

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

## **Relatório de Projeto**

### **Exercício I: Máquina de Cuba Libre**

**Grupo:**

Guilherme de Souza Bastos  
Karl Vandesman de Matos Sousa

**Disciplina:**

Microcomputadores

**Professor:**

Mauro dos Santos

Recife - PE  
Maio de 2017

# 1 Introdução

Este trabalho consiste no projeto de um sistema de controle e acionamento utilizando o microcontrolador PIC16F628A, da Microchip. O sistema realiza o controle automático de quatro máquinas para fazer uma bebida chamada Cuba Libre, feita com uma dose de rum, quatro cubos de gelo, uma fatia de limão e 250ml de Coca-Cola. As máquinas realizam as seguintes atividades:

- A máquina M1 fornece uma dose de rum toda vez que for ativada;
- A máquina M2 fornece um cubo de gelo toda vez que for ativada;
- A máquina M3 fornece uma fatia de limão toda vez que for ativada;
- A máquina M4, quando ativada, fornece Coca-Cola e após escoar 250ml gera um sinal de saída  $S=1$ , sinalizando o fim da preparação da bebida.

Essas máquinas são acionadas a partir de um pulso de  $T_a = 1segundo$ . Após a preparação de  $N=15$  Cubas Libres (capacidade de abastecimento), deve ocorrer o reabastecimento dos insumos, para que a preparação possa acontecer.

Foram realizadas duas versões para esse projeto, com uma diferença no código utilizado no microcontrolador, e fazendo pequenas alterações na interface com o usuário, adicionando entradas e saídas.

As características comuns às versões são quanto a algumas entradas e saídas:

- Entradas:
  1. *Partida: reset* do sistema;
  2. *Preparar*: responsável por realizar uma Cuba Libre;
  3. *S*: avalia se a máquina M4 terminou o derramamento de Coca-Cola. Quando acontece esse término, um sinal é enviado para essa entrada ( $S=1$ ), e consequentemente acaba o fim do ciclo de preparação da bebida.
  4. *Oscilador*: foi usado um oscilador externo de frequência  $10Hz$  para o *delay*, como alternativa ao oscilador interno do PIC16F628A que possui frequência muito alta,  $4MHz$ , e seria necessário muitos pulsos para realizar um *delay* de 1 segundo.
- Saídas:
  1. *Preparando*: acende/apaga LED sinalizando que está sendo preparada uma bebida;
  2. *Fim*: acende/apaga LED sinalizando que ocorreu o fim da preparação de uma bebida;
  3. *M1, M2, M3 e M4*: acionamento das máquinas, em que M1, M2 e M3 serão acionadas por um pulso de  $T_a = 1segundo$ , e M4 será acionada por nível lógico alto.
  4. *Display de 7 segmentos*: apresenta a quantidade de cubas libres que ainda podem ser realizadas em hexadecimal. Logo no começo da preparação de uma bebida, esse valor já é decrementado.

Na versão 1, foram usadas as entradas e saídas elencadas anteriormente. Na segunda versão, o diferencial foi o uso de interrupções para o tratamento das entradas e do contador *Timer 0*, a adição de uma entrada chamada *Carregar*, para uso do usuário informando que as máquinas foram reabastecidas, e uma saída (*Aviso carregar*) informando que não há material para realização das bebidas.

## 2 Metodologia

Para realizar o desenvolvimento do projeto de acordo com o requisitado foi necessário realizar o seguinte passo a passo:

- Realizar uma visão geral do projeto, descrevendo-o a partir de uma interpretação técnica;
- Definir o modo de operação com a interface Homem/Máquina, a partir das entradas e saídas do sistema;
- Configurar as portas do PIC16F628A de acordo como as especificações da sua folha de dados. Portas específicas possuem a habilidade de interrupção, outras são usadas como entrada analógica para o conversor A/D interno, entre outras coisas;
- Criar um fluxograma para ilustrar a lógica central do programa;
- Desenvolver o código fonte em *assembly*;
- Simular o funcionamento do sistema no Proteus.

O MPLAB foi o *software* utilizado para desenvolver aplicações para o microcontrolador, ou seja, a ferramenta usada para se desenvolver o código em *assembly*. O *software* Proteus foi utilizado a fim de simular o funcionamento do sistema, utilizando o código *assembly* no microcontrolador e as entradas e saídas necessárias.

## 3 Desenvolvimento

### 3.1 Versão 1 - Sem interrupções

Na Figura 1 está o fluxograma em relação a versão 1 do projeto, representando de forma mais fácil e direta o controle do microcontrolador. O código, como dito anteriormente, foi escrito em linguagem *Assembly* para programação do PIC16F628A. Inicialmente são feitas as configurações iniciais do microcontrolador, definindo as portas de entrada e saída, atribuindo valores iniciais para as variáveis e outras configurações gerais do PIC. A rotina do *display* apresenta o valor de cubas livres que ainda podem ser realizadas. Logo após isso, é verificado o botão *preparar*, que é uma interface com o usuário. Caso o número de bebidas que podem ser feitas seja maior que zero, o microcontrolador acende o LED avisando o início da operação e começa a coordenar o acionamento das máquinas. Assim, aplica de forma sequencial um pulso para M1, quatro pulsos para M2 (quatro cubos de gelo), um pulso pra M3, e uma mudança de nível para M4, que derramará 250ml de coca-cola, e após isso, envia um sinal S=1 para o microcontrolador avisando o término da ação. Ao final, é acendido um LED avisando do término da preparação da bebida, e

assim o sistema volta inicialmente pro ponto em que se verifica se o botão preparar foi pressionado.

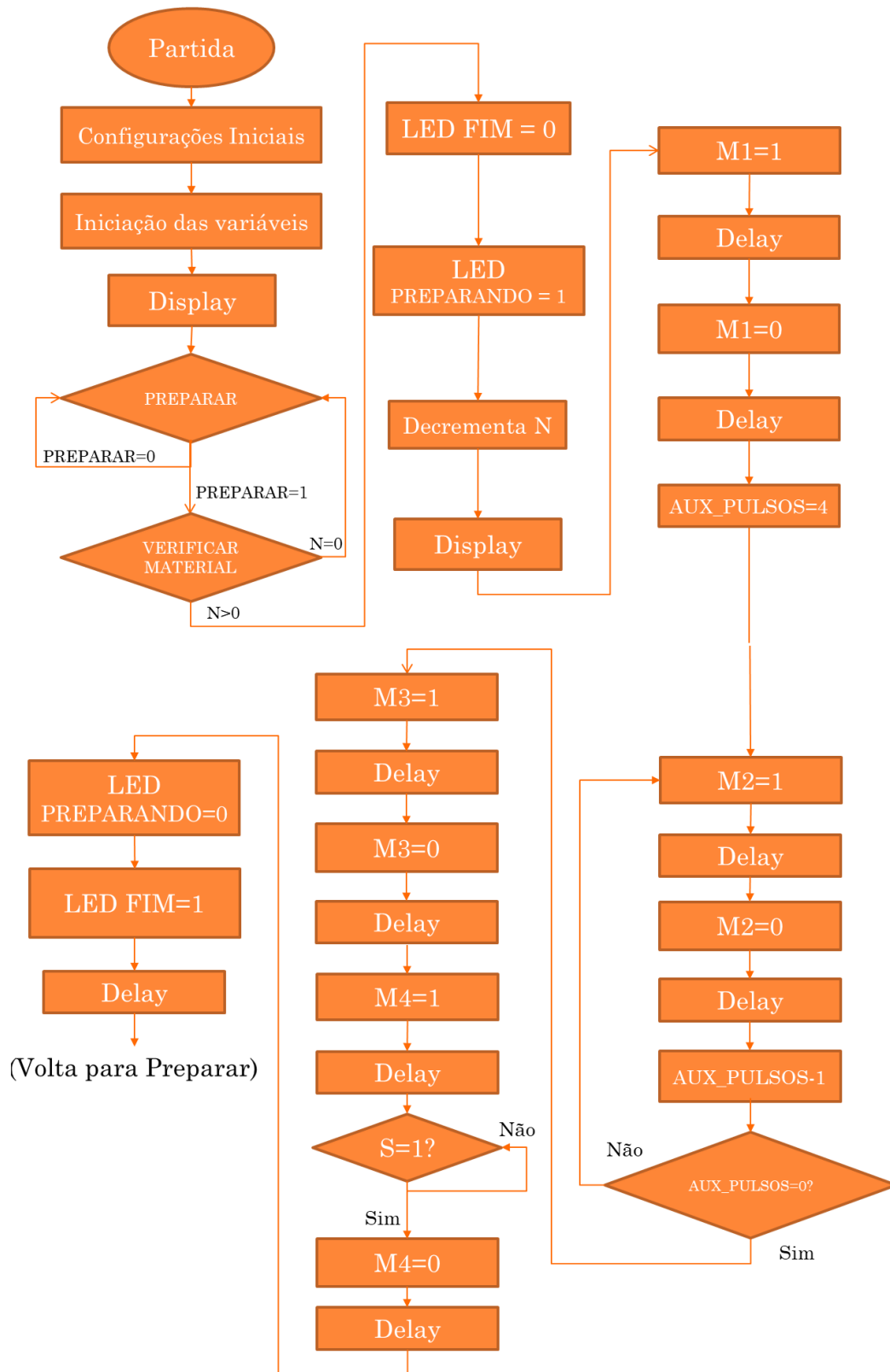


Figura 1: Fluxograma do projeto, versão 1.

### 3.2 Versão 2 - Com interrupções

Interrupções, como deixa claro em seu nome, serve para interromper o programa imediatamente, de forma assíncrona. Quando uma interrupção ocorre, o programa é paralisado e uma rotina específica é executada, localizada no endereço 0x04, e quando finalizada, o programa continua de onde parou antes da interrupção. Foram utilizadas 4 interrupções nesta segunda versão do projeto, sendo elas interrupção do *Timer 0*, externa (RB0) e por mudança de estado (foram utilizados os pinos RB6 e RB7 para interrupção).

A rotina de interrupção trata inicialmente de salvar o contexto em que o programa estava inserido, armazenando em variáveis temporárias os registros STATUS e W. Após isso, são verificados os sinalizadores de cada interrupção, que são setados quando a respectiva interrupção ocorre. Assim, se a *flag* (sinalizador) estiver em nível lógico alto, uma rotina específica será executada. Caso o *Timer 0* tenha estourado, caso o operador tenha mandado preparar uma Cuba Libre, caso a Coca-Cola tenha parado de escoar (sinal S=1) ou ainda caso o sistema seja recarregado, em todos esses casos, ocorrerá uma interrupção no microcontrolador.

A rotina de acionamento de máquinas será bastante similar a da primeira versão, com diferença no *delay*, que será feito utilizando a interrupção de *Timer 0*.

## 4 Resultados

Como dito anteriormente, para a simulação foi usado o *software* Proteus. Logo, foi implementado para a simulação tanto o microcontrolador PIC16F628A, quanto alguns recursos para simular o interfaceamento com as máquinas e com as entradas. As figuras 2 e 3 apresentam o esquemático do projeto, nas versões 1 e 2, respectivamente.

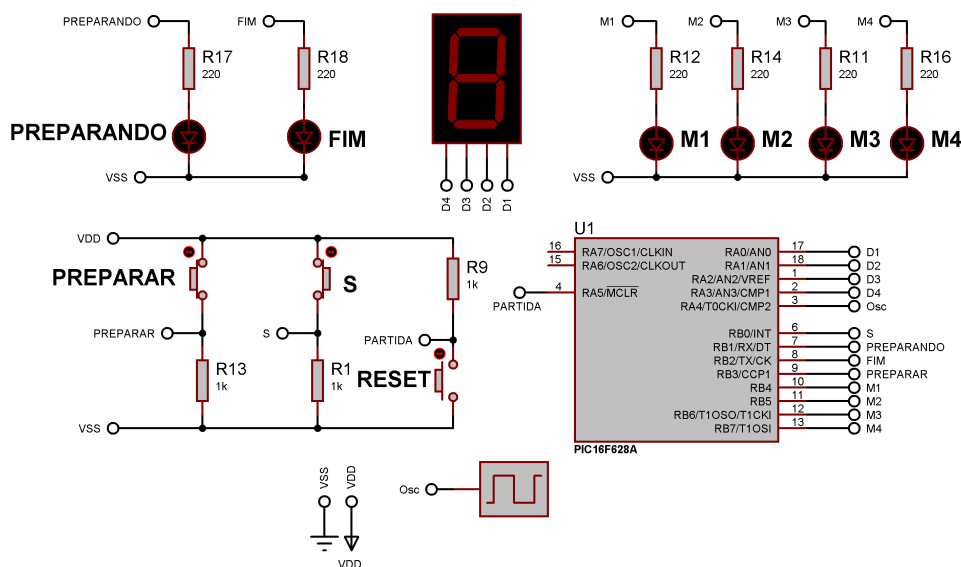


Figura 2: Visão geral da versão 1 do projeto no Proteus.

Como se pode ver nos esquemáticos do Proteus, as máquinas M1, M2, M3 e M4 estão sendo representadas por LEDs (pois são saídas), e as entradas por botões.

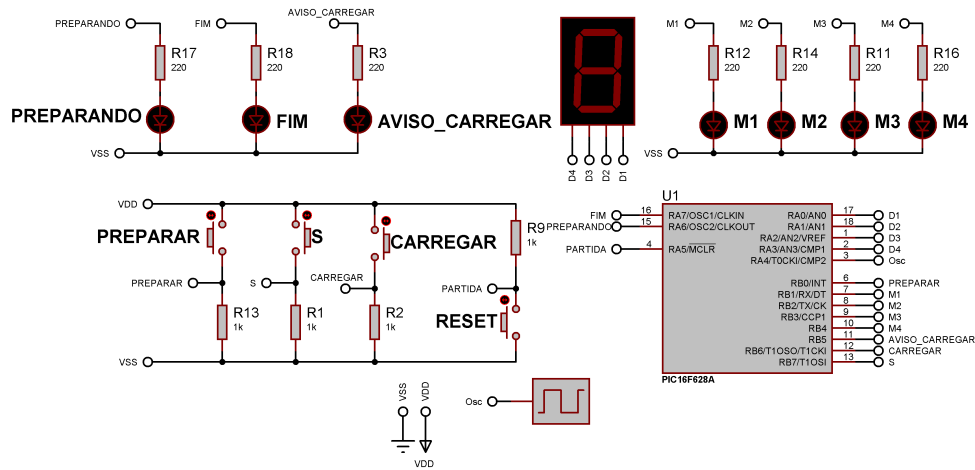


Figura 3: Visão geral da versão 2 do projeto no Proteus..

## 5 Conclusão

Foi atingido o objetivo de aprender a desenvolver um projeto, com base em certas especificações. Foi possível utilizar saídas do tipo pulso e de mudança de nível para acionamento das máquinas, e visto as várias formas de interrupção presente no PIC16F628A, como interrupção externa por RB0, por mudança de nível nas portas RB4 a RB7 e estouro do *Timer 0*. Na versão 2, como estiveram presentes mais de uma interrupção, foi importante também avaliar o tratamento de mais de uma interrupção, definindo prioridade no programa para o tratamento de cada uma. A simulação do sistema desenvolvido obteve sucesso na tarefa dada, o controle das máquinas na fabricação de uma Cuba Libre com uma interface para usuários.