

## REPRESENTAÇÃO PRÁTICA DE UM SENSOR DE RÉ DE AUTOMÓVEIS UTILIZANDO PIC16F877A, COM DESTAQUE NAS TÉCNICAS DE SOLUÇÃO DE PROBLEMAS PROGRAMÁVEIS

Kaio Vítor Gonçalves de Freitas<sup>1</sup>; Eric Guimarães Barbosa<sup>2</sup>; Rafael Mateus Carvalho de Paiva<sup>1</sup>; José Lucas Damasceno Holanda<sup>1</sup>; Tamires dos Santos Pereira<sup>3</sup>

1 Universidade Federal de Campina Grande, Graduando em Engenharia Elétrica, kaio.freitas@ee.ufcg.edu.br; rafael.carvalho@ee.ufcg.edu.br; lucasholanda95@gmail.com

2 Instituto Federal da Paraíba, Graduando em Engenharia da Computação, ericguimaraes@msn.com

3 Escola Técnica Redentorista, Eixo Tecnológico de Controle e Processos Industriais, tsantosp16@gmail.com

### Introdução

O conhecimento, o seu engenho e seu espírito é a melhor arma para enfrentar o mundo, preparar um melhor destino e a mais agradável vida (MAX STIRNER, 1844), sendo a sua essência aplicá-lo, depois que o possui. Conceitos ligados à física, eletrônica e sistemas embarcados são muito importantes e possuem aplicações indispensáveis para nosso dia a dia que contribuem para o avanço da nossa sociedade, sendo um entendimento de grande importância para estudantes da área de tecnologia, uma vez que esta área, impreterivelmente, necessita de inovações. Esses conceitos foram utilizados para projetar nosso objeto em estudo e, a partir disso, houve ênfase no contorno de problemas encontrados durante a construção do projeto, para estimular o raciocínio dos leitores.

O uso de um sensor de distância é muito útil em diversas aplicações, tais como: sensores de ré de automóveis (objeto de inspiração do projeto), corretores de rota de movimento de máquinas, medição de nível de líquidos, detector de objetos, funcionamento de uma lombada eletrônica. Dentre os existentes, poderia ser utilizado o sensor de medição por ultrassons, infravermelhos, fotoelétricos, entre outros. Foi escolhido, nesse trabalho, o sensor ultrassônico HC-SR04, pela sua alta precisão (em microssegundos) e longa distância máxima (até 4 metros), além do uso apenas de pinos digitais, de fácil implementação, tornando-se acessível para estudantes que estão começando a se empenhar nessa área. Esse sensor possui 4 pinos: GND, VCC, *Echo* e *Trigger*. Seu funcionamento consiste em enviar ondas ultrassônicas de frequência 40 kHz (fora da faixa de frequência que o ser humano é capaz de escutar), após a onda encontrar um obstáculo, volta para o sensor e a partir do tempo de envio e tempo de retorno, é calculada a distância entre o sensor e o objeto.

Para inicializar e fazer o uso dos sinais digitais do sensor, foi-se utilizado o PIC16F877A, com a ide PCW e o compilador PCM, que usa uma linguagem derivada de C. Segundo Fábio Pereira (2006, p.18) “A utilização de uma linguagem de alto nível como C permite que o programador preocupe-se mais com a programação da aplicação em si, já que o compilador assume para si tarefas como controle e localização das variáveis, operações matemáticas e lógicas, verificação de bancos de memórias”. A escolha do modelo do microcontrolador se deu por conta do grande número de pinos que ele possui a disposição, uma vez que o *display* LCD 16x2 usado para mostrar os resultados utiliza 8 pinos dele, precisando de ainda outros para ligar os sensores, LEDs. A única parte negativa do uso desse PIC foi o fato dele não vir com um oscilador interno, diferente de outros modelos como o

PIC16F628A, porém, isso pode ser facilmente resolvido, fazendo-se uso de um circuito elétrico que funciona como

*clock* desse microcontrolador, formado por um cristal oscilador de 16MHz e dois capacitores cerâmicos de 22pF para manter a oscilação estável. Além disso, é importante ressaltar que o pino 1 do microcontrolador, o *Master Clear*, precisa ser colocado em nível alto, para tirá-lo da situação de *RESET*.

## Metodologia

Para esse trabalho, foi utilizado dois sensores ultrassônicos, um display LCD 16x2, PIC16F877A, um circuito elétrico para definir o *clock* do microcontrolador, dois sensores HC-SR04, resistores e LEDs de diferentes cores. Inicialmente, é enviado um sinal com duração de 10 microssegundos ao pino *trigger*, para indicar o início de uma nova medição, em seguida o módulo envia automaticamente 8 pulsos de 40 kHz e o pino *echo* é setado para nível lógico alto, aguardando o retorno do sinal pelo receptor, caso ocorra retorno, *echo* vai para nível lógico baixo e a distância é calculada a partir da equação: distância igual ao tempo do pino *echo* em nível alto multiplicado por 340m/s (velocidade do som), sendo esse resultado dividido por 2 pelo fato do tempo contabilizar a ida e a volta. Para o cálculo da distância feito no projeto, foram utilizadas recomendações do fabricante encontradas no *datasheet* do sensor, como um intervalo de 60 ms para realizar uma nova medição e a distância encontrada em cm, fazendo distância (em cm) igual ao tempo de *echo* em nível lógico alto (em microssegundos) \* 0.017.

Na aba *interrupts* presente no utilitário *wizard da IDE*, foi ativada a interrupção por qualquer mudança na porta B, sendo necessário aterrar os pinos B5, B6 e B7 para que a mudança ocorra apenas no pino B4, que foi ligado ao pino *echo* do primeiro sensor ultrassônico, simulando, assim, uma interrupção externa, uma vez que só possui um pino para isso no microcontrolador, que já seria usado pelo outro sensor ultrassônico. Ao entrar na interrupção, um *flag* teve seu nível lógico mudado para 1 enquanto o *echo* recebia a onda ultrassônica do *Trigger*, voltando para zero apenas na *main*. Foi usado o timer 1 para a contagem do tempo em microssegundos para se fazer o cálculo da distância, iniciando a contagem assim que o *echo* era setado e desabilitando-o quando o *echo* ia para zero, para se ter um tempo preciso, em microssegundos, de ida e volta da onda ultrassônica. O *timer 1* possui 16 bits, foi configurado para uma resolução de 2µs e período de *overflow* de 131ms ( $65535 * 2\mu s = 131ms$ ). Logo, a distância máxima que poderia ser encontrada usando o *timer 1* na interrupção do pino *echo* seria de  $0.017 * 131ms = 2227cm$ . Uma vez que o alcance máximo do sensor ultrassônico é de 4m, não há nenhuma preocupação em usá-lo.

Para o segundo sensor, utilizou-se uma interrupção externa, ligando o pino *Echo* no pino B0 do PIC utilizado. Apesar do processo ser semelhante ao do outro sensor, o uso do timer 2 para a contagem do tempo fez com que fosse necessária uma interrupção periódica para a contagem de quantos *overflows* tinham acontecido no timer enquanto o pino *Echo* estava em nível lógico alto.

O *timer 2* possui 8 bits, foi configurado para resolução de 4µs e contagem até 249, tendo, assim, um período de *overflow* de  $249 * 4\mu s = 996\mu s$ . Portanto, a distância máxima que poderia ser calculada por ele seria de  $0.017 * 996\mu s = 16,9cm$ , sendo muito baixa para a aplicação de sensor de ré. A interrupção periódica dele atribuía valores à uma variável do tipo contador, que era multiplicado a 249 e somada ao valor atual do timer na interrupção externa, sendo esse valor atribuído à uma variável auxiliar que ainda era multiplicada por 4 (por conta da resolução), para se ter o tempo em microssegundos. Após uma medida da distância, a variável era zerada na própria interrupção externa.

Foram usadas condições para ligar o LED vermelho ou verde, que sinalizam se há um objeto mais próximo da esquerda ou da direita, a partir do valor da distância de cada sensor encontrado. Os valores de

distância encontrados na interrupção de cada sensor foi colocado nas linhas do *display* LCD, logo, o valor é instantaneamente atualizado sempre que o programa entrasse nas interrupções do pino *echo* a partir de um comando no código definido numa biblioteca cujo programador já havia colocado na pasta do projeto.

### Resultados e discussão

A distância pôde ser calculada com precisão a partir da medição do tempo em que o pino *echo* permaneceu em nível lógico alto, os resultados foram testados utilizando-se uma régua. Cada um dos dois sensores utilizados teve seus respectivos tempos calculados utilizando diferentes *timers*: O *timer1* e o *timer2*, e como pôde ser visto nos testes, não houve qualquer influência das contagens de tempo de um sensor em outro. Foi verificada uma interferência quando ambos os sensores foram dispostos no mesmo sentido, provocando variações equivocadas nas medições. Para solucionar este contratempo, a ligação dos sensores foi feita em sentidos opostos, de modo a simular a tangente entre a traseira do carro e a lateral esquerda e direita dele, o que acabou por trazer um maior sentido físico para o projeto, possibilitando medição de distância com relação ao lado direito e ao lado esquerdo.

### Conclusões

Foi verificado que é possível realizar medições simultâneas de distância utilizando sensores ultrassônicos e diferentes recursos do PIC, tais como: Interrupções, timers, GPIO (*General Purpose In/Out*). As dificuldades e singularidades encontradas puderam ser contornadas utilizando técnicas e lógicas de programação, e otimizações na construção do projeto podem ser feitas utilizando entendimentos externos em diversas áreas, uma vez que não podemos criar uma cultura unilateral do entendimento.

**Palavras-Chave:** Sensor; Microcontrolador; *Timer*; Tecnologia.

### Referências

- DEITEL, H. M. **Como programar em C**. Editora Pearson Education – Br. 6ª Edição, 2011.  
FRENZEL JR. L. E. **Eletrônica Moderna**. Editora Bookman. 1º Edição, 2015.  
MANO, M. M. **Digital Design**. Editora Pearson/Prentice-Hall. 5ª edição, 2013.  
Microchip Technology Inc. **PIC16F87XA Data Sheet**. DS39582B, 2003.  
PEREIRA, FABIO. **Microcontroladores PIC: programação em C**. Editora Érica. 7ª edição, 2006.