

Annexe S.6

```

1  /* ***** */
2  /** Descriptive File Name
3
4      @Company
5          ETML-ES
6
7      @File Name
8          gestionBatterie.h
9
10     @Auteurs
11         - Perret Mélissa
12
13     @Description
14         Fonctions liées à la gestion de la batterie
15     */
16 /* ***** */
17
18
19
20 #ifndef _GESTION_BATTERIE_H
21 #define _GESTION_BATTERIE_H
22
23
24 /* ***** */
25 /* ***** */
26 /* Section: Included Files */
27 /* ***** */
28 /* ***** */
29
30 /* This section lists the other files that are included in this file.
31  */
32
33 #include "main.h" // pour Etat
34
35 #ifdef __cplusplus
36 extern "C" {
37 #endif
38
39 // *****
40 // *****
41 // Section: Global Data
42 // *****
43 // *****
44
45 // Structure pour stocker les informations liée à la batterie
46 typedef struct {
47     uint32_t compteurCheckBatterie;
48     Etat etatBatterie;
49 } InfoBatterie;
50
51 // Valeurs des resistances utilisées pour le pont diviseur de tension
52 #define R3_THEORIQUE 18000 // Valeur théorique de la résistance
53 #define R6_THEORIQUE 5100 // Valeur théorique de la résistance
54
55 #define VALEUR_BATTERIE_VOLT 4.5 // Valeur batterie (4.5 ou 6.0)
56
57 #define VALEUR_3PILES_VOLT 4.5 // Tension pour 3 piles
58 #define VALEUR_BATTERIE_4_5V_MIN_VOLT 4.0 // Valeur min acceptable pour que la batterie 4.5V soit
   considérée comme chargée
59 #define VALEUR_BATTERIE_4_5V_MAX_VOLT 4.8 // Valeur max acceptable pour que la batterie 4.5V soit
   considérée comme chargée
60
61 #define VALEUR_4PILES_VOLT 6.0 // Tension pour 4 piles
62 #define VALEUR_BATTERIE_6V_MIN_VOLT 4.5 // Valeur min acceptable pour que la batterie 6V soit
   considérée comme chargée
63 #define VALEUR_BATTERIE_6V_MAX_VOLT 6.4 // Valeur max acceptable pour que la batterie 6V soit
   considérée comme chargée
64
65 // Attention: DELAI_ENTRE_DEUX_REVEILS_MS doit coïncider avec la valeur DUREE_SOMMEIL_MS du côté de
   l'ESP (Sommeil.h)
66 // #define DELAI_ENTRE_DEUX_REVEILS_MS 1*3600*1000 // Délai entre deux réveils du STM32
   (1*3600*1000 = 1h).
67 #define DELAI_ENTRE_DEUX_REVEILS_MS 5000 // Délai test entre deux réveils du STM32
68
69 // #define DELAI_VERIFICATION_BATTERIE_MS 4*3600*1000 // Délai entre deux vérifications de batterie
   (4*3600*1000 = 4h)
70 #define DELAI_VERIFICATION_BATTERIE_MS 5000 // Délai test entre deux vérifications de

```

```

    batterie (faible délai pour tester le bon fonctionnement du code)
71
72 #define DELAI_ACTIVATION_TRANSISTOR_MS 10 // Délai entre l'enclenchement du transistor et la mesure
    de l'état de la batterie
73 #define DELAI_TIMER6_MS 10 // Délai entre 2 interruptions du timer 6
74
75 #define DEBUG_ALARME_BATTERIE false // Quand vrai, change l'état de la batterie à chaque réveil
    (permet de tester l'envoi des trames)
76
77
78 /* ***** */
79 /* ***** */
80 /* Section: Constants */
81 /* ***** */
82 /* ***** */
83
84 // *****
85 // *****
86 // Section: Prototypes
87 // *****
88 // *****
89 //-----
90
91 bool GestionCheckBatterie(ADC_HandleTypeDef *hadc, TIM_HandleTypeDef *htim, InfoBatterie
    *infoBatterie);
92 Etat MesurerEtatBatterie(ADC_HandleTypeDef *hadc, TIM_HandleTypeDef *htim, InfoBatterie
    *infoBatterie);
93 void LancerCheckBatterie(TIM_HandleTypeDef* htim, InfoBatterie *infoBatterie);
94 Etat ControleEtatChargeBatterie(ADC_HandleTypeDef* hadc, InfoBatterie* infoBatterie);
95 void ArrêterLectureTensionBatterie(TIM_HandleTypeDef* htim, InfoBatterie* infoBatterie);
96
97
98 #ifdef __cplusplus
99 }
100 #endif
101
102 #endif /* _GESTION_BATTERIE_H */
103

```