****

**Avy Chueke Pinto Rodolpho Filgueiras**

**Gabriel Vaz Nicolas Fonteyne**

**Pedro Casella Raphael Bomeisel**

**Luca Facciolo**

**Professores: Silvio Szafir, Fabio Ferraz, Francisco Lourenço**

**Projeto Mecatrônico - Sistemas Eletrônicos e Microprocessadores**

**-**

**Documentação Final do Projeto Mecatrônico - Software**

**São Paulo**

**02 de junho de 2017**

Sumário

[DEFINIÇÕES 3](#_Toc484261620)

[BIBLIOTECAS E MODULOS 3](#_Toc484261621)

[ENTRADAS 4](#_Toc484261622)

[VARIAVEIS 5](#_Toc484261623)

[SENSORES 6](#_Toc484261624)

[COR 6](#_Toc484261625)

[DISTÂNCIA 8](#_Toc484261626)

[FUNÇÃO TAP 9](#_Toc484261627)

[REFERENCIAMENTO 10](#_Toc484261628)

[SALVAR POSIÇÃO 11](#_Toc484261629)

[IR PARA POSIÇÃO 11](#_Toc484261630)

[SET HOME 12](#_Toc484261631)

[CALIBRACAO PARA MILÍMETROS 12](#_Toc484261632)

[JOG 13](#_Toc484261633)

[ROTINA AUTOMÁTICA 13](#_Toc484261634)

[MENU 14](#_Toc484261635)

[FUNÇÃO PRINCIPAL 18](#_Toc484261636)

# DEFINIÇÕES

No início do código é necessário definir as variáveis globais que serão usadas ao decorrer do algoritmo

## BIBLIOTECAS E MÓDULOS

Definição das bibliotecas e módulos, junto com o sensor de cor.

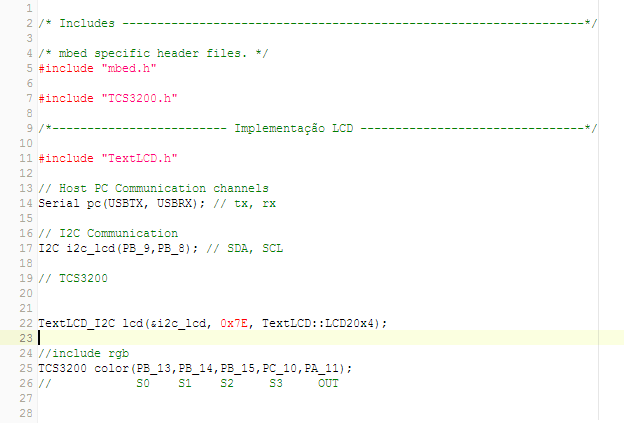


Figura 1 - Definição 1

## ENTRADAS

Definimos as entradas de nosso algoritmo:

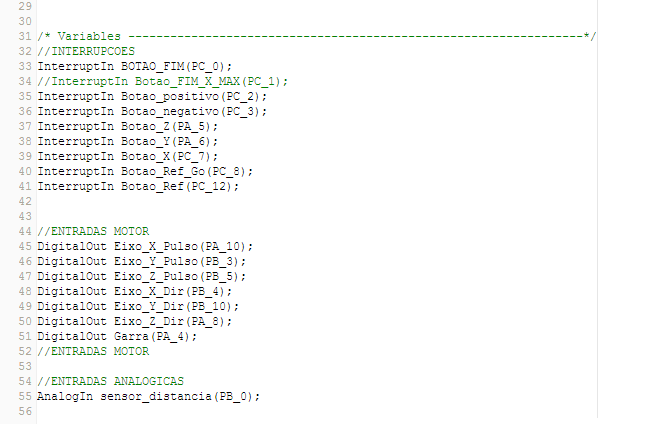


Figura 2 - Definição 2

O botão ref é o nosso botão “enter” e o botão ref go é o nosso “back”. As demais variáveis estão com nomes intuitivos o suficiente para dispensar observações.

## VARIÁVEIS

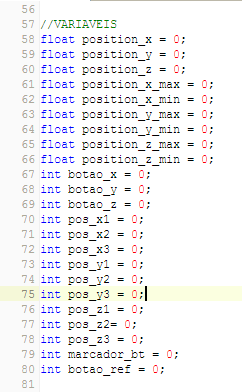
Definimos as variáveis globais

Figura 3 – Definição variáveis globais parte 1

As primeiras variáveis, position x , y e z são usadas para plotar a posição do eixo na tela lcd.

As variáveis position xmax e min são responsáveis por fazer o controle via software dos limites de cada eixo.

As variáveis botão x , y e z são responsáveis por armazenar o valor do botão ( caso o botão x , y ou z seja pressionado o código entenderá que este eixo está selecionado sem a necessidade de manter o botão apertado)

As variáveis pos\_x1 , pos\_x2 e pos\_x3 são responsáveis por salvar as posições.

A variável marcador\_bt é responsável por armazenar qual das posições deve ser salva caso o botão enter seja pressionado.

A variável botão ref foi descontinuada.

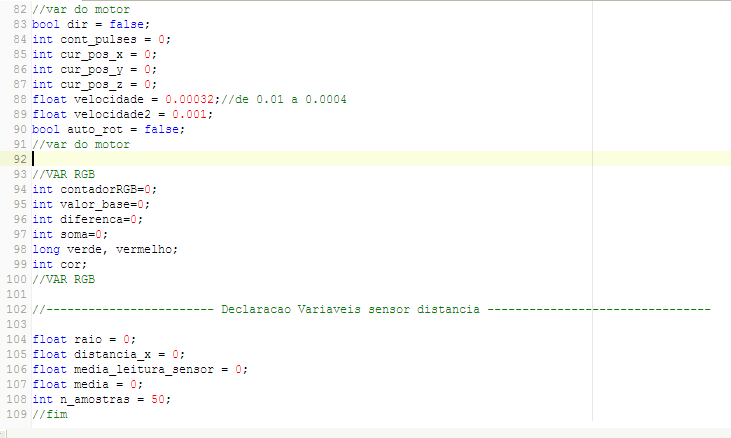


Figura 4 - Definição variáveis globais parte 2

A variável dir é responsável por manter o loop de movimentação do eixo

A variável count\_pulses foi descontinuada.

As variáveis cur\_pos\_x , y e z são responsáveis por armazenar a posição atual do eixo em passos do motor.

As variáveis de velocidade são responsáveis por ditar a frequência de curva de pulsos dada aos motores, quanto menor esta variável for mais rápido o motor irá funcionar.

As variáveis do rgb são responsáveis por fazer a calibração e a partir desta medir a diferença entre a cor medida e a cor calibrada (vermelho).

# SENSORES

Fizemos a calibração dos sensores e as funções para a leitura da cor e diâmetro do tarugo

## COR

Código de calibração do sensor

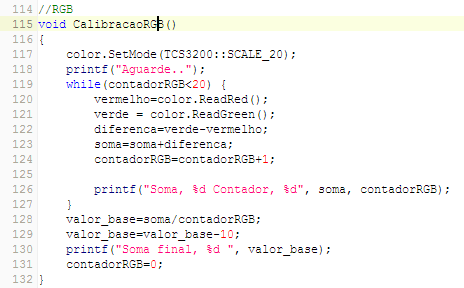


Figura 5 – Calibração

É necessário fazer várias contagens para aumentar a acurácia do sensor. E na parte da calibração é importantíssimo chegarmos num valor confiável para que na hora da medição não haja erros de identificação de cor.

Código de identificação da cor do tarugo

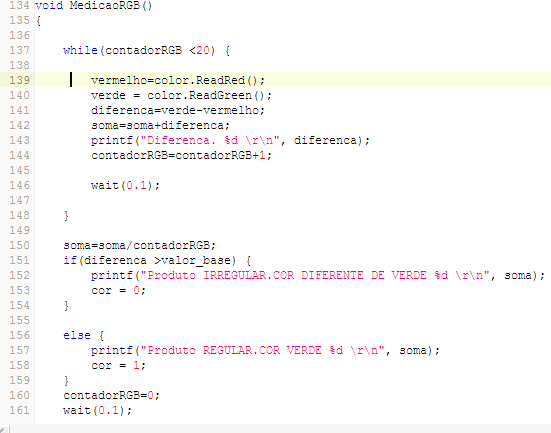


Figura 6 - Identificação da cor

## DISTÂNCIA

Códigos da medição do diâmetro do tarugo

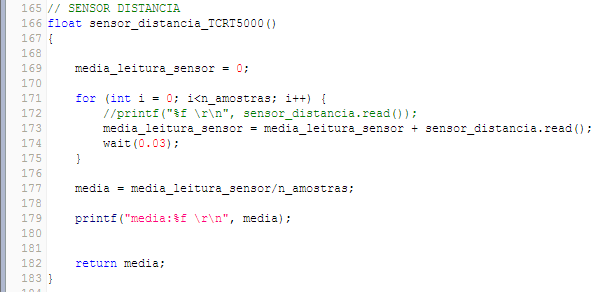


Figura 7 - Leitura das médias

Transformação da leitura para milímetros

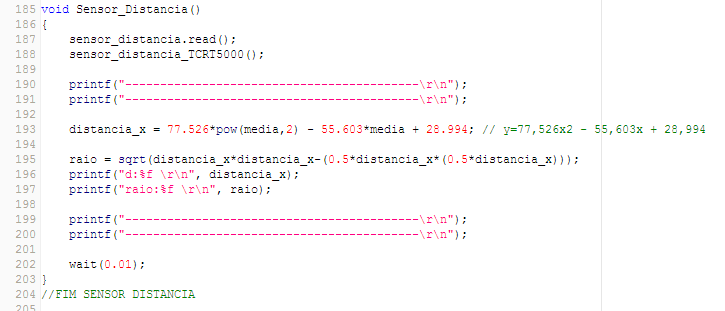


Figura 8 - Calibração do sensor

# FUNÇÃO TAP

Função responsável por fazer o motor se movimentar no sentido contrário após bater no fim de curso.

A lógica para os outros eixos é igual, apenas as variáveis mudam.

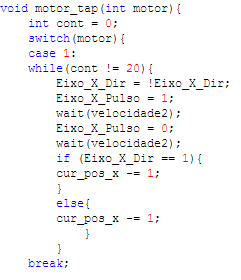


Figura 9 - Função Tap

# REFERENCIAMENTO

Os referenciamentos dos outros eixos são iguais, apenas as variáveis mudam.

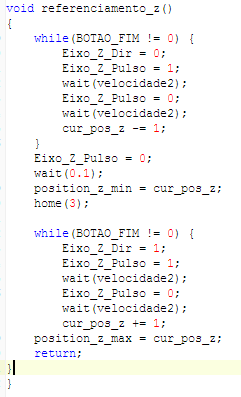


Figura 10 – Referenciamento

# SALVAR POSIÇÃO

Salvar posição é uma sub função do jog.

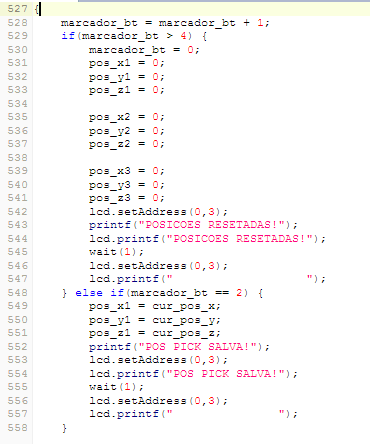


Figura 11 - Salvar posição

# IR PARA POSIÇÃO

Sub função para a rotina automática. Para os outros eixos a lógica é a mesma só há uma mudança nas variáveis. Cada número inteiro dos cases correspondem a uma posição diferente salva (PICK ,PLACE 1 E PLACE 2).

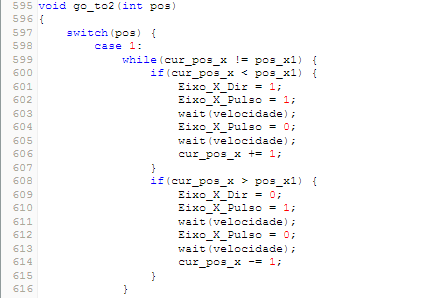


Figura 12 - Salvar posição

# SET HOME

Salvar posição no referenciamento como 0.

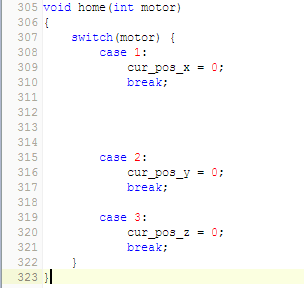


Figura 13- SET HOME

# CALIBRACAO PARA MILÍMETROS

Transforma os passos dados pelo motor de passo em milímetros de acordo com o fuso do eixo.

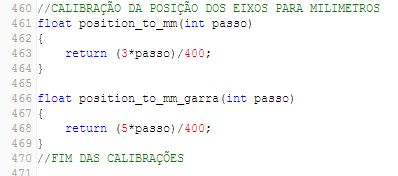


Figura 14 - Calibração para milímetros

# JOG

Movimentação manual do eixo. As outras partes do jog apenas se repetem dependendo do botão pressionado (positivo ou negativo) e do eixo selecionado (X, Y e Z).

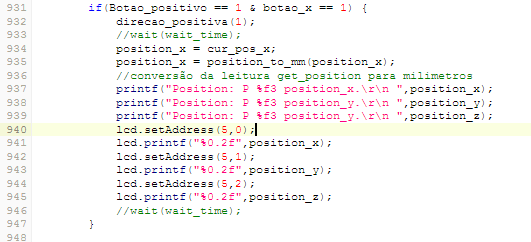


Figura 15 - Logica de movimentação manual

# ROTINA AUTOMÁTICA

A partir de todas as outras funções, e com as posições salvas a lógica da rotina automática pode ser visualizada a seguir.

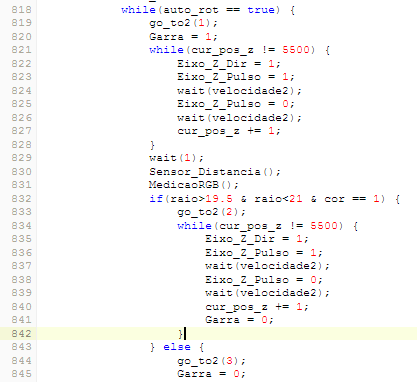


Figura 16 – Rotina automática

# MENU

O menu é a função principal do código, responsável por acionar todas outras funções

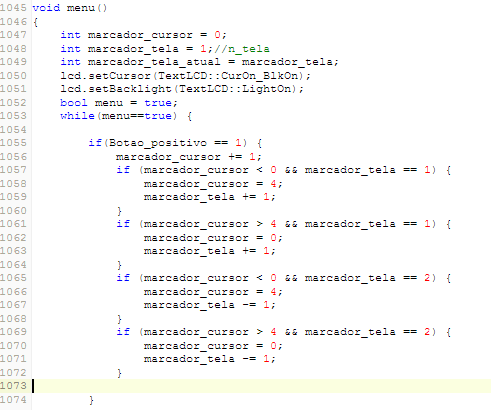


Figura 17 - Movimentação do cursor

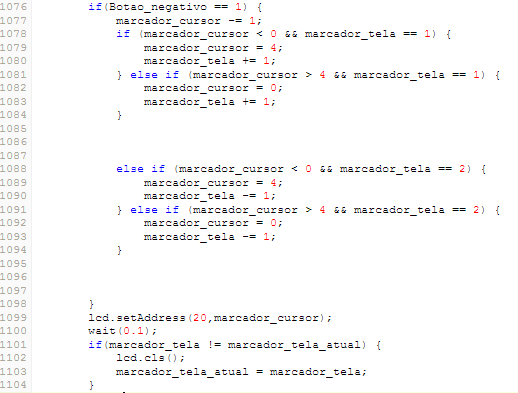


Figura 18 - Movimentação de telas e Limpar tela

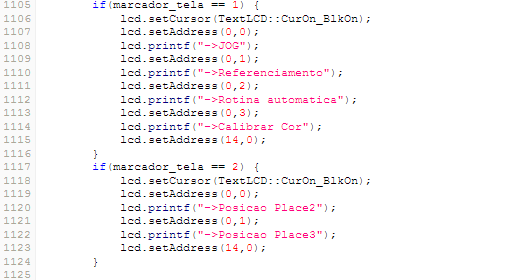


Figura 19 – Definição do que será escrito nas telas

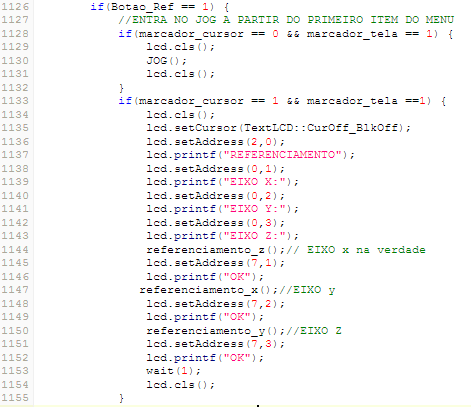


Figura 20 - Acionamento das funções

Os marcadores servem para o código identificar em qual linha do menu o usuário está. A partir daí , ao pressionar o botão “enter” o usuário entra em uma das funções como podemos observar na figura 20.

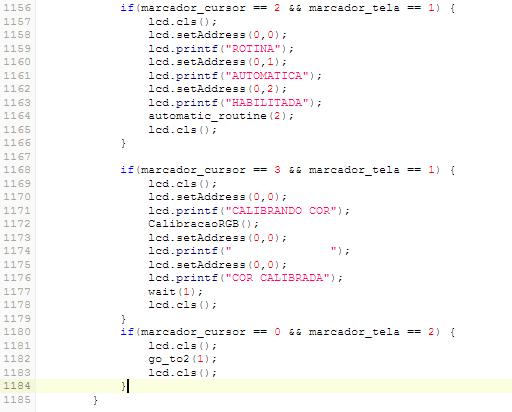


Figura 21 – Acionamento das funções parte2

# FUNÇÃO PRINCIPAL

Função main do código responsável por chamar todas as outras funções

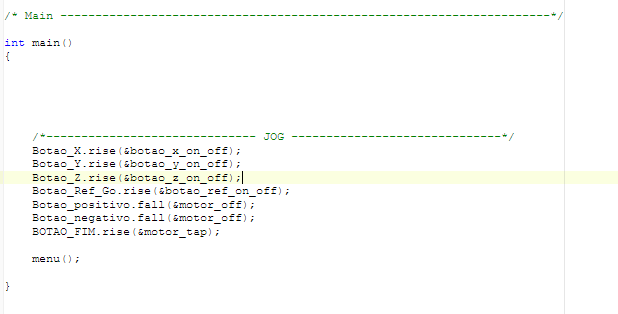


Figura 22- Função main

Optamos por usar botões como interrupção pois na prática isto torna o software mais ágil em entender comandos humanos e executa-los. Caso isto não tivesse sido feito, haveria um delay entre o comando do usuário e a execução deste comando. A função rise é acionada quando o sinal dos botões cuja ela está associada muda de 0 para 1. A função fall tem a lógica inversa da anterior, ou seja, quando o valor do botão muda de 1 para 0.

Como fizemos todos os fins de curso em série foi necessário apenas uma função e uma porta para eles. Por fim, caso haja a curiosidade de ver o código na integra, está disponibilizado logo abaixo um codeshare com o mesmo.

https://codeshare.io/GRUPO2R