

SYSTÈMES TEMPS-RÉEL (TP)

Nom	Prénom	Table
-----	--------	-------

Examen pratique

Durée : 1h30

Ressources autorisées :

- Tous documents
- Internet,
 - sauf IA générative (Chat GPT et équivalents)

Consignes

IMPORTANT : il y a plusieurs tâches et plusieurs fonctionnalités à implémenter. Ne cherchez pas à tout faire d'un coup ! Commencez par une fonctionnalité dans une tâche, puis la même fonctionnalité entre deux tâches, et ainsi de suite.

Appelez l'enseignant pour valider **CHAQUE** étape.

Ne restez pas bloqué trop longtemps : appelez l'enseignant pour avoir des indices ou corriger une erreur persistante.

1. Projet

3 points – 10 min

Créer un projet STM32 pour répondant aux caractéristiques suivantes :

- nom : « eval_rtos_VOTRENOM »
- emplacement : un endroit judicieux
- configuration du projet : USART2 désactivé
- configuration de l'OS :
 - FreeRTOS, CMSIS_v1
 - Dans l'onglet *Config Parameters*
 - *Memory Allocation* à « *Dynamic* »
 - *USE_TASK_NOTIFICATIONS* à « *Enabled* »
 - tous les autres champs à « *Disabled* »
 - Dans l'onglet *Include Parameters*
 - tous les champs à « *Disabled* »

Assurez-vous de la **bonne compilation du projet**.

Appelez l'enseignant pour validation de la compilation (pas d'erreur, pas de warning).

2. Drivers UART

3 points – 10 min

Ajoutez à votre workspace les deux fichiers de drivers de l'UART fournis sur Moodle.

Dans le fichier d'en-tête, dé-commentez la ligne permettant de sélectionner votre carte.

Initialisez le périphérique UART au démarrage du MCU puis envoyez le message suivant une fois les périphériques prêts :

```
"\r\n*** APPLICATION STARTING NOW ***\r\n"
```

Assurez-vous de la **bonne compilation du projet**.

Appelez l'enseignant pour validation sur cible (terminal série avec le texte affiché).

3. Application : pilotage de LED

Cahier des charges

Vous allez devoir piloter une LED (la LD2 de la carte NUCLEO) à partir d'un terminal série. L'application fera donc office d'IHM (Interface Homme-Machine) qui acceptera certaines commandes et contrôlera la LED en conséquence.

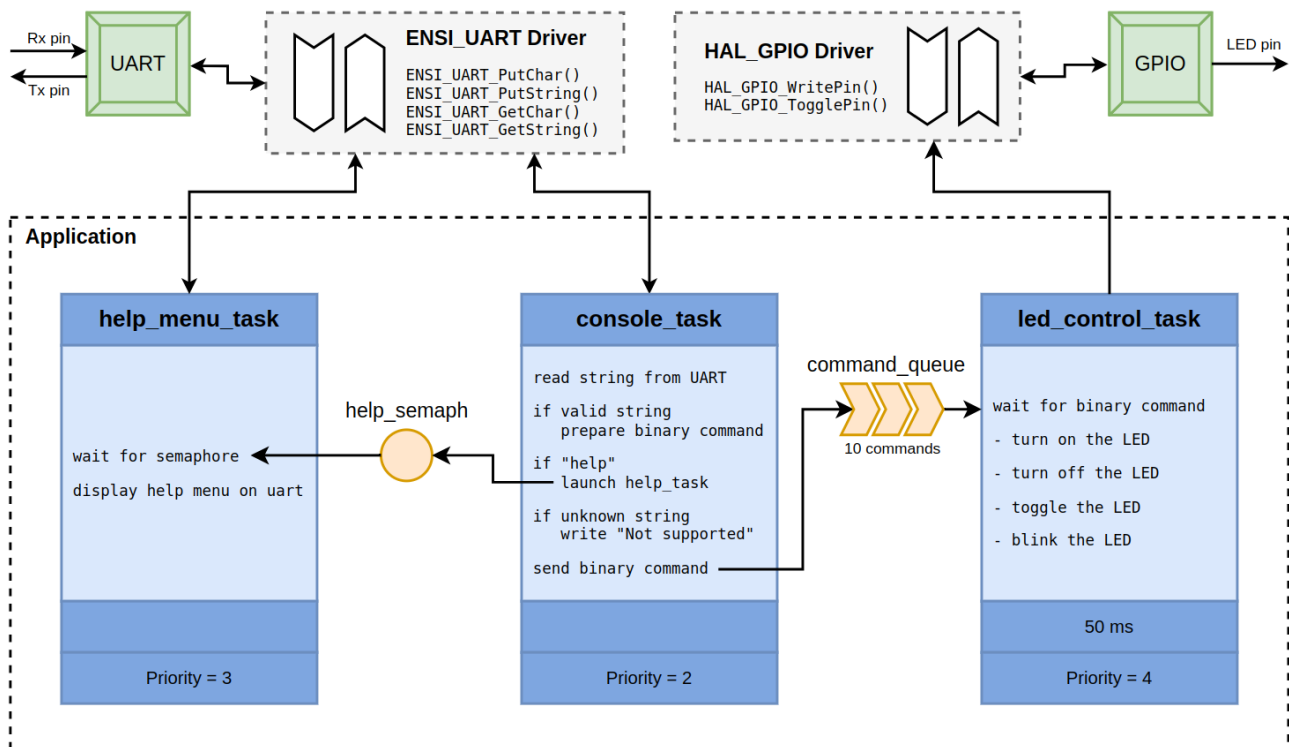
Votre application devra respecter l'affichage ci-dessous. Les saisies par l'utilisateur se font toujours après un caractère '#' (donc le reste est affiché par l'application). Avec le menu d'aide affiché ici, vous avez également les fonctionnalités que vous devrez implémenter.

```

*** APPLICATION STARTING NOW ***
LED control# on
LED control# off
LED control# toggle
LED control# blink
LED control# help
*** HELP MENU ***
  on   - turn on the LED
  off  - turn off the LED
  toggle - change the LED state
  blink - make the LED blink at 10 Hz
  help - open this menu
LED control# flash
> Command not implemented
LED control#
  
```

L'IHM utilisera le VCP (Virtual Comm Port) de la NUCLEO, relié à l'UART2 du MCU et dont les fonctions pilotes sont fournies avec les drives ENSI_UART déjà préparés dans l'étape précédente.

L'architecture logicielle à suivre est la suivante :



Travail à réaliser

10 point – 55 min

Il vous est demandé de travailler dans le fichier **Core/Src/freertos.c** pour développer votre application basée sur FreeRTOS.

Vous devrez implémenter toutes les fonctionnalités indiquées dans le menu d'aide affiché page précédente.

À titre indicatif, voici les critères qui seront évalués (1 point par item).

Les tâches sont créées	
Le menu d'aide s'affiche correctement	
Le sémaphore est fonctionnel	
Les fonctionnalités LED on/off/toggle sont implémentées	
La fonctionnalité LED blink 50 Hz est implémentée	
La queue de message est fonctionnelle	
Les commandes non supportées sont signalées sur la console	
Les noms des éléments respectent ceux de la figure	
Le code est propre (indentations, déclarations, noms explicites, ...)	
L'affichage de la console respecte le cahier des charges	

Appelez régulièrement l'enseignant pour valider ces items.

RAPPELS :

Il est important de procéder étape par étape, en implémentant les fonctionnalités une par une, d'abord au sein d'une seule tâche puis entre plusieurs tâches. Le plus simple me semble de démarrer par le menu d'aide ...

De même, il est inutile de rester bloqué trop longtemps. N'hésitez pas à demander des indices à l'enseignant.

1 point – 2 min

Modifiez votre application pour que le menu *help* s'affiche automatiquement au démarrage.

4. Driver UART

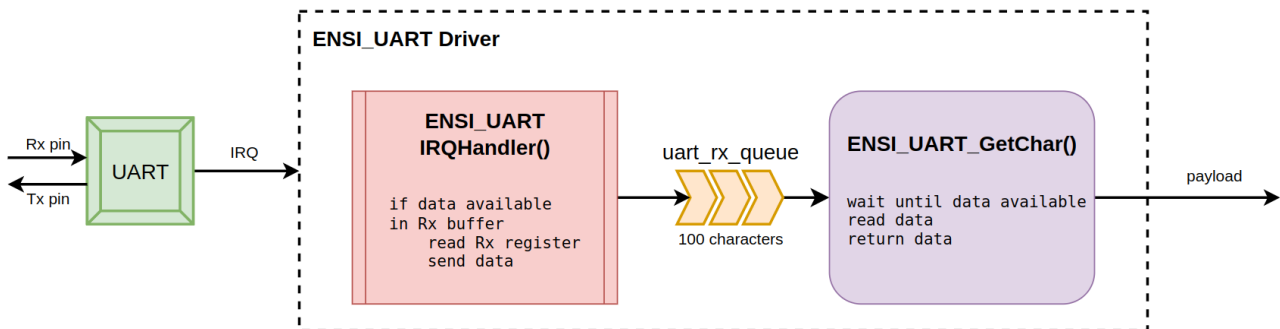
Modifiez la priorité de la tâche `help_menu_task` à 1. Lancez votre application et vérifiez que la fonctionnalité « help » n'est plus disponible. Explications.

La partie « réception » du driver ENSI UART fonctionne autour d'un buffer circulaire (solution utilisée en enseignement de Systèmes Embarqués). Pour rappel, la routine d'interruption (*Interrupt Service Routine*) remplit le buffer à chaque fois qu'un caractère est reçu par le périphérique USART, tandis que la fonction `ENSI_UART_GetChar()` récupère/vidé ce buffer à chaque appel. Ces deux fonctions travaillent indépendamment, cependant la fonction de réception effectue une scrutation active (*polling*), la tâche `console_task` sera toujours alors toujours active (*running/ready*) et la tâche `help_menu_task` ne pourra pas s'exécuter.

Prenez le temps d'étudier la solution actuelle, et appelez l'enseignant pour plus de précisions. Ne restez pas bloqué sur ce point.

3 point – 15 min

Votre travail consiste à remplacer la scrutation active par une fonctionnalité bloquante du RTOS dans le driver `ENSI_UART.c`. Vous suivrez la solution indiquée dans le schéma ci-dessous. Notez que la figure ne représente pas tout le driver ENSI UART, mais uniquement la partie réception d'un caractère.



Pour débiter, commentez toute référence au buffer circulaire. Ensuite, implémentez une solution par une queue de message, qui assurera l'interaction entre la routine d'interruption et la fonction de lecture de caractère.

Note : les appels systèmes depuis une routine d'interruption doivent impérativement utiliser les fonctions de l'API dont le nom termine par `FromISR()`.

Validez votre nouvelle version du driver (l'application n'a pas changé). Le menu d'aide doit à nouveau être disponible (même si l'affichage s'est légèrement détérioré, c'est un détail).

Appelez l'enseignant pour confirmer.