



UNIVERSIDADE FEDERAL DE ITAJUBÁ

ECOP04

PROGRAMAÇÃO EMBARCADA

RELATÓRIO DA ATIVIDADE FINAL DE PROGRAMAÇÃO EMBARCADA

*Atividade avaliativa final para a disciplina ECOP04 -
Programação Embarcada*

Docente:

Otávio de Souza Martins Gomes

Discente:

Daniel Augusto Teixeira Venâncio - 2020006079

ECOP04 - T01 / ECOP14 - T02

31 de Julho, 2021

Daniel Augusto Teixeira Venâncio

RELATÓRIO DE ATIVIDADE EXPERIMENTAL Nº03

Atividade avaliativa final
para a disciplina ECOP04 -
Programação Embarcada

31 de Julho, 2021

1 Resumo

Neste relatório serão apresentados o resultado e o processo de criação do trabalho final apresentado para as disciplinas de ECOP04 e ECOP14. O trabalho final consiste na simulação de um carro utilizando uma placa de desenvolvimento com o controlador PIC18F4520. A simulação foi desenvolvida utilizando os próprios componentes da placa.

[left=0cm,top=2.5cm,right=2.5cm,bottom=2.5cm]geometry

2 Introdução

O projeto busca, por meio das funcionalidades presentes na placa de desenvolvimento PICGenius, simular algumas funções de um carro. Para tanto são trabalhados tanto dispositivos de coleta de informações do usuário quanto dispositivos de saída de informações.

A simulação é composta por 3 estados distintos conectados entre si e cuja transição é comandada pelo usuário. Os três estados são: carro desligado; carro em funcionamento; e carro abastecendo.

Dos componentes disponíveis na placa PICGenius, 5 foram selecionados para representar o funcionamento do carro:

- Display LCD
- Teclado matricial
- Comunicação Serial
- Ventoinha (cooler)
- Relés

O **display LCD** é o componente principal de apresentação de informações ao usuário. Nele é possível observar informações distintas a depender do estado do carro. No estado "desligado" o display apresenta informações de como ligar e desligar o veículo. No estado "em funcionamento" o display é

responsável por informar a marcha do carro e o seu nível de gasolina. Por fim, no estado de "abastecimento", o display mostra ao usuário que o carro esta sendo abastecido e informa a quantidade de gasolina que foi colocada.

O **teclado matricial** é a maneira com a qual o usuário controla o carro. As teclas mapeadas são os números de 1 a 9 e a tecla '#'. Cada tecla tem uma função específica, descrita abaixo.

- Tecla '1' : coloca o carro na marcha 2
- Tecla '2' : coloca o carro na marcha 1
- Tecla '3' : seta para a esquerda
- Tecla '4' : coloca o carro na marcha 4
- Tecla '5' : coloca o carro na marcha 3
- Tecla '6' : seta para a direita
- Tecla '7' : coloca o carro na marcha Ré
- Tecla '8' : coloca o carro na marcha 5
- Tecla '9' : liga/desliga o carro
- Tecla '#' : coloca o carro em modo de abastecimento

A escolha da posição das marchas não foi aleatória. A intenção foi simular o caminho de um cambio real. As figuras 01, que mostra as teclas de controle de marcha da simulação, e 02, onde e apresentado um caminho de cambio real, foram colocadas lado a lado para possibilitar uma comparação visual.



figura 01

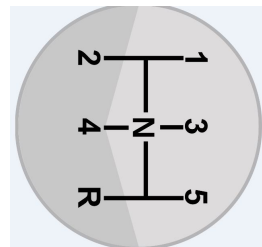


figura 02

A **comunicação serial** tem o objetivo de controlar até quanto de gasolina será adicionada no tanque do carro enquanto estiver no estado de abastecimento.

A **ventoinha** busca simular a velocidade do motor/velocidade do carro. É importante enfatizar que não foi levado em consideração o funcionamento real do motor de automóvel. A velocidade de giro da ventoinha pode ser entendida como a velocidade de giro do motor do carro e a velocidade do carro pode ser entendida como somente dependente da velocidade de giro do motor. Essa velocidade é alterada de acordo com as marchas.

Os **relés**, por sua vez, simulam as setas do carro. O relé superior representa a seta esquerda e o relé inferior representa a seta direita.

3 Desenvolvimento

3.1 Implementação do display LCD

O processo de implementação do display LCD foi auxiliado pela biblioteca "lcd.h" fornecida pelo professor. Houve, porém, adição de funções nessa biblioteca, de modo que, como é possível perceber nos arquivos do projeto, a biblioteca utilizada se chama "lcd2.h". As funções utilizadas da biblioteca "lcd2.h" são:

- void lcdInit() : realiza as configurações iniciais para a utilização do display LCD.
- void lcdCommand(unsigned char) : envia um comando ao LCD.
- void lcdData(unsigned char) : envia um caractere ao LCD.
- void lcdPosition(unsigned char, unsigned char) : posiciona o cursor do display LCD na posição desejada.
- void lcdMessage(const char*) : envia um conjunto de caracteres para o display LCD.

3.2 Implementação do teclado numérico

O processo de implementação do teclado foi auxiliado pela biblioteca "keypad.h" fornecida pelo professor. A biblioteca conta com 3 funções:

- void kpInit() : configura os 3 bits menos significativos de TRISB como saídas e os 4 bits menos significativos de TRISD como entradas, de modo a permitir o funcionamento do teclado matricial.
- void kpDebounce() : Realiza o Debounce de modo a impossibilitar que somente um clique seja processado multiplas vezes e, quando uma tecla for clicada, armazena seu valor em uma variável.
- unsigned int kpRead() : Retorna o valor da tecla clicada (valor esse processado pelo kpDebounce).

Entendido o funcionamento das funções fornecidas, a implementação do teclado se resume a ficar constantemente, ou seja, dentro do meu loop infinito, chamando a função kpDebounce() e analisando se kpRead() retorna algo diferente de 0. Caso o retorno seja diferente de 0, basta analisar qual o valor desse retorno para saber a tecla pressionada a partir do mapeamento das teclas e então fazer o processamento desejado.

3.3 Implementação da comunicação serial

O processo de implementação da comunicação serial foi auxiliado pela biblioteca "serial.h" fornecida pelo professor. As funções utilizadas da biblioteca foram:

- void serial_init() : *realiza as configurações iniciais da comunicação serial.* void serial_tx() : *realiza a leitura da informação enviada via comunicação serial.*

A comunicação serial é feita no processo de abastecimento do carro. Ao clicar na tecla '#', a simulação entra em estado de abastecimento e, por duas vezes consecutivas, realiza um "while" que espera até que um valor diferente de "0xa5" (valor padrão da comunicação serial) seja enviado, de modo que seja possível realizar a leitura de um valor de dois dígitos que, posteriormente, será armazenado no controle do nível de gasolina.

3.4 Implementação da ventoinha

O processo de implementação da comunicação serial foi auxiliado pela biblioteca "pwm.h" fornecida pelo professor. As funções utilizadas da biblioteca foram:

- void pwmInit() : realiza as configurações iniciais da saída pwm.
- void pwmSet1(unsigned char) : que configura a saída pwm 1 com a porcentagem desejada.

Uma vez que a ventoinha está conectada a saída pwm 1, a sua velocidade de movimento é controlada pela função pwmSet1(). Com isso em mente, a alteração dos valores da velocidade foi realizada de acordo com a tecla apertada, ou seja, a velocidade está diretamente relacionada com a marcha do carro.

3.5 Implementação ds relés

Os relés são controlados pela biblioteca interna do PIC18F4520, de modo que, para acionar o relé inferior basta configurar PORTCbits.RC0 como 1, e para acionar o relé superior basta configurar PORTEbits.RE0 como 1.

Sabendo disso, a implementação do relé se resumiu a mapear as teclas 3 e 6 do teclado e, quando apertadas, começar um ciclo for que faz com que o relé referente pisque 3 vezes.

3.6 Fluxograma

A figura 03 representa o fluxograma do programa desenvolvido.

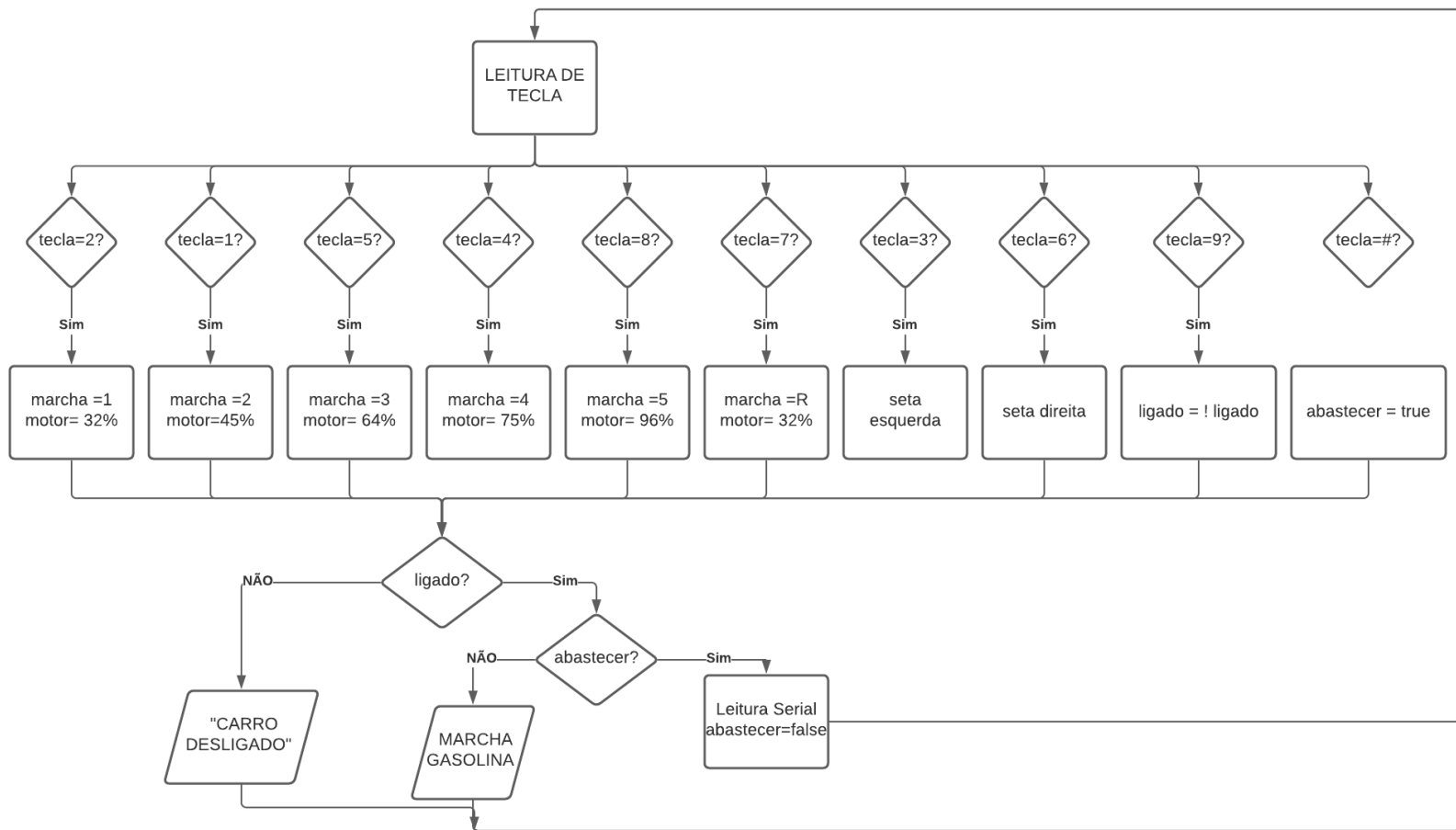


figura 03

3.7 Imagens da simulação em funcionamento

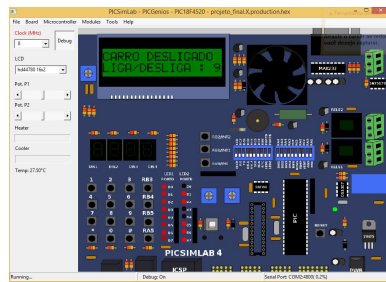


figura 04
Estado desligado

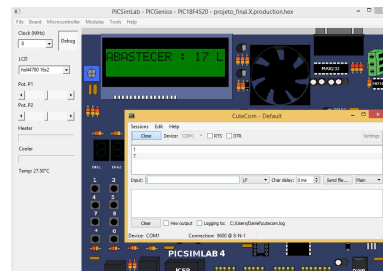


figura 05
Estado de abastecimento

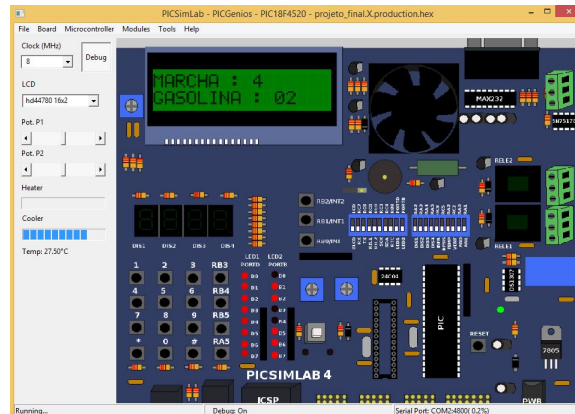


figura 05
Estado ligado

4 Conclusão

O objetivo almejado no início do processo de desenvolvimento do projeto final foi alcançado. A ideia inicial se manteve em quase sua totalidade. A única funcionalidade inicialmente pensada que não foi implementada foi a leitura do potenciômetro para simular a velocidade. Essa funcionalidade, porém, foi substituída pela representação da velocidade por meio do controle da saída pwm conectada a ventoinha.

O projeto final foi uma ótima oportunidade para integração dos conteúdos vistos ao longo do semestre nas disciplinas ECOP04 e ECOP14.

5 Referências bibliográficas

- ALMEIDA, Rodrigo M. A.; MORAES, Carlos H. V.; SERAPPHIM, Thatyana F. P.; *Programação de Sistemas Embarcados : desenvolvendo software para microcontroladores em linguagem C*. GEN LTC; 1ª edição (5 julho 2016).