

```

1  /* ***** */
2  /* */
3  /* Nome do arquivo:    communicationStateMachine.h */
4  /* */
5  /* Descricao:          Arquivo contendo a biblioteca das funcoes */
6  /* de que implementam uma */
7  /* maquina de estados para comunicacao UART */
8  /* */
9  /* Autores:            Gustavo Lino e Giacomo Dollevedo */
10 /* Criado em:           21/05/2020 */
11 /* Ultima revisao em:   25/07/2020 */
12 /* ***** */
13
14
15 #ifndef _COMMUNICATIONSTATEMACHINE_H
16 #define _COMMUNICATIONSTATEMACHINE_H
17
18
19
20
21 /* ***** */
22 /* Nome do metodo:      returnParam */
23 /* Descricao:           Imprime no terminal de comunicacao UART os parametros */
24 /* solicitados pelo comando Get */
25 /* */
26 /* Parametros de entrada: ucParam -> Parametro solicitado (de acordo com dicionario)*/
27 /* */
28 /* Parametros de saida:  n/a */
29 /* ***** */
30 void returnParam(unsigned char ucParam);
31
32
33 /* ***** */
34 /* Nome do metodo:      setParam */
35 /* Descricao:           Define valores de controle/usabilidade necessarios para */
36 /* garantir a interface e funcionamento adequado do uC: */
37 /* Temperatura Máxima */
38 /* Disponibilidade dos botoões */
39 /* */
40 /* Parametros de entrada: ucParam -> Parametro que sera alterado */
41 /* *ucValue -> Array com valores de alteracao */
42 /* */
43 /* Parametros de saida:  n/a */
44 /* ***** */
45 void setParam(unsigned char ucParam, unsigned char* ucValue);
46
47 /* ***** */
48 /* Nome do metodo:      processByteCommUART */
49 /* Descricao:           Realiza todos os processos para que a comunicacao UART */
50 /* ocorra, baseado numa maquina de estados */
51 /* */
52 /* Parametros de entrada: ucCmdByte-> Comandos em Bytes enviados por UART */
53 /* */
54 /* Parametros de saida:  n/a */
55 /* ***** */
56 void processByteCommUART(unsigned char ucCmdByte);
57
58
59
60 #endif /* _COMMUNICATIONSTATEMACHINE_H */

```