```c
/* ************************************************************************** */
/*                                                                            */
/*   Nome do arquivo:       lcd.c                                             */
/*                                                                            */
/*   Descricao:           Arquivo contendo as implementacoes das funcoes      */
/*                      necessarias para a interface do LCD com o kit         */
/*                                                                            */
/*   Autores:             Gustavo Lino e Giacomo Dollevedo                    */
/*   Criado em:             07/04/2020                                        */
/*   Ultima revisao em:     31/07/2020                                        */
/* ************************************************************************** */

/*Correções implementadas: Pino 3 da GPIO não setado resolvido setando  do Pino 3 como GPIO
na função lcd_initLcd;
No comando "GPIOC_PDOR |= LCD_RS_CMD;" poderia ser considerado valores anteriores, resolvido
zerando a saida da porta C responsável pelo pino RS ao entrar  na função lcd_write2Lcd
inserido na board.h a constante LCD_RS_WAINTG;
Não é alocado memória para a strings usada, resolvido dando malloc.
*/

#include "lcd.h"
#include "board.h"

/* Bibliotecas da linguagem */

#include <string.h>


/* line and columns */
#define LINE0       0U
#define LINE1       1U

#define COLUMN0     0U

#define L0C0_BASE   0x80 /* line 0, column 0 */
#define L1C0_BASE   0xC0 /* line 1, column 0 */
#define MAX_COLUMN  15U

/* ************************************************************************** */
/* Nome do metodo:         lcd_initLcd                                        */
/* Descricao:            Inicializa as funcoes do LCD                         */
/*                                                                            */
/* Parametros de entrada:   n/a                                               */
/*                                                                            */
/* Parametros de saida:     n/a                                               */
/* ************************************************************************** */
void lcd_initLcd(void)
{
    /* pins configured as outputs */

    /* un-gate port clock*/
    SIM_SCGC5 |= PORTC_CLOCK_GATE;


    /* set pin as gpio */
    PORTC_PCR0 |= uiSetPinAsGPIO;
    PORTC_PCR1 |= uiSetPinAsGPIO;
    PORTC_PCR2 |= uiSetPinAsGPIO;
    PORTC_PCR3 |= uiSetPinAsGPIO;
    PORTC_PCR4 |= uiSetPinAsGPIO;
    PORTC_PCR5 |= uiSetPinAsGPIO;
    PORTC_PCR6 |= uiSetPinAsGPIO;
    PORTC_PCR7 |= uiSetPinAsGPIO;
    PORTC_PCR8 |= uiSetPinAsGPIO;
    PORTC_PCR9 |= uiSetPinAsGPIO;

```

```
67
68      /* set pin as digital output */
69
70      GPIOC_PDDR |= uiPin0MaskEnable;
71      GPIOC_PDDR |= uiPin1MaskEnable;
72      GPIOC_PDDR |= uiPin2MaskEnable;
73      GPIOC_PDDR |= uiPin3MaskEnable;
74      GPIOC_PDDR |= uiPin4MaskEnable;
75      GPIOC_PDDR |= uiPin5MaskEnable;
76      GPIOC_PDDR |= uiPin6MaskEnable;
77      GPIOC_PDDR |= uiPin7MaskEnable;
78      GPIOC_PDDR |= uiPin8MaskEnable;
79      GPIOC_PDDR |= uiPin9MaskEnable;
80
81
82      // turn-on LCD, with no cursor and no blink
83      lcd_sendCommand(CMD_NO_CUR_NO_BLINK);
84
85      // init LCD
86      lcd_sendCommand(CMD_INIT_LCD);
87
88      // clear LCD
89      lcd_sendCommand(CMD_CLEAR);
90
91      // LCD with no cursor
92      lcd_sendCommand(CMD_NO_CURSOR);
93
94      // cursor shift to right
95      lcd_sendCommand(CMD_CURSOR2R);
96
97  }
98
99
100
101 /* ************************************************************************** */
102 /* Nome do metodo:          lcd_write2Lcd                                     */
103 /* Descricao:               Escreve caracter no LCD                           */
104 /*                                                                            */
105 /* Parametros de entrada:   ucBuffer    -> char do dado que sera enviado      */
106 /*                          cDataType   -> commando (LCD_RS_CMD) ou dado      */
107 /*                                         (LCD_RS_DATA)                      */
108 /* Parametros de saida:     n/a                                               */
109 /* ************************************************************************** */
110 void lcd_write2Lcd(unsigned char ucBuffer,  unsigned char cDataType)
111 {
112     //Vamos colocar apenas o pino RS como indefinido (em zero)
113     GPIOC_PDOR &= LCD_RS_WAITING;
114
115     /* writing data or command */
116     if(LCD_RS_CMD == cDataType)
117         /* will send a command */
118         GPIOC_PDOR |= LCD_RS_CMD;
119     else
120         /* will send data */
121         GPIOC_PDOR |= LCD_RS_DATA;
122
123     /*Zera as portas de dados que sera utilizada e insere o valor binario do caracter*/
124     GPIOC_PDOR &= 0xFFFFFF00;
125     GPIOC_PDOR |= ucBuffer;
126
127     /* enable, delay, disable LCD */
128     /* this generates a pulse in the enable pin */
129
130     GPIOA_PDOR |= LCD_ENABLED;
131     util_genDelay1ms();
132     GPIOA_PDOR &= LCD_DISABLED;
133     //util_genDelay1ms();
```

```
134     //util_genDelay1ms();
135 }
136
137
138
139 /* ********************************************************************** */
140 /* Nome do metodo:          lcd_writeData                                 */
141 /* Descricao:               Escreve um dado no LCD                        */
142 /*                                                                        */
143 /* Parametros de entrada:   Um unsigned char que serÃƒÂ¡ escrito           */
144 /*                                                                        */
145 /* Parametros de saida:     n/a                                           */
146 /* ********************************************************************** */
147 void lcd_writeData(unsigned char ucData)
148 {
149     /* just a relay to send data */
150     lcd_write2Lcd(ucData, LCD_RS_DATA);
151 }
152
153
154
155 /* ********************************************************************** */
156 /* Nome do metodo:          lcd_sendCommand                               */
157 /* Descricao:               Escreve um comando no LCD                     */
158 /*                                                                        */
159 /* Parametros de entrada:   Um unsigned char descrevendo o comando que sera feito */
160 /*                                                                        */
161 /* Parametros de saida:     n/a                                           */
162 /* ********************************************************************** */
163 void lcd_sendCommand(unsigned char ucCmd)
164 {
165     /* just a relay to send command */
166     lcd_write2Lcd(ucCmd, LCD_RS_CMD);
167 }
168
169
170
171 /* ********************************************************************** */
172 /* Nome do metodo:          lcd_setCursor                                 */
173 /* Descricao:               Move o cursor no LCD para uma posicao especifica */
174 /*                                                                        */
175 /* Parametros de entrada:   Dois unsigned char, contendo a linha (cLine) e coluna */
176 /*                          (cColumn) para onde o cursor sera movido no display */
177 /*                                                                        */
178 /* Parametros de saida:     n/a                                           */
179 /* ********************************************************************** */
180 void lcd_setCursor(unsigned char cLine, unsigned char cColumn)
181 {
182     char cCommand;
183
184     if(LINE0 == cLine)
185         /* line 0 */
186         cCommand = L0C0_BASE;
187     else
188         /* line 1 */
189         cCommand = L1C0_BASE;
190
191     /* maximum MAX_COLUMN columns */
192     cCommand += (cColumn & MAX_COLUMN);
193
194     // send the command to set the cursor
195     lcd_sendCommand(cCommand);
196 }
197
198
199 /* ********************************************************************** */
200 /* Nome do metodo:          lcd_dummyText                                 */
```

```
/* Descricao:              Escreve um texto padrao no LCD                    */
/*                                                                           */
/* Parametros de entrada:   n/a                                              */
/*                                                                           */
/* Parametros de saida:     n/a                                              */
/* ************************************************************************* */
void lcd_dummyText(void)
{
   // clear LCD
   lcd_sendCommand(CMD_CLEAR);

   // set the cursor line 0, column 1
   lcd_setCursor(LINE0,1);

   // send string
   lcd_writeString("*** ES670 ***");

   // set the cursor line 1, column 0
   lcd_setCursor(1,0);
   lcd_writeString("Prj Sis Embarcad");
}


/* ************************************************************************* */
/* Nome do metodo:         lcd_writeString                                   */
/* Descricao:              Escreve uma string no LCD                         */
/*                                                                           */
/* Parametros de entrada:   Um array dinamico de char, contendo a string que sera   */
/*                         escrita                                           */
/*                                                                           */
/* Parametros de saida:     n/a                                              */
/* ************************************************************************* */
void lcd_writeString(const char *cBuffer){
   while(*cBuffer){
      lcd_writeData(*cBuffer++);
   }
}

/* ************************************************************************* */
/* Nome do metodo:         lcd_writeText                                     */
/* Descricao:              Escreve um texto especifico em uma das duas linhas      */
/*                        do LCD                                             */
/*                                                                           */
/* Parametros de entrada:   Uma string contendo o texto a ser escrito e um inteiro  */
/*                         indicando a linha (0 ou 1) do LCD para escrita          */
/*                                                                           */
/* Parametros de saida:     n/a                                              */
/* ************************************************************************* */
void lcd_writeText(const char *cBuffer, int iLine)
{

   int ilen = strlen(cBuffer);
   char *cLine1, *cLine2;
   cLine1 = (char*)malloc(sizeof(char) * 16);
   cLine2 = (char*)malloc(sizeof(char) * 16);
   // clear LCD
   lcd_sendCommand(CMD_CLEAR);
   // identifica a linha desejada
   if(0 == iLine)
      lcd_setCursor(LINE0,1);
   else
      lcd_setCursor(LINE1,1);

   // send string


   if(ilen < 17){
```

```
268         lcd_writeString(cBuffer);
269     }
270 
271     else if(ilen < 37){
272         strncpy(cLine1, cBuffer, 16);
273         strcpy(cLine2, &cBuffer[17]);
274         lcd_writeString(cLine1);
275 
276         lcd_setCursor(LINE1,1);
277         lcd_writeString(cLine1);
278 
279     }
280 
281     else{
282         lcd_setCursor(LINE0,1);
283         lcd_writeString("Too Many Car");
284 
285     }
286 
287 
288 }
```