Universidade Federal de Itajubá

Engenharia de Computação

Programação Embarcada e Laboratório de Programação Embarcada

Documento Técnico – ECOP04 e ECOP14

Simulação de um Micro-ondas

Aluno:

Bruno Vilela Batista – Matrícula: 2020003845

01 de agosto de 2021





Bruno Vilela Batista

Relatório do Projeto Final de Programação Embarcada

Simulação do funcionamento de um micro-ondas no micro controlador pic18f4520.

01 de Agosto, 2021

Sumário

RESUMO .....................................................................................................................04

1 INTRODUÇÃO.........................................................................................................05

2 DESENVOLVIMENTO............................................................................................05

2.1 Processo de desenvolvimento.....................................................................05

2.2 Funcionamento............................................................................................06

3 CÓDIGOS..................................................................................................................09

3.1 Bibliotecas e Funções..................................................................................09

3.2 Main.............................................................................................................11

4 DIFICULDADES ENCONTRADAS........................................................................12

5 CONCLUSÃO............................................................................................................12

Resumo

Neste relatório será descrito o processo de criação de um algoritmo para simulação de um micro-ondas no micro controlador PIC18F4520, com intuito de aplicar e testar os conhecimentos adquiridos na disciplina de Programação embarcada e no laboratório dessa disciplina. Como linguagem de programação foi usada a linguagem C, utilizando-se também o MPLAB X como IDE, XC8 como compilador e PicSimlab como emulador do microcontrolador real. O objetivo final da simulação presente nesse projeto era assemelhar-se com a interface de comunicação de um micro-ondas, apresentando ao usuário funções predefinas para o aquecimento de diversos tipos de alimentos.

Abstract

This report will describe the process of creating an algorithm for simulating a microwave in the PIC18F4520 microcontroller, in order to apply and test the knowledge acquired in the Embedded Programming discipline and in the laboratory of that discipline. As programming language the C language was used, using also MPLAB X as IDE, XC8 as compiler and PicSimlab as emulator of the real microcontroller. The final objective of the simulation present in this project was to resemble the communication interface of a microwave, presenting the user with predefined functions for heating different types of food.

**1.Introdução**

O micro-ondas é um eletrônico doméstico amplamente disseminado no mundo, presente em uma grande quantidade de lares. Só no Brasil, são aproximadamente 9 milhões de domicílios que possuem esse equipamento. Seu funcionamento se resume em agitar as moléculas de água presentes em quase todos os tipos de alimento, gerando então calor que, consequentemente, aquece o alimento.

Para que esse processo seja executado, é necessário que o eletrodoméstico “converse” com o usuário, fornecendo diversas informações, como estado do dispositivo(ligado ou desligado), tempo para o fim do preparo, qual potência deve ser usada ou que tipo de alimento está sendo esquentado; para que seu propósito seja executado corretamente e sem acidentes.

A simulação do hardware foi feita no PicSimlab com uma placa PicGenios, utilizando-se do microcontrolador PIC18F4520. Para o projeto, as funcionalidades da placa utilizadas foram: Display LCD 16x4, 4 Displays de Sete Segmentos, um Cooler, o Teclado Matricial, LEDs, relés e um potenciômetro.

**2.Desenvolvimento**

2.1-Processo de desenvolvimento

Antes de ser escrita a primeira linha de código, teve-se que pensar em um produto que fosse simples o suficiente para ser simulado pela quantidade limitada de dispositivos, memória e processamento da placa, que representam limitações coerentes com a realidade em que, por questões financeiras e espaciais, a quantidade de recursos utilizados em um projeto é limitado a não mais nem menos que o necessário para o seu funcionamento correto. Partindo desse princípio, o micro-ondas foi escolhido para ser o sistema simulado neste projeto.

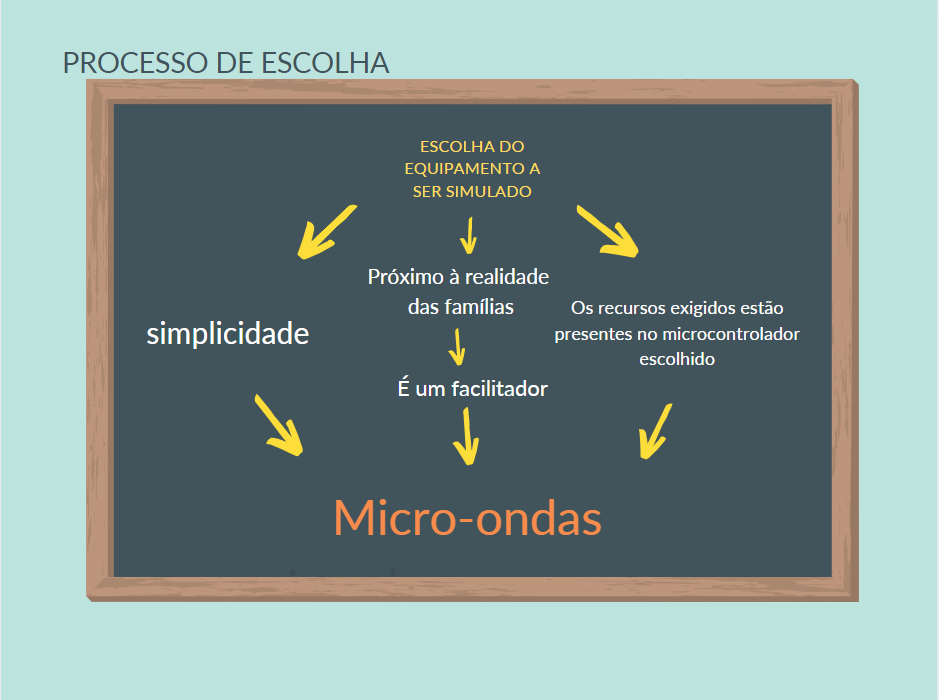
 O fluxograma abaixo indica esse processo:

Figura 1: Fluxograma da escolha de criação

Tomada a decisão de qual seria o objeto simulado, iniciou-se o desenvolvimento do código, começando por um “esqueleto” geral com vários comentários que foram posteriormente substituídos por funções, modularizadas no próprio arquivo main para evitar ao máximo bugs e erros desnecessários, que realizam a manipulação dos dispositivos presentes na placa, resultando em uma simulação funcional.

2.2-Funcionamento

O micro-ondas feito na simulação possui 6 opções que definem tempos predefinidos para o aquecimento de 6 tipos “genéricos” de alimentos, que cobrem uma ampla gama de alimentos que podem vir a ser encontrados nos ambientes domiciliares, tornando a opção de definição de um tempo personalizado menos atrativa de se incluir no projeto.

A situação inicial apresenta no Display LCD uma mensagem esperando o pressionamento de uma tecla para que seja iniciado/ligado o micro-ondas. Após pressionada tal tecla, é feita uma pequena “animação” envolvendo o acendimento progressivo dos leds presentes no PORTD e 3 mensagens: “ligando. ”, “ligando.. ” e “ligando...” que quando colocadas em sequenciada representam um processo comum de início de um dispositivo eletrônico.

Logo depois, é apresentada a primeira, de duas, tela do menu contendo as 3 primeiras opções de pratos para serem esquentados, reservando uma tecla para cada opção. Em seguida, após um pequeno atraso, é apresentada a segunda tela do menu, contendo mais 3 opções de alimentos e 1 reservada para o desligamento. Nesse momento o programa aguarda o pressionamento de qualquer tecla apresentada nas opções, da primeira e segunda telas de menu, para executar uma série de ações já programas. Vale lembrar que o relé 1, representando as portas do micro-ondas abertas, está ligada desde o início dessa etapa.

Divide-se agora 2 caminhos distintos, se foi pressionada alguma tecla ligada ao aquecimento de alimentos, um sub menu é apresentado no LCD para a escolha da potência a ser usada no processo de aquecimento e que influenciará no tempo necessário para tal tarefa, cuja intensidade é escolhida pelo potenciômetro1. Escolhida e confirmada a potência, são exibidas algumas mensagens no LCD, o cooler gira em sua velocidade máxima(simulando o prato do micro-ondas girando), o relé1 é desligado(embora apresente uma pequena interferência) o relé2 é ligado(representado o travamento das portas) e uma contagem regressiva do tempo restante de aquecimento é apresentado no display de sete segmentos. Findada a contagem, os relés são alternados, o cooler para, o display de 7-seg é desligado e uma mensagem dizendo que o alimento está pronto e que as portas estão abertas/destravadas é exibida no LCD permanecendo por um tempo de 8 segundos, depois do atraso, uma mensagem no LCD indica uma tecla para voltar ao menu principal e o programa aguarda o pressionamento dessa tecla e reinicia sua rotina principal. O segundo caso é o caso de pressionamento da tecla reservada para o desligamento, caso seja pressionada, todos os dispositivos usados na simulação são desligados e uma mensagem informa o correto desligamento; após 3 segundos a mensagem é substituída outra que informa uma tecla para que o “micro-ondas” seja ligado novamente, aguardando esse pressionamento por um tempo indeterminado, simulando um estado de stand-by. Pressionada tal tecla, a rotina principal é reiniciada, lembrando que a inicialização não é executada novamente, uma vez que os dispositivos já estão, teoricamente, devidamente iniciados, apenas aguardando novos comandos, exceto se o dispositivo for desligado sem ter esquentado nenhum alimento, apresentado novamente a animação de inicialização.

Abaixo serão mostradas as telas geradas durante a execução:

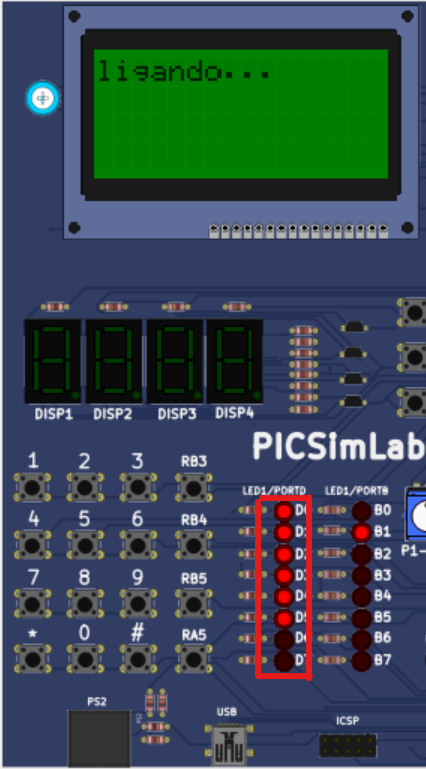


Figura2: Telas iniciais

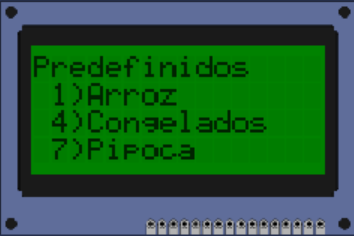
 

Figura3: Menu



Figura 4: Menu de potência

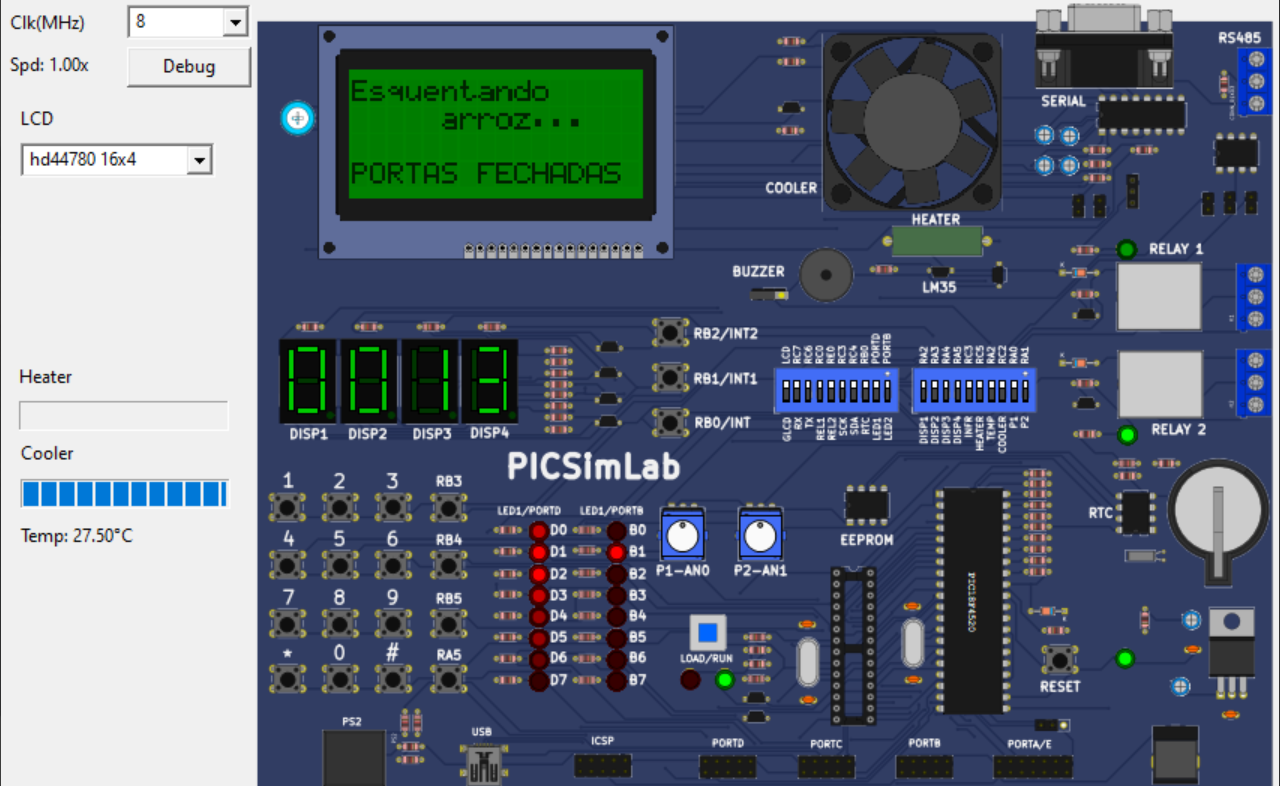


Figura 5: Situação no aquecimento(exemplo)

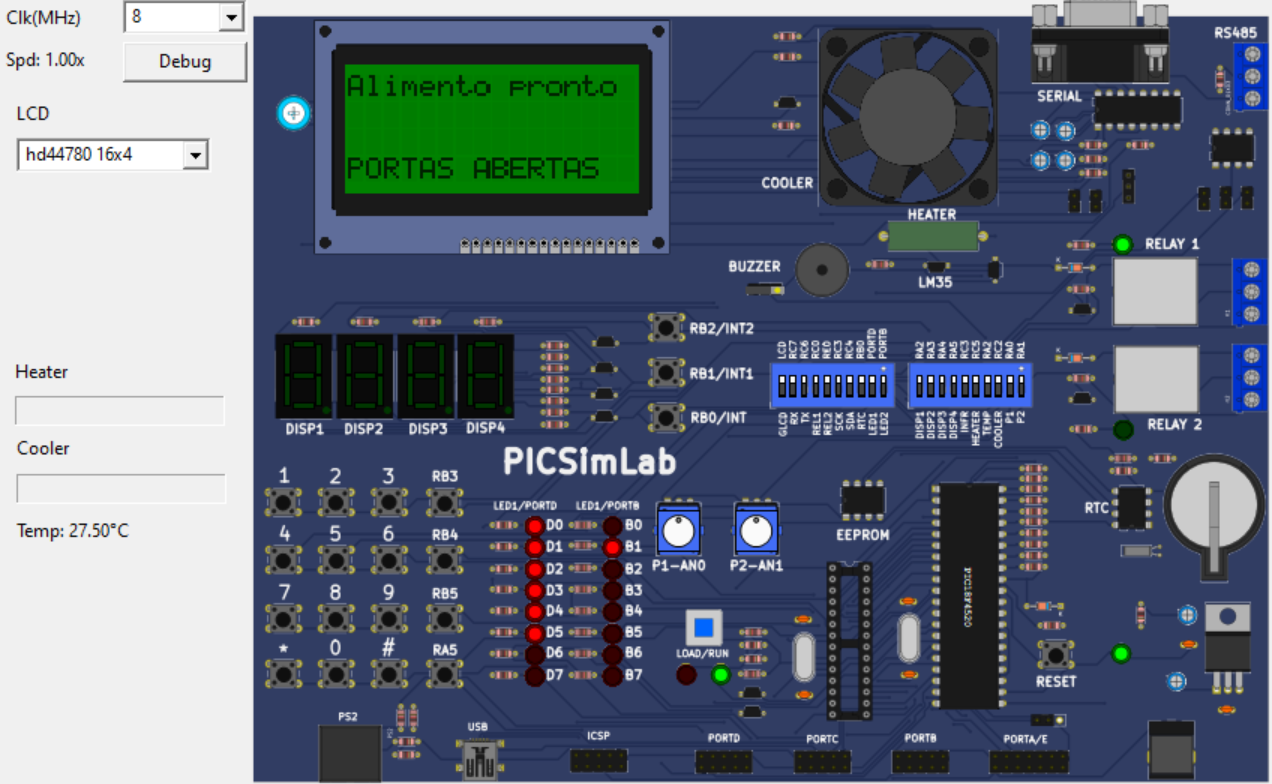
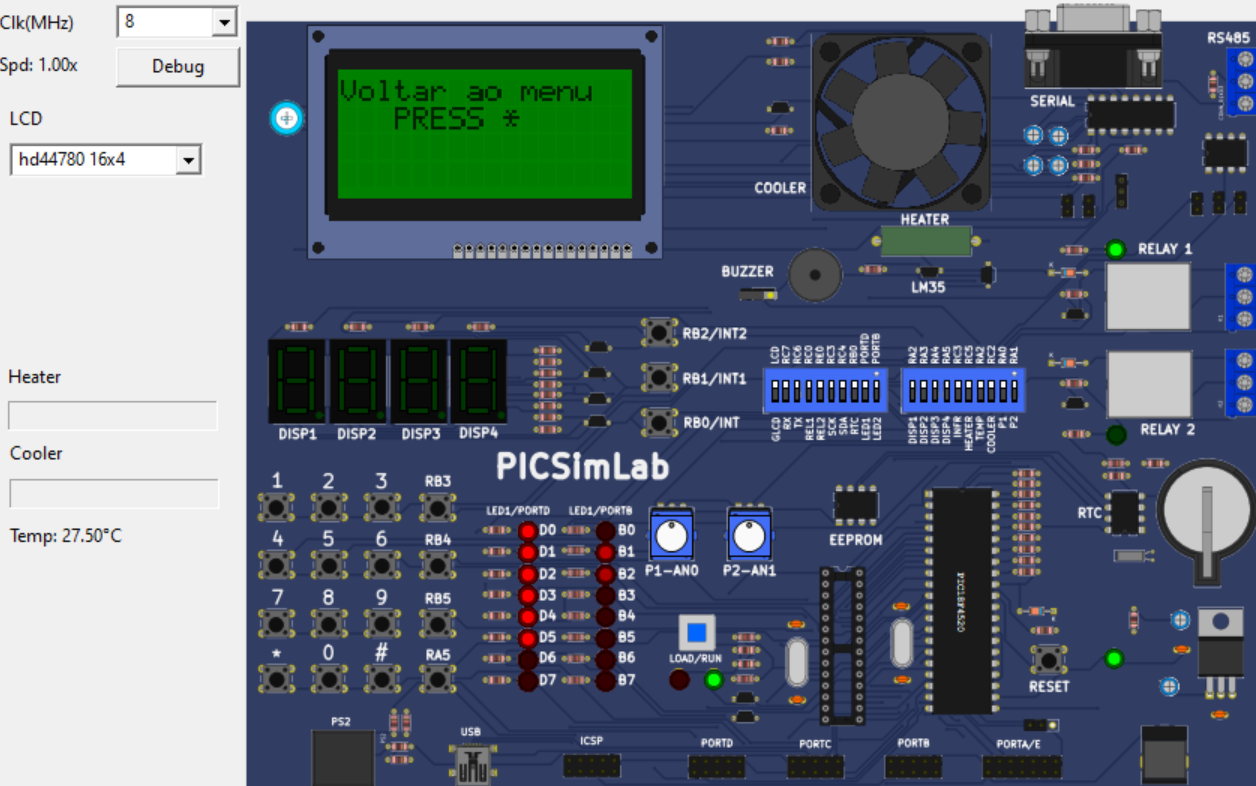
 

Figura 6: Situação fim aquecimento

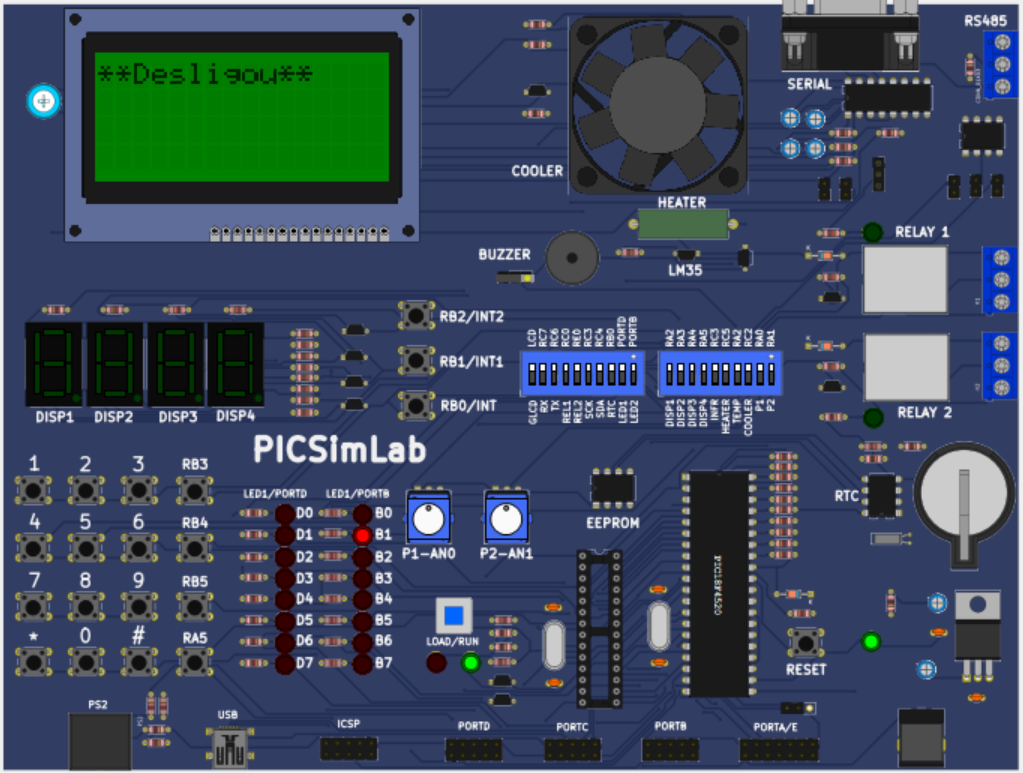


Figura7: Situação desligamento

**3. Código**

3.1- Bibliotecas e funções

Para a produção do código, foram utilizadas 13 bibliotecas criadas e mais 2 extras, próprias do compilador, além de algumas funções criadas no próprio arquivo main, separadas da rotina principal.

Dentre as bibliotecas, a explicação será limita somente à aquelas que não são disponibilizas pelo PIC18F4520 ou pelo compilador, ou seja, as bibliotecas criadas ao longo do semestre nessa disciplina.

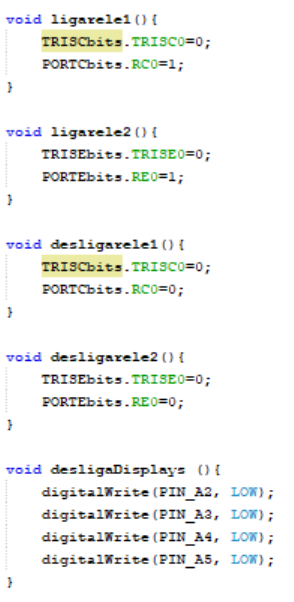
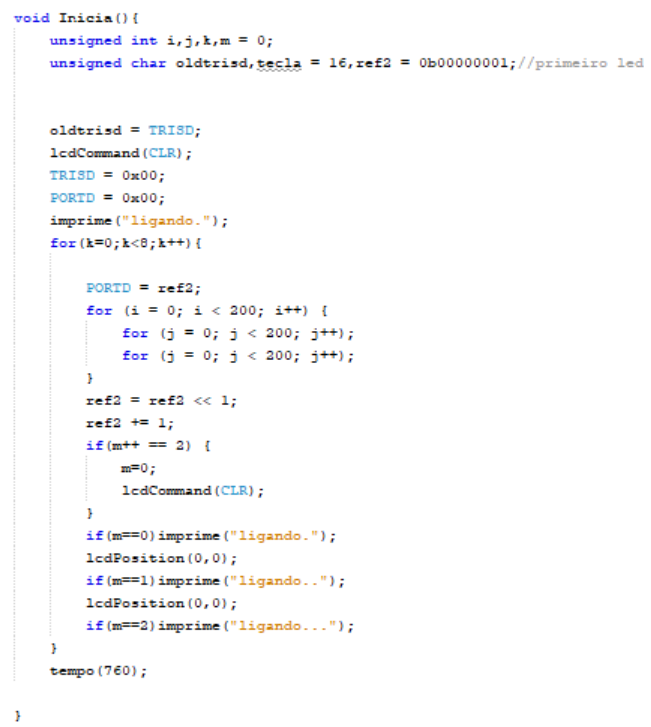
* Delay.h: Gera atrasos.
* lcdPosition.h: Move o “ponteiro” do LCD para a posição desejada.
* Stdio.h: Função para imprimir mensagens no LCD.

Passando agora para as funções criadas no arquivo main:

* void ligarele1(): Liga o relé 1.
* void ligarele2(): Liga o relé 1.
* void desligarele1(): Desliga o relé 1.
* void desligarele2(): Desliga o relé 2.
* void ligarcooler(): liga o cooler.
* void desligacool(0: desliga o cooler.
* void desligadisplay(): Desliga todos os displays de 7 segmentos.
* void Inicia(): Faz uma animação que simula a inicialização de um dispositivo genérico.
* void potencia(): Exibe um menu e realiza a leitura da potência através potenciômetro P1.

void temporizador(unsigned long int valor): Mostra no display de 7 segmentos uma contagem regressiva para o fim do tempo recebido como parâmetro.

* int verfligamento(): Verifica a intenção de sair do modo stannd-by/desligado.
* void menuPrincipal(): Exibe o menu principal. Chama a função inicia na primeira vez que essa função (menuPrincipal()) é chamada na main.

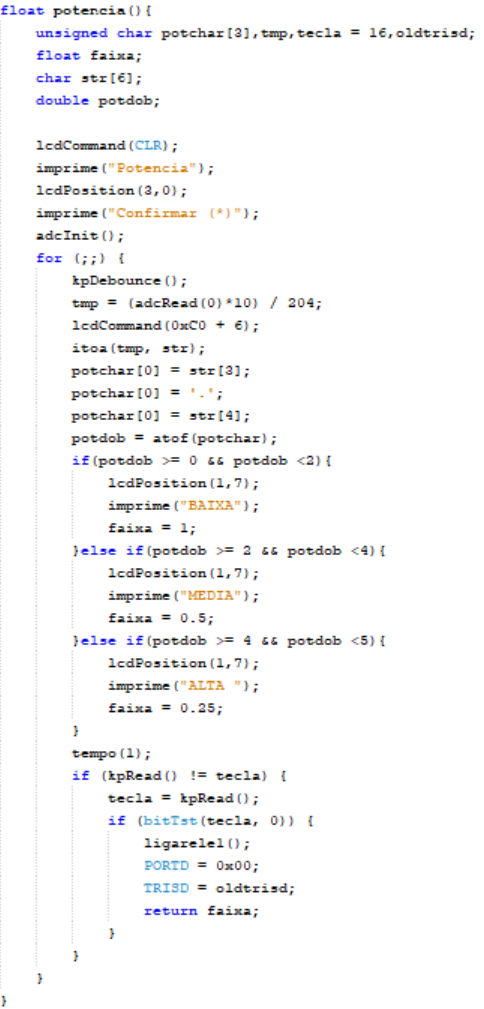
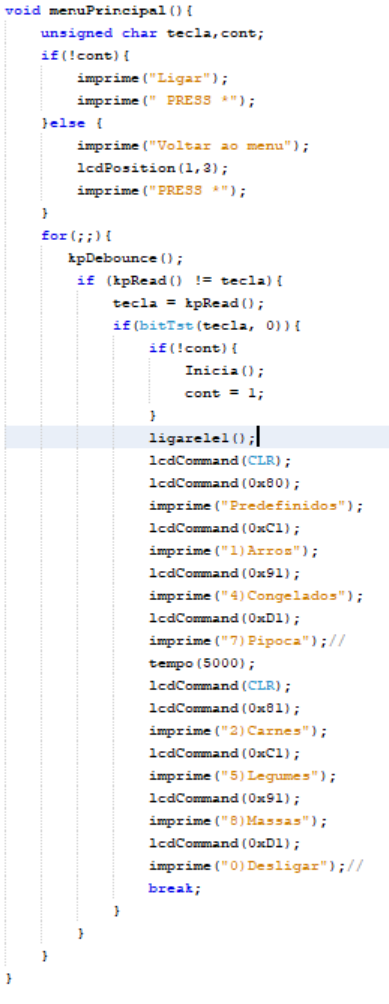
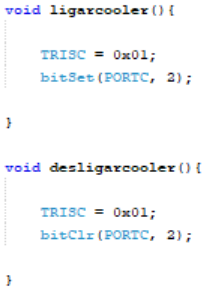
  

Figura 8: Funções criadas no arquivo main, fora da rotina principal.

3.2-Main

Na construção da rotina principal, são criadas variáveis necessárias, inicializados os dispositivos e feita a chamada do menu principal, após isso o programa entra em um loop infinito, saindo dele somente quando uma tecla for pressiona(verificação feita pela varredura proporciada pela função kpDebounce()) e as ações ligadas a esse pressionamento sejam propriamente executadas. Como não há nenhum comando após esse dito loop infinito, a rotina main é reinicida após um comando break presente em cada situação possível nesse loop.

Dentro do loop, são feitos diversas comparações através da função bisTst() que, guiado pela tecla pressiona, direciona o programa para executar uma sequência de ações envolvendo o cooler, os relés, LCD e displays de 7 segmentos para simular um micro-ondas, enquanto está esquentando algum alimento.

Um caso especial é do comando relacionado ao desligamento, ao pressionar a tecla reservada para esse fim, uma instrução while “trava” a execução do código até que uma tecla específica seja pressionada, simulando um modo stand-by.

Abaixo será inserida a instruções iniciais da rotina main, um exemplo da estrutura criada para cada opção disponibilizada no menu e as instruções do “desligamento”.

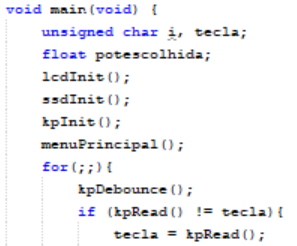
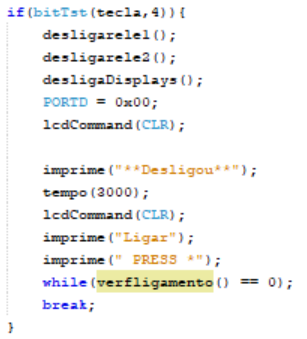
  

Figura 9: instruções iniciais da rotina main, um exemplo da estrutura criada para cada opção disponibilizada no menu e as instruções do “desligamento”

4.Dificuldades Encontradas

Durante o processo de construção do código, o maior problema foi, sem dúvidas, o uso do teclado. Em diversos pontos, foi-se necessário mudar a abordagem programada inicialmente a fim de contornar obstáculo envolvendo a leitura matricial do teclado e mesmo assim vários problemas persistiram, levando a um código funcional porém não muito otimizado.

Além disso, o uso dos displays de 7 segmentos concomitantemente com os leds do PORTD gerou graves erros, forçando novas mudanças de abordagem a fim de evitar a necessidade de utilizar esses dois dispositivos juntos.

Vale destacar ainda, a ocorrência de alguns bugs da IDE utilizada, que dificultava encontra a fonte de alguma falha na compilação, uma vez que havia dúvidas se o erro era na sintaxe, lógica ou na própria plataforma.

5.Conclusão

Apesar das dificuldades encontradas , foi-se capaz de gerar um código funcional, que, por sua vez produziu uma simulação razoavelmente condizente com a realidade, ou seja, uma simulação satisfatória.

Além disso, as informações relacionadas ao consumo de memória disponibilizadas pela interface de desenvolvimento demonstra claramente a necessidade de, em próximos projetos, buscar melhores formas de construir o código final uma vez que o limite de recursos disponíveis na placa pode ser facilmente atingido levando ao fracasso de um projeto maior, que não foi capaz de otimizar a utilização dos recursos disponíveis.

Por fim, é indiscutível que as habilidade de manipulação de sistemas embarcados, adquiridas ao longo do semestre letivo, foram devidamente testadas e, inclusive aprimoradas pela produção deste trabalho.