Ministério da Educação
**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**
*Campus Curitiba*
Departamento Acadêmico de Eletrônica (Daeln-CT)
**ELF52 – Sistemas Microcontrolados**

Prof.: Guilherme de S. Peron
Prof.: Marcos E. P. Monteiro
Prof.: Paulo D. G. da Luz

# Atividade Prática 6 - Timers / Counters

## Contexto:

Temporizadores ou *Timers* são utilizados para permitir aos microcontroladores contagem por *hardware* seguindo um *clock* interno ou um sinal externo, neste caso atuando como contadores. Outra utilização é a geração de **PWM**.

A comunicação **UART** é amplamente utilizada para trocar dados entre 2 equipamentos, neste caso entre a placa **EK-TM4C1294XL** e o **PC**.

## Objetivos:

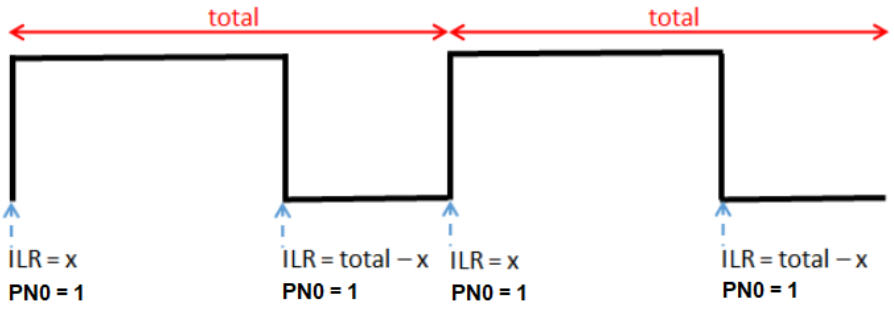
Controlar o brilho de um **LED** da placa **EK-TM4C1294XL** utilizando a interface **UART** e utilizando um *timer* no modo periódico. Por meio de um programa terminal, por exemplo, **Putty** (win32bits: http://elf52.daeln.com.br/Labs/Putty_32b.exe.zip ou win64bits: http://elf52.daeln.com.br/Labs/Putty_64b.exe.zip) ou **SSCom** (http://elf52.daeln.com.br/Labs/sscom3.2.zip), controlar a intensidade de brilho conforme os caracteres enviados.

Utilizando Linguagem **C** e/ou instruções **Assembly** para **Cortex-M4** e o kit de desenvolvimento **EK-TM4C1294XL**, desenvolver um programa que utilize **um *timer*** e **UART**. O *timer* deve controlar um **PWM** para controlar o brilho de um LED, e os dados recebidos pela **UART** devem alterar o **PWM**.

## Tarefas:

1. Ao inicializar a placa ou "resetar" o **LED1** (PN1) deve estar no seu brilho mínimo (**PWM** com *duty cycle* de **1%**). Neste caso, a placa deve enviar via **UART** para o programo de terminal no **PC** a mensagem "**PWM INI 1%\r\n**".
2. O programa de terminal no **PC** deve esperar uma tecla ser pressionada ou enviar o dado para definir o brilho do LED, conforme a seguir:
   a) '0' - Brilho do LED a 1% do máximo;
   b) '1' - Brilho do LED a 20% do máximo;
   c) '2' - Brilho do LED a 40% do máximo;
   d) '3' - Brilho do LED a 60% do máximo;
   e) '4' - Brilho do LED a 80% do máximo;
   f) '5' - Brilho do LED a 99% do máximo;
3. Na placa a cada recepção de dados pela **UART** enviados pelo programa terminal do **PC**, deve-se efetivar a troca de brilho do **LED1** (PN1), através da alteração do **PWM** no ***timer*** (*timer* 0), deve ser mostrada no terminal a mensagem "**LED a X%\r\n**".

4 Para controlar o brilho do LED fazer um PWM:
Na Placa um **PWM** pode ser gerado de várias formas. Uma das formas é por meio de um *timer* periódico em que em cada estouro do *timer* o valor de uma porta é alternado e o valor de contagem é alterado conforme o valor total do período. Por exemplo: Suponha um **PWM** com período 1ms (80000 *ticks*), para um *duty cycle* de 60% o período em alta deve permanecer por 600us (48000 *ticks*) e o período em baixa por 400us (32000 *ticks*). Desta forma, pode-se configurar para o ***timer0*** contar inicialmente para 600us e ligar a porta **PN0** (que neste caso está ligada ao **LED1**). Quando o ***timer0*** estourar, na rotina de tratamento de interrupção pode-se recarregar com o valor que o ***timer0*** deveria permanecer em baixa, neste caso por 400us, e desligar a porta **PN0** e repetir o procedimento indefinidamente.




5 O ideal é declarar duas variáveis globais que sejam acessíveis dentro da rotina de estouro do temporizador. Uma para controlar se o **LED1** está aceso ou apagado e outra para controlar o ***duty cycle***.

6 Cada vez que o ***timer0*** estourar e entrar na rotina de tratamento de interrupção, verificar se o **LED1** está apagado (neste caso acender e trocar o **GPTMAILR** para o valor atual do ***duty cycle:*** "[(80000**duty_cycle)*/100]") ou está aceso (neste caso apagar e trocar o **GPTMAILR** para o complemento do ***duty cycle:*** "{[80000*(100-duty_cycle]/100}"). *Obs: Trabalhando com números inteiros, deve-se fazer sempre a divisão por último, para que o resultado não seja zero!!!*

7 Fazer as rotinas de transmissão e recepção da **UART** conforme os fluxogramas mostrados na aula. No laço infinito da função main(), ficar verificando se algum caractere já foi recebido e neste caso tratar conforme o **item 2**.

**Dicas:**

Verificar o exemplo do *timer* em C e o exemplo da serial em C. Entregar a pasta do projeto do Keil com todos os arquivos zipado. Nomear o arquivo com o nome e o último sobrenome dos dois alunos da dupla.

Ex.: fulanodetal1_fulanodetal2_ap4.zip