

NOM:

PRENOM:

N° PLACE:

ENSICAEN - année 2021-2022 – semestre 2

2^{ème} année Électronique et Physique Appliquée

Majeure SATE

28/03/2022 - durée 1h30 – hugo descoubes

Documents autorisés

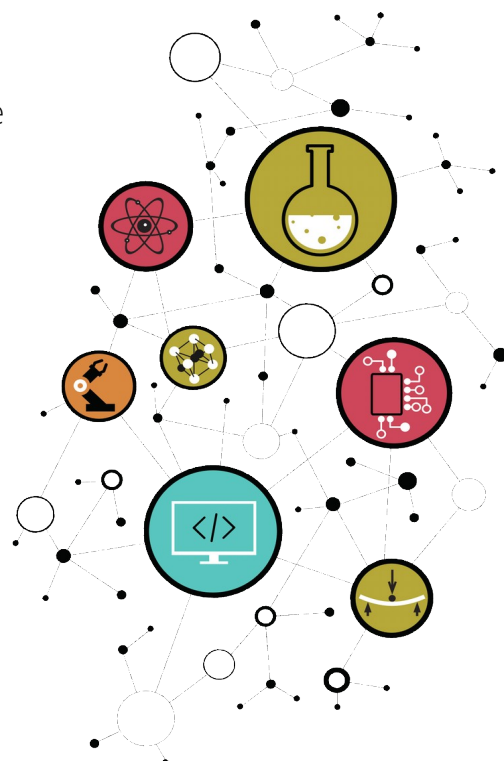
- Pages manuscrites

Conseils

Ce document tient lieu à la fois de sujet et de copie d'examen. Attention, toute recopie directe de phrases du polycopié entraînera une sanction. Utilisez vos propres mots afin de répondre aux questions. Si nécessaire, vous trouverez un espace de prise de notes complémentaires en dernière page. Les pondérations de chaque question étant données, il est conseillé de parcourir le sujet avant de débuter l'épreuve. Une attention toute particulière sera apportée à la clarté et la simplicité des solutions proposées et présentées. Ce sujet d'examen est découpé en trois parties indépendantes :

1. SAVOIR – 7pts : Questions de culture générale pouvant traiter sur tout point abordé en séance de cours présentiel ou présent dans le support de travail. *Connaissances fondamentales et culture scientifique de l'ingénieur en systèmes embarqués*

2. ANALYSER – 13pts : Analyse d'un programme réalisant une application exploitant un RTOS sur une technologie non découverte en enseignement. *Adaptabilité de l'ingénieur aux concepts étudiés sur de nouvelles technologies*



1. SAVOIR

Étayer vos propos ! Détaillez chacune de vos réponses et aidez-vous impérativement de schémas afin d'illustrer les concepts présentés. Bonne épreuve à vous ...

hugo

1. (3,5pts) Question surprise !

2. (*3,5pts*) Question surprise !

2. ANALYSER

L'analyse de programme qui suit porte sur 2 technologies, logicielle et matérielle, parmi les plus utilisées sur le marché de l'embarqué, notamment en UE et US/Canada. Il s'agit d'un programme de démonstration pour processeur MCU STM32F051 proposé par la société Franco/Italienne STMicroelectronics et embarquant un cœur ARM Cortex-M0. Néanmoins, l'applicatif de démonstration reste relativement abstrait au matériel utilisé. La carte d'évaluation STM320518-EVAL est présentée ci-dessous. Les schémas électriques sont donnés en ANNEXES du sujet d'examen.



La technologie RTOS découverte dans cette application de test est uc/OS-III, l'une des principales solution open source du marché concurrente à FreeRTOS.



Le programme à analyser est donné en ANNEXE du sujet d'évaluation. Si votre analyse est sujette à des doutes, spécifier les ainsi que les hypothèses réalisées. Si vos doutes sont justifiés et vos hypothèses justes ou partiellement justes, la notation sera adaptée.

7. (2pt) Entourer en couleur sur la photo précédemment présentée de la carte d'évaluation STM320518-EVAL les interfaces physiques (LED, bouton poussoir, etc) utilisées par l'application et permettant de tester et d'évaluer le fonctionnement du programme. *Bien entourer le(s) composant(s)/broche(s)/interface(s) électronique utilisés par l'application uniquement et non les inscriptions et lettrages présents sur le PCB. Spécifier s'il s'agit d'entrées ou de sorties de l'application. S'aider des schémas électriques et des extraits du guide utilisateur fournis en ANNEXE*

8. (1pt) Après analyse du programme multi-tâches, préciser le nombre de tâches applicatives et leurs priorités (hors tâche d'initialisation) ?

Comme pour *FreeRTOS*, *uc/OS-III* propose des services de *timer* logiciel. Comme pour un *timer* matériel, un Timer logiciel n'est qu'un compteur géré par le *RTOS* (période associée au TICK), qui une fois arrivé à débordement réveille une fonction applicative. Tout comme les *ISR's* réveillées sur événements matériels (*IRQ's*), ces fonctions ne peuvent être réveillées que sur événement logiciel généré par le système. Ces fonctions sont généralement nommées fonctions de rappel ou *callback*. De même, la fréquence de comptage de ces *timers* logiciels, et donc de réveil des fonctions de *callback*, est forcément un multiple du *tick* système. Pour information, dans le cadre de cette application, le **tick a été fixé à 1ms soit 1KHz de fréquence**. Le *scheduler* de *ucOS-III* fonctionne en mode préemptif.

9. (5pts) Proposer une représentation graphique et exhaustive du squelette de l'application. Phase d'initialisation et d'amorçage à ne pas présenter. Nommages des interfaces, limitations et contraintes imposées à présenter et à respecter. Votre modèle devra faire apparaître les éléments suivants et respecter le formalisme imposé :

- Les tâches applicatives
- Les outils système et leurs rôles (synchronisation, protection et communication)
- Les fonctions de callback (associées aux timers logiciels)
- Les ressources partagées (périphériques et variables globales)



10. (1pt) Proposer une procédure de test permettant de valider le fonctionnement de ce programme. *Préciser les appareillages de mesure utilisés (oscilloscope, GBF, etc) ainsi que les calibres utilisés (tension/période). Proposer ci-dessous un schéma ou dessin présentant votre procédure de test*

11. (2pts) Expliquer le fonctionnement du programme en vous aidant obligatoirement d'un chronogramme commenté sur une durée de capture de 300ms à partir du démarrage de l'application (ne pas tenir compte de la phase de démarrage et d'initialisation du programme).

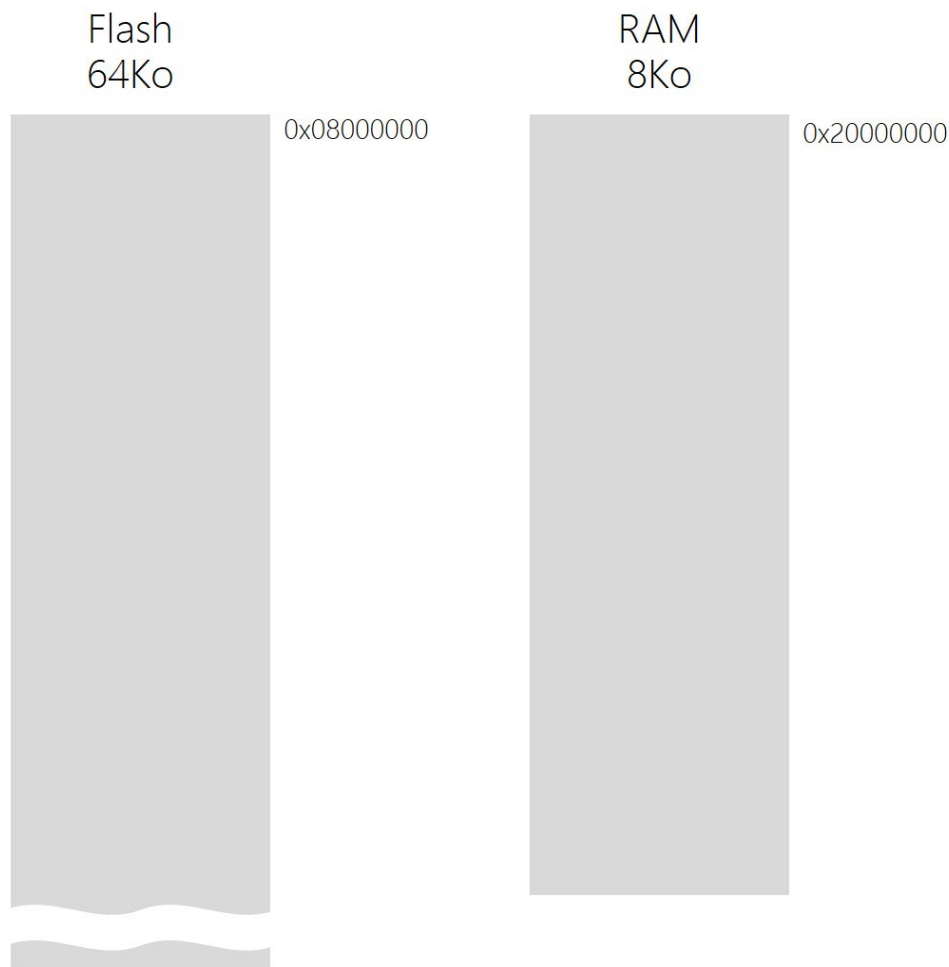
12. (2pt) Après quelques ajustements, voici ci-dessous le *mapping* mémoire généré par l'éditeur de liens durant le processus de compilation de l'application précédemment analysée. Celui-ci présente la segmentation logique de la mémoire physique. Représenter graphiquement sur le schéma ci-dessous cette segmentation logique et les natures de chaque segment et section. Respecter "grossièrement" les échelles et tailles de chaque section et segment.

Memory Configuration

Name	Origin	Length	Attributes
FLASH	0x08000000	0x00010000	xr
RAM	0x20000000	0x00002000	xrw

Linker script and memory map

.isr_vector	0x08000000	0xc4
.text	0x080000c4	0x4588
.data	0x20000000	0x10
.bss	0x20000010	0x1628
.stack	0x20001638	0x80



13. (1pt) Comme pour FreeRTOS (stratégies heap1.c et heap2.c), sous uc/OS-III, dans l'application précédente, le tas système est une variable globale statique non initialisée. Quelle est la section statique concernée ? Proposer en complétant le schéma ci-dessus et pour l'application précédente, une segmentation logique applicative du tas système de uc/OS-III (avec TCB et piles des tâches applicatives et système)

