

**Sigle : INF1643 Gr. 01****Titre : Architecture des ordinateurs II****Session : Hiver 2026 Horaire et local****Professeur : Rahmani, Naim Mohamed****1. Description du cours paraissant à l'annuaire :****Objectifs**

Comprendre le fonctionnement global d'un ordinateur et de ses composants. Comprendre la communication entre les différentes parties matérielles et logicielles d'un ordinateur à l'aide du langage assembleur.

**Contenu**

Rappel sur la représentation des nombres, arithmétique en compléments et codes numériques. Structure interne des ordinateurs : processeur, mémoire, entrées/sorties, bus. Modèle du processeur : registres, unité arithmétique et logique, modes d'adressage, gestion de la pile. Introduction à la programmation sur un microcontrôleur et au langage assembleur. Programmation des ports d'entrée/sortie sur un microcontrôleur en langage de haut niveau. Ce cours comporte des séances obligatoires de travaux pratiques (TP) de trois heures par semaine.

Descriptif – Annuaire

**2. Objectifs spécifiques du cours :**

À la fin de ce cours, l'étudiant(e) sera en mesure de :

- Présenter les concepts fondamentaux des microprocesseurs.
- Décrire le rôle des composantes et le fonctionnement d'un microprocesseur et d'un ordinateur.
- Introduire les techniques de base de programmation en assembleur HCS12 et en langage C.

Le cours couvre 6 des 12 qualités requises des diplômé(e)s telles que définies dans les normes d'agrément des programmes de génie au Canada (<http://www.engineerscanada.ca/fr/ressources-en-matiere-dagrément>).

a. Qualité 1 : Connaissances en génie

**b. Qualité 2 : Analyse de problèmes**

**c. Qualité 3 : Investigation**

**d. Qualité 4 : Conception**

e. Qualité 5 : Utilisation d'outils d'ingénierie

f. Qualité 6 : Travail individuel et en équipe

**Les qualités 2, 3, et 4 sont mesurées dans ce cours pour fins de rétroaction.**

Objectifs spécifiques	Qualité	Indicateurs	Introduit	Développé	Utilisé
<ul style="list-style-type: none"> <li>Résoudre des problèmes spécifiques à l'aide d'un microprocesseur.</li> </ul>	2	2. Formuler un processus de résolution de problèmes, comprenant des approximations et des hypothèses.	X		
<ul style="list-style-type: none"> <li>Concevoir, développer, tester et évaluer des prototypes sur les microcontrôleurs HCS12 et Arduino.</li> </ul>	3	2. Mettre en œuvre des investigations documentaires, des expériences et/ou des prototypes.	X		
	4	3. Créer des modèles, simulations, prototypes, et faire des tests.	X		
	4	4. Vérifier la conformité de la conception par rapport au cahier des charges.	X		

### 3. Stratégies pédagogiques :

L'approche générale est interactive et conçue en vue d'impliquer l'étudiant(e) dans la construction et l'utilisation des savoirs. Divers modèles d'enseignement seront utilisés dont :

- Mode d'enseignement en présentiel, sauf trois séances qui seront données en non-présentiel.
- L'exposé interactif
- L'apprentissage par problèmes
- L'apprentissage coopératif
- La discussion et les exercices pratiques
- Les travaux pratiques et un projet de conception

#### Séances de cours :

- Les connaissances seront présentées sous forme de cours magistraux.
- Le matériel du cours sera mis à la disposition des étudiant(e)s sur Moodle.

#### Séances de TP et projet :

- Les séances de TP auront lieu en présentiel sous la supervision de l'assistant(e) à l'enseignement. La présence à ces séances est obligatoire selon les dates prescrites dans la section plan de cours détaillé ;
- Le projet et les séances de TP permettront la mise en œuvre des concepts acquis ;
- Les logiciels requis seront disponibles au laboratoire pour la partie pratique ;
- Une liste des composantes sera fournie aux étudiant(e)s pour réaliser leurs laboratoires et le projet ;
- Les étudiant(e)s peuvent aussi pratiquer à distance avec les simulateurs (Trace 32, <https://www.lauterbach.com/frames.html?home.html> et Tinkercad, <https://www.tinkercad.com/#/>) pour les travaux prescrits et le projet. Des tutoriels sur l'utilisation de ces simulateurs seront mis à la disposition des étudiant(e)s ;
- Si les étudiant(e)s disposent de leur propre matériel (p. ex. carte Arduino Mega ou Uno et composantes électroniques), ils/elles peuvent l'utiliser pour les TP et/ou le projet ;
- La chronologie des séances de laboratoires sera disponible sur le site Moodle du cours ;
- Un rapport final est à remettre une semaine après la séance de travaux pratiques.

NB : Le responsable de laboratoire fournira de l'aide uniquement pour la mise à disposition et l'utilisation du matériel. La présence à l'heure indiquée au laboratoire est obligatoire.

**Examens de mi-session et final seront en mode présentiel.**

**Les attentes :**

- Les étudiant(e)s investissent au moins 90 heures de travail personnel en plus des 45 heures de cours et des heures de TP.
- Les étudiant(e)s qui s'inscrivent à ce cours doivent s'assurer qu'ils ont accès à : un ordinateur (avec un système d'exploitation Windows); une connexion Internet; une webcam; un microphone; la suite Office 365 (les étudiant(e)s ont un accès gratuit à la suite Office 365).

**Lien utile :** [Modalités de tenue des séances de travaux pratiques \(TP\) et de projets dans les laboratoires de génie](#)

**4. Heures de disponibilité ou modalités pour rendez-vous :**

Prendre rendez-vous en envoyant un courriel à [NaimMohamed.Rahmani@uqo.ca](mailto:NaimMohamed.Rahmani@uqo.ca).

Le rendez-vous aura lieu soit en présentiel (Bureau B-2093) ou par vidéoconférence via Zoom.

**5. Plan détaillé du cours sur 15 semaines :**

Semaine	Thèmes	Dates
1	<b>Introduction</b> Présentation du plan de cours <ul style="list-style-type: none"><li>• Mise en contexte</li><li>• Objectifs du cours</li><li>• Organisation du cours</li><li>• Rappels sur les systèmes de numération</li><li>• Arithmétique binaire des nombres signés et non signés</li></ul>	12 janv. 2026
2	<b>Structure interne des ordinateurs</b> <ul style="list-style-type: none"><li>• Opérations dans l'unité Arithmétique et Logique (ALU)</li><li>• Logiciel et matériel : Langage machine et de haut niveau. Dispositifs d'entrée et de sortie. Stockage de données. Communication. Technologie. Performance.</li></ul> <b>Architecture d'un ordinateur monoprocesseur</b> <ul style="list-style-type: none"><li>• Schéma bloc et fonctionnement global d'un ordinateur</li><li>• Architectures RISC et CISC</li></ul> <b>TD1 : Rappel sur les bases d'encodage. Performance. Principes généraux des microcontrôleurs. Conception structurée des programmes.</b> <b>Groupe A et Groupe B : mercredi 21 janv. 2026</b>	19 janv. 2026
3	<b>Principes généraux des microcontrôleurs</b> <ul style="list-style-type: none"><li>• Un microcontrôleur typique</li><li>• Exécution du code (instructions)</li><li>• Le développement de logiciel (<i>firmware</i>) pour les microcontrôleurs</li></ul> <b>Introduction au microcontrôleur HCS12</b> <ul style="list-style-type: none"><li>• Aperçu</li><li>• Les registres et les mots d'états</li><li>• Modes d'adressage</li><li>• Le jeu d'instructions de base</li><li>• Structure d'une instruction : adressage et format</li></ul>	26 janv. 2026

	<b>TD2 : Registres. Mode d'adressage. Programmation en langage assembleur.</b> <b>Groupe A : lundi 26 janv. 2026</b> <b>Groupe B : mercredi 28 janv.2026</b>	
4	<b>Introduction à la programmation en assembleur HCS12</b> <ul style="list-style-type: none"> <li>• Instructions de stockage</li> <li>• Instructions de transfert</li> <li>• Instructions arithmétiques</li> <li>• Instructions logiques</li> <li>• Instructions de test</li> <li>• Instructions de branchement</li> <li>• Outils de développement logiciels et matériels</li> <li>• Outils de développement pour HCS12. Débogage</li> </ul>	02 fév. 2026
5	<b>Programmation avancée en langage assembleur</b> <ul style="list-style-type: none"> <li>• Sous-programmes. La gestion de la pile</li> <li>• Structures remarquables</li> <li>• Tableaux &amp; chaînes de caractères</li> <li>• Piles</li> <li>• Interfaçage E/S</li> <li>• Outils de développement pour HCS12 : Débogage.</li> <li>• Sous-programmes. La gestion de la pile. Interfaçage E/S.</li> </ul> <b>TP1 : Introduction au microcontrôleur HCS12.</b> <b>Groupe A : lundi 09 févr. 2026</b> <b>Groupe B : mercredi 11 févr. 2026</b>	09 févr. 2026
6	<b>Introduction à la programmation en langage C</b> <ul style="list-style-type: none"> <li>• Structure générale du langage C</li> <li>• Organisation générale des programmes</li> <li>• Les types, constantes et variables</li> <li>• Les structures de contrôle</li> <li>• Les bibliothèques standards</li> </ul> <b>Présentation de l'énoncé des projets aux étudiants – Le cycle de développement du projet – Les exigences – Sélection des équipes</b>	16 févr. 2026 <b>(Non-présentiel)</b>
7	<b>Arduino : Programmation en langage C</b> <ul style="list-style-type: none"> <li>• Déclaration et utilisation des fonctions</li> <li>• Gestion d'interruptions</li> <li>• Interfaçage E/S avec l'Arduino : DEL, potentiomètre, bouton-poussoir, afficheur 7-segments</li> </ul> <b>TP2 : Programmation en langage assembleur.</b> <b>Groupe A : lundi 23 févr. 2026</b> <b>Groupe B : mercredi 25 févr. 2026</b>	23 févr. 2026 <b>(Non-présentiel)</b>

8	<b>Semaine d'études</b>	02 mars 2026
9	<b>Examen de mi-session</b>	09 mars 2026
10	<b>Arduino : Programmation en langage C (suite)</b> <ul style="list-style-type: none"> <li>• Interfaçage E/S avec l'Arduino : afficheurs LCD, capteur de distance, manette de jeux, boucliers</li> <li>• Interfaçage avec <i>Processing</i></li> </ul> <b>TP3 : Introduction à l'Arduino et à la programmation en langage C.</b> <b>Groupe A : lundi 16 mars 2026</b> <b>Groupe B : mercredi 18 mars 2026</b>	16 mars 2026 <b>(Non-présentiel)</b>
11	<b>Arduino : Programmation avancée en langage C</b> <ul style="list-style-type: none"> <li>• Introduction aux pointeurs</li> <li>• Type du pointeur</li> <li>• Initialisation d'un pointeur</li> <li>• Utilisation des pointeurs</li> <li>• Structures. Unions.</li> </ul> <b>TP4 : Interfaçage E/S avec l'Arduino : DEL et bouton-poussoir.</b> <b>Groupe A : lundi 23 mars 2026</b> <b>Groupe B : mercredi 25 mars 2026</b>	23 mars 2026
12	<b>Arduino : Programmation avancée en langage C (suite).</b> <ul style="list-style-type: none"> <li>• Présentations : sujets spéciaux en programmation avancée en langage C sur Arduino. Discussion sur le projet</li> </ul> <b>TP5 : Arduino : Programmation avancée en langage C.</b> <b>Groupe A : lundi 30 avril 2026</b> <b>Groupe B : mercredi 01 avril 2026</b>	30 mars 2026
13	<b>Lundi de Pâques - Férié</b>	06 avril 2026
14	<b>Présentation des projets d'étudiants</b>	13 avril 2026
15	<b>EXAMEN FINAL</b>	20 avril 2026

## 6. Évaluation du cours :

La pénalité de retard pour la remise d'un travail est de **20 % des points** par jour (y compris les jours fériés et les fins de semaine).

Outils d'évaluation	Pondération	Indicateurs mesurés
Examen de mi-session	20 %	2.2
Examen final	30 %	2.2

Travaux pratiques	<b>25 %</b>	<b>4.3</b>
Projet	<b>25 %</b>	<b>3.2 et 4.4</b>

Par **indicateur mesuré**, on entend qu'à la fin du cours, un niveau de performance (0, 1, 2, 3) est donné pour chaque indicateur et pour chaque étudiant(e), selon la grille ci-dessous :

<b>Indicateurs</b>	<b>Niveau 0</b>	<b>Niveau 1</b>	<b>Niveau 2</b>	<b>Niveau 3</b>
2.2. Formuler un processus de résolution de problème, comprenant des approximations et des hypothèses.	Formulation du processus de résolution inacceptable et traitement inadéquat des approximations et des hypothèses	Formulation du processus de résolution acceptable, mais traitement partiel des approximations et des hypothèses	Formulation du processus de résolution et traitement des approximations et des hypothèses acceptables	Formulation du processus de résolution et traitement des approximations et des hypothèses remarquables
3.2. Mettre en œuvre des investigations documentaires, des expériences et/ou des prototypes.	Mise en œuvre inacceptable	Mise en œuvre partielle	Mise en œuvre acceptable	Mise en œuvre remarquable
4.3. Créer des modèles, simulations, prototypes, et faire des tests.	Création de modèles, simulations, prototypes et/ou exécution de tests inadéquates ou inexistantes	Création acceptable de modèles, simulations, prototypes, mais exécution de tests insuffisante	Création de modèles, simulations, prototypes et/ou exécution de tests adéquats	Création de modèles, simulations, prototypes et/ou exécution de tests remarquables
4.4. Vérifier la conformité de la conception par rapport au cahier des charges.	Vérification inadéquate ou inexistante	Vérification partielle	Vérification acceptable	Vérification exhaustive

## 7. Politiques départementales et institutionnelles :

- Politiques relatives à la tenue des examens
- Note sur le plagiat et les fraudes
- Politique relative à la qualité de l'expression française écrite chez les étudiants et les étudiantes de premier cycle à l'UQO

- Absence aux examens : cadre de gestion, demande de reprise d'examen (formulaire)

Tolérance **ZÉRO** en matière de violence à caractère sexuel.

Le Bureau d'intervention et de prévention en matière de harcèlement (BIPH) a pour mission d'accueillir, soutenir et guider toute personne vivant une situation de harcèlement, de discrimination ou de violence à caractère sexuel. Le BIPH oriente ses actions afin de prévenir les violences à caractère sexuel pour que nous puissions étudier, travailler et s'épanouir dans un milieu sain et sécuritaire.

Vous vivez ou êtes une personne témoin d'une situation de violence à caractère sexuel ? Vous êtes une personne membre de la communauté étudiante ou une personne membre du personnel, autant à Gatineau qu'à Ripon et St-Jérôme, l'équipe du BIPH est là pour vous, sans jugement et en toute confidentialité.

Ensemble, participons à une culture de respect.

Pour de plus amples renseignements consultez UQO.ca/biph ou écrivez-nous au [Biph@uqo.ca](mailto:Biph@uqo.ca)

## 8. Principales références :

### Notes de cours :

- Des notes de cours seront fournies couvrant la matière du cours.

### Référence recommandée :

- H-W. Huang, *HCS12/9S12: An Introduction to Software and Hardware Interfacing*, Delmar Cengage Learning, 2010.

### Autres références :

- F D.A. Patterson, J.L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann, 2013.
- A. Tanenbaum, *Architecture de l'ordinateur*, Pearson, 2005.
- F.M. Cady, *Microcontrollers and Microcomputers: Principles of Software and Hardware Engineering*, Oxford University Press, 2010.
- S. Monk, *Arduino : Les bases de la programmation*, Pearson, 2013.
- C. Tavernier, *Arduino : Maîtriser sa programmation et ses cartes d'interfaces*, Dunod, 2011.

## 9. Page Web du cours :

<https://moodle.uqo.ca>