**Université du Québec en Outaouais** **Département d'informatique et d'ingénierie**

**Sigle : GEN1483 Gr. 01**
**Titre : Systèmes en temps réel**
**Session : Automne 2026** **Horaire et local**
**Professeur : Bennai, Mustapha**

## 1. Description du cours paraissant à l'annuaire :

**Objectifs**

Au terme de cette activité, l'étudiant(e) sera en mesure d'analyser et de concevoir des systèmes informatiques en temps réel.

**Contenu**

Introduction aux systèmes embarqués temps réel : définitions, caractéristiques et applications. Contraintes temporelles : temps réel dur, souple, ferme. Systèmes d'exploitation temps réel (RTOS): planification des tâches, gestion des ressources, synchronisation, communication inter-processus, et exemples de RTOS. Modélisation et vérification : conception basée sur les profils UML, vérification formelle incluant la logique temporelle. Conception conjointe matériel/logiciel : développement de logiciels embarqués avec contraintes temps réel, intégration matérielle pour garantir les performances, algorithmes d'ordonnancement et d'arbitrage, et résolution du problème de l'inversion de priorité (ex : héritage de priorité). Systèmes embarqués en réseau : protocoles pour systèmes critiques (CAN-Bus, AFDX), sécurité et tolérance aux fautes. Études de cas : applications critiques (aéronautique, médical, automobile). Projet de conception : planification, spécification, implémentation, vérification et validation d'un système embarqué temps réel, incluant la conception matérielle et logicielle. Ce cours comporte des séances obligatoires de travaux pratiques (TP) de 3 heures par semaine.

Descriptif – Annuaire

## 2. Objectifs spécifiques du cours :

Le cours couvre 4 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-dagrement) :

**a. Qualité 1 : Connaissances en génie**
**b. Qualité 2 : Analyse de problèmes**
**d. Qualité 5 : Utilisation d'outils d'ingénierie**
**e. Qualité 6 : Travail individuel et en équipe**

**Toutes les qualités énumérées ci-dessus sont mesurées dans ce cours pour fins de rétroaction.**

| Objectifs spécifiques | Qualité | Indicateurs | Introduit | Développé | Appliqué |
|---|---|---|---|---|---|
| Apprendre à identifier ce que c'est qu'un système à temps réel et comprendre les défis de conception et les domaines d'applications. | 1 | 4. Comprendre et appliquer les concepts de l'ingénierie propres au programme. | | X | |
| Analyser et concevoir des systèmes en temps réel en se basant sur des choix logiciels et matériels disponibles selon l'application. | 2 | 3. Choisir un modèle et appliquer l'analyse appropriée pour résoudre un problème. | | X | |
| Apprendre à utiliser les outils logiciels adaptés temps réel et les appliquer à un matériel avec des caractéristiques temps réel. | 5 | 2. Utiliser les outils techniques de mesure, modèles ou simulations appropriés. | | X | |

| | | | | | |
|---|---|---|---|---|---|
| Adopter une approche individuelle dans le cadre d'une recherche poussée sur les composantes logicielles et matérielles existantes. | 6 | 1. Travailler de manière autonome. | | X | |

## 3. Stratégies pédagogiques :

Le cours adopte une approche intégrée reliant les fondements théoriques, l'analyse quantitative, la mise en œuvre sur une plateforme embarquée et la vérification expérimentale des propriétés temporelles.

**1. Mode de prestation**

- Le cours comprend une séance de cours de 3 heures et une séance pratique encadrée de 3 heures selon le calendrier détaillé. La semaine 14 est réservée à une séance unique de 3 heures combinant la présentation PowerPoint et la démonstration pratique du projet.
- Le cours sera dispensé en mode hybride. Les séances de cours prévues les mardis des semaines 4, 10 et 12 se tiendront à distance, par vidéoconférence, tandis que les séances de TP, tenues les mercredis, demeureront en présentiel. Les examens ainsi que la présentation et la démonstration du projet auront également lieu en présentiel. Les liens de connexion aux séances à distance seront diffusés sur Moodle.

**2. Approche pédagogique**

- Exposés structurés accompagnés d'exemples de systèmes critiques.
- Résolution de problèmes d'ordonnancement, de synchronisation, d'arbitrage et d'analyse temporelle.
- Études de cas provenant notamment des domaines automobile, médical et aéronautique.
- Apprentissage progressif par travaux pratiques et projet de conception.
- Mesures expérimentales de période, latence, gigue, temps de réponse, charge et respect des échéances.

**3. Plateforme technologique**

- La plateforme principale du cours est l'Arduino UNO Q.
- Le microcontrôleur STM32U585, exécutant Zephyr, est utilisé pour les fonctions déterministes et les mécanismes RTOS.
- Le processeur QRB2210 sous Linux est utilisé pour la supervision, la journalisation et l'intégration matériel/logiciel.
- Les outils logiciels, bibliothèques et versions utilisés au laboratoire sont figés et validés avant le début du trimestre.

**4. Travaux pratiques**

- Le trimestre comprend 5 séances de travaux pratiques (TP).
- Les TP portent sur l'instrumentation temporelle, l'ordonnancement, la synchronisation, la vérification et les communications critiques.
- Les TP sont réalisés en binôme fixe. Un trinôme peut exceptionnellement être autorisé lorsque l'effectif du groupe l'exige.
- Les rapports de TP sont remis, individuellement, dans les 7 jours suivant la séance. Une pénalité de 20 % par jour de retard s'applique.
- Une absence non justifiée à une séance obligatoire entraîne la note zéro pour l'activité concernée.

**5. Projet de conception**

- Le projet est réalisé en binôme et porte sur un système embarqué temps réel intégrant les composantes matérielles et logicielles. Un trinôme peut exceptionnellement être autorisé lorsque l'effectif du groupe l'exige.
- Il comprend la planification, le cahier des charges, les exigences temporelles, l'architecture, l'implantation, la modélisation, la vérification et la validation.
- Le projet doit comporter, entre autres, des tâches concurrentes, une ressource partagée, une communication interprocessus, une exigence temporelle mesurable, une communication réseau et un scénario de faute.
- L'évaluation comporte une partie collective et une partie individuelle incluant la contribution personnelle et la défense technique.

**6. Compétences visées :**

Au terme des activités d'apprentissage, du projet et des travaux pratiques, la personne étudiante devra être en mesure de :

| Code | Compétence mesurable |
|---|---|
| C1 | Caractériser un système embarqué temps réel et traduire ses besoins fonctionnels, temporels, de sûreté et de sécurité en exigences mesurables. |
| C2 | Modéliser un ensemble de tâches périodiques, sporadiques ou apériodiques et analyser son ordonnançabilité au moyen d'algorithmes et de tests appropriés. |
| C3 | Concevoir et implanter une application multitâche sous un système d'exploitation temps réel en utilisant les tâches, les priorités, les interruptions, les temporisateurs et les mécanismes de communication interprocessus. |
| C4 | Analyser et corriger les problèmes de concurrence, d'interblocage, de famine, d'inversion de priorité et d'accès aux ressources partagées. |
| C5 | Concevoir une architecture conjointe matériel-logiciel, répartir les fonctions entre les composants de calcul et mesurer la latence, la gigue, la charge processeur, l'utilisation mémoire et le respect des échéances. |
| C6 | Modéliser le comportement fonctionnel et temporel d'un système au moyen de machines à états, de diagrammes UML/MARTE et d'automates temporisés, puis vérifier des propriétés de sûreté, de vivacité et d'absence d'interblocage. |
| C7 | Analyser une communication embarquée critique en considérant l'arbitrage, la charge du réseau, les délais de transmission, les erreurs, la tolérance aux fautes et les mécanismes de sécurité. |
| C8 | Concevoir, intégrer, tester et valider un prototype de système embarqué temps réel, puis justifier les choix d'architecture et défendre les résultats obtenus. |

**Travaux et progression**

| Activité | Titre | Compétences dominantes | Production attendue |
|---|---|---|---|
| TP1 | Caractérisation temporelle de l'Arduino UNO Q | C1, C5, C8 | Application fonctionnelle, chronogrammes, mesures de période, de latence et de gigue, puis validation des résultats. |
| TP2 | Tâches Zephyr, priorités, surcharge et échéances | C2, C3, C5 | Application multitâche, modèle de tâches, traces d'ordonnancement, mesures de surcharge et analyse du respect des échéances. |
| TP3 | Synchronisation, communication interprocessus et inversion de priorité | C3, C4, C8 | Application concurrente, utilisation de mécanismes IPC, mise en évidence d'une inversion de priorité et validation de la correction. |
| TP4 | Intégration, instrumentation et validation temporelle d'un contrôleur | C3, C5, C8 | Contrôleur intégré, mesures de latence et de gigue, journalisation, scénario de défaut et rapport de validation temporelle. |
| TP5 | Communication CAN, arbitrage, latence et tolérance aux fautes | C7, C8 | Échange de trames CAN, mesures de charge et de latence, analyse de l'arbitrage, injection d'erreur et validation du comportement. |

| | | | |
|---|---|---|---|
| Projet | Projet de conception d'un système embarqué temps réel | C1 à C8 | Cahier des charges, exigences temporelles, architecture matérielle et logicielle, prototype fonctionnel, modèles, mesures expérimentales, plan de vérification et de validation, présentation PowerPoint, démonstration pratique et défense technique. |

**7. Travail personnel**

- Prévoir en moyenne de 10 à 15 heures par semaine pour la révision, les exercices, la préparation des séances pratiques et l'avancement du projet.

**8. Intégrité académique et usage des outils numériques**

- Toute source externe, y compris le code, les données, les schémas et les contenus générés à l'aide d'outils d'intelligence artificielle, doit être déclarée et citée conformément aux directives du cours et aux politiques institutionnelles présentées à la section 7 de ce plan de cours.
- Chaque étudiant doit être en mesure d'expliquer et de défendre individuellement tout travail soumis.

**9. Ressources et communication**

- Moodle est utilisé pour les notes de cours, les énoncés, les dépôts, les annonces et les forums.
- La plateforme doit être consultée régulièrement. Ceci est de la responsabilité de l'étudiant.
- Préalable académique : INF3723, réussi ou suivi simultanément.

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

- Les disponibilités sont offertes sur rendez-vous à distance, selon une plage convenue avec l'enseignant.
- Les demandes de rencontre doivent être transmises par courriel institutionnel ou par la messagerie Moodle. Les rencontres auront normalement lieu par Microsoft Teams.
- Les questions générales relatives au contenu, aux TP ou au projet doivent d'abord être publiées dans le forum Moodle afin que les réponses puissent bénéficier à l'ensemble du groupe.
- Assistant à l'enseignement : Sara Benseba – bens129@uqo.ca
- Responsable des laboratoires : Abdallah Guire Ali - alixab01@uqo.ca.
- Technicien de laboratoire : Abdelkrim Chebihi - abdelkrim.chebihi@uqo.ca.
- Enseignant : Mustapha Bennai – mustapha.bennai@uqo.ca.

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

| Semaine | Thèmes | Dates |
|---|---|---|
| 1 | **Présentation du plan de cours**<br>**Ch. 1 : Introduction aux systèmes embarqués temps réel**<br>• Définitions, caractéristiques et domaines d'application.<br>• Temps réel dur, ferme et souple.<br>• Contraintes temporelles : période, échéance, latence et gigue.<br>• Architecture générale d'un système embarqué temps réel.<br>• Présentation de l'architecture de l'Arduino UNO Q.<br>→ ***Formation des binômes et publication de l'énoncé du projet.*** | 08 sept. 2026 |
| 2 | **Ch. 1 : Fondements, criticité et exigences temporelles**<br>• Déterminisme, prédictibilité et comportement temporel.<br>• Événements périodiques, sporadiques et apériodiques.<br>• Exigences fonctionnelles, temporelles, de sûreté et de sécurité.<br>• Introduction à la tolérance aux fautes et aux états sûrs. | 15 sept. 2026 |
| 3 | **Ch. 2 : Systèmes d'exploitation temps réel et modèle de tâches**<br>• Architecture et services d'un RTOS.<br>• États des tâches, priorités, préemption et changements de contexte. | 22 sept. 2026 |

| | | |
|---|---|---|
| | • Tâches périodiques, sporadiques et apériodiques.<br>• Interruptions, traitements différés et temporisateurs sous Zephyr.<br>→ ***TP1 : 23 sept. 2026 - caractérisation temporelle de l'Arduino UNO Q.*** | |
| 4 | **Ch. 2 : Algorithmes d'ordonnancement**<br>• Exécutif cyclique, priorités fixes et priorités dynamiques.<br>• Rate Monotonic, Deadline Monotonic et Earliest Deadline First.<br>• Tests d'utilisation et conditions d'ordonnançabilité.<br>• Introduction au WCET et à l'analyse du temps de réponse.<br>→ ***Jalon 1 du projet : planification et exigences.***<br>**SÉANCE DE COURS EN MODE NON PRÉSENTIEL.** | 29 sept. 2026 |
| 5 | **Ch. 2 et Ch. 3 : Analyse temporelle et gestion des ressources**<br>• Analyse du temps de réponse et temps de blocage.<br>• Gestion des surcharges et détection des échéances manquées.<br>• Sections critiques et ressources partagées.<br>• Sémaphores, mutex, événements et files de messages. | 06 oct. 2026 |
| 6 | **Semaine d'études** | 12-16 oct. 2026 |
| 7 | **Examen intra** | 20 oct. 2026 |
| 8 | **Ch. 3 : Concurrence, synchronisation et inversion de priorité**<br>• Communication et synchronisation entre tâches.<br>• Interblocage, famine et conditions de concurrence.<br>• Inversion de priorité.<br>• Héritage de priorité et principe du plafond de priorité.<br>• Allocation mémoire et prévisibilité.<br>→ ***TP2 : 28 oct. 2026 - tâches Zephyr, priorités, surcharge et échéances.***<br>→ ***Jalon 2 du projet : architecture matériel/logiciel et modèle de tâches.*** | 27 oct. 2026 |
| 9 | **Ch. 4 : Conception conjointe matériel/logiciel**<br>• Partitionnement des fonctions entre microcontrôleur et processeur Linux.<br>• Architecture STM32U585–QRB2210 de l'Arduino UNO Q.<br>• Communication entre les domaines temps réel et Linux.<br>• Arbitrage des ressources et interfaces matérielles.<br>• Instrumentation de la latence, de la gigue, de la charge et de la mémoire. | 03 nov. 2026 |
| 10 | **Ch. 4 : Intégration et analyse des performances**<br>• Acquisition, commande, journalisation et supervision.<br>• Temporisateurs, interruptions et périphériques.<br>• Budgets de performance et critères de partitionnement matériel/logiciel.<br>• Mesure de la latence, de la gigue, de la charge processeur et de l'utilisation mémoire.<br>• Watchdog, détection de fautes et fonctionnement dégradé.<br>• Introduction aux principes de sécurité des systèmes embarqués.<br>• ***TP3 : 11 nov. 2026 - synchronisation, IPC et inversion de priorité.***<br>**SÉANCE DE COURS EN MODE NON PRÉSENTIEL.** | 10 nov. 2026 |
| 11 | **Ch. 5 : Modélisation des systèmes embarqués temps réel**<br>• Exigences fonctionnelles et temporelles<br>• Traçabilité des exigences.<br>• Machines à états UML.<br>• Diagrammes de séquence et diagrammes temporels.<br>• Introduction au profil UML/MARTE.<br>• Propriétés de sûreté et de vivacité.<br>• Introduction à la logique temporelle.<br>• Correspondance entre modèle, code et tests.<br>→ ***Jalon 3 du projet : architecture détaillée, modèle comportemental et exigences temporelles.*** | 17 nov. 2026 |
| 12 | **Ch. 5 : Vérification formelle et validation** | 24 nov. 2026 |

| | | |
|---|---|---|
| | • Automates temporisés : états, horloges, gardes et invariants.<br>• Vérification du respect des échéances.<br>• Vérification de l'absence d'interblocage.<br>• Vérification des propriétés de sûreté et de vivacité.<br>• Plan de vérification et de validation.<br>• Validation expérimentale du comportement temporel.<br>• Matrice de traçabilité entre exigences, modèles, implantation et tests.<br>• Comparaison entre résultats analytiques et mesures expérimentales.<br>→ ***TP4 : 25 nov. 2026 - intégration, instrumentation et validation temporelle d'un contrôleur sur Arduino UNO Q.***<br>→ ***Jalon 4 du projet : stratégie complète de vérification et de validation.***<br>**SÉANCE DE COURS EN MODE NON PRÉSENTIEL.** | |
| 13 | **Ch. 6 : Réseaux embarqués critiques : CAN, AFDX et sécurité**<br>• Architecture d'un système distribué temps réel.<br>• Trames, identifiants, priorités et arbitrage CAN.<br>• Charge du bus, temps de transmission, latence et temps de réponse.<br>• Détection des erreurs et confinement des fautes.<br>• Introduction aux réseaux AFDX : liens virtuels, contrôle du trafic et redondance.<br>• Menaces de sécurité et mécanismes de protection des communications embarquées.<br>• Comparaison CAN–AFDX et synthèse à partir de cas automobile, médical et aéronautique.<br>→ ***TP5 : 02 déc. 2026 - communication CAN, arbitrage, mesure de latence et tolérance aux fautes.*** | 01 déc. 2026 |
| 14 | **Présentation et démonstration pratique du projet.**<br>**Présentation PowerPoint**<br>• Problème traité, contexte d'application et principales exigences.<br>• Architecture matérielle et logicielle, modèle de tâches et mécanismes de synchronisation.<br>• Démarche de vérification et de validation, principaux résultats, limites et améliorations possibles.<br>**Démonstration pratique**<br>• Mise en service du prototype et démonstration du fonctionnement nominal.<br>• Présentation d'une mesure temporelle significative et comparaison avec l'exigence correspondante.<br>• Injection d'une faute ou d'une perturbation et démonstration du fonctionnement dégradé, de l'état sûr ou de la reprise contrôlée.<br>**Durée maximale par équipe : 20 minutes, incluant la présentation PowerPoint, la démonstration pratique, les questions et la transition.** | 08 déc. 2026 |
| 15 | **Examen final** | 15 déc. 2026 |

## 6. Évaluation du cours :

L'attribution de la note finale se fera selon la répartition suivante :

| Outils d'évaluation | Pondération | Indicateurs mesurés |
|---|---|---|
| **Travaux pratiques : 5 TP** | 20 % | 1.4, 2.3, 5.2 |
| **Projet de conception avec réalisation pratique** | 20 % | 1.4, 2.3, 5.2 |
| **Examen de mi-session** | 25 % | 1.4, 2.3, 6.1 |
| **Examen final** | 35 % | 1.4, 2.3, 6.1 |

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.

| Indicateurs | Niveau 0 | Niveau 1 | Niveau 2 | Niveau 3 |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1.4 – Comprendre et appliquer les concepts de l'ingénierie propres au programme. | Moins de 52 % | Entre 52 % et 63 % | Entre 64 % et 83 % | Plus de 84 % |
| 2.3 – Choisir un modèle et appliquer l'analyse appropriée pour résoudre un problème. | Choix du modèle et analyse inacceptables. | Choix du modèle acceptable, mais analyse partielle. | Choix du modèle acceptable et analyse adéquate. | Choix du modèle et analyse remarquables. |
| 5.2 – Utiliser les outils, techniques de mesure, modèles ou simulations appropriés. | Utilisation inadéquate ou inexistante. | Utilisation partielle. | Utilisation adéquate. | Utilisation remarquable. |
| 6.1 – Travailler de façon autonome. | Incapable d'effectuer le travail individuel sans assistance. | Effectue le travail individuel avec peu d'assistance. | Effectue le travail individuel sans assistance. | Effectue le travail individuel de façon remarquable sans assistance. |

## 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

## 8. Principales références :


1. Notes de cours GEN1483, Mustapha Bennai.
2. Buttazzo, G. C. Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, 4e éd., Springer, 2024.
3. Kopetz, H. et Steiner, W. Real-Time Systems: Design Principles for Distributed Embedded Applications, 3e éd., Springer, 2022.
4. Burns, A. et Wellings, A. Real-Time Systems and Programming Languages, 4e éd., Addison-Wesley, 2009.
5. Arduino. Arduino UNO Q — documentation matérielle, manuel d'utilisation, fiche technique, brochage et schémas officiels. https://docs.arduino.cc/hardware/uno-q
6. Zephyr Project. Zephyr RTOS Documentation — noyau, ordonnancement, synchronisation, communication interprocessus et prise en charge de l'Arduino UNO Q. https://docs.zephyrproject.org
7. Object Management Group. UML Profile for MARTE, version 1.3. https://www.omg.org/spec/MARTE/1.3
8. ISO 11898-1:2024. Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical coding sublayer.

9. SAE International / AEEC. ARINC Specification 664, Part 7: Avionics Full-Duplex Switched Ethernet Network.

**Arduino UNO Q, ArduinoCore-zephyr, Zephyr RTOS et communications CAN**

| Organisation | Référence technique | Utilité dans le cours | Lien direct |
|---|---|---|---|
| **Arduino** | Arduino UNO Q — page officielle et ressources matérielles | Point d'entrée regroupant le manuel, la fiche technique, le brochage, les schémas, les fichiers CAO et les caractéristiques de la plateforme hybride QRB2210–STM32U585. | Ouvrir la ressource officielle |
| **Arduino** | Arduino UNO Q — manuel d'utilisation | Installation, mise en service, architecture générale, alimentation, connectivité et utilisation des environnements de développement. | Ouvrir la ressource officielle |
| **Arduino** | Arduino UNO Q — fiche technique officielle (PDF) | Description détaillée de l'architecture matérielle, des composants, des interfaces, de l'alimentation et des caractéristiques électriques. | Ouvrir la ressource officielle |
| **Arduino** | Arduino UNO Q — brochage complet (PDF) | Affectation des GPIO, interfaces UART, SPI, I²C, PWM, entrées analogiques et signaux CAN/FDCAN du STM32U585, ainsi que les niveaux logiques et limitations électriques. | Ouvrir la ressource officielle |
| **Arduino** | Arduino UNO Q — schémas électriques officiels (PDF) | Analyse des connexions entre le STM32U585, le QRB2210, les interfaces d'alimentation, les connecteurs, les circuits de communication et les périphériques intégrés. | Ouvrir la ressource officielle |
| **Arduino** | Arduino App Lab — documentation officielle | Environnement de développement pour les projets combinant le MPU sous Linux, le MCU STM32U585, les programmes Arduino et les fonctions de supervision. | Ouvrir la ressource officielle |
| **Arduino** | ArduinoCore-zephyr — dépôt officiel | Code source, installation du paquet Arduino Zephyr Boards, configuration de l'IDE et de la CLI, chargeur, variantes de cartes, dépannage et intégration avec App Lab. | Ouvrir la ressource officielle |
| **Arduino** | ArduinoCore-zephyr 0.56.0 — publication officielle | Version logicielle proposée comme référence pour figer l'environnement du laboratoire et assurer la reproductibilité des TP et du projet. | Ouvrir la ressource officielle |
| **Arduino** | Arduino IDE 2 — documentation officielle | Installation, gestionnaire de cartes, compilation, téléversement, moniteur série et gestion des bibliothèques pour le sous-système MCU. | Ouvrir la ressource officielle |
| **Arduino** | Arduino CLI — documentation officielle | Compilation automatisée, installation des cœurs, gestion des cartes et bibliothèques, téléversement, scripts de validation et reproductibilité de l'environnement logiciel. | Ouvrir la ressource officielle |
| **Zephyr Project** | Arduino UNO Q — documentation officielle de la carte | Description de la cible arduino_uno_q, du STM32U585, des périphériques pris en charge, du débogage et des configurations de compilation Zephyr. | Ouvrir la ressource officielle |
| **Zephyr Project** | Zephyr RTOS — services du noyau | Documentation sur les threads, interruptions, ordonnancement, sémaphores, mutex, événements, files de messages, temporisateurs et gestion de la mémoire. | Ouvrir la ressource officielle |

| | | | |
|---|---|---|---|
| **Zephyr Project** | Zephyr RTOS — ordonnancement | Référence pour les priorités, la préemption, les tâches coopératives, le partage temporel et les mécanismes de sélection des tâches exécutables. | Ouvrir la ressource officielle |
| **Zephyr Project** | Zephyr RTOS — mutex et héritage de priorité | Référence pour les ressources partagées, l'exclusion mutuelle, l'inversion de priorité et le mécanisme d'héritage de priorité. | Ouvrir la ressource officielle |
| **Zephyr Project** | Zephyr RTOS — files de messages | Communication interprocessus entre les threads et les routines d'interruption, avec définition, émission, réception et dimensionnement des files. | Ouvrir la ressource officielle |
| **Zephyr Project** | Zephyr RTOS — fonctions de mesure temporelle | Mesure du temps d'exécution de sections de code afin d'analyser la latence, la charge, la gigue et les performances temporelles. | Ouvrir la ressource officielle |
| **Zephyr Project** | Zephyr RTOS — API CAN | Configuration des contrôleurs et émetteurs-récepteurs CAN, transmission et réception de trames, filtrage, surveillance des erreurs et récupération d'un état bus-off. | Ouvrir la ressource officielle |
| **Zephyr Project** | Zephyr RTOS — exemples CAN | Exemples officiels d'émission et de réception de trames CAN, de fonctionnement en boucle locale et d'utilisation de files de messages. | Ouvrir la ressource officielle |

## 9. Page Web du cours :

https://moodle.uqo.ca