**FUNÇÕES DA BIBLIOTECA DE LINGUAGEM C DA DISCIPLINA TEEE – SE**

**void led(int led, int comando)** → Função que aciona um LED da placa principal. Descrição dos parâmetros:

int led – Numero de 1 a 8 que indica qual LED se deseja operar;

int comando – Parâmetro que indica a operação desejada. 0: desliga o LED selecionado, 1: liga o LED selecionado e 2: comuta o estado do LED selecionado.

**void leds(unsigned char byte)** → Função que aciona os LEDs da placa principal de acordo com o byte passado como parâmetro. Cada bit representa um LED na placa principal, sendo que o MSB (bit mais significativo) representa o primeiro LED de baixo para cima na placa e o LSB (bit menos significativo) representa o último LED. O valor de cada bit indica o comando para o LED correspondente, sendo: bit 0 – desligar e bit 1 – ligar.

**int button(int botao)** → Função que realiza a leitura do estado lógico de um botão da placa principal especificado pelo parâmetro ‘int botao’ (valor de 1 a 8). A função retorna 1 se o botão estiver pressionado e 0 caso contrário.

**unsigned char buttons()** → Função que realiza a leitura do estado atual dos botões da placa principal. Retorna um byte no qual cada bit representa o estado de um botão, sendo: 1 – pressionado e 0 – não pressionado. O MSB do byte representa o primeiro botão de baixo para cima na placa, e o LSB representa o último botão.

**int potenciometro(int channel)** → Função que realiza a leitura do potenciômetro selecionado por ‘int channel’ (0 ou 1). Retorna um valor entre 0 e 100.

**int poti2c(int channel)** → Função que realiza a leitura do potenciômetro selecionado por ‘int channel’ (0 ou 1). Retorna um valor entre 0 e 100.

**void buzzer(int estado) →** Função que aciona o buzzer da placa principal de acordo com o inteiro ‘estado’ passado como parâmetro:

- estado = 0 → Desligar;

- estado = 1 → Ligar;

**void mapSemaforo(int semaf, char cor)** → Função que aciona os semáforos da placa de semáforos. A descrição dos parâmetros de entrada é feita a seguir:

- int semaf: seleção do semáforo na placa (número de 1 a 10);

- char cor: ligar o LED correspondente à cor selecionada, sendo: ‘R’ (vermelho), ‘G’ (verde) e ‘Y’ (amarelo).

**void ledSemaforo(int led, int comando)** → Função que aciona um LED na placa de semáforos. O parâmetro ‘int led’ seleciona o LED desejado (número de 1 a 30) e o parâmetro ‘int comando’ seleciona a operação desejada, sendo: 1 → Ligar, 0 → Desligar.

**void iniciarSemaforo(int modo)** → Função que liga (int modo = 1) ou desliga (int modo = 0) todos os LEDs da placa de semáforos.

**void LCD\_init()** → Função que inicia o LCD, liga o backlight, limpa todos os caracteres da tela e move o cursor para a posição inicial.

**void LCD\_write(unsigned char str[80])** → Função que escreve a string ‘str’ na tela do LCD. OBS.: O caractere ‘\n’ é suportado.

**void comandoLCD(unsigned char\* comandos)** → Realiza comandos I2C no LCD. O parâmetro ‘unsigned char\* comandos’ é um vetor que contém comandos para o LCD conforme tabela disponível no link: <<http://www.robot-electronics.co.uk/htm/Lcd05tech.htm>>. O envio de comandos para esta função deve seguir o seguinte protocolo:

- A última posição do vetor deve conter o valor ‘;’ (ponto-e-virgula entre aspas simples);

**unsigned char LCD\_read()** → Realiza a leitura do keypad (teclado matricial). O valor retornado conforme é pressionada cada tecla é mostrado nas correspondências a seguir:

* tecla 1 → ‘1’;
* tecla 2 → ‘2’;
* tecla 3 → ‘3’;
* tecla 4 → ‘4’;
* tecla 5 → ‘5’;
* tecla 6 → ‘6’;
* tecla 7 → ‘7’;
* tecla 8 → ‘8’;
* tecla 9 → ‘9’;
* tecla 0 → ‘0’;
* tecla \* → ‘ ‘;
* tecla # → 8;
* nenhuma tecla → 0;

**void set\_timer()** → Inicia a contagem do temporizador.

**int get\_timer(long int usec)** → Verifica se o temporizador iniciado previamente por ‘set\_timer’ atingiu o tempo especificado por ‘long int usec’ em microssegundos. ATENÇÃO: O valor de ‘usec’ deve ser, necessariamente, menor que 1 segundo (1.000.000 microssegundos).