

Annexe T.11

```
1 // Sommeil.ino
2 //
3 // Description : fonctions liées à l'entrée et la sortie du mode sommeil
4 // Auteur : Perret Mélissa
5 // Création : 22/09/2024
6 // Modifications : --
7
8 // Version : V1.0
9 /*-----*/
10
11
12 #include "Sommeil.h"
13
14
15 ///// Fonction EntreeModeSommeilProfond: logique pour entrer en mode sommeil profond
16 ///// Description: active les conditions de réveil (pin pour permettre le réveil
17 ///// Entrées: -
18 ///// Sorties: -
19 void EntreeModeSommeilProfond() {
20
21     // Activation réveil possible via GPIO
22     const gpio_config_t config = {
23         .pin_bit_mask = BIT(PIN_REVEIL_PAR_STM),
24         .mode = GPIO_MODE_INPUT,
25     };
26     ESP_ERROR_CHECK(gpio_config(&config)); // Pour configurer la pin de réveil
27     ESP_ERROR_CHECK(esp_deep_sleep_enable_gpio_wakeup(BIT(PIN_REVEIL_PAR_STM),
28         ESP_GPIO_WAKEUP_GPIO_HIGH)); // Pour indiquer la pin permettant de sortir du mode
29     // sommeil (Permet au STM de réveiller l'ESP si besoin)
30
31     // http://esp-idf.readthedocs.io/en/latest/api-reference/system/deep\_sleep.html
32     esp_sleep_enable_timer_wakeup(DUREE_SOMMEIL_MS * CONVERSION_MS_EN_uS); // Pour
33     // indiquer le délai avant de sortir du mode sommeil (si l'on est pas réveillé avant
34     // via la pin)
35
36     if (MODE_DEBUG) {
37         printf("Entree en mode sommeil pendant %d ms (ou via réveil STM pin GPIO%d)\n",
38             DUREE_SOMMEIL_MS, PIN_REVEIL_PAR_STM);
39     }
40
41     // https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/sleep\_modes.html
42     esp_deep_sleep_pd_config(ESP_PD_DOMAIN_RTC_PERIPH, ESP_PD_OPTION_OFF);
43     esp_deep_sleep_start(); // Pour entrer en mode sommeil profond
44 }
45
46 ///// Fonction GestionReveil: logique exécutée au moment du réveil
47 ///// Description: détermine la raison du réveil (lancement, STM ou timer), lance
48 ///// l'exécution de la logique correspondante au type de réveil
49 ///// Entrées: -
50 ///// Sorties: -
51 void GestionReveil() {
52     esp_sleep_wakeup_cause_t raisonReveil; // Déclaration variable locale
53     raisonReveil = esp_sleep_get_wakeup_cause(); // Pour obtenir la raison du réveil
54     if (MODE_DEBUG) {
55         printf("\nRaison du réveil: %d\n", raisonReveil);
56     }
57
58     // Si la raison du réveil est inconnue, c'est qu'il s'agit du premier démarrage de
59     // l'ESP
60     if (raisonReveil == ESP_SLEEP_WAKEUP_UNDEFINED) {
61         if (MODE_DEBUG) {
62             printf("Initialisation au premier réveil\n");
63         }
64
65         // Initialisation au premier réveil
66         // On met des valeurs incohérentes pour s'assurer que les premières valeurs qui
67         // seront reçues du serveur seront considérées comme différentes
68         valeursServeur[0] = -9999;
69         valeursServeur[1] = -9999;
```

```
63     valeursServeur[2] = -9999;
64     valeursServeur[3] = -9999;
65     valeursServeur[4] = -9999;
66     valeursServeur[5] = -9999;
67
68     ExecutionReveilTimer();
69 } else {
70     // Si la raison du réveil provient de la pin GPIO, c'est que l'ESP a été réveillé
    par le STM
71     if (raisonReveil == ESP_SLEEP_WAKEUP_GPIO) {
72         if (MODE_DEBUG) {
73             printf("Réveil causé par le STM (trames UART à réceptionner)\n");
74         }
75
76         ExecutionReveilSTM();
77     } else {
78         // ESP réveillé par le timer
79         if (MODE_DEBUG) {
80             printf("Réveil causé par le timer\n");
81         }
82
83         ExecutionReveilTimer();
84     }
85 }
86 }
```