



Aquisição de tempo entre pulsos consecutivos utilizando microcontrolador

Felipe Pereira Rodrigues¹, Geofly de Lima Adonias¹, Gustavo Gonçalves de Sousa Forte¹, Winnie de Lima Torres¹

¹Bolsista do Programa de Educação Tutorial do curso de Bacharelado em Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba.

Resumo: O presente artigo trata de um método pesquisado para descobrir o tempo entre dois pulsos sucessivos. Essa informação é útil para diversas aplicações, sendo aqui utilizada para calcular, através de um programa implementado e gravado em um microcontrolador, a velocidade de uma cadeira de rodas. A velocidade obtida deve ser comparada com um valor limite e, caso ele seja ultrapassado, o usuário deve ser avisado. Foram utilizados os *softwares* Proteus ISIS e ARES, CCS e PICPgm para simulação, projeto de placa de circuito impresso, criação e compilação do código do programa e gravação no PIC18F4520, respectivamente. O circuito montado em matriz de contatos funcionou como desejado. O custo do projeto não foi considerado alto e poderia ter sido menor, dependendo dos componentes necessários. Os componentes são utilizados mundialmente, existindo assim muito suporte para dúvidas e podendo ser encontrados facilmente.

Palavras-chave: microcontrolador, pulsos, temporização, velocidade

1. INTRODUÇÃO

A obtenção do tempo entre pulsos sucessivos é interessante no que se refere às aplicações. Para esclarecer, trata-se da determinação aproximada de quanto tempo se passou desde que um pulso ocorreu até o instante em que outro pulso acontece. Pode-se, por exemplo, conseguir o período e a frequência de uma onda quadrada ou analisar adequadamente a resposta de um sensor. No caso deste artigo, o problema consistiu em calcular a velocidade de uma cadeira de rodas e verificá-la de acordo com um limite estabelecido, sendo o usuário avisado de alguma forma se este limite for ultrapassado, procurando evitar velocidades desnecessárias que possam resultar em algum acidente.

Com o uso de um microcontrolador, as possibilidades aumentam enquanto é diminuída a complexidade do projeto. A família PIC de microcontroladores da Microchip® mostrou-se adequada e de fácil acesso para a implementação do projeto, visto que possui pinos de entrada/saída, temporizadores (*timers*), módulos de captura e comparação, assim como possibilita comunicação com um *display* de cristal líquido. Como essa família é utilizada em diversas aplicações mundialmente, não é difícil encontrar suporte para dúvidas que possam surgir.

Logo, o projeto procura identificar o tempo entre pulsos consecutivos vindos de um sensor, para que possa ser calculada a velocidade e depois compará-la com um limite, sinalizando para o usuário se está abaixo ou acima do valor considerado seguro. É interessante ressaltar que o princípio de aquisição dessas informações pode ser útil para exercer outras funções como, por exemplo, a de descobrir a velocidade de um motor, sendo conhecido o tempo que ele leva para dar uma volta completa.

2. MATERIAL E MÉTODOS

Foi utilizado o microcontrolador PIC18F4520, da Microchip®, que possui 40 pinos, memória de programa *flash* (podendo, portanto, ser regravado), portas de entrada/saída, módulos de conversão analógico-digital, vários tipos de interrupções, temporizadores e, entre muitas outras coisas, módulos necessários para a realização do projeto, tal como o de captura/comparação/PWM (CCP). O *software* foi escrito em linguagem C utilizando o compilador CCS, que gera um arquivo em hexadecimal para posterior gravação no microcontrolador, através de um gravador adequado que usa a porta serial do computador e um programa que gerencie o envio de dados para o PIC18F4520, sendo para este fim utilizado o programa PICPgm.

Foi utilizado, para exibição dos valores calculados, um *Liquid Crystal Display* (LCD) de referência JHD 162A, o qual possui 2 linhas de 16 colunas de caracteres. Ele é controlado pelo envio

de sinais (que podem ser de dados ou de comandos) nos seus pinos. Dispõe de microprocessadores internos que interpretam e executam a função desejada. Esse componente tem um custo mais elevado, mas foi utilizado porque estava disponível e seria bom para visualização de informações. O *display* deve ser inicializado adequadamente por *hardware* ou por *software*, sendo a segunda a opção que foi utilizada. O envio de dados e comandos é gerenciado por uma biblioteca que possui funções como a de enviar *byte*, enviar *nibble*, inicializar o *display*, ligar/desligar o cursor, ir para uma determinada linha/coluna, entre outras. Essa biblioteca é também escrita em linguagem C e é incluída no programa a ser gravado no microcontrolador, sendo ela adaptada e personalizada para otimizar a disposição dos componentes numa futura placa de circuito impresso.

Um *reed switch* (pequena chave sensível a um campo magnético) foi utilizado para detecção de quando um ímã passa por um local. Quando o ímã passa por perto do sensor, ocorre o fechamento da chave e um pulso é enviado ao pino do PIC18F4520. Quando o ímã se afasta do *reed switch*, é enviado nível lógico zero, pois há um resistor *pull-down* que define este estado. Certo tempo depois, o ímã passa pelo sensor novamente e é nessa hora que o microcontrolador calcula quanto tempo se passou entre o pulso anterior e o atual, fazendo isso toda vez que um novo pulso for detectado. Uma ilustração do esquema pode ser vista na Figura 1.

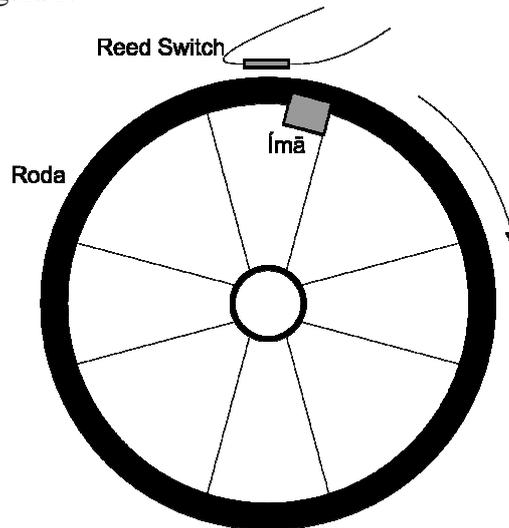


Figura 1 - Ilustração do esquema do sensor para geração dos pulsos

Uma matriz de contatos (*protoboard*) foi utilizada para teste do circuito, onde foi possível agregar todas as partes do projeto e verificar se estava funcionando como desejado e se algum melhoramento precisava ser feito.

Um diodo emissor de luz (LED) serviu como sinalizador para indicar se o limite havia sido ultrapassado ou não. Caso aceso, o usuário deveria diminuir a velocidade. Caso contrário, significaria que a velocidade não era excessiva. Foram necessários também um cristal de quartzo (10 MHz), um botão, alguns resistores, fios, multímetro e osciloscópio digitais, fonte de alimentação (5 Volts contínuos), alicates, entre outras coisas.

Para que o microcontrolador possa identificar o tempo entre os pulsos, é necessário ativar o modo de captura do módulo CCP, que tem a função de capturar o valor de 16 *bits* de um temporizador (Timer1 ou Timer3, que são dois dos quatro temporizadores disponíveis) no momento em que um evento ocorre. Esse evento pode ser escolhido pelo programador, sendo as opções (todas relativas a um sinal de entrada em um pino específico): a cada borda de descida, a cada borda de subida, a cada 4 bordas de subida e a cada 16 bordas de subida. Para o projeto, foi selecionado o evento como sendo a borda de subida de um sinal. A cada vez que o evento ocorre é gerada uma interrupção que irá capturar o valor do temporizador naquele instante, armazenando-o em um registrador. Dentre os dois módulos CCP, foi utilizado o CCP1 e o temporizador escolhido foi o Timer1.



Os *timers* do PIC18F4520 funcionam como contadores que incrementam a cada ciclo de *clock* (dependente da fonte de *clock*, neste caso o cristal de quartzo de 10 MHz). Possuem diversas utilidades, sendo necessário neste projeto apenas para marcação de tempo. Quando o microcontrolador faz a captura do valor presente no Timer1, pode-se calcular o tempo que se passou, pois são conhecidos o valor inicial e o tempo entre incrementos. Um problema que teve de ser resolvido é o do “estouro” (*overflow*) do contador Timer1, pois ele, como tem 16 *bits*, só armazena até o valor 65535. Após isso, ele começará do valor zero. Como o tempo de incremento é suficientemente pequeno, ele poderá “estourar” várias vezes e esses estouros devem ser levados em consideração no cálculo do tempo. Foi resolvido criando uma variável que conta quantas vezes ocorreu esse *overflow*.

A configuração do microcontrolador foi feita visando à otimização do código, sendo procurado estabelecer valores diretamente nos registradores do dispositivo, valores estes que determinam como ele irá funcionar, quais funções estarão ativadas, a direção dos pinos de entrada/saída (configurando-os como *input* ou *output*), se interrupções estarão ativas ou inativas em determinado instante, entre muitas outras coisas. Tomou-se cuidado também com as operações lógicas e procurou-se, quando fosse melhor, utilizar operações *bit a bit* para diminuição de tempo de execução, pois estas instruções são mais simples de serem executadas. Através de um botão, a opção de *reset* do programa é disponibilizada, devendo ser usada caso aconteça algum tipo de erro ou mau funcionamento do microcontrolador.

Para o cálculo da velocidade, basta dividir a distância percorrida pelo tempo gasto. O raio da roda foi medido, sendo de aproximadamente 24,5 cm. Logo, a cada volta completa, é percorrida a distância de 1,54 m, equivalente à circunferência da roda.

O projeto foi simulado utilizando o *software* Proteus ISIS antes de ser montado. Este programa foi importante porque permitiu economia de tempo e maior segurança, pois evitou erros que na prática fariam com que o projeto não funcionasse adequadamente.

3. RESULTADOS E DISCUSSÃO

Após a etapa de simulações, o circuito foi montado na matriz de contatos (*protoboard*) e o resultado se deu como esperado, porém modificações poderiam ser feitas para otimizar o trabalho, modificações estas que não eram tão visíveis na etapa anterior. O circuito foi testado e verificado várias vezes, observando as faltas e erros inerentes à programação que só puderam ser vistos na prática, regravando o microcontrolador com o programa corrigido.

Na certeza de que tudo estava funcionando satisfatoriamente, veio a etapa de projeto para uma placa de circuito impresso. Para tal, fez-se uso do *software* Proteus ARES. Neste passo ocorreram mudanças significativas, tendo em vista que o objetivo era minimizar o espaço físico que o circuito ocuparia e otimizar o posicionamento dos componentes observando como ficariam dispostas as trilhas do circuito. Como citado anteriormente, o PIC18F4520 pode se comunicar com o LCD, sendo esta comunicação através de 4 ou 8 pinos, sendo que foi escolhida a primeira opção. Personalizações no código da biblioteca do *display* foram imprescindíveis para simplificação dessas ligações de comunicação entre os dois componentes. Com base nas dimensões do LCD, foi decidido que o mesmo ficaria por cima de parte dos outros componentes, pois estes não precisavam ser visíveis e dessa forma houve economia de espaço.

Após essas atualizações, o circuito foi testado novamente e funcionou como era desejado. Com a utilização de um ímã, verificou-se que os pulsos estavam sendo gerados normalmente, o microcontrolador os detectava e executava os cálculos, exibindo o valor da velocidade corretamente. A comparação da velocidade com o limite também funcionou, podendo ser vista através do LED.

6. CONCLUSÕES

- A medição de tempo entre dois eventos consecutivos pode ser feita de forma simples;
- Essa medição é útil em diversos projetos;
- O uso de um microcontrolador é vantajoso para essa aplicação;
- Os componentes utilizados podem ser encontrados facilmente no comércio;
- A velocidade desejada pode ser calculada sem grandes dificuldades;



O uso de programas de simulação facilita o desenvolvimento do projeto;
O custo de um projeto que utilize os mesmos componentes não é alto;
O trabalho está na fase de confecção de uma placa de circuito impresso, sendo os próximos passos a soldagem dos componentes, verificação e integração do circuito à cadeia de rodas.

AGRADECIMENTOS

Ao Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB). Ao Programa de Educação Tutorial (SESu/MEC). Às pessoas que contribuíram para o projeto.

REFERÊNCIAS

MICROCHIP TECHNOLOGY INC. **PIC18F2420/2520/4420/4520 Data Sheet**. Versão DS39631A. Estados Unidos da América: [s.n.], 2004. 391 p.

OPTREX CORPORATION. **Dot matrix character LCD module user's manual**. [S.l.: s.n.]. [200-?]. 58 p.