Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Systèmes Embarqués

TP n°2

Programmation avec FreeRTOs

1 Utilisation des « *files de messages* »

a. Vous implémenterez sur votre ESP32s3 la correction de l'exercice 2 de la fiche de TD 2 :

```
#define TAILLE_MESSAGE 32

static const uint8_t msg_queue_len = 5;

static QueueHandle_t msg_queue;

void printMessages(void *parameters) {
  char buffer_message[32];

  while (1) {
  xQueueReceive(msg_queue, (void *)buffer_message, portMAX_DELAY);
  Serial.println(buffer_message);
  }
}

void sendMessage(void *paramaters) {
  int nb_messages = 0;
  char buffer_tache[32];

  while(1){
    sprintf(buffer_tache, "[Depuis 1 message %d]", ++nb_messages);
    xQueueSend(msg_queue, (void *)buffer_tache, portMAX_DELAY);
    vTaskDelay(1000 / portTICK_PERIOD_MS);
  }
}

void sendMessage2(void *paramaters) {
  int nb_messages = 0;
  char buffer_tache[32];

  while(1){
    sprintf(buffer_tache, "[Depuis 2 message %d]", ++nb_messages);
    xQueueSend(msg_queue, (void *)buffer_tache, portMAX_DELAY);
    vTaskDelay(500 / portTICK_PERIOD_MS);
  }
}

void setup() {

  // Configure Serial
  delay(1000);
  Serial.begin(115200);
  while (!Serial) delay(10);

  // Wait a moment to start (so we don't miss Serial output)
  vTaskDelay(1000 / portTICK_PERIOD_MS);
  Serial.println();
  Serial.println("---FreeRTOS Queue Demo---");

  // Create queue
  msg_queue = xQueueCreate(msg_queue_len, TAILLE_MESSAGE*sizeof(char));

  // Start print task
  xTaskCreate(sendMessage, "Send Message", 4096, NULL, 1, NULL);
  xTaskCreate(sendMessage2,"Send Message", 4096, NULL, 1, NULL);
  xTaskCreate(printMessages,"Print Messages", 4096, NULL, 1, NULL);
}

void loop() {

}
```

b. Est-ce qu'il peut y avoir des problèmes pour les buffer gérés par chaque tâche ?
Vous regarderez la documentation à `https://www.freertos.org/Documentation/02-Kernel/04-API-references/06-Queues/03-xQueueSend`

c. Rajoutez la possibilité d'identifier la **provenance du message**, en vous inspirant de la fiche de TP précédente (avec les couleurs réparties en 4 zones), et en affectant une zone pour chaque tâche et une couleur pour chaque opération ;

d. On veut rajouter un codage de couleur :
   - ◇ vert quand la file de message est vide ;
   - ◇ bleu quand la tâche « *sendMessage* » 1 ou 2 envoi un message ;

   Proposez **deux versions différentes** utilisant des outils différents proposés par FreeRTOs pour mettre en œuvre cette modification.

e. Rajoutez **l'affichage d'un message spécial** lié à l'appui du **bouton**.
La couleur associée au bouton est blanc.

f. On veut maintenant que le bouton **réinitialise les compteurs** associés aux deux tâches.
Donnez une solution correcte à ce problème.

g. Observez si **tout fonctionne bien** lors de l'appui répété et rapide du bouton.
Avez vous bien géré la réinitialisation des compteurs ?