

ISIE4 – LOO C++ - Fiche de séance (TD / AP)

AP7 – C++ embarqué sur cible MBED

Référence/type ¹	LOO C++ AP7		<input type="checkbox"/> TD <input checked="" type="checkbox"/> AP
Thématique principale	Développement C++ embarqué sur cible compatible MBED		
Thématique(s) secondaire(s)	<ul style="list-style-type: none"> ✓ Génie logiciel (UML + SART) ✓ Mise en œuvre d'un RTOS 		
Durée	8h	Niveau²	A
Prérequis			
Compétences opérationnelles et niveau cible³	<ul style="list-style-type: none"> ✓ Analyser une application embarquée s'appuyant sur un langage objet (A) ✓ Implémenter une analyse logicielle embarquée/objet en C++ (A) ✓ Mettre en œuvre les ressources d'un RTOS C++ (A) ✓ Produire une documentation complète et de qualité (M) 		
Modalités et critères d'évaluation	CR sous forme de dossier technique (/20)		
Matériel(s), équipement(s), composant(s) nécessaires	Module MBED-LPC1768 ou carte Nucleo (L476RG, L073RZ) ou autre cible matérielle compatible MBED (si utilité justifiée).		
Logiciel(s) nécessaire(s)⁴	Environnement de développement MBED (local ou en ligne)		
Ressources	https://os.mbed.com/docs/mbed-os/v6.16/introduction/index.html https://os.mbed.com/platforms/mbed-LPC1768/ https://github.com/ARMmbed/mbed-os/blob/master/targets/TARGET_NXP/TARGET_LPC176X/TARGET_MBED_LPC1768/PinNames.h https://os.mbed.com/platforms/ST-Nucleo-L476RG/ https://os.mbed.com/platforms/ST-Nucleo-L073RZ/		

Avant la séance...

L'objectif de cette activité pratique est le développement d'une application embarquée sur cible type MCU 32bits (ARM Cortex-M) mettant en œuvre a minima un module matériel (capteur, actionneur...). Une des premières tâches à effectuer est de trouver une idée d'application à développer.

Il est demandé, en amont de la première séance de :

- ✓ Identifier un-e équipièr-e pour le travail (recommandé, même si rien n'interdit un travail en autonomie).
- ✓ Passer un peu de temps à parcourir le fichier recensant les matériels (potentiellement disponibles et qu'il serait possible de mettre en œuvre (attention cependant aux problématiques d'alimentation : il sera compliqué, par exemple de mettre en œuvre un matériel nécessitant du 12V).
- ✓ Commencer à réfléchir à l'application à développer, avec comme **objectif principal l'originalité**. Il s'agit en effet d'imaginer des **cas d'utilisation totalement improbables**, dans des applications de type jeu ou ludiques par exemple.

¹ TD : ½ groupe, apports de cours, mise en pratique guidée – TP : ½ groupe, travail autonome majoritaire.

² Le niveau se rapporte à la thématique principale. Il peut ici être entendu comme un indicateur de la difficulté du travail (N-Facile, A-Sans grande difficulté, M-Quelques points complexes, MA-Difficile)

³ Les niveaux cible correspondent pour chaque compétence opérationnelle au niveau d'acquisition de la celle-ci à l'issue du travail dans son intégralité (en incluant les phases préparatoires et de synthèse).

⁴ En plus d'un environnement de développement C++

Travail encadré en séance

Apports de cours

- ✓ Selon besoins (Génie logiciel, RTOS, C++...)

Activités

Phase 1 – Choix et définition du sujet

Cette phase, devant obligatoirement faire l'objet d'une validation par l'encadrant de TP, va permettre de préciser les contours de l'application à développer, identifier les matériels nécessaires et définir les objectifs du travail.

Phase 2 – Analyse préliminaire

Il s'agit ici de mettre en place les premières briques de l'analyse logicielle. Sont attendus (a minima) :

- ✓ Une matrice MOSCOW précisant le cahier des charges des fonctionnalités du dispositif
- ✓ Un diagramme des cas d'utilisation UML
- ✓ Un Diagramme de Contexte de Données SART

Phase 3 – Faisabilité / Prise en main des matériels et drivers

Sauf cas particulier, aucun des matériels à mettre en œuvre ne sera parfaitement connu/maîtrisé. Une phase de tests/essais permettant de s'approprier le fonctionnement des matériels et surtout des éventuelles bibliothèques / drivers disponibles (bibliothèques spécifiques MBED-OS, mais envisager aussi le portage de bibliothèques autres comme Arduino, FreeRTOS...).

Phase 4 – Analyse détaillée

Il est maintenant temps de rentrer dans les détails de l'analyse logicielle. Les diagrammes suivants sont (a minima) à produire :

- ✓ Diagramme des Flots de Donnée (et de Contrôle) de niveau 0 (DFDo) SART.
 - Avec dictionnaire de données
- ✓ Diagramme des classes UML
 - Avec spécifications détaillées des classes

Au-delà de ces diagrammes obligatoires, il faudra envisager de s'appuyer sur tous les outils et diagrammes connus et disponibles :

- ✓ Diagrammes de séquence UML
- ✓ D.E.T. (UML / SART)
- ✓ Algorithmes / Ordinogrammes
- ✓ ...

Phase 5 – Développement et tests unitaires

Cette phase a pour objectif de développer les modules logiciels (classes principalement) conformément aux spécifications issues de l'analyse détaillée et de s'assurer de leur bon fonctionnement à l'aide d'une batterie de tests unitaires.

Phase 6 – Développement application et intégration

S'appuyant sur un plan de tests d'intégration, il s'agit maintenant de finaliser la partie applicative du sujet.

Annexes

Annexe 1 – Consignes pour la rédaction du compte-rendu

Le CR à produire doit prendre la forme d'un « dossier technique » devant idéalement permettre de redévelopper le dispositif. On devra donc trouver **a minima**, et dans cet ordre les éléments suivants :

- ✓ Présentation du dispositif, grandes lignes du cahier des charges, objectifs...
- ✓ Précision du cahier des charges avec une **matrice MOSCOW**.
- ✓ Présentation de l'analyse préliminaire avec :
 - **Diagramme de Contexte de Données**
 - **Diagramme des cas d'utilisation**
 - Commentaires et explications
- ✓ Identification et présentation (rapide) des bibliothèques logicielles à utiliser. Compte-rendu rapide de la prise en main et des éléments saillants de ces modules.
- ✓ Présentation de l'analyse détaillée avec :
 - **Diagramme des Flots de Données / Contrôles**
 - **Diagramme des classes**
 - Commentaires et explications
- ✓ Compte-rendu rapide du développement et des tests unitaires menés pour valider chaque module logiciel.
- ✓ Compte-rendu du développement de l'applicatif et de l'intégration.
- ✓ Etat des lieux précis à la fin des travaux, on indiquera notamment de manière claire
 - L'état d'avancement de la prise en main des matériels (qu'est-ce qui a été mis en œuvre avec succès, qu'est-ce qui n'a pas pu être mis en œuvre...)
 - L'état d'avancement au niveau unitaire (quels sont les modules validés unitairement, lesquels ne le sont pas, et pour quelle raison...)
 - L'état d'avancement au niveau intégration