

Programmation FreeRTOS : les problèmes liés à la mémoire

■ ■ ■ Programmation freeRTOS : dépassement de capacité

```
// Utilisation d'un seul core pour la démo
#if CONFIG_FREERTOS_UNICORE
static const BaseType_t app_cpu = 0;
#else
static const BaseType_t app_cpu = 1;
#endif

// Les deux tâches

// tâche 1 : affiche sur le port série avec une priorité basse
void startTache1(void *parameter) {
    while(1) {
        int a = 1;
        int b[100];
        for(int i = 0; i < 100; i++) {
            b[i] = a + 1;
        }
        Serial.println(b[0]);
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}

void setup() {
    Serial.begin(115200);
    // Attente pour éviter de rater la sortie sur le port série
    vTaskDelay(1000 / portTICK_PERIOD_MS);
    Serial.println();
    Serial.println("Demo taches et memoire FreeRTOS");
    // lancement des taches
    xTaskCreatePinnedToCore( // Use xTaskCreate() in vanilla FreeRTOS
        startTache1, // Function to be called
        "Tache 1", // Name of task
        1024, // Stack size (bytes in ESP32, words in FreeRTOS)
        NULL, // Parameter to pass to function
        1, // Task priority (0 to configMAX_PRIORITIES - 1)
        NULL, // no Task handle
        app_cpu); // Run on one core for demo purposes (ESP32 only)
    // Détruit la tâche courante => setup
    vTaskDelete(NULL);
}

void loop(){
    // Execution should never get here
}
```

- 1 – Testez le programme ? Que se passe-t-il ?
Modifiez le programme pour qu'il fonctionne.
- 2 – Ajoutez les lignes suivantes après l'affichage de b[0] :

```
Serial.print("High water mark (words) :");
Serial.println(uxTaskGetStackHighWaterMark(NULL));

Serial.print("Heap before malloc (bytes) :");
Serial.println(xPortGetFreeHeapSize());

int *pointeur = (int *)pvPortMalloc(1024 * sizeof(int));
pointeur[0] = 1;
Serial.print("Heap after malloc (bytes) :");
Serial.println(xPortGetFreeHeapSize());
```

Que se passe-t-il ? Comment corriger ?

3 – Écrire un programme qui crée deux tâches A et B :

Tâche A

1. lit les entrées sur le port série ;
2. sur le caractère '\n' stocke tous les caractères dans une zone mémoire dans le tas ;
3. notifie B qu'un message est disponible.

Tâche B

1. attend une notification de A ;
2. affiche le message trouvé dans le tas ;
3. libère la mémoire du tas.