**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**

**Jonas José, Monaline Chaves, Paulo Frade e Victor Dias**

**ROLETA DO BOTAFOGO**

**Belo Horizonte**

**2017**

**Jonas José, Monaline Chaves, Paulo Frade e Victor Dias**

**Turma: ELT2A**

**ROLETA DO BOTAFOGO**

Relatório técnico de projeto apresentado ao Curso Técnico em Eletrônica do Centro de Educação Tecnológica de Minas Gerais.

Orientador: Prof. Enderson Neves Cruz

Disciplina: Introdução à Programação

**Belo Horizonte**

**2017**

**RESUMO**

Esse artigo trata da explicação e a descrição do funcionamento minuciosa da primeira parte do projeto “Roleta do estádio do Botafogo”, utilizando conceitos e fundamentos aprendidos no decorrer do Primeiro ano do Curso Técnico em Eletrônica do CEFET-MG na disciplina teórica de Eletrônica Digital, pelo professor Enderson Neves e conceitos que serão utilizados na disciplina teórica de Introdução a Programação.

O projeto consiste em um sistema dedicado baseado no microcontrolador MSP430G2553 e em um software feito na interface de desenvolvimento IAR, capaz de gerenciar três roletas de um estádio hipotético, exibir em um display usando técnicas de multiplexação, e se comunicar com um software feito no programa LabView, enviando informações sobre preço arrecadado e lotação, e recebendo informações sobre a capacidade do estádio.

Palavras-chave: Roleta, MSP430, Estádio.

**SUMÁRIO**

[1 - INTRODUÇÃO 5](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283918)

[1.1. Motivação 5](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919)

[1.2. Projeto 5](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283920)

[2 - DESENVOLVIMENTO](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283923) 6

[2.1 Fluxograma](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919) 6

[2.2 Descrição do funcionamento](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919) 7

[2.3 Circuito](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919) 7

[2.4 Componentes](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919) 9

[2.4.1 MSP430](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919) 9

[2.4.2 Outros Componentes](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919) 11

[2.4.3 Linguagem C](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919) 11

[2.5 Multiplexação 1](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919)2

[3 – Divisão de Tarefas](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283924) 13

[4 – Conclusão](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283925) 13

[6- REFERÊNCIAS BIBLIOGRÁFICAS](file:///C:\Users\JKL\Desktop\FONTE%20SIMÉTRICA%205.docx#_Toc465283919) 15

**1. INTRODUÇÃO**

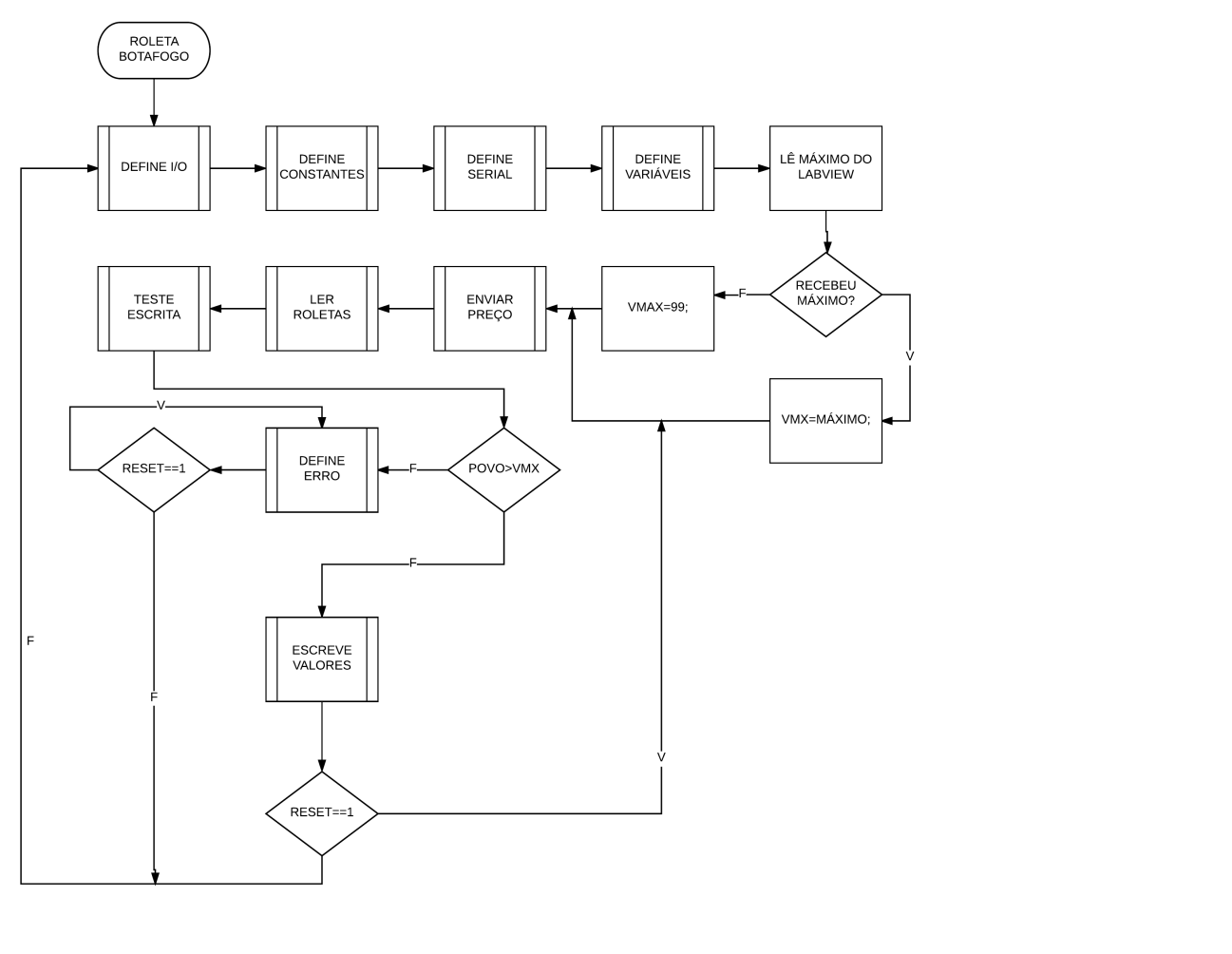
**1.1. MOTIVAÇÃO**

O projeto foi proposto pelo Professor Enderson Neves como parte da pontuação do 3º e 4º bimestre, e será realizado para a obtenção dos pontos nas disciplinas de Introdução a programação (com a produção dos softwares do sistema dedicado) e Laboratório de Práticas Aplicadas (com o projeto, e confecção da placa de circuito impresso para o sistema).

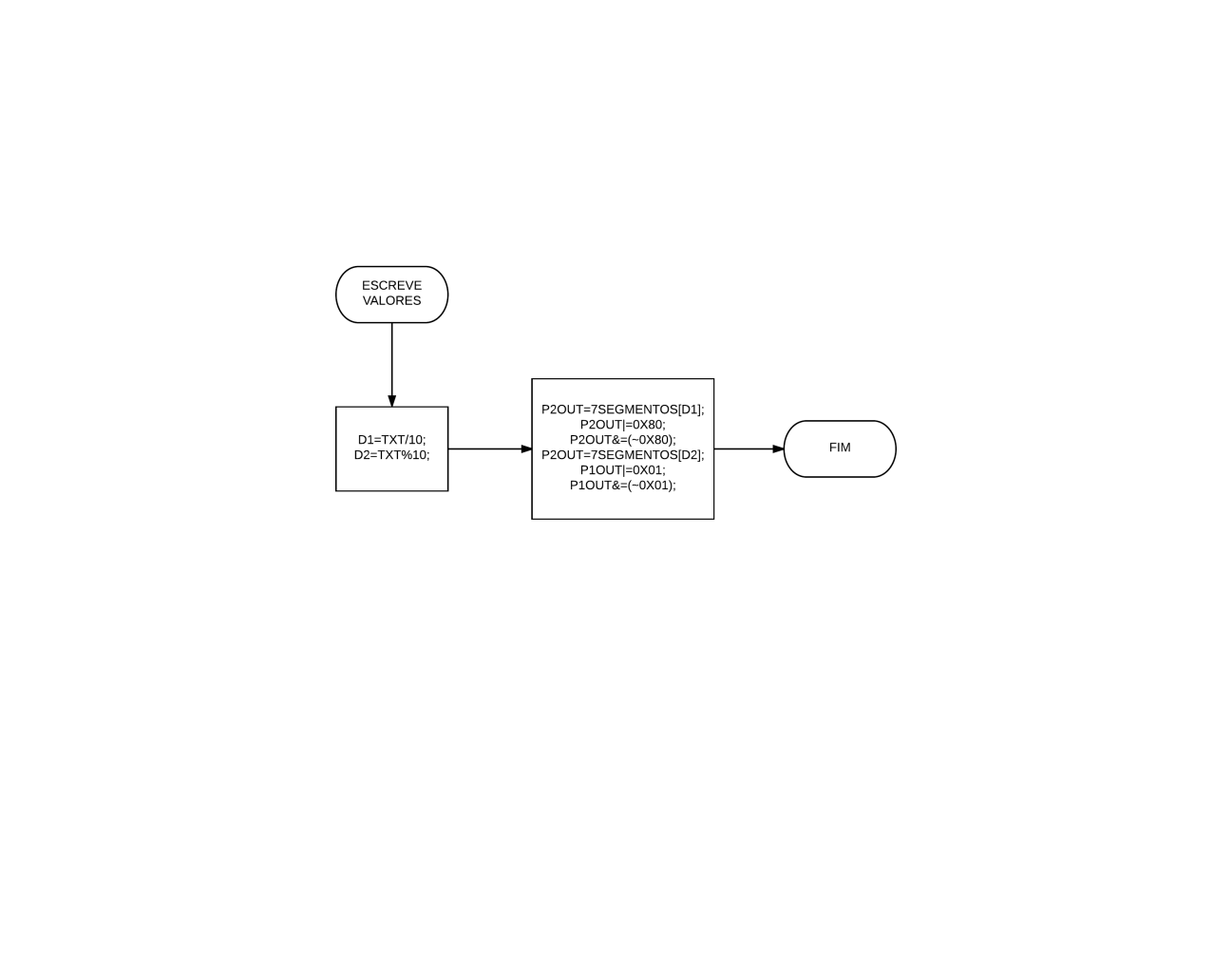
**1.2. O PROJETO**

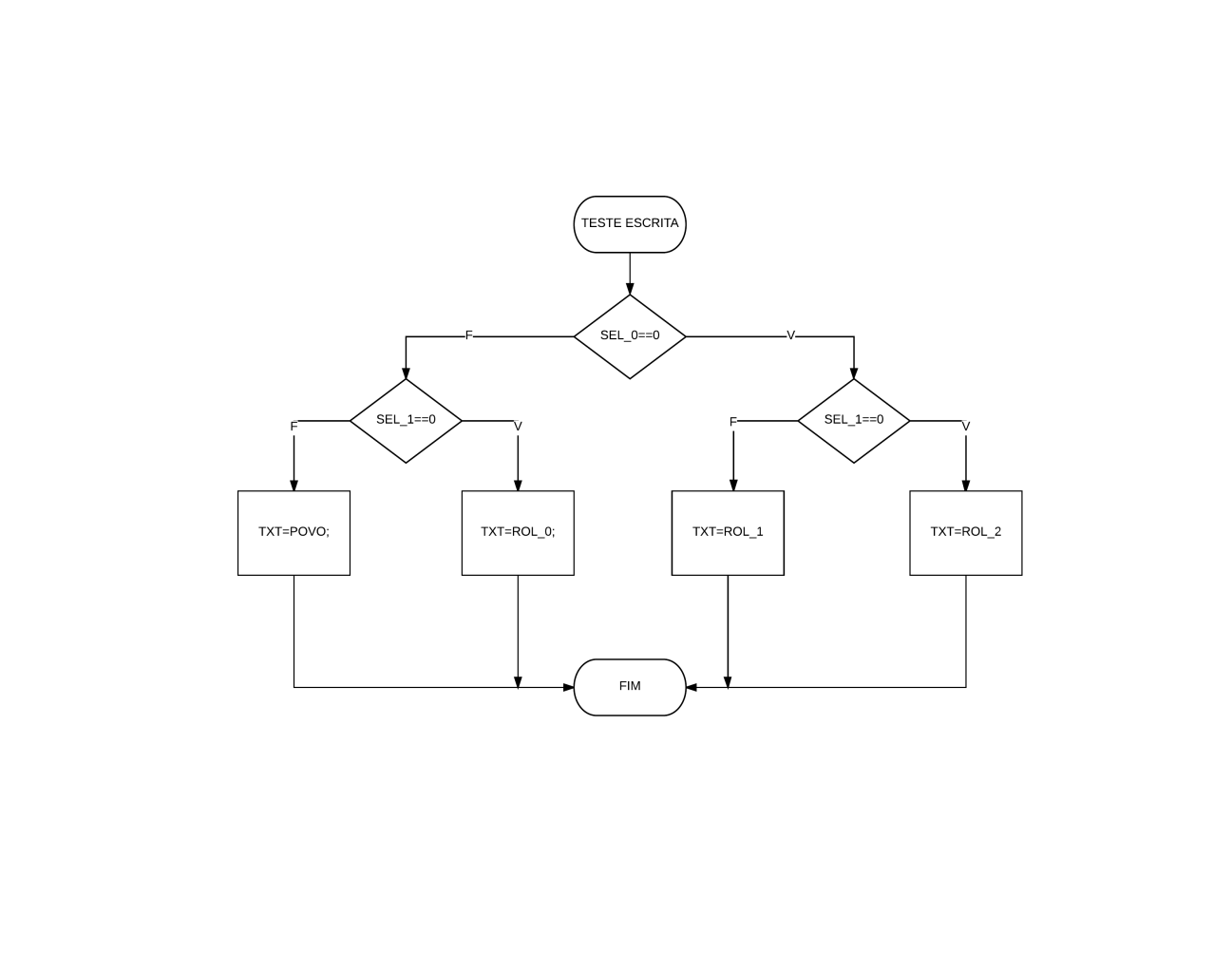
O projeto “Roleta do Botafogo” foi feito utilizando o MSP430 LaunchPad e é basicamente um sistema capaz de gerenciar um estádio hipotético. O mesmo tem a função de verificar 3 roletas (com preços de R$ 7,50; R$ 15,00; R$ 35,80; respectivamente) receber de um computador através de um programa construído em LabView a capacidade máxima até 99 (caso não receba nenhum valor receberá 99 por default). Além disso, a combinação de duas Switches permitirá selecionar entre o público total, e os públicos de cada roleta, para serem exibidos em displays de 7 segmentos utilizando técnicas de multiplexação de display. O sistema ainda é capaz de detectar o estouro da capacidade máxima e exibir um alerta de ERRO no software em LabView e no display.

**2 – DESENVOLVIMENTO**

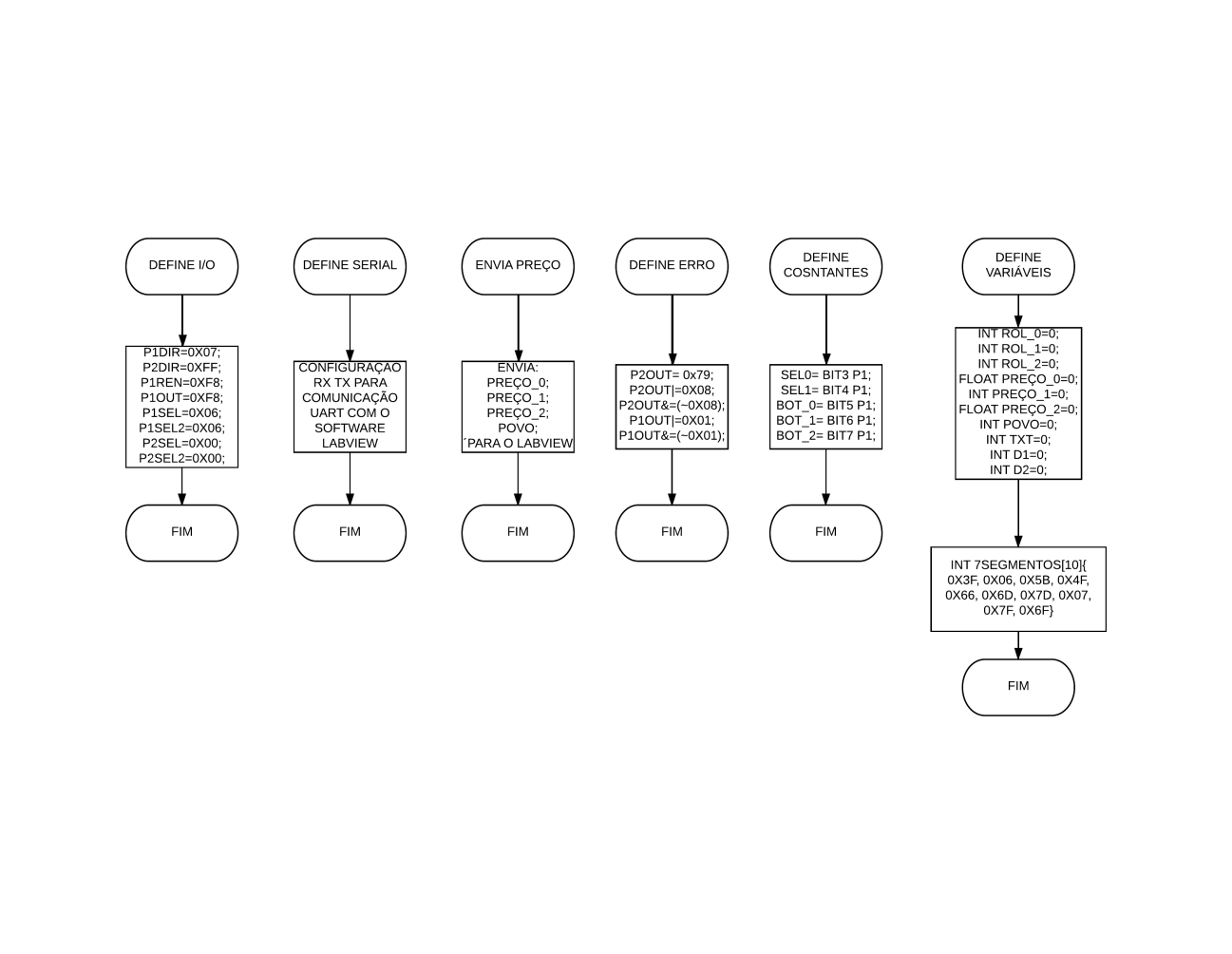
Inicialmente, seguindo as recomendações do Professor foi realizado um fluxograma detalhado do software para na sequência facilitar a produção do software em C no IAR.

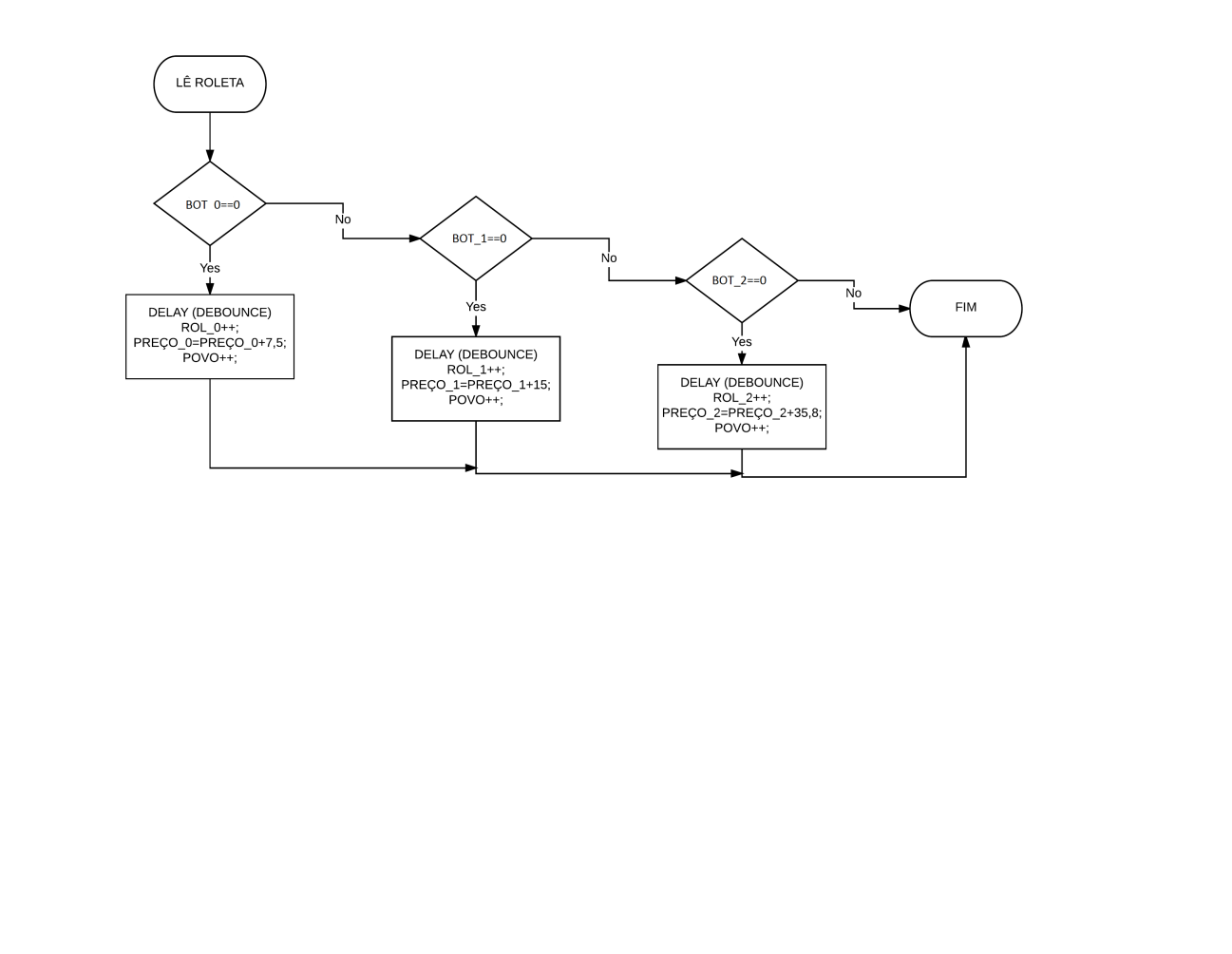
**2.1 Fluxograma**





v





**2.2. Descrição do funcionamento**

O Fluxograma é uma distribuição gráfica do programa e consiste em várias partes:

O programa principal, e suas subfunções.

O programa principal começa com as configurações do microcontrolador, configurando os pinos de I/O, as constantes, as variáveis e as definições da comunicação serial.

Em seguida ele verifica se o LabView enviou algum valor de máximo. Caso tenha recebido, esse passa a ser a capacidade máxima, caso não tenha recebido a capacidade máxima é de 99 por padrão.

Em seguida, o preço atual arrecadado é enviado para o LabView, e se inicia a leitura dos botões.

Então, verificando as combinações das teclas SEL\_0 e SEL\_1 o programa define qual valor será exibido no display (total, pessoas da roleta 0, pessoas da roleta 1 ou pessoas da roleta 2).

Uma verificação de estouro é realizada e caso o estouro tenha sido atingido, os displays mostrarão EE simbolizando Erro de Estouro até que o sistema seja resetado.

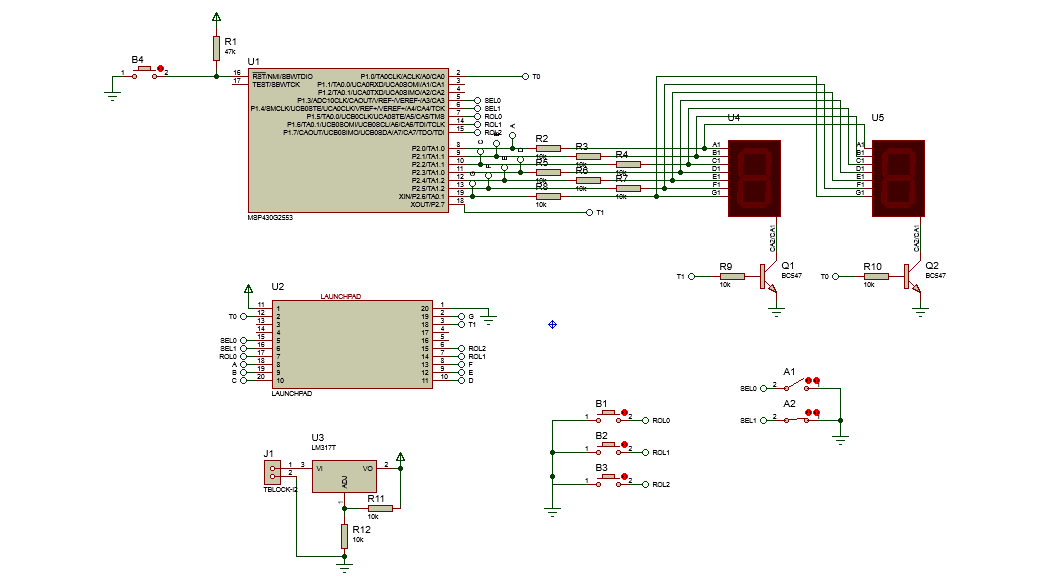
Caso o estouro não tenha sido atingido, o circuito segue escrevendo os valores nos displays.

Dessa forma o sistema entra em loop retornando a parte de enviar os valores para o LabView, ou se reseta e volta ao princípio.

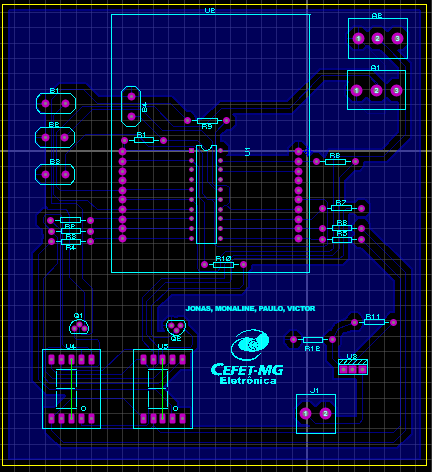
**2.3. Circuito**

Na imagem abaixo está o circuito feito no software Proteus e sua PCB.

**Figura 1 - Circuito**



**Figura 2 – PCB**

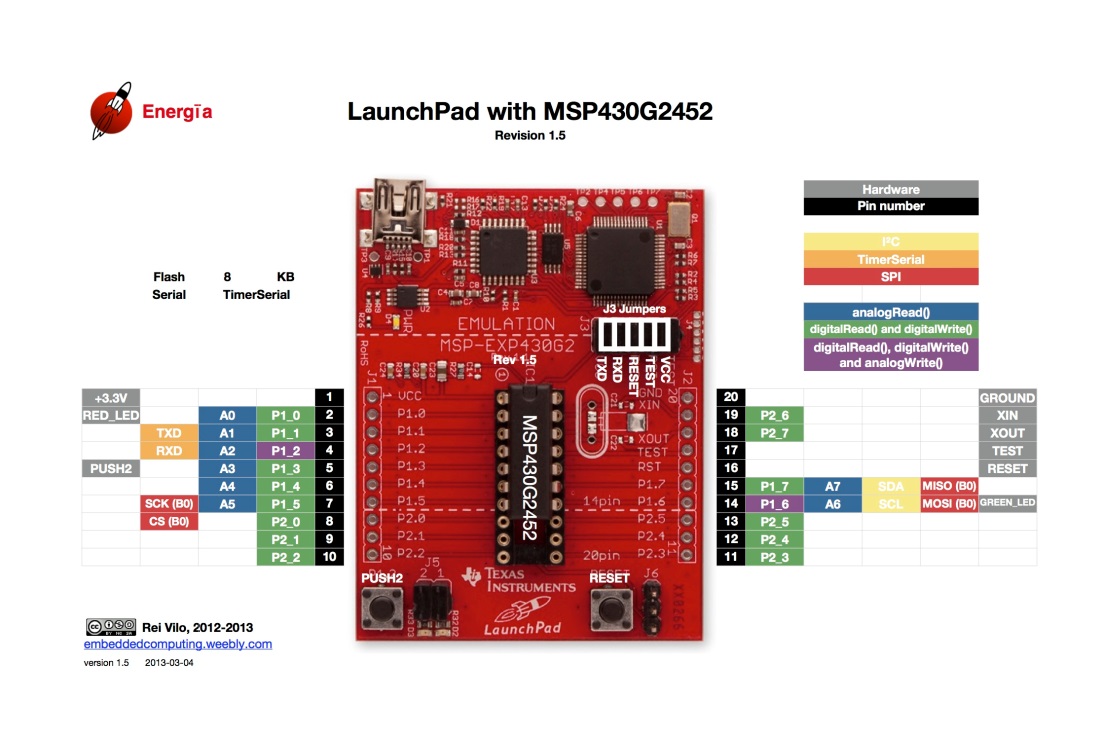


**2.4. Componentes**

**2.4.1 MSP430**

O microcontrolador MSP430G2xx possui 2KB de Flash, 128B RAM, 32khz de velocidade de clock. O MSP também possui 16 bits e socket de 14 pinos para I/O. Há também uma porta USB para conexão com uma fonte externa. Como artigo de comparação, o Arduino UNO tem 32 KB de Flash (mas 2 KB são utilizados para o bootloader),2 KB de SRAM e também uma porta USB. Opera em 16 MHz de clock. Possui 14 pinos que podem ser utilizados como I/O. O UNO precisa de, no mínimo, 5 volts (7v recomendado) contra os 2v do MSP430 (3.6v recomendado). Além do baixo consumo, o MSP pode ser debugado, ou seja, pode se analisar o programa em tempo real ( executando linha por linha) , dependendo do compilador, coisa que não é possível no Arduino. Isso faz com que seja mais fácil encontrar possíveis erros.

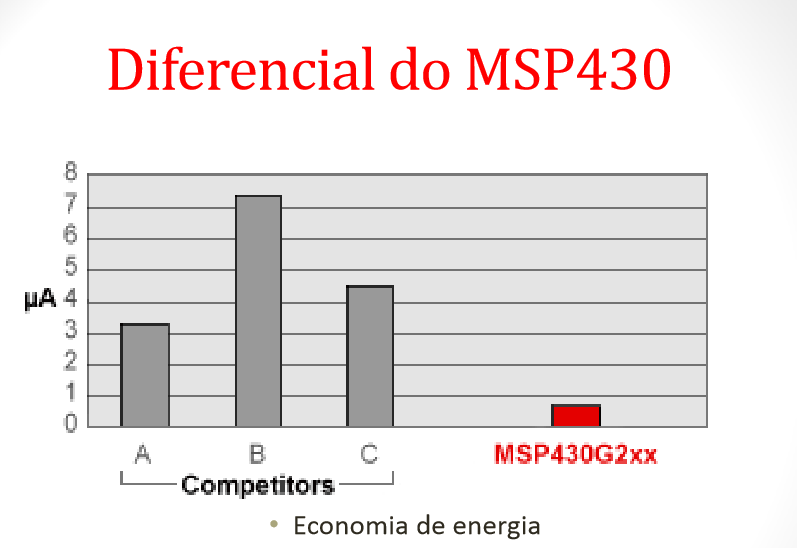
**Figura 4- MSP430 LaunchPad**



**Fonte: energia.nu**

Na figura 5 mostra um gráfico comparando o consumo de energia do MSP430 e de outros concorrentes.

**Figura 5- Consumo de energia**



**Fonte:** **sites2.jf.ifsudestemg.edu.br**

**2.4.2 Outros componentes**

O microcontrolador por si só, é uma ferramenta potente, porém para este projeto, alguns recursos externos foram necessários para o bom funcionamento.

**Resistores**: Componentes cuja função é limitar a corrente no circuito. São utilizados para definir estados lógicos (Pull-up no pino de RESET) e limitação de corrente (resistores de proteção dos LEDS do display), e para o ajuste do regulador de tensão (resistores conectados ao LM317).

**Displays de 7 segmentos:** São leds agrupados em formato de display de forma que é possível gerar todos os caracteres Hexadecimais (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

**LM317:** Regulador de tensão variável que pode ser ajustado com resistores para se obter uma tensão desejada, nesse caso, os resistores de 390 e 240, garantem uma tensão de saída de 3,5V (o necessário para a alimentação do MSP430G2553).

**Transistores BC547:** Transistores funcionam como chaves amplificadoras eletrônicas. Nesse circuito eles são utilizados como chave para ativação dos displays individualmente para a técnica de multiplexação.

**Switchs e Pushbuttons:** São chaves conectadas o microcontrolador ou seja, sensores capazes de enviar comandos para o software, (swtiches tem ação permanente e pushbuttons ação momentânea).

**2.4.3 Linguagem C**

A primeira versão da Linguagem C foi criada por Dennis Ritchie, em 72, nos laboratórios Bell.

O surgimento do C iniciou com a linguagem ALGOL 60. ALGOL era uma linguagem de altíssimo nível, que permitia ao programador trabalhar "longe da máquina", sem se preocupar com os aspectos de como cada comando ou dado era armazenado ou processado.

ALGOL não obteve sucesso, por tentar ser de muito alto nível em uma época em que a maioria dos sistemas operacionais exigiam um grande conhecimento de hardware, tornando o uso da linguagem inviável.

A partir disso, criasse a linguagem C, que é um intermédio do ALGOL e as linguagens de baixo nível, proporcionando o contato direto com o software e o uso comercial.

A linguagem de programação C permite ao programador endereçar a memória,como no Assembly. Outras linguagens, como Java, permitem que o programador programe e ignore a atribuição de memória, que é requisitado muito trabalho. Porém, quando lidamos com tarefas de baixo-nível,tarefas que um núcleo tem que desempenhar, como a de copiar um conjunto de bytes para uma placa de rede, torna-se altamente necessário um acesso direto à memória, o que não é possível fazer com o Java. C tambem pode ser diretamente compilado em código de máquina, o tornando mais rápido e eficiente. Além disso, C permite alterações na forma em que o hardware trabalha, aumentando o seu desempenho.

**2.5. Multiplexação dos displays.**

A multiplexação de displays é uma técnica que economiza energia e pinos de I/O do microcontrolador. Ela funciona da seguinte forma:

Os displays são conectados em paralelo e recebem um sinal do microcontrolador através dos resistores limitadores de correntes. Teoricamente, sempre o mesmo valor seria mostrado nos dois displays, entretanto os displays são acionados separadamente através de dois transistores. Dessa forma para se exibir, por exemplo o número 27, a saída se torna o número 2 e o transistor aciona o display das dezenas. Em seguida, a saída recebe o número 7 e o transistor das unidades é acionado. Essa troca ocorre em alta frequência, impedindo que nossos olhos percebam o que está acontecendo.

Além disso a criação do Vetor 7segmentos[10] facilita a conversão entre decimal para a saída 7 segmentos. Foi desenvolvido uma tabela onde cada número em hexadecimal foi atribuído à uma casa do vetor:

TABELA 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0x3F | 0X06 | 0X5B | 0X4F | 0X66 | 0X6D | 0X7D | 0X07 | 0X7F | 0X67 |

**3. DIVISÃO DE TAREFAS**

A parte inicial do projeto permitiu que os integrantes aprendessem lições sobre o trabalho em grupo e a divisão de tarefas.

A divisão de tarefas foi a seguinte:

Paulo Henrique: Objetivos, Divisão de tarefas, informações sobre os componentes

Jonas José: Conclusões, Resumo, compra dos componentes, contribuições no fluxograma.

Monaline: introdução, conferir e reorganizar os textos dos outros integrantes, montagem do relatório.

Victor Hugo: Construção do fluxograma, simulação e confecção da placa no software Proteus 8.6.

**4. CONCLUSÃO**

Mesmo sendo apenas a primeira etapa pôde-se utilizar vários conhecimentos adquiridos anteriormente em teoria, agora aplicados em prática.

A experiência de desenvolvimento da primeira etapa do projeto foi um pouco conturbada e com alguns empecilhos, como por exemplo o desenvolvimento do fluxograma que foi um pouco complexo, entretanto a confecção do esquema e do layout foram bem tranquilas. Contudo pode-se concluir que o trabalho em equipe é fundamental para resolução do projeto e dos possíveis problemas, e que os conhecimentos utilizados e adquiridos nesse trabalho serão fundamentais para projetos seguintes.

**6. REFERÊNCIAS**

[1] Slides hare < <https://pt.slideshare.net/hvdoria/energia-14814604> > Acesso em 22 de setembro de 2017

[2] Hardware < <http://www.hardware.com.br/analises/msp430-lp-concorrente-arduino/> > Acesso em 22 de setembro de 2017

[3] Microntrolados e platatormas de prototipagem < <https://pt.slideshare.net/zelmasantos2015/microcontroladores-e-plataformas-de-prototipagem>> Acesso em 26 de setembro de 2017

[4] Arduino vs MSP430 <[http://sites2.jf.ifsudestemg.edu.br/sites/default/files/Minicurso%20- %20MSP430.pdf](http://sites2.jf.ifsudestemg.edu.br/sites/default/files/Minicurso%20-%20%20MSP430.pdf)> Acesso em 29 de setembro de 2017

[5] MSP 430 benefícios <<http://www.hardware.com.br/analises/msp430-lp-concorrente-arduino/>> Acesso em 29 de setembro de 2017

[6] Energia Home <energia.nu> Acesso em 30 de setembro de 2017

[7] Texas Instruments Home <http://www.ti.com/> Acesso em 30 de setembro de 2017

[8] Draw.io <http://www.draw.io> acesso em 30 de setembro de 2017