

Annexe T.8

```
1  // Execution.ino
2  //
3  // Description : fonctions liées à l'exécution principale de la logique de l'ESP
4  // Auteur : Perret Mélissa
5  // Création : 22/09/2024
6  // Modifications : --
7
8  // Version      : V1.0
9  /*-----*/
10
11
12 #include "Execution.h"
13 #include "Wifi.h"      // pour wifiMulti
14 #include "Sommeil.h"   // pour PIN_REVEIL_PAR_STM
15
16
17 ///// Fonction ExecutionReveilTimer: logique princiaple de l'ESP lorsqu'il est
18 ///// reveillé par le timer
19 ///// Description: connexion Wifi, réveiller le STM, récupération et transmissions
20 ///// des valeurs du serveur, envoie notification Discord, retour en mode sommeil
21 ///// Entrées: -
22 ///// Sorties: -
23 void ExecutionReveilTimer() {
24
25     // connexion WiFi
26     if ((wifiMulti.run() == WL_CONNECTED)) {
27
28         HTTPClient http;
29
30         if (MODE_DEBUG) {
31             printf("[HTTP] begin...\n");
32         }
33
34         http.begin(SERVEUR_ADRESSE_HTTP); // pour indiquer le début de la communication
35         HTTP
36
37         ReveillerSTM();
38         delay(DELAI_TRANSMISSION_STM_MS); // pour s'assurer que le STM soit bien prêt à
39         recevoir les trames UART
40
41         ReceptionServeur(&http); // réception et traitement des données du serveur
42
43         EnvoyerSignalFinSTM(); // indiquer au STM que l'ESP à fini d'envoyer ses
44         éventuelles trames UART
45
46         // considérer les valeurs du STM comme non changées avant de recevoir les
47         éventuels changements de la part du STM
48         valeursSTM[ETAT_SEUILS].valeur = IDENTIQUE;
49         valeursSTM[ETAT_BATTERIE].valeur = IDENTIQUE;
50         ReceptionSTM(&http);
51
52         EnvoyerNotificationsDiscord(&http); // envoyer les éventuelles notificartions
53         Discord en cas de changements de valeurs de la part du STM
54
55         http.end(); // pour indiquer la fin de la communication HTTP
56     }
57
58     delay(DELAI_MODE_SOMMEIL_MS); // délai éventuel avant de passer en mode sommeil
59     (par précautions)
60
61     EntreeModeSommeilProfond(); // entrer en mode sommeil
62 }
63
64 ///// Fonction ExecutionReveilSTM: logique princiaple de l'ESP lorsqu'il est reveillé
65 ///// par le STM
66 ///// Description: réception trames STM, envoie notification Discord, retour en mode
67 ///// sommeil
68 ///// Entrées: -
69 ///// Sorties: -
70 void ExecutionReveilSTM() {
71     // connexion WiFi
72     if ((wifiMulti.run() == WL_CONNECTED)) {
```

```

64 HTTPClient http;
65
66 if (MODE_DEBUG) {
67     printf("[HTTP] begin...\n");
68 }
69
70 http.begin(SERVEUR_ADRESSE_HTTP); // pour indiquer le début de la communication
    HTTP
71
72 // marquer les valeurs du STM comme non changées avant de recevoir les éventuels
    changements de la part du STM
73 valeursSTM[ETAT_SEUILS].valeur = IDENTIQUE;
74 valeursSTM[ETAT_BATTERIE].valeur = IDENTIQUE;
75
76 // procéder à la réception des données UART tant que le STM ne nous indique pas
    qu'il a terminé d'envoyer d'éventuelles trames UART
77 do {
78     ReceptionSTM(&http);
79     delay(DELAI_RECEPTION_STM_MS); // Délai entre deux essais
        pour recevoir les trames UART attendues de la part du STM
80 } while (digitalRead(PIN_REVEIL_PAR_STM) == HIGH); // Attendre que le STM nous
    signale qu'il a terminé d'envoyer les trames UART
81
82 ReceptionSTM(&http); // Important de refaire une nouvelle réception de trames
    UART au cas où le STM aurait envoyé des trames juste avant d'indiquer le signal
    de fin de communication
83
84 EnvoyerNotificationsDiscord(&http); // envoyer les éventuelles notifications
    Discord en cas de changements de valeurs de la part du STM
85
86 http.end(); // pour indiquer la fin de la communication HTTP
87 }
88
89 delay(DELAI_MODE_SOMMEIL_MS); // délai avant de passer en mode sommeil (par
    précautions)
90
91 EntreeModeSommeilProfond();
92 }

```