Sistema de Balanças Automotivo

Generated by Doxygen 1.9.2

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 main.cpp File Reference

```
#include <Arduino.h>
#include "main.h"
#include <Arduino_FreeRTOS.h>
#include <semphr.h>
#include <queue.h>
#include <task.h>
```

Functions

- void **TaskLeitura** (void *pvParameters)

  include FreeRTOS librarys
- void **TaskUpdateLCD** (void *pvParameters)
- void **setup** ()

  mutex que vai controlar porta serial
- void **loop** ()
- void **readButtons** ()

  read state of each button
- void **printScales** ()

  print data in LCD about individual scales data
- void **printLong** ()

  print data in LCD about longitudinal mass distribution
- void **printLat** ()

  print data in LCD about lateral mass distribution
- void **printTotal** ()

  print data in LCD about total mass of the car
- void **readData** ()

  read analog data and store its in an vector
- void **tara** ()
- void **processData** ()
- int32_t **mediaMovel** (int32_t *array)

  calculate media movel of an vector

Variables

- SemaphoreHandle_t **threadSemaphore**

2.1.1 Function Documentation

2.1.1.1 loop()

```
void loop ( )
```

2.1.1.2 mediaMovel()

```
int32_t mediaMovel (
            int32_t * array )
```

calculate media movel of an vector

2.1.1.3 printLat()

```
void printLat ( )
```

print data in LCD about lateral mass distribution

2.1.1.4 printLong()

```
void printLong ( )
```

print data in LCD about longitudinal mass distribution

2.1.1.5 printScales()

```
void printScales ( )
```

print data in LCD about individual scales data

functions

2.1.1.6 printTotal()

```
void printTotal ( )
```

print data in LCD about total mass of the car

2.1.1.7 processData()

```
void processData ( )
```

process data, calculating media movel, removing tara, and converting to kg

2.1.1.8 readButtons()

```
void readButtons ( )
```

read state of each button

2.1.1.9 readData()

```
void readData ( )
```

read analog data and store its in an vector

2.1.1.10 setup()

```
void setup ( )
```

mutex que vai controlar porta serial

initialize serial communication at 9600 bits per second:

read EEPROM data

cria a mutex

Now set up two tasks to run independently.

2.1.1.11 tara()

```
void tara ( )
```

get the current analog data and store its to EEPROM this value will be subtract from de current data that is being redden

2.1.1.12 TaskLeitura()

```
void TaskLeitura (
            void * pvParameters )
```

include FreeRTOS librarys

define two tasks for read and print data

2.1.1.13 TaskUpdateLCD()

```
void TaskUpdateLCD (
            void * pvParameters )
```

2.1.2 Variable Documentation

2.1.2.1 threadSemaphore

```
SemaphoreHandle_t threadSemaphore
```

2.2 main.h File Reference

```
#include <LiquidCrystal.h>
#include <EEPROM.h>
```

Macros

- #define **FE_PORT** 0

  defines analog inputs pin
- #define **FD_PORT** 1
- #define **TE_PORT** 2
- #define **TD_PORT** 3
- #define **FUNC_PIN** 2

  defines digital inputs pin
- #define **T_PIN** 3
- #define **BUFFER_SIZE** 32

  defines constants of buffer and media movel calculation
- #define **bitshift** 5
- #define **UPDATE_LCD_HZ** 10

Functions

- LiquidCrystal **lcd** (**rs**, **en**, **d4**, **d5**, **d6**, **d7**)
- void **printScales** ()

  functions
- void **printLong** ()

  print data in LCD about longitudinal mass distribution
- void **printLat** ()

  print data in LCD about lateral mass distribution
- void **printTotal** ()

  print data in LCD about total mass of the car
- void **tara** ()
- void **readButtons** ()

  read state of each button
- void **readData** ()

  read analog data and store its in an vector
- void **processData** ()
- int32_t **mediaMovel** (int32_t *array)

  calculate media movel of an vector

Variables

- const int **rs** = 13
- const int **en** = 12
- const int **d4** = 11
- const int **d5** = 10
- const int **d6** = 9
- const int **d7** = 8
- const double **calibrationFactorDe** = 0.48481

  initialize calibration data that converts Voltage (0-1023) to Kg
- const double **calibrationFactorDd** = 0.48481
- const double **calibrationFactorTe** = 0.48481
- const double **calibrationFactorTd** = 0.48481
- byte **state** = 0

  state of LCD (witch function is beeing displayed)
- boolean **funcState** = false

  state of each button
- boolean **tState** = false
- int32_t **dataDe** [**BUFFER_SIZE**]

  data buffers
- int32_t **dataDd** [**BUFFER_SIZE**]
- int32_t **dataTe** [**BUFFER_SIZE**]
- int32_t **dataTd** [**BUFFER_SIZE**]
- int32_t **taraDe** = 0

  data constants
- int32_t **taraDd** = 0
- int32_t **taraTe** = 0
- int32_t **taraTd** = 0
- double **total** = 0
- double **de** = 0
- double **dd** = 0
- double **te** = 0
- double **td** = 0

2.2.1 Macro Definition Documentation

2.2.1.1 bitshift

```
#define bitshift 5
```

2.2.1.2 BUFFER_SIZE

```
#define BUFFER_SIZE 32
```

defines constants of buffer and media movel calculation

2.2.1.3 FD_PORT

```
#define FD_PORT 1
```

2.2.1.4 FE_PORT

```
#define FE_PORT 0
```

defines analog inputs pin

2.2.1.5 FUNC_PIN

```
#define FUNC_PIN 2
```

defines digital inputs pin

2.2.1.6 T_PIN

```
#define T_PIN 3
```

2.2.1.7 TD_PORT

```
#define TD_PORT 3
```

2.2.1.8 TE_PORT

```
#define TE_PORT 2
```

2.2.1.9 UPDATE_LCD_HZ

```
#define UPDATE_LCD_HZ 10
```

2.2.2 Function Documentation

2.2.2.1 lcd()

```
LiquidCrystal lcd (
              rs ,
              en ,
              d4 ,
              d5 ,
              d6 ,
              d7  )
```

2.2.2.2 mediaMovel()

```
int32_t mediaMovel (
             int32_t * array )
```

calculate media movel of an vector

2.2.2.3 printLat()

```
void printLat ( )
```

print data in LCD about lateral mass distribution

2.2.2.4 printLong()

```
void printLong ( )
```

print data in LCD about longitudinal mass distribution

2.2.2.5 printScales()

```
void printScales ( )
```

functions

functions

2.2.2.6 printTotal()

```
void printTotal ( )
```

print data in LCD about total mass of the car

2.2.2.7 processData()

```
void processData ( )
```

process data, calculating media movel, removing tara, and converting to kg

2.2.2.8 readButtons()

```
void readButtons ( )
```

read state of each button

2.2.2.9 readData()

```
void readData ( )
```

read analog data and store its in an vector

2.2.2.10 tara()

```
void tara ( )
```

get the current analog data and store its to EEPROM this value will be subtract from de current data that is being redden

2.2.3 Variable Documentation

2.2.3.1 calibrationFactorDd

```
const double calibrationFactorDd = 0.48481
```

2.2.3.2 calibrationFactorDe

```
const double calibrationFactorDe = 0.48481
```

initialize calibration data that converts Voltage (0-1023) to Kg

2.2.3.3 calibrationFactorTd

```
const double calibrationFactorTd = 0.48481
```

2.2.3.4 calibrationFactorTe

```
const double calibrationFactorTe = 0.48481
```

2.2.3.5 d4

```
const int d4 = 11
```

2.2.3.6 d5

```
const int d5 = 10
```

2.2.3.7 d6

```
const int d6 = 9
```

2.2.3.8 d7

```
const int d7 = 8
```

2.2.3.9 dataDd

```
int32_t dataDd[ BUFFER_SIZE]
```

2.2.3.10 dataDe

```
int32_t dataDe[ BUFFER_SIZE]
```

data buffers

2.2.3.11 dataTd

```
int32_t dataTd[ BUFFER_SIZE]
```

2.2.3.12 dataTe

```
int32_t dataTe[ BUFFER_SIZE]
```

2.2.3.13 dd

```
double dd = 0
```

2.2.3.14 de

```
double de = 0
```

2.2.3.15 en

```
const int en = 12
```

2.2.3.16 funcState

```
boolean funcState = false
```

state of each button

2.2.3.17 rs

```
const int rs = 13
```

2.2.3.18 state

```
byte state = 0
```

state of LCD (witch function is beeing displayed)

2.2.3.19 taraDd

```
int32_t taraDd = 0
```

2.2.3.20 taraDe

```
int32_t taraDe = 0
```

data constants

2.2.3.21 taraTd

```
int32_t taraTd = 0
```

2.2.3.22 taraTe

```
int32_t taraTe = 0
```

2.2.3.23 td

```
double td = 0
```

2.2.3.24 te

```
double te = 0
```

2.2.3.25 total

```
double total = 0
```

2.2.3.26 tState

```
boolean tState = false
```

2.3 main.h

Go to the documentation of this file.

```
1 //
2 // Created by eugen on 26/08/2021.
3 //
4
5 // include the library code:
6 #include <LiquidCrystal.h>
7 #include <EEPROM.h>
8
10 #define FE_PORT 0
11 #define FD_PORT 1
12 #define TE_PORT 2
13 #define TD_PORT 3
14
16 #define FUNC_PIN   2
17 #define T_PIN      3
18
20 #define BUFFER_SIZE 32
21 #define bitshift    5
22 #define UPDATE_LCD_HZ 10
23
24 // initialize the LCD by associating any needed LCD interface pin
25 // with the arduino pin number it is connected to
26 const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
27 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
28
30 const double calibrationFactorDe = 0.48481;
31 const double calibrationFactorDd = 0.48481;
32 const double calibrationFactorTe = 0.48481;
33 const double calibrationFactorTd = 0.48481;
34
36 byte state = 0;
37
39 boolean funcState = false;
40 boolean tState = false;
41
43 int32_t dataDe[BUFFER_SIZE];
44 int32_t dataDd[BUFFER_SIZE];
45 int32_t dataTe[BUFFER_SIZE];
46 int32_t dataTd[BUFFER_SIZE];
47
49 int32_t taraDe = 0;
50 int32_t taraDd = 0;
51 int32_t taraTe = 0;
52 int32_t taraTd = 0;
53
54 double total = 0;
55 double de = 0;
56 double dd = 0;
57 double te = 0;
58 double td = 0;
59
61 void printScales();
62 void printLong();
63 void printLat();
64 void printTotal();
65 void tara();
66
67 void readButtons();
68 void readData();
69 void processData();
70 int32_t mediaMovel(int32_t *array);
```

Index