Universidade Federal de Pernambuco
Departamento de Engenharia Mecânica

Laboratório de Sistemas Digitais — 1º Semestre 2019
Prof. João Paulo Cerquinho Cajueiro — 11 de abril de 2019

Aluno: _____

Experimento II – Motor DC

Objetivo – Construir um controlador de um motor DC.

Especificação

A partir da placa FRDM-KL25Z, de dois botões, de um potenciômetro e de um driver de motor, construir um controlador de um motor DC.

Dois botões teram a função de (S) iniciar e parar o motor e (R) definir a direção de rotação do mesmo.

A velocidade de rotação será definida através de um potenciômetro, indo de 0 rpm a 250 rpm.

O motor a ser controlado tem um encoder magnético, gerando 341 pulsos por revolução, que servirá para medir sua velocidade real de rotação. O controle efetivo de velocidade será feito através de um controlador PID implementado no microcontrolador.

O microcontrolador ficará continuamente enviando uma string de status pela porta de comunicação, com os seguintes campos separados por tab ('' no código C/C++): modo, sendo 'off' para desligado, 'ramp' para acelerando' e 'on' para em velocidade nominal; velocidade medida e set-point de velocidade.

primeira parte – Medição da velocidade

O encoder gera 341 bits por volta. Deve-se definir um contador numa interrupção para ficar acumulando a quantidade de pulsos gerados e uma interrupção de tempo (talvez usando o sysTick) para calcular periodicamente a velocidade do motor.

Esta informação servirá, junto com os botões S e R para acionar o motor de acordo com a máquina de estados da figura 1.

Nesta parte não será ainda feita a parte de aceleração ou desaceleração do motor, mas apenas será checado quando receber um pedido de inversão da rotação que o motor tenha parado antes de acionar no outro sentido.

O encoder diferencial opera segundo a figura 2, de onde para saber o sentido de rotação (e saber se incrementa ou não o contador), se coloca um dos sinais gerando uma interrupção e dentro dela se checa o valor do outro sinal.

Segunda parte – Ajuste de velocidade do motor

É necessário configurar um timer para gerar um sinal PWM ligado a uma saída. O microcontrolador tem módulos TPM que tem esta função (capítulo 31 do datasheet). Devemos configurar um módulo deste com 2 pinos de saída disponíveis na placa e configurá-los para gerarem um sinal PWM.

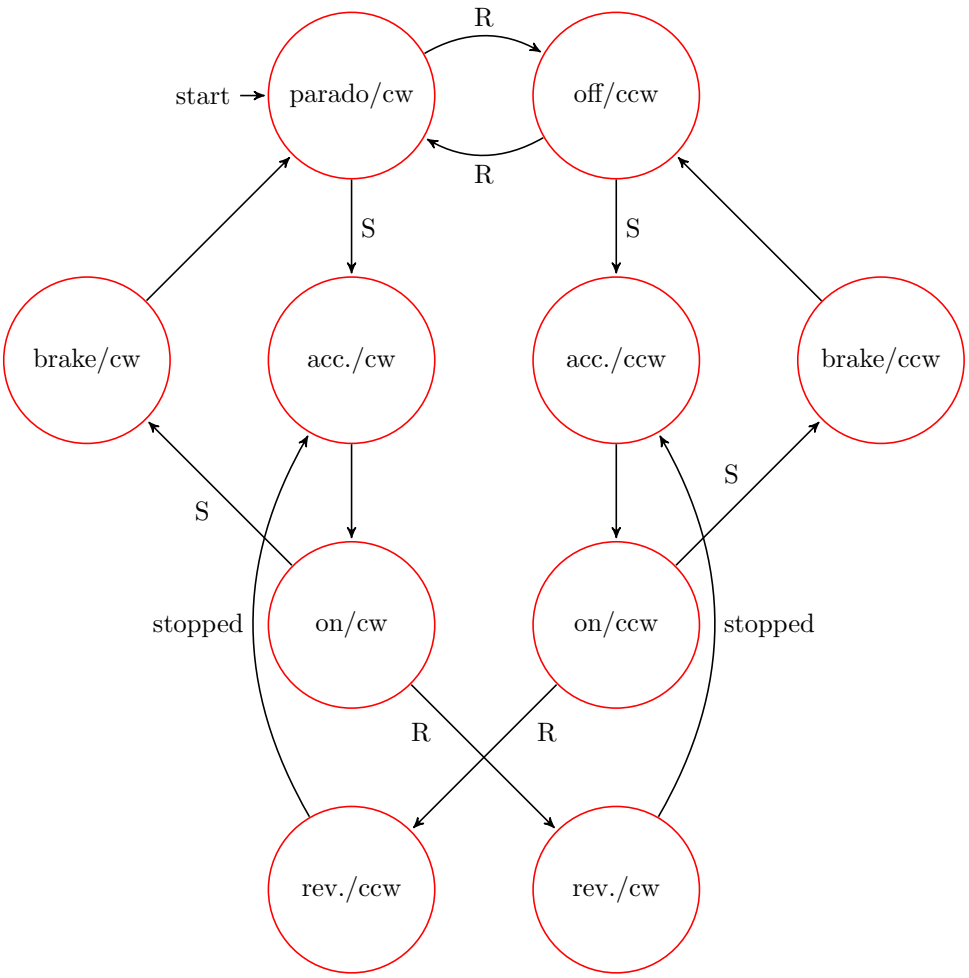


Figura 1: Máquina de estado do acionamento do motor.

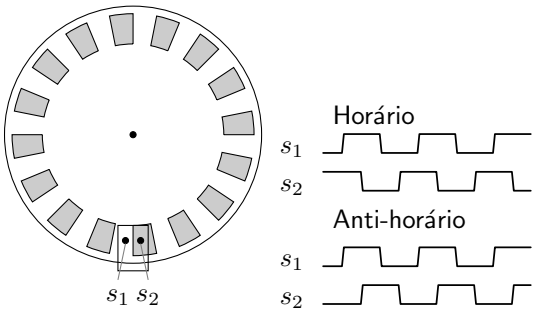
Como a maioria dos periféricos, precisamos inicialmente enviar o sinal de clock ao TPMx para poder usá-lo. Isto é configurado no registrador SIM_SCGC6. Além disso, a configuração de qual o clock é enviado aos TPM é feita no registrador SIM_SOPT2. Podemos usar a configuração 01 neste campo.

No registrador TPMx_SC, devemos configurar: modo PWM alinhado no centro (CPWMS=1), LPTM clock (CMOD=01), com pré-escala de 8 (PS=011).

No registrador TPMx_MOD, coloca-se o valor máximo que o contador irá contar. Para trabalhar com um PWM de 12 bits, podemos escrever nele um valor de 0x0FFF.

Para um canal gerar um PWM, deve-se configurar no TPMx_CnSC: MSB em 1, MSA em 0, ELSB em 1 e ELSA em 0, onde n é o canal.

O valor do PWM de cada canal será então definido pelo registrador TMPx_CnV.



(a) diagrama (b) Sinais

Figura 2: Visualização esquemática de um encoder mecânico diferencial.

Terceira parte – Medição do set-point e controle

Deve-se configurar o periférico de conversão AD para realizar uma conversão da tensão do potênciometro e utilizá-la como setpoint do PID. Tal periférico é descrito no capítulo 28 do datasheet.

Também é necessário habilitar o clock deste periférico. Isto é feito no registrador SIM_SCGC6.

O conversor analógico tem diversas opções, entre as quais:

- Conversão diferencial (entre duas entradas) ou de uma entrada ao terra.
- Resolução de 8, 10, 12 ou 16 bits.
- Início de conversão por software, de modo contínuo ou iniciado por algum outro periférico (pino, temporizador, comparador, etc.).
- Realização de média de várias aquisições.

Para nossa aplicação, ligaremos um potênciometro entre terra e 3,3 V para gerar o sinal a ser convertido, logo basta ser no modo *single ended*. Como já configuramos o PWM para 12 bits, faz sentido configurarmos o ADC também para 12 bits. O sinal de entrada não é de alta velocidade então não precisamos nos preocupar num clock muito alto e inclusive podemos usar o recurso de fazer automaticamente a média de várias aquisições para gerar o sinal desejado. Podemos então usar o modo de aquisição contínua e quando cada aquisição terminar, capturar o valor medido.

A maior parte das configurações é feita no registrador ADC0_CFG1. Neste podemos colocar ADICLK=01 para escolher o *busclock*/2 e o ADIV=11 para dividir este clock por 8 e o MODE=01 para escolher resolução de 12 bits.

Para fazer a conversão contínua e a média de várias medidas, é necessário mexer também no ADC0_SC3, que tem o bit ADCO para habilitar a conversão contínua e o bit AVGE para habilitar a média de múltiplas conversões. A quantidade de conversões é definida no campo AVGS deste mesmo registrador (AVGS=11 para 32 amostras).

Para converter de um único canal, precisamos configurar o registrador ADC0_SC1A. Nele configuramos a conversão *single-ended* (DIFF 0), se a conversão gera ou não uma interrupção (AIEN), e o canal usado (ADCH = 8 para o pino PTB6).

O ato de escolher um canal inicia o processo de conversão, mesmo no caso de multiplas conversões, logo este deve ser o último registrador a ser configurado.

No mesmo registrador ADC0_SC1A tem um bit COCO - COnversion COmplete, que checaremos para saber quando o resultado está pronto. Quando COCO for 1, podemos ler o valor no registrador ADC0_RA.

Quarta parte – Comunicação

Deve-se trabalhar com o periférico de UART0 (capítulo 39) para realizar a transmissão das strings de status.