

```

1  /* ***** */
2  /* */
3  /* Nome do arquivo:    ledSwi.h */
4  /* */
5  /* Descrição:        Arquivo Header contendo a declaração */
6  /*                   das funções de atuação do microcontrolador */
7  /*                   sobre os LEDs e chaves do kit */
8  /* */
9  /* Autores:          Gustavo Lino e Giacomio Dollevedo */
10 /* Criado em:        29/03/2020 */
11 /* Última revisão em: 03/04/2020 */
12 /* ***** */
13 #ifndef SOURCES_LEDSWI_
14
15 #define SOURCES_LEDSWI_
16
17 /* ***** */
18 /* Nome do metodo:    ledSwi_init */
19 /* Descrição:        Inicializa os GPIO como entrada (botao) ou saída */
20 /*                   (LED) */
21 /* */
22 /* Parametros de entrada: Quatro variaveis chareanas que indicam se sera */
23 /*                   inicializado como entrada (false) ou saída (true) */
24 /* */
25 /* Parametros de saída:  n/a */
26 /* ***** */
27 void ledSwi_init(char led1, char led2, char led3, char led4);
28
29
30
31
32 /* ***** */
33 /* Nome do metodo:    readSwitch */
34 /* Descrição:        Le o status de um switch para saber se o mesmo */
35 /*                   está pressionado ou não */
36 /* */
37 /* Parametros de entrada: Um inteiro (0<n<5) que indica qual botão será lido */
38 /*                   inicializado como entrada (botao) ou saída (LED) */
39 /*                   0 -> Leitura PTA1; 1 -> Leitura PTA2, */
40 /*                   2 -> Leitura PTA4; 3 -> Leitura PTA5; */
41 /* */
42 /* Parametros de saída:  Um chareano indicando se o botao lido está sendo */
43 /*                   pressionado (true), se não está ou se é inválido */
44 /*                   (false) */
45 /* ***** */
46 char readSwitch(int n);
47
48
49
50 /* ***** */
51 /* Nome do metodo:    writeLED */
52 /* Descrição:        Liga ou desliga o LED selecionado conforme as */
53 /*                   entradas */
54 /* */
55 /* Parametros de entrada: Um inteiro (0<n<5) indicando sobre qual LED sera */
56 /*                   efetuado o comando; */
57 /*                   Um chareano (status) indicando se o LED sera */
58 /*                   aceso (true) ou apagado (false) */
59 /* */
60 /* Parametros de saída:  n/a */
61 /* ***** */
62 void writeLED(int n, char status);
63
64
65
66 /* ***** */

```

```

67  /* Nome do metodo:      turnOnLED      */
68  /* Descrição:          Liga um LED especificado pela entrada      */
69  /*                      */
70  /* Parametros de entrada: Um inteiro (0<n<5) indicando qual LED sera aceso */
71  /*                      */
72  /* Parametros de saida:   n/a          */
73  /* ***** */
74  void turnOnLED(int n);
75
76
77
78
79  /* ***** */
80  /* Nome do metodo:      turnOffLED      */
81  /* Descrição:          Desliga um LED especificado pela entrada      */
82  /*                      */
83  /* Parametros de entrada: Um inteiro (0<n<5) indicando qual LED sera apagado */
84  /*                      */
85  /* Parametros de saida:   n/a          */
86  /* ***** */
87  void turnOffLED(int n);
88
89
90
91  /* ***** */
92  /* Nome do metodo:      toggleLED      */
93  /* Descrição:          Inverte o status atual de um LED especificado pela */
94  /*                      entrada      */
95  /*                      */
96  /* Parametros de entrada: Um inteiro (0<n<5) indicando qual LED tera seu */
97  /*                      status invertido      */
98  /*                      */
99  /* Parametros de saida:   n/a          */
100 /* ***** */
101 void toggleLED(int n);
102
103
104
105 #endif

```