



**UNIVERSIDADE
FEDERAL DO CEARÁ**
CAMPUS QUIXADÁ

Universidade Federal do Ceará - Campus Quixadá

Relatório sobre TIMERS
QXD0149 - Técnicas de Programação para
Sistemas Embarcados I

Prof. Dr. Francisco Helder Candido
Outubro 2020

Aluno: Samuel Henrique - **Matrícula:** 473360

Quixadá-CE

Sumário

1	Introdução	3
2	DMTimer	3
2.1	Introdução	3
2.1.1	Visão Geral	3
2.1.2	Recursos	3
2.2	Descrição Funcional	4
2.2.1	Funcionalidade do Modo Cronômetro	4
2.2.2	Funcionalidade do Modo de Captura	5
2.2.3	Funcionalidade do Modo de Comparação	6
3	Principais Registradores	6
4	Como configurar	6

1 Introdução

Relatório sobre o uso do módulo DMTimer, baseado no capítulo 20.1 do Manual do ARM335X, que apresenta várias funções e modos junto com ele. É bastante importante o seu uso, pois ele confere precisão à aplicação que está sendo desenvolvida, pois tudo é baseado no tempo.

Esse relatório contém a explicação de como funciona esse módulo e de como é feita a configuração dele.

2 DMTimer

2.1 Introdução

2.1.1 Visão Geral

O módulo de *timer* contém um contador com capacidade de recarregamento automático, caso ocorra um *overflow*. O contador pode ser lido e escrito em tempo-real (enquanto está contando). Um sinal de saída dedicado pode ser usado para propósito geral, basta programar o bit 14 do TCLR Register. Um sinal dedicado de entrada é usado para acionar a captura automática do contador do cronômetro e interromper o evento, no tipo de transição de sinal que for programado. Esse módulo é controlável através do barramento periférico OCP.

2.1.2 Recursos

O cronômetro possui alguns recursos como:

- Modo de comparação e de captura
- Recarregamento automático
- Modo *start-stop*

- Divisor de *clock* programável
- Endereçamento de 16-32-bits
- Registradores de leitura/escrita dinâmicos
- Modo de despertar habilitado
- Compatível com interface OCP

2.2 Descrição Funcional

Três modos são suportados:

- Modo Cronômetro
- Modo Captura
- Modo de Comparação

Por padrão, o modo de captura e o modo de comparação são desativados.

2.2.1 Funcionalidade do Modo Cronômetro

O cronômetro é um contador crescente, que pode ser iniciado e parado a qualquer momento através do **registrador TCLR** no bit ST (TCLR ST bit). O **registrador TCRR** pode ser carregado quando parado ou em tempo real. O cronômetro é interrompido e o valor do contador é zerado quando a reinicialização do módulo é confirmada. O cronômetro é mantido parado após o *reset* ser liberado. Quando o cronômetro é interrompido, o **registrador TCRR** também é interrompido. O cronômetro pode ser reiniciado a partir do valor colegado, a não ser que TCRR tenha sido escrito com um novo valor.

O modo *one shot*, o contador é parado quando ocorre um *overflow*, esse modo é ativado pelo **registrador TCLR**, no bit AR = 0.

O modo *auto-reload*, é quando o contador é recarregado com o valor presente no **Timer Load Register (TLDR)** após um *overflow* ter ocorrido. Esse modo pode ser habilitado através do **registrador TCLR**, no bit **AR** = 1.

Uma interrupção pode ser emitida caso ocorra um *overflow*, para isso basta habilitar o bit **OVF_IT_FLAG** dentro do **registrador IRQENABLE_SET**. Um pino de saída dedicado (PORTIMERPWM) é programado através do **TCLR** (bits **TRG** e **PT**) para gerar um pulso positivo ou para inverter o valor atual quando o *overflow* acontecer.

2.2.2 Funcionalidade do Modo de Captura

O modo de captura serve para captura o valor do cronômetro quando ocorrer alguma transição, seja transição de subida, descida ou ambas. Para isso, basta programar o campo **TCM[8-9]** no **registrador TCLR** para programar o tipo de transição que acione o modo de captura. Esse modo pode ser programado através do **registrador TCLR**, no bit **CAPT_MODE**.

O módulo define o **registrador IRQSTATUS**, no bit **TCAR_IT_FLAG** quando uma transição é detectada, e ao mesmo tempo o valor contador em **TCRR** é salvo no **registrador TCAR1** ou **registrador TCAR2**.

Existem 2 tipos de configurações no bit **CAPT_MODE**, 0 ou 1. Se o bit for 0 então, no primeiro evento de captura o valor do contador será armazenado em **TCAR1** e todos os outros serão ignorados até que a lógica de detecção seja reiniciada ou seja escrito 1 no bit **TCAR_EN_FLAG** do **registrador IRQENABLE_CLR**, apagando o valor guardado. Se o bit for 1, dois eventos serão capturados, o primeiro será salvo em **TCAR1** e o segundo em **TCAR2**. Todos os próximos eventos serão ignorados ao menos que a lógica seja reiniciada ou **IRQENABLE_CLR** seja escrito.

2.2.3 Funcionalidade do Modo de Comparação

Quando o bit **CE** do registrador **TCLR** é definido em 1, o modo de comparação é habilitado. Quando esse modo está habilitado, o valor do contador (**TCRR**) é comparado com o valor escrito em **TMAR**. Quando os valores de **TCRR** e **TMAR** combinam, uma interrupção pode ser gerada, mas para isso é preciso que o bit **MAT_EN_FLAG** no registrador **IRQENABLE_SET** seja definido.

A implementação correta é definir primeiro o valor de comparação em **TMAR** antes de definir **TCLR** (bit **CE**).

3 Principais Registradores

- **IRQENABLE_SET** (offset=2Ch)
- **IRQENABLE_CLR** (offset=30h)
- **TCLR** (Timer Control Register) Register (offset = 38h)
- **TCRR** (Timer Counter Register) Register (offset = 3Ch)
- **TLDR** (Timer Load Register) Register (offset = 40h)
- **TMAR** (Timer Match Register) Register (offset = 4Ch)
- **TCAR1** (Timer Capture Register 1) Register (offset = 50h)
- **TCAR2** (Timer Capture Register 2) Register (offset = 58h)

4 Como configurar

Todos os dispositivos funcionam a base de um *clock* central. Com isso, é necessário configurar esse *clock*.

Portanto, a primeira coisa a se fazer é configurar o *clock* do módulo DMTimer[0-7] que será utilizado. Próximo passo é configurar o módulo DMTimer escolhido, para isso é necessário "resetá-lo", depois basta dar uma *auto-reload* nele e habilitá-lo. Agora fazer a função que fique contando o tempo, utilizando os registradores de DMTimer. Agora ele conta o tempo certo, mas ainda está ocupando a CPU, fazendo *polling*. Com isso, o uso de interrupção se faz necessário.

Então, a interrupção será configura para DMTimer. Após configurar o *clock* do módulo DMTimer, é preciso habilitar o IRQ e inicializar a interrupção. Posteriormente, configurar o módulo para interrupção, habilitar o ISR, seta prioridade e habilita o sistema de interrupção. Com isso, a CPU não fica mais em *polling*.