



Systèmes Embarqués 2 pour les classes T-2/I-2

Objectifs de l'examen oral 2019

Les étudiant-e-s devront être capables :

Prérequis : programmation en assembleur et en C

- de coder dans les règles de l'art un traitement simple en assembleur ARM
- d'utiliser et de reconnaître les différents modes d'adressage présent sur le ARM
- de manipuler correctement les types de base de C
- de coder un algorithme de traitement de données en C
- de coder dans les règles de l'art des appels de fonctions en C
- de décrire le passage d'arguments par valeur et référence lors d'appel de fonction en C
- de différencier les fonctions globales des fonctions locales dans une application en C
- de différencier les variables globales des variables locales et des variables rémanentes dans une application en C
- d'expliquer les conditions pour qu'une fonction C soit réentrante
- de manipuler correctement les types complexes (énumérations, tableaux, structures, unions ...) de C
- de concevoir une interface C permettant d'accéder aux registres d'un périphérique
- de manipuler correctement les pointeurs en C
- de décrire les conversions des types en C
- de manipuler correctement les pointeurs de fonction en C

Remarque : un exemplaire du jeu d'instructions du processeur ARM sera mis à disposition, document « 05_ARM_Instruction_Set_Summary.pdf »

100. Interfaçage assembleur – C

- 100.01 d'utiliser correctement la pile (stack) dans un programme codé en assembleur
- 100.02 de programmer une sous-routine en assembleur ARM en utilisant la pile pour le passage de paramètres et pour les variables locales
- 100.03 de représenter le passage de paramètres à des sous-routines et d'expliquer le retour de valeurs et du résultat pour interfacer avec des programmes écrits en C
- 100.04 de représenter l'image de la pile lors de l'appel d'une sous-routine en assembleur
- 100.05 de coder dans les règles de l'art l'appel de routines développées en C depuis un code en assembleur ARM
- 100.06 de spécifier correctement l'appel de routines en assembleur depuis un code C
- 100.07 de décrire l'utilisation des registres des μ P ARM en assembleur



Systèmes Embarqués 2 pour les classes T-2/I-2

Objectifs de l'examen oral 2019

200. Interruptions

- 200.01 d'expliquer les différentes phases de traitement d'une interruption (séquence)
- 200.02 de classer les types d'interruptions et d'exceptions
- 200.03 d'expliquer la fonctionnalité de la table des vecteurs d'interruptions sur les processeurs ARM, ainsi que l'implémentation spécifique du μ P TI AM335x
- 200.04 de déterminer le mode de fonctionnement du μ P à partir de son registre de statut
- 200.05 d'expliquer comment l'on peut passer du mode superviseur au mode utilisateur et vice versa et de concevoir le code assembleur permettant ces passages
- 200.06 de décrire la fonction des pointeurs de piles des différents modes du processeur ARM et de concevoir le code permettant de les initialiser
- 200.07 de décrire le concept de commutation de contexte, de latence et de gigue
- 200.08 de décrire le système d'interruption des processeurs ARM et du processeur AM335x (μ P (core), INTC, GPIO, etc.)
- 200.09 d'expliquer les mécanismes d'activation et de désactivation des interruptions matérielles (μ P (core), INTC, GPIO, etc.)
- 200.10 de décrire le principe de niveaux de priorité (μ P, INTC et GPIO)
- 200.11 de décrire la procédure de reconnaissance d'interruption multiple (scrutation, priorisée, vectorisée)
- 200.12 de décrire et de concevoir un gestionnaire d'interruption (du μ P à l'application)
- 200.13 de programmer une application utilisant une interruption
- 200.14 de décrire et de concevoir des opérations atomiques
- 200.15 de simuler des interruptions logicielles et des exceptions
- 200.16 de simuler des interruptions matérielles
- 200.17 de décrire le concept de « listener », de le réaliser et de le mettre en œuvre

220. Systèmes d'exploitation

- 220.01 de citer quelques techniques et méthodes de développement pour des systèmes embarqués
- 220.02 de décrire les différents types de systèmes multitâches (systèmes d'exploitation)
- 220.03 de décrire les différentes composantes d'un noyau
- 220.04 de décrire les ressources globales, partagées et privées d'un thread
- 220.05 de décrire les états principaux d'un thread
- 220.06 de décrire la commutation de contexte entre deux threads
- 220.07 de concevoir un algorithme de transfert de contexte entre deux threads
- 220.08 de concevoir l'initialisation du contexte d'un thread



Systèmes Embarqués 2 pour les classes T-2/I-2

Objectifs de l'examen oral 2019

- 220.09 de concevoir un ordonnanceur (scheduler) élémentaire
- 220.10 de concevoir un mécanisme de synchronisation simple (sémaphore, message queue)
- 220.11 de décrire les éléments et structures nécessaires à la gestion d'un thread
- 220.12 de décrire la fonction d'un ordonnanceur (scheduler) élémentaire
- 220.13 de décrire et concevoir un mécanisme pour la mise en pause d'un thread pour un certain laps de temps

240. Entrées/Sorties

- 240.01 de décrire le concept et la structure générale des entrées/sorties
- 240.02 de décrire les différents modes et techniques pour piloter des périphériques d'entrées/sorties
- 240.03 de différencier une programmation interruptive d'une programmation par scrutation d'une entrée (interrupt vs. polling)
- 240.04 de concevoir et réaliser une application gérant des entrées/sorties en mode interruptif
- 240.05 de concevoir et réaliser une application gérant des entrées/sorties par scrutation
- 240.06 de dimensionner (calculer) la taille des tampons d'émission et de réception sous des conditions données

300. Architecture générale

- 300.01 de décrire l'architecture générale des systèmes à microprocesseurs
- 300.02 de décrire l'architecture générale de l'unité centrale
- 300.03 de décrire le principe de traitement de l'information
- 300.04 de citer les différentes architectures selon la classification de Flynn
- 300.05 de décrire la structure et l'architecture de Von Neuman et Harvard
- 300.06 de décrire succinctement l'architecture RISC et CISC
- 300.07 de décrire les différentes mémoires volatiles et permanentes
- 300.08 de décrire la structure mémoire (layout) d'un programme
- 300.09 de décrire l'organisation de la mémoire des systèmes embarqués et systèmes on chip
- 300.10 de représenter des données dans un espace mémoire suivant une numérotation des adresses par byte (8 bits), halfword/short (16 bits) et word/long (32 bits).
- 300.11 de comparer une organisation de la mémoire de type « Big Endian » et celle d'une organisation « Little Endian »
- 300.12 d'expliquer l'alignement des données en mémoire



Systèmes Embarqués 2 pour les classes T-2/I-2

Objectifs de l'examen oral 2019

310. Architecture interne

- 310.01 de décrire l'architecture interne des processeurs ARM
- 310.02 de décrire les éléments composant la structure interne des processeurs ARM
- 310.03 de décrire le principe de fonctionnement des processeurs ARM
- 310.04 d'expliquer les différents modes de fonctionnement du μ P ARM (usr, svc, irq, ...)
- 310.05 de comprendre la fonction des registres internes des processeurs ARM
- 310.06 d'expliquer le déroulement d'une instruction assembleur à l'intérieur de la structure du microprocesseur ARM et de la décomposer en cycles de fonctionnement
- 310.07 d'expliquer le rôle du pipeline et de décrire son principe de fonctionnement
- 310.08 d'expliquer le comportement du pipeline lors d'un branchement conditionnel

400. Memory Management Unit (MMU)

- 400.01 de décrire la fonctionnalité (les rôles) de la MMU
- 400.02 de décrire l'architecture de la MMU
- 400.03 de décrire les mécanismes de translation d'adresses
- 400.04 de décrire le rôle de la TLB
- 400.05 d'expliquer le concept de pages, tables de pages et tables sur plusieurs niveaux
- 400.06 de configurer les tables de translation de la MMU
- 400.07 de retrouver l'adresse physique à partir de l'adresse virtuelle dans les tables de translation de la MMU
- 400.08 de décrire l'implémentation de la MMU sur le μ P TI AM335

410. Mémoire cache

- 410.01 de décrire la hiérarchie de la mémoire sur des systèmes à μ P
- 410.02 de décrire la fonctionnalité (les rôles) d'une mémoire cache
- 410.03 de décrire les 2 principes de localité spatiale et temporelle
- 410.04 de décrire les mécanismes de la mémoire cache (identification d'une ligne, placement d'une ligne, recherche d'une ligne, etc.)
- 410.05 de décrire la différence entre une mémoire cache virtuelle et physique
- 410.06 de décrire l'implémentation de la mémoire cache sur le μ P TI AM335x
- 410.07 d'expliquer pourquoi certains algorithmes ont de meilleures ou moins bonnes performances au niveau du μ P et de sa mémoire cache
- 410.08 de calculer le nombre de bits du tag en fonction des dimensions de la cache
- 410.09 de décrire l'architecture du mémoire cache (voies, set, lignes, tags, flags,...)



Systèmes Embarqués 2 pour les classes T-2/I-2

Objectifs de l'examen oral 2019

500. Direct Memory Access (DMA)

- 500.01 d'expliquer la fonctionnalité d'un DMA
- 500.02 de décrire le principe et l'architecture DMA
- 500.03 de décrire la fonctionnalité des DMA sur le processeur TI AM335x

6xx. Travaux pratiques et projet intégré

610. Git

- 610.01 de décrire les différentes étapes et opérations pour mettre en place un dépôt Git
- 610.02 de décrire les opérations pour ajouter un nouveau fichier dans un dépôt Git
- 610.03 de décrire les opérations pour synchroniser les dépôts locaux avec le dépôt centralisé
- 610.04 d'expliquer comment traiter les conflits de la synchronisation des dépôts Git

620. Debugging

- 620.01 d'expliquer comment trouver la taille et l'emplacement en mémoire d'une variable
- 620.02 d'expliquer comment trouver la taille d'une application stockée en mémoire

630. Interfaces de communication

- 630.01 de citer les caractéristiques principales (signaux et protocole) de l'interface de communication UART et de donner un exemple d'utilisation / d'application
- 630.02 de citer les caractéristiques principales (signaux et protocole) de l'interface de communication I2C et de donner un exemple d'utilisation / d'application
- 630.03 de citer les caractéristiques principales (signaux et protocole) de l'interface de communication SPI et de donner un exemple d'utilisation / d'application

640. GPIO du μ P AM3358

- 640.01 de citer les composants mis en œuvre sur le μ P AM3358 pour accéder aux broches d'entrée/sortie
- 640.02 de décrire le rôle du multiplexeur PAD-Mux
- 640.03 de décrire le rôle et le principe de fonctionnement du contrôleur d'entrée/sortie GPIO
- 640.04 de décrire le fonctionnement des GPIO en mode interruptif

650. DMTimer du μ P AM3358

- 650.01 de décrire le fonctionnement des DMTimer en mode par scrutation
- 650.02 de décrire le fonctionnement des DMTimer en mode interruptif



Systèmes Embarqués 2 pour les classes T-2/I-2

Objectifs de l'examen oral 2019

- 650.03 de concevoir une fonction permettant de mesurer le temps sur plusieurs années avec une granularité de ~40ns (fréquence 24MHz), ceci par scrutation et par interruption
- 650.04 de décrire et concevoir un algorithme permettant de mesurer le temps d'exécution de parties de code ou de fonctions

660. Périphériques de la carte d'extension

- 660.01 de décrire le principe de fonctionnement de l'afficheur 7-segments (plusieurs digits)
- 660.02 de concevoir un programme pour piloter l'afficheur 7-segments (plusieurs digits)
- 660.10 de décrire le principe de fonctionnement d'un bouton-poussoir
- 660.11 de concevoir la machine d'états logicielle pour la détection de changement d'état d'un bouton-poussoir (scrutation)
- 660.20 de décrire le principe de fonctionnement d'un encodeur rotatif
- 660.21 de concevoir la machine d'états logicielle pour la détection du sens de rotation de l'encodeur rotatif (scrutation)
- 660.22 de concevoir la détection du sens de rotation de l'encodeur rotatif en mode interruptif

670. Click Board OLED-C

- 670.01 de décrire le principe de fonctionnement de l'écran LCD OLED
- 670.02 de décrire le système de codage des couleurs RGB et plus particulièrement RGB565
- 670.03 de décrire et concevoir les fonctions permettant de dessiner des figures géométriques simples (carré, rectangle)
- 670.04 de décrire et concevoir la fonction permettant de dessiner des caractères ASCII
- 670.05 de décrire et concevoir la fonction permettant d'effectuer une rotation d'image

680. Click Board nRF24 (selon choix fait par l'étudiant)

- 680.01 de décrire le principe de transmission et réception des trames du nRF24 (adressage, bandes, pipes)
- 680.02 de concevoir une application mettant en œuvre des modules nRF24

690. Click Board RFid (selon choix fait par l'étudiant)

- 690.01 de concevoir une application mettant en œuvre un module RFid