

BinToDec

LUCAS RISTER, GRÉGORY ROSSATO

Versão 1.0.0

Domingo, 22 de Agosto de 2021

Sumário

Table of contents

Índice dos Arquivos

Lista de Arquivos

Esta é a lista de todos os arquivos e suas respectivas descrições:

C:/Users/Home/Desktop/source/main/main.c pagenum

Arquivos

Referência do Arquivo

C:/Users/Home/Desktop/source/main/main.c

```
#include <LiquidCrystal.h>
#include <stdio.h>
#include <Arduino_FreeRTOS.h>
#include "task.h"
#include <queue.h>
```

Definições e Macros

0) #define **tempoDebounce** 50

Funções

- 0) LiquidCrystal **lcd** (8, 9, 4, 5, 6, 7)
- 1) void **setup** ()
- 2) void **loop** ()
- 3) static void **Input1** (void *pvParameters)
- 4) static void **process1** (void *pvParameters)
- 5) static void **Show1** (void *pvParameters)

Variáveis

- 0) volatile int32_t **result** =0
 - 1) volatile int **decimal** = 0
 - 2) QueueHandle_t **queue**
 - 3) QueueHandle_t **queue2**
 - 4) TaskHandle_t **process**
 - 5) TaskHandle_t **Show**
 - 6) TaskHandle_t **Input**
-

Definições e macros

#define tempoDebounce 50

Funções

static void Input1 (void * *pvParameters*)[static]

```
40                                     { //Task para coletar os btns de entrada
41
42 int state;
43     for(;;){
44         int valBotoes = analogRead(A0);
45
46         if ((valBotoes < 800) && (valBotoes >= 600)) {
47             state = 1; // condicao para o btn SELECT
48             xQueueSend(queue, &state, portMAX_DELAY);
49
50         } else if ((valBotoes < 600) && (valBotoes >= 400)) {
51             state = 2; // condicao para o btn LEFT;
52             xQueueSend(queue, &state, portMAX_DELAY);
53
54         } else if ((valBotoes < 400) && (valBotoes >= 200)) {
55             state = 4; // condicao para o btn DOWN;
56             xQueueSend(queue, &state, portMAX_DELAY);
57
58         } else if ((valBotoes < 200) && (valBotoes >= 60)) {
59             state = 3; // condicao para o btn UP;
60             xQueueSend(queue, &state, portMAX_DELAY);
61
62         } else if (valBotoes < 60) {
63             state = 5; // condicao para o btn RIGHT;
64             xQueueSend(queue, &state, portMAX_DELAY);
65
66         } else {
67             state = 0; // condicao para o btn EMPTY;
68             xQueueSend(queue, &state, portMAX_DELAY);
69         }
70     }
71 }
```

LiquidCrystal lcd (8 , 9 , 4 , 5 , 6 , 7)

void loop ()

```
{
37 //Fluxo acontece nas Tasks
38 }
```

static void process1 (void * *pvParameters*)[static]

```
{
75
76 unsigned long delayBotao; //Delay do debounce
77 int estadoBotaoAnt = 0; //Var temp para o debounce
78 int32_t remainder =0; //var para o resto da divisao do conversor de bin para dec
79 int32_t base =1; // var para a base multiplicativa do conversor de bin para dec
80 int statel; //var que recebe qual btn foi apertado
81 int input =0; //var que determina o que mostrar no display 0 - entrada 1 - conversao
82
83 for(;;){
84     xQueueReceive(queue, &statel, portMAX_DELAY);
85     if ((millis() - delayBotao) > tempoDebounce) {
86         if ((statel != 0) && (estadoBotaoAnt == 0) ) {
87             if (statel == 2){ //se btn LEFT pressionado
88                 result = result*10 + 1; //recebe 1 e pula para o proximo digito
```

```

89         input = 0; // Seta o display para mostrar os dados de entrada
90         xQueueSend(queue2, &input, portMAX_DELAY);
91     }
92     if(state1 == 5){ // se btn RIGHT pressionado
93         result = result*10; // recebe 0 e pula para o proximo digito
94         input = 0; //Seta o display para mostrar os dados de entrada
95         xQueueSend(queue2, &input, portMAX_DELAY);
96     }
97     if(state1 == 3){ //se btn UP pressionado convert Bin to Dec
98         remainder =0;
99         decimal = 0;
100        base =1;
101        while(result > 0){
102            remainder =result % 10;
103            decimal = decimal + remainder* base;
104            result = result/10;
105            base = base*2;
106        }
107        input = 1; // seta o display para mostrar a conversao de bin para dec
108        xQueueSend(queue2, &input, portMAX_DELAY);
109    }
110    delayBotao = millis();
111 }
112 if ((state1 == 0) && (estadoBotaoAnt != 0) ) {
113     delayBotao = millis();
114 }
115 }
116 estadoBotaoAnt = state1;
117 input =0; // Seta o display para mostrar os dados de entrada
118 xQueueSend(queue2, &input, portMAX_DELAY);
119 }
120 }

```

void setup ()

```

{
21     pinMode(10, OUTPUT);
22     digitalWrite(10, HIGH);
23
24     lcd.begin(16, 2);
25     Serial.begin(9600);
26
27     queue = xQueueCreate(1, sizeof(int));
28     queue2 = xQueueCreate(1, sizeof(int));
29
30     xTaskCreate(Input1, "Input" , 128, NULL, 2 , &Input );
31     xTaskCreate(process1, "process" , 128, NULL, 2 , &process );
32     xTaskCreate(Show1, "Show" , 128, NULL, 1 , &Show );
33 }

```

static void Show1 (void * pvParameters)[static]

```

{ // Task para mostrar os dados na tela
123
124     int input1 =0;
125
126     for(;;){
127         xQueueReceive(queue2, &input1, portMAX_DELAY);
128         if(input1 ==0 ){ //Condicao para mostrar os dados da entrada
129             lcd.setCursor(0,0);
130             lcd.print("Input:");
131             lcd.setCursor(2,1);
132             lcd.print(result); //Mostra os dados da entrada
133         }else if(input1 ==1){ //Condicao para mostrar os dados convertidos
134             lcd.clear();
135             lcd.print("Dec:");
136             lcd.setCursor(2,1);
137             lcd.print(decimal); //Mostra a conversao
138             vTaskDelay(pdMS_TO_TICKS(5000)); //Deixa os dados na tela por 5s
139             result =0; //Reseta a var dos dados convertidos
140             lcd.clear(); //Reseta o display

```

```
141     }  
142   }  
143 }
```

Variáveis

volatile int decimal = 0

TaskHandle_t Input

TaskHandle_t process

QueueHandle_t queue

QueueHandle_t queue2

volatile int32_t result =0

TaskHandle_t Show

Sumário

INDEX