

```

1  /* ***** */
2  /* Nome do arquivo:      display7seg.c */
3  /*
4  /* Descricao:      Arquivo contendo a implementacao */
5  /*                  das funcoes de interface do microcontrolador */
6  /*                  com o display de 7 segmentos do kit */
7  /*
8  /* Autores:      Gustavo Lino e Giacomo Dollevedo */
9  /* Criado em:      13/04/2020 */
10 /* Ultima revisao em:      31/07/2020 */
11 /* ***** */
12
13 /* Correções implementadas:
14
15 Pinos foram definidos no arquivo display7seg.h e não no board.h, resolvido declarando
16 as constantes no arquivo board.h;
17 Função tc_installLptmr0 não foi chamada, resolvido implementado dentro da board_init na main;
18 Período em microsegundos e não em milisegundos, resolvido adequando a constante de tempo
19 para 400;*/
20
21 #include "board.h"
22 #include "display7seg.h"
23 #include "lptmr.h"
24
25 //Variaveis para controle dos displays
26
27
28
29
30
31 /* ***** */
32 /* Nome do metodo:      display7Seg_init */
33 /* Descricao:      Inicializa os registradores para funcionamento do D7S */
34 /*
35 /* Parametros de entrada:      n/a */
36 /*
37 /* Parametros de saida:      n/a */
38 /* ***** */
39 void display7seg_init(void){
40
41
42 /* Liberando o Clock para porta C*/
43 SIM_SCGC5 |= PORTC_CLOCK_GATE;
44
45 /* Declarando os pinos como GPIO */
46 PORTC_PCR0 |= uiSetPinAsGPIO; //Segmento A
47 PORTC_PCR1 |= uiSetPinAsGPIO; //Segmento B
48 PORTC_PCR2 |= uiSetPinAsGPIO; //Segmento C
49 PORTC_PCR3 |= uiSetPinAsGPIO; //Segmento D
50 PORTC_PCR4 |= uiSetPinAsGPIO; //Segmento E
51 PORTC_PCR5 |= uiSetPinAsGPIO; //Segmento F
52 PORTC_PCR6 |= uiSetPinAsGPIO; //Segmento G
53 PORTC_PCR7 |= uiSetPinAsGPIO; //Segmento DP
54
55 PORTC_PCR13 |= uiSetPinAsGPIO; //Display 1
56 PORTC_PCR11 |= uiSetPinAsGPIO; //Display 2
57 PORTC_PCR12 |= uiSetPinAsGPIO; //Display 3
58 PORTC_PCR10 |= uiSetPinAsGPIO; //Display 4
59
60 /* Declarando os pinos como Saida*/
61 GPIOC_PDDR |= D7S_GPIO_CONFIG;
62
63 /*Incializa o temporizador de interrupção*/
64

```

```

65 }
66 }
67
68 /* ***** */
69 /* Nome do metodo:      display7seg_writeChar      */
70 /* Descricao:          Escreve uma letra em um D7S      */
71 /*                      */
72 /* Parametros de entrada: ucDisplay -> indica o D7S no qual sera escrito (1 a 4) */
73 /*                      ucValue -> valor indicando a letra que sera escrita */
74 /*                      */
75 /* Parametros de saida:   n/a                      */
76 /* ***** */
77 void display7seg_writeChar(unsigned char ucDisplay, unsigned char ucChar){
78
79     selectDisp(ucDisplay);
80     GPIOC_PDOR &= DISP_CLEAR;
81
82     if(97 <= ucChar && 102 >= ucChar){
83         switch(ucChar){
84             case 'a':
85                 GPIOC_PDOR |= DISP_A;
86                 break;
87             case 'b':
88                 GPIOC_PDOR |= DISP_B;
89                 break;
90             case 'c':
91                 GPIOC_PDOR |= DISP_C;
92                 break;
93             case 'd':
94                 GPIOC_PDOR |= DISP_D;
95                 break;
96             case 'f':
97                 GPIOC_PDOR |= DISP_F;
98                 break;
99         }
100     }
101 }
102
103 /* ***** */
104 /* Nome do metodo:      display7seg_writeSymbol      */
105 /* Descricao:          Escreve um caracter em um D7S      */
106 /*                      */
107 /* Parametros de entrada: ucDisplay -> indica o D7S no qual sera escrito (1 a 4) */
108 /*                      ucValue -> valor indicando o caracter que sera escrito */
109 /*                      */
110 /* Parametros de saida:   n/a                      */
111 /* ***** */
112 void display7seg_writeSymbol(unsigned char ucDisplay, unsigned char ucValue){
113
114     selectDisp(ucDisplay);
115
116     /*CLEAR no display*/
117     /* A mascara mantem o estado dos outros pinos */
118
119     GPIOC_PDOR &= DISP_CLEAR;
120
121     if(ucValue < 20){
122         switch(ucValue%10){
123             case 0:
124                 GPIOC_PDOR |= DISP_0;
125                 break;
126
127             case 1:
128                 GPIOC_PDOR |= DISP_1;
129                 break;
130
131

```

```

132     case 2:
133         GPIOC_PDOR |= DISP_2;
134         break;
135
136     case 3:
137         GPIOC_PDOR |= DISP_3;
138         break;
139
140     case 4:
141         GPIOC_PDOR |= DISP_4;
142         break;
143
144     case 5:
145         GPIOC_PDOR |= DISP_5;
146         break;
147
148     case 6:
149         GPIOC_PDOR |= DISP_6;
150         break;
151
152     case 7:
153         GPIOC_PDOR |= DISP_7;
154         break;
155
156     case 8:
157         GPIOC_PDOR |= DISP_8;
158         break;
159
160     case 9:
161         GPIOC_PDOR |= DISP_9;
162         break;
163     }
164 }
165
166 }
167
168 /* Acendendo o ponto decimal */
169 if(ucValue >= 10 && ucValue <= 20){
170     GPIOC_PDOR |= DISP_DP;
171 }
172
173
174 /* Caso CLEAR do display */
175 else if(ucValue == 21){
176     GPIOC_PDOR &= DISP_CLEAR;
177 }
178
179 /* Caso acenda todos os segmentos do display */
180 else if(ucValue == 22){
181     GPIOC_PDOR |= DISP_ALL;
182 }
183 }
184
185 /* ***** */
186 /* Nome do metodo:      selectDisp                               */
187 /* Descricao:          Seleciona o D7S que sera escrito          */
188 /*                                                              */
189 /* Parametros de entrada: ucDisplay -> indica o D7S no qual sera escrito (1 a 4) */
190 /*                                                              */
191 /* Parametros de saida:   n/a                                     */
192 /* ***** */
193 void selectDisp(unsigned char ucDisplay){
194
195     /* CLEAR, zerando pinos de 10 a 13 (1111 1111 1111 1111 1111 0000 1111 1111) */
196     /* A mascara mantem o estado dos outros pinos */
197     GPIOC_PDOR &= 0xFFFF0FF;
198

```

```
199 switch(ucDisplay){
200     /* Display 1, pino 13*/
201     case 1:
202         GPIOC_PDOR |= 0x000C;
203         break;
204
205     /* Display 2, pino 11*/
206     case 2:
207         GPIOC_PDOR |= 0x000A;
208         break;
209
210     /* Display 3, pino 12*/
211     case 3:
212         GPIOC_PDOR |= 0x000B;
213
214         break;
215
216     /* Display 4, pino 10*/
217     case 4:
218         GPIOC_PDOR |= 0x0009;
219         break;
220
221     default:
222         break;
223 }
224
225
226 }
```