Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Systèmes Embarqués

FreeRTOS

Système embarqué & Programmation RTOS

Programmation RTOS

1 – a. Pourquoi disposer de « ***timer hardware*** » est intéressant ?

b. Quel intérêt par rapport à une **tâche freeRTOS** ?

2 – Est-ce que la présence d'un **port série** est toujours nécessaire sur un IoT ?
Donnez des arguments « pour » *et* « contre ».

3 – a. Les processeurs **ARM** et **Risc-V** disposent-ils pour toute leur gamme des mêmes instructions ?

b. Comment est-ce géré ?

4 – a. Si une tâche freeRTOS doit **démarrer** plus tard dans la « *vie* » du système, est-il nécessaire ou non de la créer au démarrage/allumage du système et pourquoi ?

b. Comment **l'activer** au moment où on en a besoin ?

5 – a. Comment **entrelacer** l'exécution de plusieurs tâches T_1, T_2 et T_3 qui s'exécutent indéfiniment ?

b. Comment peut-on faire si on veut, en plus, **contrôler** dans quel ordre l'exécution passe d'une tâche à l'autre ?
Par exemple, $T_1 \longrightarrow T_2 \longrightarrow T_3 \longrightarrow T_1 \longrightarrow T_2 \longrightarrow \ldots$

6 – Un **ESP32** est installé dans un boitier plastique avec les éléments suivants :

- ☐ un écran LCD de contrôle capable d'afficher deux lignes de texte :
  - ◇ l'écran affiche de manière permanente le contenu d'un tampon mémoire de 32 caractéres réparti sur deux lignes de 16 caractères ;
  - ◇ la fonction `void write_screen(char *buffer);` mets à jour le contenu du tampon ;
- ☐ une LED rouge : reliée à une broche GPIO en sortie ;
- ☐ un bouton poussoir : relié à une broche GPIO en entrée.

Le travail de l'ESP32 est le suivant :

- ▷ l'heure `HH:MM` doit être affiché sur la seconde ligne de l'écran avec une mise à jour toute les minutes ;
- ▷ l'appui sur le bouton poussoir doit allumer la LED pendant 3s, afficher un texte sur la première ligne de l'écran « `START WORK` », et déclenche la tâche `task_work()` ;
- ▷ l'écran doit être capable d'afficher des messages en provenance de la tâche `task_work()` sur la ligne 1 tout en continuant à afficher l'heure sur la ligne 2.
- ▷ le travail de `task_work()` peut prendre plusieurs minutes.

Vous allez définir comment, en freeRTOS, vous allez mettre en place le travail de cet ESP32.

Questions

a. Est-ce que la mise à jour de l'écran peut créer des conflits ? Lesquels ?

b. Comment gérer le lien entre le bouton et la LED ? Est-ce qu'il y a des risques ?

c. Comment déclencher le travail de la tâche `task_work()` avec l'appui sur le bouton ?

d. Comment « *bloquer* » la prise en compte du bouton tant que la tâche `task_work()` s'exécute ?

e. Comment allez vous gérer la mise à jour du temps ?

f. Avec quels éléments de freeRTOS allez vous gérez les différents composants matériels et leur(s) interaction(s) compte tenu de votre analyse précédente ?

7 – Soit le programme suivant :

```
1  #define APPCPU 1
2
3  static SemaphoreHandle_t mutex_1;
4  static SemaphoreHandle_t mutex_2;
5
6  void doTaskA(void *parameters) {
7    while (1) {
8      xSemaphoreTake(mutex_1, portMAX_DELAY);
9      vTaskDelay(1 / portTICK_PERIOD_MS);
10
11     xSemaphoreTake(mutex_2, portMAX_DELAY);
12     vTaskDelay(500 / portTICK_PERIOD_MS);
13
14     xSemaphoreGive(mutex_2);
15     xSemaphoreGive(mutex_1);
16
17     vTaskDelay(500 / portTICK_PERIOD_MS);
18   }
19 }
20
21 void doTaskB(void *parameters) {
22   while (1) {
23     xSemaphoreTake(mutex_2, portMAX_DELAY);
24     vTaskDelay(1 / portTICK_PERIOD_MS);
25
26     xSemaphoreTake(mutex_1, portMAX_DELAY);
27     vTaskDelay(500 / portTICK_PERIOD_MS);
28
29     xSemaphoreGive(mutex_1);
30     xSemaphoreGive(mutex_2);
31
32     vTaskDelay(500 / portTICK_PERIOD_MS);
33   }
34 }
35
36 void setup() {
37
38   mutex_1 = xSemaphoreCreateMutex();
39   mutex_2 = xSemaphoreCreateMutex();
40
41   xTaskCreatePinnedToCore(doTaskA, "Task A", 1024, NULL, 2, NULL, APPCPU);
42   xTaskCreatePinnedToCore(doTaskB, "Task B", 1024, NULL, 1, NULL, APPCPU);
43   vTaskDelete(NULL);
44 }
45
46 void loop() {
47 }
```

Décrivez le fonctionnement des tâches à l'aide d'un chronogramme.
Est-ce que tout fonctionne bien ?