 0x02
WT3CCP0	PD2	GPIO_PORTD_AFSEL_R = 0x04
WT3CCP1	PD3	GPIO_PORTD_AFSEL_R = 0x04
WT4CCP0	PD4	GPIO_PORTD_AFSEL_R = 0x10
WT4CCP1	PD5	GPIO_PORTD_AFSEL_R = 0x20
WT5CCP0	PD6	GPIO_PORTD_AFSEL_R = 0x40
WT5CCP1	PD7	GPIO_PORTD_AFSEL_R = 0x80

Table 9.3: Timers Alternate Pin Assignments

Timer Pin	**I/O pin**	**How to select the timer function on the pin**
T0CCP0	PB6	GPIO_PORTB_PCTL_R = 0x0700.0000
	PF0	GPIO_PORTF_PCTL_R = 0x0000.0007
	PB7	GPIO_PORTB_PCTL_R = 0x7000.0000

T0CCP1	PF1	GPIO_PORTF_PCTL_R = 0x0000.0070
	PB4	GPIO_PORTB_PCTL_R = 0x0007.0000
T1CCP0	PF2	GPIO_PORTF_PCTL_R = 0x0000.0700
	PB5	GPIO_PORTB_PCTL_R = 0x0070.0000
T1CCP1	PF3	GPIO_PORTF_PCTL_R = 0x0000.7000
	PB0	GPIO_PORTB_PCTL_R = 0x0000.0007
T2CCP0	PF4	GPIO_PORTF_PCTL_R = 0x0007.0000
T2CCP1	PB1	GPIO_PORTB_PCTL_R = 0x0000.0070
T3CCP0	PB2	GPIO_PORTB_PCTL_R = 0x0000.0700
T3CCP1	PB3	GPIO_PORTB_PCTL_R = 0x0000.7000
T4CCP0	PC0	GPIO_PORTC_PCTL_R = 0x0000.0007
T4CCP1	PC1	GPIO_PORTC_PCTL_R = 0x0000.0070
T5CCP0	PC2	GPIO_PORTC_PCTL_R = 0x0000.0700
T5CCP1	PC3	GPIO_PORTC_PCTL_R = 0x0000.7000
WT0CCP0	PC4	GPIO_PORTC_PCTL_R = 0x0007.0000
WT0CCP1	PC5	GPIO_PORTC_PCTL_R = 0x0070.0000
WT1CCP0	PC6	GPIO_PORTC_PCTL_R = 0x0700.0000
WT1CCP1	PC7	GPIO_PORTC_PCTL_R = 0x7000.0000
WT2CCP0	PD0	GPIO_PORTD_PCTL_R = 0x0000.0007

WT2CCP1	PD1	GPIO_PORTD_PCTL_R = 0x0000.0070
WT3CCP0	PD2	GPIO_PORTD_PCTL_R = 0x0000.0700
WT3CCP1	PD3	GPIO_PORTD_PCTL_R = 0x0000.7000
WT4CCP0	PD4	GPIO_PORTD_PCTL_R = 0x0007.0000
WT4CCP1	PD5	GPIO_PORTD_PCTL_R = 0x0070.0000
WT5CCP0	PD6	GPIO_PORTD_PCTL_R = 0x0700.0000
WT5CCP1	PD7	GPIO_PORTD_PCTL_R = 0x7000.0000

Table 9.4: Timers pin assignment using GPIO_PCTL

Example 9.1: Write the code to provide the T0CCP1 function on PF1.

Solution: The I/O pin is used as a peripheral function pin by setting the AFSEL register of PORTF and the desired peripheral function is selected by setting the PCTL register of PORTF:

```
GPIO_PORTF_AFSEL_R |= 0x02;
GPIO_PORTF_PCTL_R = 0x00000070;
```

Some Implementations on GPTM Modules

Timer system is a complex system with many different components and functions. We only discuss some very popular and important implementations. These include:

- Using timer **capture** functions to detect the number of input edges.
- Using timer **detecting** functions to measure the period for periodic signals.
- Using timer **control** functions to generate PWM signals to control motors.

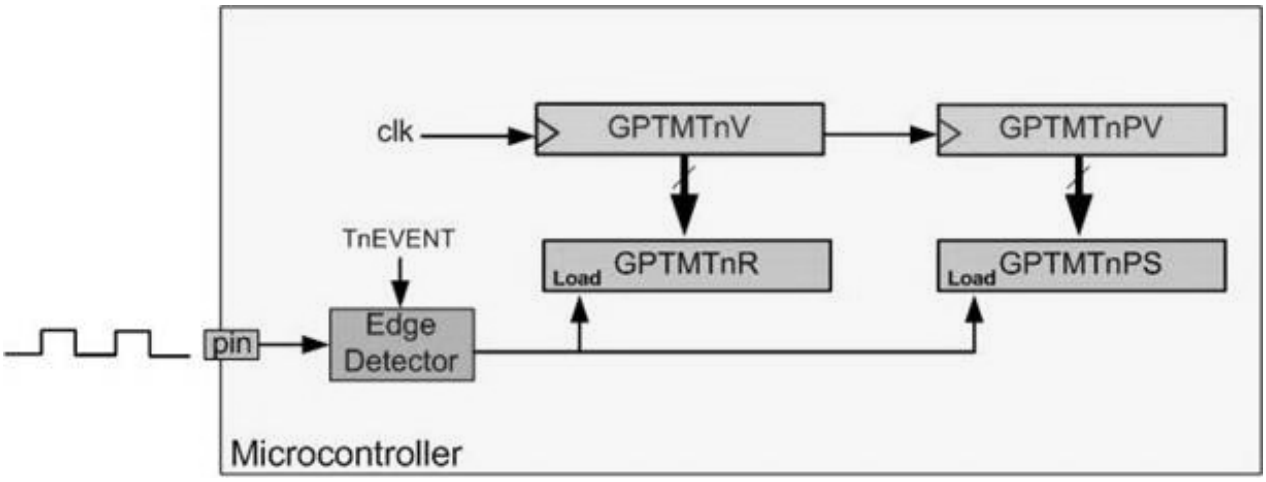
Using Timer for input edge-time capturing

Input edge-time capture mode

In input edge-time mode, an I/O pin is used to capture the signal transition events. When an event occurs, the content of the timer counter is captured in another register while the counter keeps counting. The program can then read the counter value when the event occurs at a slightly later time.

To configure the timer to input edge time mode the TnMR and TnCMR bits of GPTMTnMR should be set to capture edge-time mode (TnMR = 3 and TnCMR = 1). See Table 8.5 (/edwiki/EmSys:TM4C123_Timer_Programming#TimerA_Mode_selection_register_.28GPTMTAMR.29).

In this mode, the timer counter value is stored in the GPTMTnR register whenever the input pin is triggered by an external event. See Figure 9.1



(/edwiki/File:Tm4c_input_edge_time_capture.png)

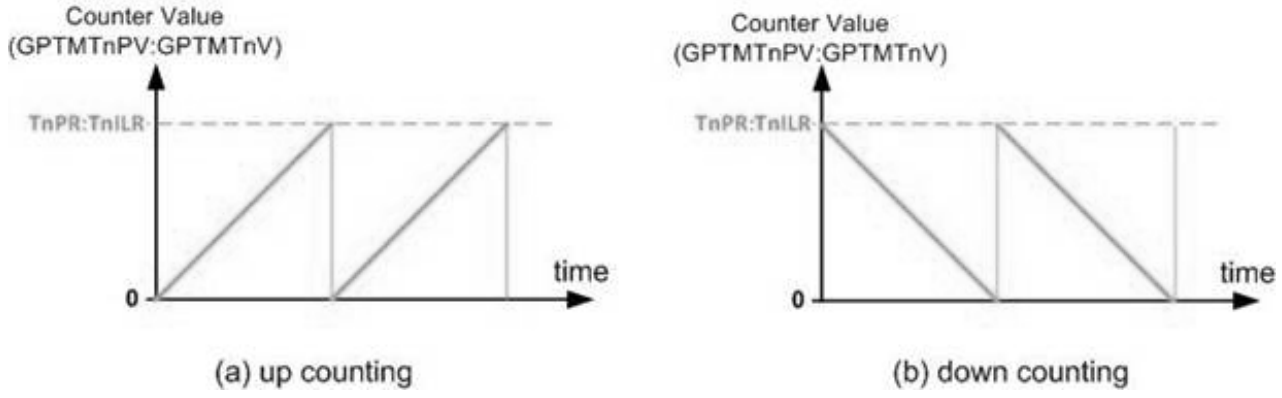
Figure 9.1: Input Edge Time Capturing

The timer can be configured to capture on the falling edge, rising edge, or both. To determine the type of edge that is captured, the TnEVENT bits of the GPTMCTL register should be initialized. See Table 8.4 (/edwiki/EmSys:TM4C123_Timer_Programming#TimerA_Mode_selection_register_.28GPTMTAMR.29)

Timer counting in input edge time mode

In edge time mode GPTMTnV and the optional prescaler are combined to make a 24-bit or 48-bit timer. The timer is 24-bit when 16-bit timers are used. The timer is 48-bit when 32-bit timers are used.

In up counting mode, the GPTMTnV and GPTMTnPV registers are initialized with 0s and they count up until they reach to GPTMTnILR and GPTMTnPR, respectively. Then, they are reset to 0s again.



(/edwiki/File:Tm4c_counting_in_input_edge_time_mode.png)

Figure 9.2: Counting in Input Edge-Time Mode

In down counting mode, the GPTMTnV and GPTMTnPV registers are initialized with GPTMTnILR and GPTMTnPR, respectively and they count down until they reach 0. Then, they are reloaded with GPTMTnILR and GPTMTnPR, again. Notice that capturing has no effect on counting and the timer continues counting when the capturing event takes place.

Initialization and Configuration for Input Edge-Time Mode

Perform the following operational steps to complete the initialization and configuration process for this mode (n = A or B):

1. Disable the selected timer by clearing the TnEN bit in the GPTMCTL register.
2. Initialize the GPTMCFG register by writing 0x4 to setup all timers as 16-bit default timers.
3. Configure the TnMR and TnCMR fields in the GPTMTnMR register by writing:
    - TnMR = 0x3 for capture mode.
    - TnCMR = 0x1 for edge time mode.
    - Select a count direction by programming the TnCDIR bit (0=count-down; 1=count-up).
4. Configure the event type (positive-going, negative-going, or both) that the timer captures by writing the TnEVENT field of the GPTMCTL register.
5. If a prescaler is to be used, write the prescale value to the GPTMTnPR register.
6. Load the timer start value into the GPTMTnILR register.
7. If interrupts are required, set the CnEIM bit in the GPTMIMR register.
8. After these initializations and configurations are done, set the TnEN bit in the GPTMCTL register to enable the timer and begin waiting for edge events.
9. If no interrupt is used, one can poll the CnERIS bit in the GPTMRIS register to wait for the edge event to occur. If interrupt is used, put appropriate codes inside the interrupt handler to process the interrupt. In both cases, the status flags are cleared by writing a 1 to the CnECIR bit on the GPTMICR register. The time at which the event happened can be obtained by reading the GPTMTnR register.

In the Input Edge Timing mode, the timer continues running after an edge event has been detected, but the timer interval can be changed at any time by writing the GPTMTnILR register and clearing the TnILD bit in the GPTMTnMR register. The change takes effect at the next cycle after the write taking placing.

Using Timer As a Counter

As shown in Table 8.4, a timer works as a counter when the TAMR bits of the GPTMTnMR are configured to capture mode and the TACMR bit is cleared. In this situation, the timer counts whenever the input pin is triggered. See Figure 9.3. The pin can be configured to count on the falling edge, rising edge, or both. To determine the type of edge that is counted, the TnEVENT bits of the GPTMCTL register should be initialized. See Table 8.3 and Table 8.4.