



Prof.: Guilherme de Santi Peron

LAB 4 - CONTROLE BIDIRECIONAL DE VELOCIDADE DO MOTOR DC

Objetivo:

- Implementar um temporizador;
- Interfacear com motor DC;
- Interfacear com o conversor AD;
- Implementar o controle bidirecional de velocidade do motor DC utilizando Linguagem *Assembly* Cortex-M4 / Linguagem C e o kit de desenvolvimento EK-TM4C1294-XL. O motor deverá girar de acordo com um potenciômetro que está ligado a entrada de uma interface analógica.

Tarefas:

- Estudar como se interfacear com motores DC utilizando o circuito L293;
- Estudar como se implementar o conversor analógico-digital;
- Estudar as interfaces da PAT DAELN com o motor DC.
- Fazer o Diagrama de Estados e Transições OU fluxograma do código planejado conforme o roteiro;
- Implementar o código conforme o roteiro e o DET ou fluxograma utilizando instruções *Assembly* para Cortex-M4;
- **Mostrar para o professor e depois entregar a pasta do projeto Keil com todos os arquivos zipada, a imagem do DET ou fluxograma (pdf, jpg ou png) da ideia proposta também dentro da pasta (preferencialmente em algum site ou aplicativo, e.g. <http://draw.io>). Nomear o arquivo com o nome e o último sobrenome dos dois alunos da dupla. Ex.: **fulanodetal1_fulanodetal2_ap4.zip**. Apenas um membro da dupla precisa enviar.**



Roteiro:

- 1) Ao inicializar a placa ou “resetar” o motor deve estar parado (velocidade = 0) e o terminal ou lcd deve mostrar a mensagem “MOTOR PARADO”.
- 2) O terminal/lcd então deve pedir o sentido de rotação (podendo ser sentido horário ou anti-horário). O usuário deve digitar o sentido pelo terminal ou teclado matricial.
- 3) O motor deve girar sua velocidade conforme o valor de tensão de entrada do potenciômetro, que deve ser lido pelo conversor A/D. Este valor deve servir como porcentagem de giro do motor de 0 a 100% em relação a 3,3V.
- 4) As velocidades e o sentido de rotação deverão ser mostrados no terminal ou no lcd conforme a posição do potenciômetro;
- 5) Para controlar a velocidade do motor fazer um PWM (*verificar ao final deste arquivo como gerar um PWM*);
- 6) A qualquer momento, durante o giro do motor, o programa deve permitir que o usuário altere o sentido de rotação do motor, pelo terminal ou pelo teclado matricial. Neste caso, o motor deve parar gradativamente e acelerar gradativamente para o outro lado.

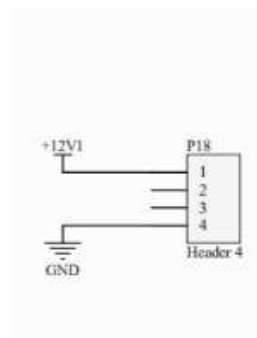
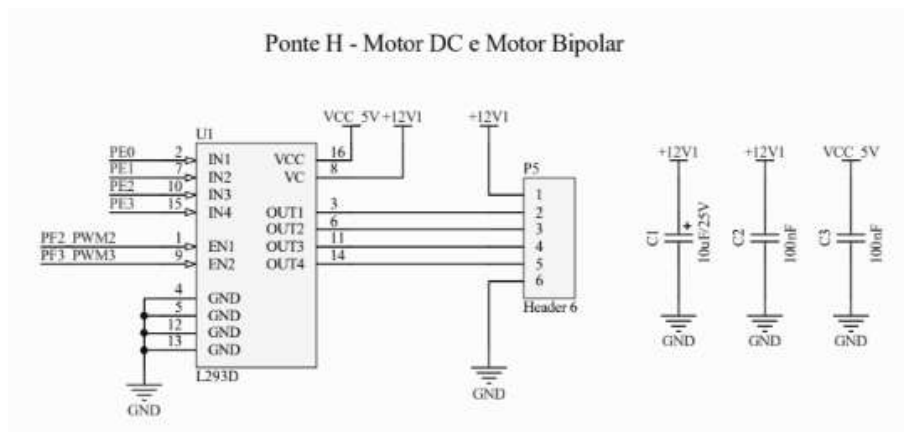
OBS: A seguir, é mostrado o driver L293 para controle bidirecional do motor DC.

- A alimentação do motor deverá ser externa, por meio de uma fonte de tensão conforme a especificação do motor. Esta fonte deverá ser ligada no P18. O Vcc da fonte deve ser ligada no +12V1 e o GND no GND. Verificar a tensão máxima de alimentação do motor DC;



- Para fazer o controle do sentido de rotação do motor, deixar habilitado o pino PF2 e colocar 1 no pino PE0 e 0 no pino PE1 para um sentido e 0 no pino PE1 e 1 no pino PE0 para outro sentido.
- O controle de velocidade pode ser feito fazendo o PWM em um dos pinos PE0 ou PE1 dependendo do sentido de rotação ou no pino PF2 (Porta de Enable do L293), mas a porta de *enable* é lenta e pode não responder corretamente.

L293 - Motor DC



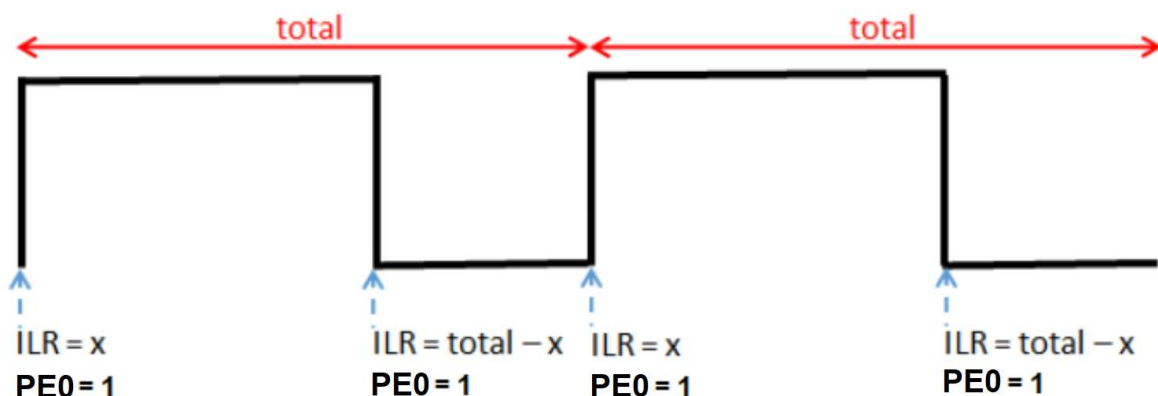
Gerando um PWM com um Timer

Na Tiva 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 alternada 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 *timer* contar inicialmente para 600us e ligar a porta PE0 (PE1). Quando o *timer* estourar, na rotina de tratamento de interrupção pode-se recarregar com o valor que o *timer* deveria permanecer em baixa, neste caso por 400us, e desligar a porta PE0 (PE1) e repetir o procedimento indefinidamente.



1. O ideal é declarar duas variáveis globais que sejam acessíveis dentro da rotina de estouro do temporizador. Uma para controlar se o pino está *on* ou *off* e outra para controlar o *duty cycle*.
2. Cada vez que o *timer* estourar e entrar na rotina de tratamento de interrupção, verificar se o LED está apagado (neste caso acender e trocar o GPTMAILR para o valor atual do duty cycle $80000 * \text{duty cycle} / 100$) ou está aceso (neste caso apagar e trocar o GPTMAILR para o complemento do duty cycle $80000 * (100 - \text{duty cycle}) / 100$).