

UNIVERSIDADE FEDERAL DE ITAJUBÁ



ECOP04 - PROGRAMAÇÃO EMBARCADA

PROJETO FINAL

Mesa Giratória para Furação de Peças

Rebeca Ribeiro Pinto

01 de Agosto, 2021

1 Resumo

Neste relatório estão descritos os procedimentos e códigos realizados em C através do MPLAB para elaborar a simulação de uma mesa giratória que realiza a furação da peça, e em seguida, testa se o furo foi feito com êxito.

2 Introdução

A tecnologia se faz cada vez mais presente na vida cotidiana, seja para lazer como jogos, comunicação como celulares, ou até mesmo na automação em indústrias por exemplo. Dessa forma, um das ferramentas utilizadas em uma ampla gama de produtos tecnológicos, é o sistema embarcado composto por microcontroladores, sensores, interface com usuário, atuadores, periféricos entre outros. Para controlar esse sistema é necessário que um programa seja executado na placa, em que a linguagem utilizada varia de acordo com cada sistema e condições.

Além disso, os sistemas embarcados possui determinadas vantagens como o tamanho e preço reduzido, facilidade de programação, já que não se utiliza linguagens de baixo nível como a linguagem de máquina, além de uma boa eficiência uma vez que realiza um número fixo de tarefas programadas.

Ademais, na matéria de Programação Embarcada ECOP04 foi apresentado diversas funcionalidades da placa PIC18F4520 utilizando linguagem C. Sendo assim como parte da nota, foi proposto a realização de um projeto utilizando pelo menos cinco desses conhecimentos de forma a desenvolver algum processo simulados pela placa.

Dessa forma, teve-se a ideia de simular uma mesa giratória que realiza as furação das peças, movimenta-se, e faz-se o teste do furo dessa peça inserida. Para isso foi utilizado ao todo seis funcionalidades da placa, o MPLABX para escrita e compilação do código além do PicSimLab para simulação.

3 Desenvolvimento

3.1 Inspiração:

Como dito anteriormente, o projeto consiste na simulação de uma mesa que realiza a furação e teste das peças, que foi inspirado de em um modelo existente no Laboratório de Automação da UNIFEI. A figura a seguir exibe o processo original do laboratório, as diferenças consiste na diminuição do número de sensores, uma vez que na simulação será utilizada a ideia do sensor apenas no teste, a redução de 4 slots para apenas 1 e o uso do display LCD e de 7 segmentos que não existem na versão original.



Figura 1: Mesa Original

3.2 Operações da Mesa:

Após adotada as modificações apresentadas no tópico anterior, definiu-se quais sequências de ações seriam feitas. Dessa forma, os itens a seguir apresenta a ordem de operações que são realizadas pelo sistema:

1. Acionamento do Sistema;
2. Setup do número de peças defeituosas aceitáveis;
3. Inserção da Peça;
4. Rotação da mesa para que a peça chegue na posição da furadeira
5. Furação da Peça;
6. Rotação da mesa para que a peça chegue na posição do pistão de teste;

7. Teste do Furo;
8. Rotação da Mesa para retirada da peça.

Após o último movimento, a máquina continua nesse processo cíclico até que o número de peças com defeitos atinja o valor configurado, podendo então ser desligada.

3.3 Funcionalidades da Placa:

Para adaptar esse sistema para a placa PIC18F4520 utilizou-se determinadas funcionalidades aprendidas durante a disciplina de Programação Embarcada. A seguir apresenta-se as funcionalidades utilizadas e o que elas representam no sistema:

- **LEDs:** Os LEDs são utilizados para representar o sistema na condição ligado ou desligado. Sendo assim, um LED do PORTB (B6) é utilizado para sinalizar que o sistema está desativado. O LED B7 representa o sistema ligado, permanecendo dessa forma durante todo o processo.
- **Display LCD:** O display foi utilizado para representar cada ação que está ocorrendo, dessa forma sinaliza todas as operações, como inserção e remoção de peças, movimentação da mesa, o furo e o teste. Essa sinalização é feita através da escrita de qual atividade está sendo realizada.
- **Display 7-Segmentos:** Utilizado para mostrar que o valor máximo de defeitos foi atingido, e então aparecendo este número do display no momento em que isso ocorrer.
- **Buzzer:** Assim como o display de 7 segmentos, o buzzer funciona como alarme para sinalizar esse número máximo de falhas, sendo parado apenas quando o sistema for desligado.
- **Teclado:** Age botões para iniciar, desligar e entrar com a configuração do número de defeitos. Ademais, atua como um sensor para representar que o furo não foi realizado com sucesso, como se o sensor do pistão de teste encontra-se a peça sem furo, consequentemente ativando o sensor.
- **Ventoinha:** Utilizada para representação do giro da mesa, sendo ativada por um tempo apenas para simbolizar que a mesa está sendo rotacionada.

3.4 Criação do Código

3.4.1 GitHub

Depois de definido a ideia do projeto e o que seria usado da placa, iniciou-se a elaboração desse. Primeiramente foi criado um diretório no GitHub a fim de obter um código compartilhado. Em seguida, criou-se um branch em que seria realizado o upload do código de acordo com as mudanças realizadas, porém como havia apenas um desenvolvedor, o commit foi realizado apenas no final do processo, para facilitar o processo.

3.4.2 Bibliotecas

Após o projeto está ligado com o diretório do GitHub começou-se a escrita do código, sendo iniciado pela definição das bibliotecas que seriam utilizadas. A lista a seguir apresenta as bibliotecas utilizadas:

- *pic18f4520.h* - Nativa das configurações do MPLABX
- *config.h* - Fornecida pelo professor
- *bits.h* - Fornecida pelo professor
- *io.h* - Fornecida pelo professor
- *keyboard.h* - Fornecida pelo professor
- *lcd.h* - Fornecida pelo professor
- *led.h* - Desenvolvida durante laboratório
- *delay.h* - Desenvolvida durante laboratório
- *stdio.h* - Desenvolvida durante laboratório
- *functions.h* - Desenvolvida para o projeto

Cada uma das bibliotecas apresentadas pode ser acessada pelo GitHub do projeto a partir do link: https://github.com/RebecaRibeiro98/Projeto_Final_ECOP04.git

Em seguida, trabalhou-se na criação da biblioteca *functions.h* uma vez que essa é específica do projeto, utilizada principalmente para sinalização dos acontecimentos do processo. Esse arquivo é composto pelas funções a seguir:

- *turn_on*: Essa função é responsável por ligar o sistema, dessa forma, configura os leds do PORTB de forma que o led que representa o sistema seja ligado.
- *turn_off*: Semelhante a função anterior, tal comando realiza o desligamento do led que sinaliza a máquina ligada e acende o que representa o sistema desligado. Além disso, limpa o display LCD.
- *setup*: Responsável pela exibição da mensagem de configuração do número de defeitos que são aceitáveis. Essa função imprime no lcd a seguinte mensagem:

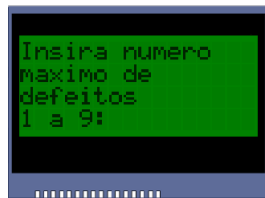


Figura 2: SetUp

- *input*: Função responsável por representar a inserção de peças na mesa, sinalizando essa ação através do display LCD. A seguir apresenta-se a mensagem exibida:

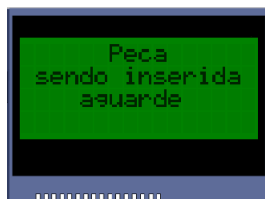


Figura 3: Input

- *move*: Essa função é responsável pela movimentação da mesa, além de sinalizar o giro no LCD, também aciona o cooler a fim de representar a rotação da plataforma em que as peças estão fazendo as deslocar de uma posição para a outra. A mensagem mostrada no display é apresentada a na figura abaixo:



Figura 4: Rotação da Mesa

- *drill*: Representa o processo em que a furadeira é acionada e realiza-se então o furo na peça. apresentando no LCD essa informação como apresentado a baixo.

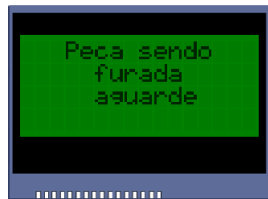


Figura 5: Furação da Peça

- *test*: Assim como as demais funções dessa biblioteca, o objetivo desta é sinalizar que está sendo realizado um furo na peça através do display LCD através de uma mensagem, sendo o teste é realizado na main.

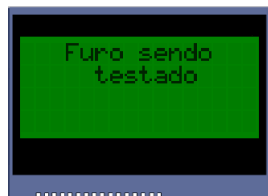


Figura 6: Teste do Furo

- *error*: Função chamada caso o furo não tenha sido realizado com sucesso, portanto sinalizando essa informação no LCD através da mensagem mostrada a seguir.

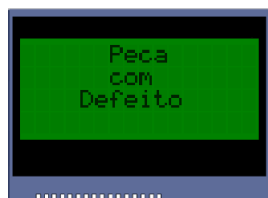


Figura 7: Peça com furo indevido

- *take*: Responsável por sinalizar a presença de um atuador retirando a peça da mesa após o ciclo do processo ser finalizado.



Figura 8: Remoção da Peça

3.4.3 Função Main

Após a criação das funções que seriam usadas no projeto, iniciou-se a criação da main, que baseou-se na ideia de uma máquina de estado, em que um estado aciona um outro posterior ou anterior (caso seja um processo cíclico). A seguir explica-se a função de cada um dos estados que no código são representados pela variável *estado*.

- Estado Inicial: Apresenta a mensagem inicial para o sistema ser ligado, além chamar a função *turn_on* e realizar a leitura do teclado. Ademais, chama-se também a função *setup* apenas para iniciar a próxima etapa. O valor da variável *estado* é nulo nesse caso.
- Estado 1: Realiza a leitura do teclado de forma a determinar qual número de defeitos tolerados pelo sistema foi pressionado pelo usuário.
- Estado 2: Responsável pela inserção de peças na esteira através da chamada da função *input*.
- Estado 3, Estado 5 e Estado 9: Chama a função que movimenta a mesa, ou seja, *move*.
- Estado 4: Etapa que chama a função da furação da peça, portanto, *drill*.
- Estado 6: Chama a função *test*.
- Estado 7: Faz leitura do teclado para vê se o furo foi bem sucedido ou não, sendo * realizado com sucesso e # furo com defeito.
- Estado 8: Ocorre apenas quando o furo é defeituoso, chamando função *error* e incrementando o contador de defeitos representado pela variável *cont*.

- Estado 10: Remove a peça da mesa chamando a função *take* e retorna para o Estado 2.

Uma vez que o processo cíclico é encerrado trata-se os casos especiais, ou seja, após o número de defeitos ser atingido. Sendo assim primeiramente, liga-se o display de 7 segmentos mostrando que o número de defeitos contados neste. Ademais, para sinalizar esse ocorrido o buzzer atua alterando diretamente o valor do bit do PORTC responsável por esse componente. Em seguida, realiza-se a leitura do teclado para esperar pelo processo ser desligado pelo pressionamento da telca *, que chama a função *turn_off* e altera o valor do PORTC desligando o buzzer, além de desligar o display de 7 segmentos.

3.5 Fluxograma:

A fim de ilustrar o funcionamento deste processo, e algumas ações tomadas, foi elaborado o seguinte fluxograma contendo a ideia geral do projeto.

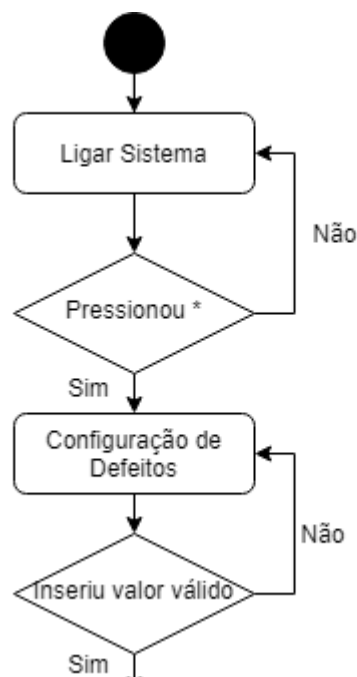


Figura 9: Fluxograma do Processo Parte I

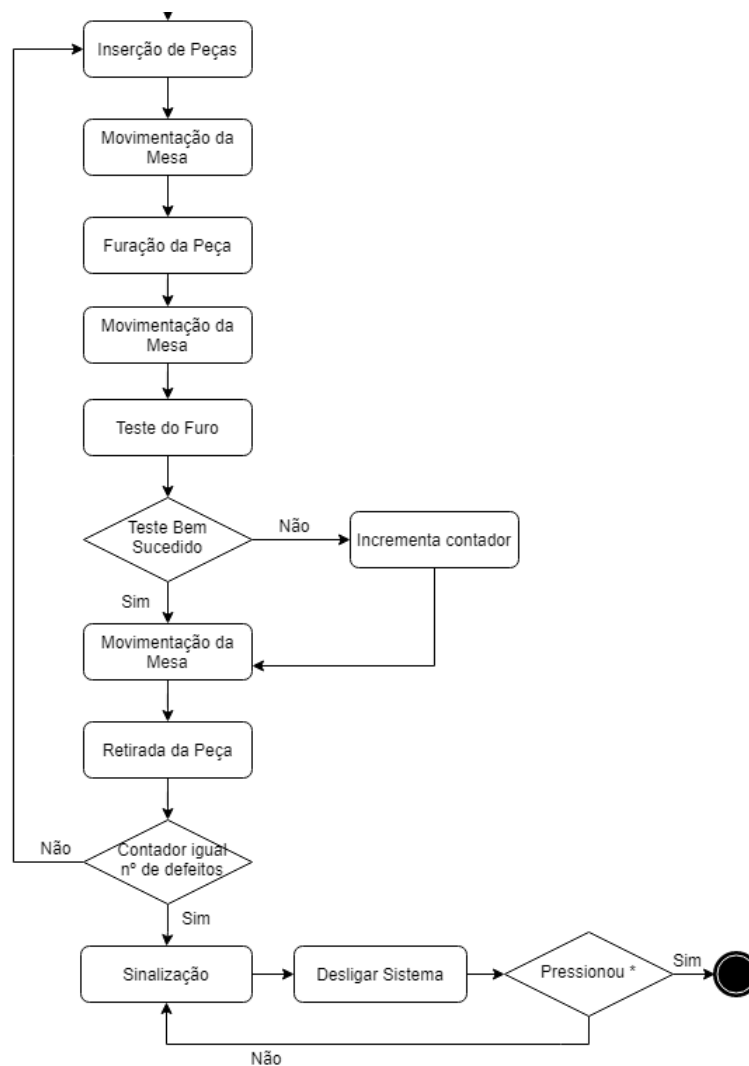


Figura 10: Fluxograma do Processo Parte II

3.6 Dificuldades e Soluções

Durante a elaboração do código ocorreu alguns problemas, sendo o mais recorrente a impossibilidade de usar certas funcionalidades ao mesmo tempo, sendo necessário realizar a multiplexação dos componentes. Além disso, Outra dificuldade foi o uso dos LED's do PORTD que inicialmente sinalizaria cada um dos atuadores do processo, entretanto devido ao uso do display de 7 segmentos o uso dessa porta não era possibilitado.

Sendo assim optou-se pelo uso do PORTB porém apenas é possível o uso dos bits mais significativos (B4 a B7), não havendo LED's suficiente para todos atuadores. Dessa forma, como a mensagem para cada ação é mostrada no LCD, alterou-se a utilização dos LED's apenas para repre-

sentar o sistema ligado e o sistema desligado.

Ademais, outras dificuldades encontradas foram utilização de equipamentos que até o momento da elaboração do código não haviam sido visto nas aulas, como por exemplo o cooler e o buzzer. Tal obstáculo foi solucionado através de pesquisa de exemplos em que esses componentes eram usados, além da leitura do datasheet da placa de forma a entender os bits e configurações necessárias para esses elementos. Outros problemas ocorridos foram devido a sintaxe da linguagem, sendo resolvidos com debug deste e a releitura do código de forma a encontrar os erros.

4 Modo de Usar

1. Ao carregar o programa, a placa encontra-se da forma apresentada na figura a seguir, devendo ser pressionada a tecla * para começar.

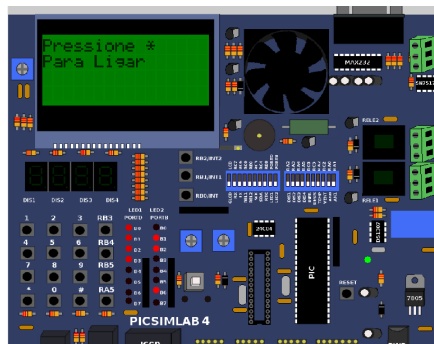


Figura 11: Situação Inicial

2. Em seguida, deve-se inserir um número de 1 a 9 no teclado para representar o número de defeitos tolerados pelo sistema, qualquer outro botão será ignorado pelo programa.

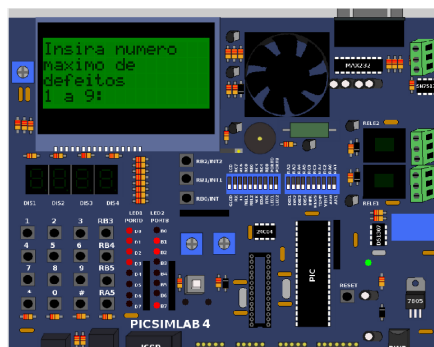


Figura 12: Configuração do Número de Defeitos

3. A outra interação com o usuário é o momento do teste, ao exibir a tela a seguir, deve-se pressionar * para indicar que não ocorreu defeito ou # indicando que essa peça será defeituosa.

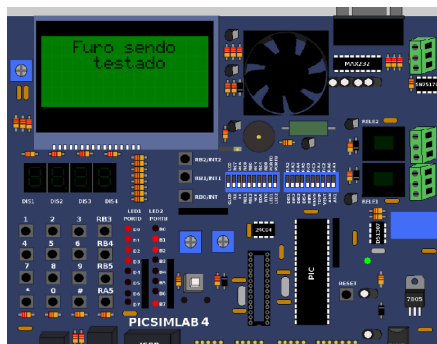


Figura 13: Realização do teste

4. A ultima participação do usuário é para desligar o sistema através do pressionamento da tecla * após o buzzer e o display de 7 segmentos serem acionados. Em qualquer outra parte do processo essa tecla possui outras funções ou apenas será ignorada.

5 Conclusão

Com esse trabalho conclui-se que através da linguagem de programação e o conhecimento da placa e funções pode-se elaborar diversos projetos aplicáveis a qualquer área. Ademais, entende-se também que independente do modelo da placa ou linguagem o que altera é a sintaxe e algumas funções ou variáveis sendo o conceito principal mantido o mesmo, sendo capaz de aplicar esses conhecimentos em outros casos desde que entenda-se as modificações necessárias. Finalmente, acrescenta-se que com o presente projeto foi possível aprofundar no conteúdo visto durante as aulas do semestre de forma a aprender mais sobre determinados componentes conforme foram usados.