
Communication entre μ C Applications embarquées

GUIDE DE L'ÉTUDIANT

S4e – APP4

Hiver 2024

Département de génie électrique et de génie informatique
Faculté de génie
Université de Sherbrooke

Note : En vue d'alléger le texte, le masculin est utilisé pour désigner les femmes et les hommes.

18 février 2022

Version 1.6 Hiver 2024 – M. Besrour

Version 1.5 Hiver 2023 – Michel L-D

Version 1.4 Été 2022 – Michel L-D

Version 1.3 Hiver 2022 – JP Gouin

Version 1.2 Été 2021 – JP Gouin

Version 1.1 Hiver 2021 – A. C. Therrien

Version 1.0 Été 2020 – J-B Michaud

Tous droits réservés ©2024 Département de génie électrique et de génie informatique, Université de Sherbrooke.

TABLE DES MATIÈRES

1	ACTIVITÉ PÉDAGOGIQUE ET COMPÉTENCES.....	1
2	Synthèse de l'évaluation	2
3	Qualités de l'ingénieur	2
4	Énoncé de la problématique	3
4.1	Architecture matérielle	5
4.2	Schéma de séquence de l'application.....	7
4.3	Autres spécifications	8
5	Connaissances nouvelles	9
6	Guide de lecture	10
6.1	Connaissances antérieures à réviser	10
6.2	Nouvelles connaissances	10
7	Logiciels et matériel.....	12
8	Santé et sécurité.....	12
8.1	Dispositions générales	12
8.2	Dispositions particulières.....	12
9	Sommaire des activités.....	13
10	Production à remettre	14
10.1	Planification MVP et tests d'intégration	14
10.2	Présentation.....	14
10.3	Validation.....	15
10.4	Dépôt	15
11	Évaluations.....	16
11.1	Production à remettre	16
11.2	Évaluation sommative de l'unité.....	17
11.3	Évaluation sommative finale.....	17
12	Politiques et règlements.....	17
13	Intégrité, plagiat et autres délits.....	17
14	procédurale 1	18
15	laboratoire	20
16	procédurale 2	22

LISTE DES FIGURES

Figure 1 : Architecture matériel.....	5
---------------------------------------	---

LISTE DES TABLEAUX

Tableau 1 : Synthèse de l'évaluation de l'unité	2
Tableau 2 : Tableau des qualités de l'ingénieur	2
Tableau 3 : Sommaire de l'évaluation de la présentation et de la validation	16

1 ACTIVITÉ PÉDAGOGIQUE ET COMPÉTENCES

GEL452 – Microcontrôleurs

1. Programmer et déployer un microcontrôleur.
2. Mettre en oeuvre et employer une méthodologie de développement de systèmes embarqués à microcontrôleur et ses applications, en utilisant des outils de développement physique et logiciel.

Description officielle : <https://www.usherbrooke.ca/admission/fiches-cours/GEL452/>

GEL442 – Logique programmable et interfaces

1. Modéliser et concevoir des interfaces numériques par une représentation mathématique de l'information discrète et par des méthodes de synthèse des équations booléennes.
2. Réaliser des interfaces numériques en fonction de critères de performance, évaluer la compatibilité électrique d'interfaces numériques à signaux différentiels, mesurer les performances d'échange d'informations numériques.

Description officielle : <https://www.usherbrooke.ca/admission/fiches-cours/GEL452/>

2 SYNTHÈSE DE L'ÉVALUATION

Tableau 1 : Synthèse de l'évaluation de l'unité

Évaluation	GEL442-1	GEL442-2	GEL452-1	GEL452-2
Validation d'APP et livrables associé	40	40	30	30
Évaluation formative théorique				
Évaluation formative pratique				
Évaluation sommative théorique/pratique	60	60	45	45
Évaluation finale	100	100	100	100
Total	200	200	175	175

3 QUALITÉS DE L'INGÉNIEUR

Les qualités de l'ingénieur visées par cette unité d'APP sont les suivantes. D'autres qualités peuvent être présentes sans être visées ou évaluées dans cette unité d'APP. Pour une description détaillée des qualités et leur provenance, consultez le lien suivant :

<https://www.usherbrooke.ca/genie/etudiants-actuels/au-baccalaureat/bcapg/>

Tableau 2 : Tableau des qualités de l'ingénieur

Numéro	Libellé	Touchée	Évaluée
Q01	Connaissances en génie	✓	✓
Q02	Analyse de problèmes	✓	✓
Q03	Investigation		
Q04	Conception		
Q05	Utilisation d'outils d'ingénierie	✓	✓
Q06	Travail individuel et en équipe		
Q07	Communication		
Q08	Professionalisme		
Q09	Impact du génie sur la société et l'environnement		
Q10	Déontologie et équité		
Q11	Économie et gestion de projets		
Q12	Apprentissage continu		

4 ÉNONCÉ DE LA PROBLÉMATIQUE

LEDanse.inc

Vous et deux de vos collègues en génie électrique avez démarré une nouvelle entreprise en sonorisation et lumière pour les DJ. Votre idée innovatrice est de concevoir un module avec accéléromètre sur une manette que vous gardez dans vos poches et qui permet de transmettre sans fil vos mouvements de danse. Les variations d'accélération en XYZ contrôleront les jeux de lumières RGB. Votre entreprise est en plein croissance, votre stagiaire en administration a signé plusieurs contrats de vente, mais vous n'avez toujours pas débuté la conception de votre produit. Il vous faudra travailler vite puisqu'il ne vous reste que quelques jours avant votre première démo chez un client.

Pour ce faire, vous débuterez la conception sur deux processeurs embarqués, un PIC32 et un MicroBlaze, sur lesquels vous déploierez une application distribuée et une communication par protocole UDP entre les deux. Le PIC32, sur la carte de développement Basys MX3, sera le « client ». Dans notre application, le client saisi des « données » qu'il doit faire traiter par le « serveur », soit le MicroBlaze instancié sur la Zybo Z7-10. Les données traitées sont retournées au client qui les utilisera localement. Sur chaque processeur, vous développerez une application multitâche utilisant une machine d'états logicielle coopérative plutôt qu'un système d'exploitation préemptif.

Dans ce cas-ci, les « données » sont les accélérations en X, Y et Z de l'accéléromètre de la carte MX3, lues à 400 échantillons par seconde par le bus I2C, puisque l'accéléromètre génère une interruption lorsque de nouvelles données sont prêtes. Les données seront récupérées durant la routine d'interruption sur le PIC32, ceci permettra de protéger la communication I2C des éventuelles coupures des autres interruptions. Les données seront assemblées en paquets de 40 données XYZ¹, puis transmises au MicroBlaze. Vous devrez formater vos données, puisque la communication transmet des octets. Les données retournées au client après filtrage seront récupérées par la suite converties en modulation de largeur d'impulsion (*Pulse width modulation, PWM*) pour produire, sur la LED RGB, une intensité lumineuse proportionnelle aux accélérations XYZ mesurées. Il faudra évaluer les temps de transmission et de traitement (latence du système) afin d'accumuler assez de données pour éviter que la LED s'éteigne en attendant un paquet.

Vous implémenterez le filtrage des données du côté FPGA en logique programmable et non en logiciel, en utilisant tout module, librairie ou code VHDL que vous jugerez pertinent, à partir d'un module « My_IP ». Ce module déjà instancié, esclave sur le bus AXI de Xilinx, possède initialement 4 registres intégrés et un fanion (*flag*) indiquant qu'un registre a été modifié.

¹ Les données doivent être sous forme de string et non seulement de chaîne de caractères, sinon cela peut fausser le calcul de la fonction `strlen` donnant nombre d'octets à transmettre. Pour résoudre ce problème, vous devez utiliser la fonction `sprintf()`, `memset()` ou `strncpy()` pour créer, modifier ou copier des chaînes de caractères. Il est aussi possible de fixer le nombre d'octets à transmettre.

Une fois le traitement terminé, ce module écrira le résultat dans un autre registre et déclenchera une interruption au MicroBlaze. Vous porterez une attention particulière au fait que les données parviennent au Zynq en paquets ; or, un filtre opère en continu, sans discontinuité entre les paquets.

La communication utilisera une pile (*stack*) TCP/IP déjà codée (bibliothèque Microchip Harmony pour le PIC32, CycloneTCP pour le Microblaze), implémentant plusieurs couches du modèle OSI. Pour simplifier le déploiement, vous utiliserez une adresse IP fixe dont le préfixe sera 192.168.13.0/24. Pour les deux processeurs, la couche d'accès au réseau IEEE 802.3u (*Media access controller, MAC*) et la couche physique (*Physical layer, PHY*) sont fournies par le même circuit intégré (ENC424J600) sur le PMOD NIC100 de Digilent, interfacé par bus SPI. Vous aurez besoin d'un câble Ethernet à paires croisées (*crossover*) puisque que ce PHY ne supporte pas la négociation automatique auto-MDIX.

Un code de départ partiellement fonctionnel est fourni pour les deux plateformes. Le développement doit respecter les règles de l'art en termes de planification de l'architecture du code, du plan de validation et d'intégration. À cet égard, vous devrez faire des diagrammes UML nécessaires pour la conception de votre application avant de débiter à coder. De plus, vous devez fournir un plan d'intégration (MVP) des modules à concevoir en y intégrant des tests pour vérifier la fonctionnalité de votre système. Des plans de tests modulaires sont également demandés pour la gestion de l'accéléromètre sur le client et le module de filtrage sur le serveur, pour lequel vous devrez implémenter un banc de test VHDL.

Finalement, vous devrez démontrer la fonctionnalité de votre système complet et par module. On y trouvera des mesures électriques SPI, I2C et broche état logiciel. De plus, vous devrez mesurer le temps d'exécution de votre boucle principale sur la carte MX3. Cette mesure pourra se faire avec les changements d'états de la patte du JB1 du PMOD B et la LED0.

4.1 Architecture matérielle

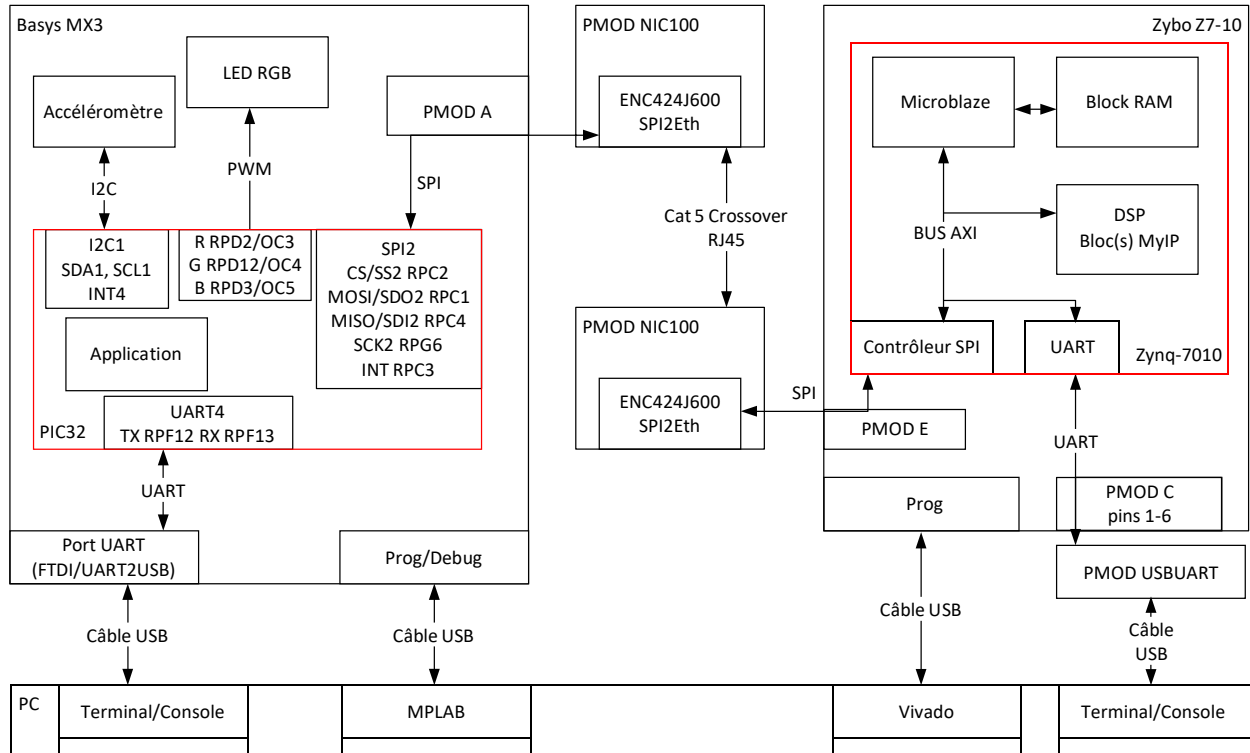
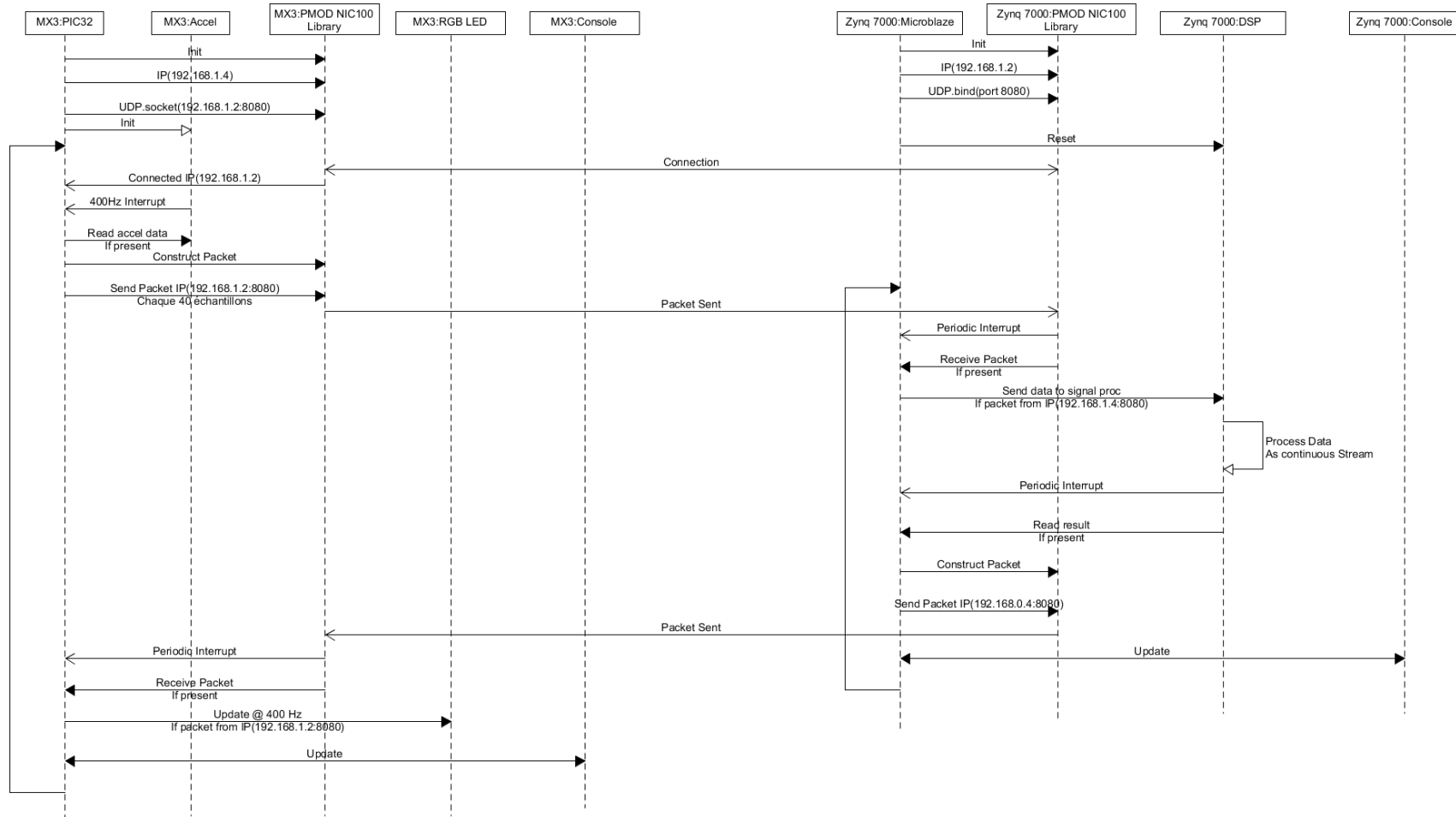


Figure 1 : Architecture matériel

PÉRIPHÉRIQUES UTILISÉS		
Processeur	Périphérique	Détails
PIC32	UART4	Baudrate 921600 bps, 8bits, 1stop, no parity Console
	Timer1	Base de temps pour la pile TCP/IP
	Timer2	Base de temps pour la génération PWM (LED RGB)
	Timer4	Base de temps pour l’affichage 7-segments
	Timer5	Base de temps pour générer des délais (e.g. dans la librairie LCD)
	Output Compare 3, 4, 5	Génération PWM pour les composantes RGB de la LED
	INT4	Interruption externe de l’accéléromètre
	PMODB patte 1 (JB1) et LED0	Heartbeat coopératif
	I2C1	400 kHz, accéléromètre
	SPI2	10 MHz, ENC424J600 sur PMODA
	INT2	Interruption externe, ENC424J600
MicroBlaze	UART Lite	Baudrate 115 200 bps, 8bits, 1stop, no parity Console via PMOD USBUART sur PMOD C
	SPI	ENC424J600 sur PMOD E
	myIP_AXI	Bloc de départ pour l’interface DSP
	Block RAM	Mémoire du Microblaze 128KB, organisée en 32768 x 32 Programme et données
	Interrupt Controller	Attention de laisser l’interruption du ENC424J100 « Level trigger » et « Active Low »

4.2 Schéma de séquence de l'application



4.3 Autres spécifications

Paquets envoyés (mots de 32 bits)

Index de séquence (entier non signé 32 bits incrémenté à chaque paquet)
Données accéléromètre X 40 entiers signés de 32 bits
Données accéléromètre Y 40 entiers signés de 32 bits
Données accéléromètre Z 40 entiers signés de 32 bits

Chaque donnée transmise accéléromètre

31 : 12	11 : 0
Extension de signe	Donnée lue (entier 12 bits signé)

DSP

La formule à calculer pour chaque axe de l'accéléromètre est le FIR suivant (en fait une moyenne mobile de l'amplitude)

$$y[k] = \frac{1}{N} \sum_{i=0}^{N-1} |x[k - i]|$$

avec $N = 8$, note². Il ne doit y avoir aucune interruption entre 2 paquets, i.e. quand pour un paquet donné $k - i < 0$ il faut remonter aux données du paquet précédent.

² Tirez profit de la division par une puissance de 2, qui en binaire équivaut à un décalage de bits à droite.

5 CONNAISSANCES NOUVELLES

GEL442

Connaissances déclaratives (QUOI ?) :

- Connaître les couches réseaux matérielles (PHY), accès média (MAC), interface réseau (IPv4), et transport (UDP) pour le protocole Ethernet 100 Mbps.

Connaissances procédurales (COMMENT?) :

- Utiliser un stack TCP/IP.
- Utiliser des bibliothèques logicielles à plusieurs couches pour périphériques.
- Concevoir et implémenter un périphérique propriétaire en VHDL.
- Utiliser un contrôleur d'interruption.

Connaissances conditionnelles (QUAND?) :

- Quand utiliser un signal différentiel plutôt qu'unipolaire.

GEL452

Connaissances déclaratives (QUOI ?) :

- Environnement embarqué multitâches coopératif.
- Environnement et outils de développement logiciel; bonnes pratiques de programmation.
- Intégration logiciel-matériel.

Connaissances procédurales (COMMENT?) :

- Exploiter un ensemble de périphériques dans le cadre d'une application complexe.
- Structurer une application coopérative.
- Structurer une application distribuée.

Connaissances conditionnelles (QUAND?) :

- Structurer les priorités d'interruption en fonction de la criticité des opérations.

6 GUIDE DE LECTURE

6.1 Connaissances antérieures à réviser

I2C

- <http://www.ti.com/lit/an/slva704/slva704.pdf>
- https://www.i2c-bus.org/fileadmin/ftp/i2c_bus_specification_1995.pdf

SPI

- <https://www.analog.com/media/en/technical-documentation/application-notes/AN-1248.pdf>

Processus d'intégration en cascade, en V ou MVP

Plans de vérification, plans de tests et banc de test VHDL

6.2 Nouvelles connaissances

Procédural 1

	Document	Notes
#1	Embedded multitasking with small MCUs (partie 1 & partie 2)	Design de machine d'état pour système embarqué multitâche
#2	Multitasking	Sur le site de l'APP Sauf sections sur Unix et VxWorks (trop spécifique)
#3	Datasheet PIC32	p. 139 à 144 (Peripheral Pin Select)

Laboratoire

	Document	Notes
#5	MPLAB Harmony (PIC32)	p. 436 <i>Manager Module Library Interface</i> p. 675 <i>UDP Module Library Interface</i>
#6	CycloneTCP (Microblaze)	Pour le MX3, les fonctions les plus importantes sont dans core/udp, core/socket, core/ethernet et core/ip.

Procédural 2

	Document	Notes
#7	A Beginner's Guide to Ethernet 802.3	Au complet
#8	Ethernet Theory of Operation	Au complet
#9	Ethernet Technology	Pages 8-16
#10	Ethernet	Jusqu'à la page 25
#11	<i>Réseaux</i> , Tenenbaum	en PDF sur le site web
	Couches de la pile Ethernet	Au complet, en diagonal d'abord puis en détail pour le formatif
#12	Ethernet System Hardware	Pages 1-12, 16-23
	Bus MII	
#13	IEEE 802.3 Ethernet	Jusqu'à la page 9
	Introduction aux PHY	
#14	Differential Signaling	
#15	ENC424J600/624J600 Data Sheet	Figure 1-1, sections 2.4, 2.7 en mode SPI, sections 6 et 12
#16	Transformateurs Ethernet	Au complet
#17	Pinout RJ45	Page 3

7 LOGICIELS ET MATÉRIEL

Vous avez besoin de la carte MX3 et de la carte FPGA pour faire l'APP, ainsi que leur logiciel de programmation (MPLAB et Vivado/Vitis).

8 SANTÉ ET SÉCURITÉ

8.1 Dispositions générales

Dans le cadre de la présente activité, vous êtes réputés avoir pris connaissance des politiques et directives concernant la santé et la sécurité. Ces documents sont disponibles sur les sites web de l'Université de Sherbrooke, de la Faculté de génie et du département. Les principaux sont mentionnés ici et sont disponibles dans la section Santé et sécurité du site web du département: <http://www.gel.usherbrooke.ca/santesecurite/>.

- Politique 2500-004 : Politique de santé et sécurité en milieu de travail et d'études
- Directive 2600-04 : Directive relative à la santé et à la sécurité en milieu de travail et d'études
- Sécurité en laboratoire et atelier au département de génie électrique et de génie informatique

8.2 Dispositions particulières

Si nécessaire, il faut mettre les dispositions spécifiques à l'unité APP pour les questions de santé et sécurité. Sinon, cette section peut être omise et il n'est pas nécessaire de numéroter et de titrer la section précédente, celle des dispositions générales.

9 SOMMAIRE DES ACTIVITÉS

Semaine 1

- Première rencontre de tutorat
- Étude personnelle et exercices
- Formation à la pratique procédurale 1
- Formation à la pratique en laboratoire
- Formation à la pratique procédurale 2

Semaine 2

- Étude personnelle et exercices
- Formation à la pratique en laboratoire : validation pratique de la solution
- Présentation orale et validation
- Deuxième rencontre de tutorat
- Évaluation formative théorique et pratique
- Évaluation sommative théorique et pratique

10 PRODUCTION À REMETTRE

- La résolution de la problématique se fait en équipe de 3.
- Les équipes d'APP doivent se faire sur Moodle.
- La présentation doit être remise avant 23h59 la veille de la présentation.
- Le code doit être remis après votre validation et avant la présentation.
- Pénalités de 10% par journée pour les retards
- Productions originales, sinon pénalité

10.1 Planification MVP, tests d'intégration et machines à états (FSM)

Décrire la planification de votre intégration (MVP) incluant un plan de tests indiquant comment vous comptez tester chaque étape d'intégration ainsi que les résultats attendus.

- Tests d'intégration : Maximum 1 pages et sous forme d'un tableau.
 - Chaque ligne du tableau doit indiquer une fonction à intégrer et au minimum un test / résultat attendu.
- Planification MVP
- Diagrammes brouillon des machines à états finis
- **Remis avant le procédural 2 afin d'être commenté en grand groupe à la fin du procédural 2.**

10.2 Présentation

Vous devez présenter votre solution pour la problématique sous forme de présentation orale d'une durée de 10 minutes. Le tuteur posera des questions afin de valider votre compréhension. L'horaire des présentations sera disponible sur la page Moodle de l'APP. Voici les sujets que vous devez aborder durant votre présentation.

- Survol de l'architecture matériel avec schéma bloc :
 - Ce court survol sert de carte d'orientation pour la suite de votre présentation indiquant où chaque bloc se situe dans le système global.
 - Quels blocs sont gérés ou génèrent des interruptions.
- Conception et validation du bloc DSP avec architecture du matériel, des machines d'états matérielle/logicielle et les problèmes rencontrés lors des tests d'intégration.
- Architecture logicielle : à l'aide d'organigrammes, de diagramme de séquence, de diagramme d'états ou de tout autre support approprié, discutez des **modifications apportées au code de départ et des résultats obtenus.**
 - Machines d'états logicielles (MX3 et Micro Blaze)
 - Interruptions (MX3 et Micro Blaze)
 - Échanges de données par UDP (MX3)
 - Traitement des paquets par myIP et DSP (Zybo)
 - Intégration LED RGB (MX3)
- Synthèse du plan de vérification de la problématique.

10.3 Validation

L'objectif ici est de démontrer que vous avez maîtrisé et complété le mandat selon les spécifications fournies.

Assurez-vous aussi d'avoir vos projets ouverts dans les environnements de développement appropriés pour consultation du code, votre matériel fonctionnel et déjà programmé, ainsi que l'analyseur logique branché. **Vous pourrez entrer 10 minutes avant votre heure de passage pour vous installer.**

- Démonstration générale du fonctionnement (au moyen d'affichages, console, etc.)
- Interprétation des signaux à l'aide de l'analyseur logique ou d'un oscilloscope :
 - Capturez une trame de communication SPI en lien avec le ENC424J600 et interprétez-la d'après la fiche technique du composant.
 - Capturez une trame de communication I2C en lien avec l'ACL et interprétez-la d'après la fiche technique du composant.
 - Observer le signal représentant le temps d'exécution de la boucle principal.
- Démontrez la conformité des fonctionnalités suivantes dans le debugger ou un autre moyen approprié :
 - Regroupement des données par paquets de 40
 - Envoi/réception UDP par connexion unique
 - Filtrage en continu sans interruption entre les paquets (banc de tests Vivado)
 - Contrôle LED RGB

10.4 Dépôt

Votre planification, votre présentation et vos codes doivent être déposés sur le serveur de dépôt dans les dossiers respectifs. Assurez-vous que votre archive porte le nom *cip1-cip2-cip3.zip*. L'absence de dépôt entraîne la note 0, peu importe le résultat de l'oral/validation. Tout retard est assorti d'une pénalité.

11 ÉVALUATIONS

11.1 Production à remettre

L'évaluation des productions à remettre portera sur les compétences figurant dans la description des activités pédagogiques. La pondération des différents éléments est indiquée au tableau suivant. L'évaluation est directement liée aux livrables demandés à la section 10 et le tableau suivant y réfère à l'aide d'une courte description.

Tableau 3 : Sommaire de l'évaluation de la présentation et de la validation

Éléments demandés	GEL442-1	GEL442-2	GEL452-1	GEL452-2
Présentation				
Survol de l'architecture matériel avec schéma bloc				10
Synthèse du plan de tests d'intégration				10
Architecture logicielle :				
- Machines d'états (MX3 et Micro Blaze)	8			10
- Interruption (MX3 et Micro Blaze)	8			
- UDP (MX3)	8			
- MyIP et DSP (Zybo)	8			
- RGB (MX3)	8			
Validation				
Démonstration générale du fonctionnement			10	
Analyseur logique				
- Trame SPI		8		
- Trame I2C		8		
- Temps d'exécution de la boucle principale			10	
Démonstrations spécifiques :				
- Paquet de 40 données			10	
- Envoi/réception UDP		8		
- Filtrage en continu sans interruption (banc de tests)		8		
- Contrôle LED RGB		8		
Total	40	40	30	30

Quant à la qualité de la communication technique, elle ne sera pas évaluée de façon sommative, mais si votre présentation est fautive sur le plan de la qualité de la communication et de la présentation, il vous sera retourné et vous devrez le reprendre pour être noté.

11.2 Évaluation sommative de l'unité

L'évaluation de l'unité d'APP4 est un examen écrit qui porte sur tous les éléments de compétences de l'unité. Il s'agit d'une évaluation théorique. C'est un examen qui se fait avec la documentation sur ordinateur. L'évaluation porte sur GEL442 et GEL452 tels que vus pendant l'unité.

11.3 Évaluation sommative finale

L'évaluation finale est un examen écrit qui porte sur tous les éléments de compétences de l'unité. Il s'agit d'une évaluation théorique. L'évaluation porte sur GEL442 et GEL452 tels que vus pendant l'unité et elle se fait sans documentation.

12 POLITIQUES ET RÈGLEMENTS

Dans le cadre de la présente activité, vous êtes réputés avoir pris connaissance des politiques, règlements et normes d'agrément suivants.

Règlements et politiques de l'Université de Sherbrooke

- Règlement des études
<https://www.usherbrooke.ca/registraire/>

Règlements facultaires

- Règlement facultaire d'évaluation des apprentissages / Programmes de baccalauréat
- Règlement facultaire sur la reconnaissance des acquis

Norme d'agrément

- Informations pour les étudiants au premier cycle :
<https://www.usherbrooke.ca/genie/etudiants-actuels/au-baccalaureat/bcapg>
- Informations sur l'agrément :
<https://engineerscanada.ca/fr/agrement/a-propos-de-l-agrement>

Si vous êtes en situation de handicap, assurez-vous d'avoir communiqué avec le Programme d'intégration des étudiantes et étudiants en situation de handicap à l'adresse de courriel prog.integration@usherbrooke.ca.

13 INTÉGRITÉ, PLAGIAT ET AUTRES DÉLITS

Dans le cadre de la présente activité, vous êtes réputés avoir pris connaissance de la déclaration d'intégrité relative au plagiat :

<https://www.usherbrooke.ca/ssf/antiplagiat/jenseigne/declaration-dintegrite/>

14 PROCÉDURALE 1

Buts de l'activité

- Comprendre comment décortiquer la documentation pour répondre aux questions suivantes.
- Comprendre comment appliquer la théorie des applications multitâches coopératives.
- Apprendre par un exemple et un contre-exemple à construire une machine d'états coopérative.

Exercice 1 Interfaçage d'un périphérique

- À partir du schéma de la MX3 et de la fiche technique du PIC32 et de la fiche technique de l'accéléromètre, trouvez comment sont brancher l'accéléromètre et le PIC32. Faites un dessin du branchement.
 - À quoi ressemble le « data flow » de la collecte de données de l'accéléromètre? (Décrivez les grandes lignes, pas besoin de détail exhaustif).
- À partir de la fiche technique de l'accéléromètre, déterminer comment configurer la génération automatique d'un cycle de lecture des données qui génère une interruption.
 - Quelles sont les spécifications visées de l'accéléromètre pour l'APP?
 - Dans quelle section de la fiche technique peut-on trouver la liste des registres et leur description?
 - Quelles sont les registres à modifier pour configurer l'accéléromètre?
 - Quelles sont les registres à lire?
- À partir du schéma de la MX3 et de la fiche technique du PIC32, configurez les pattes et le gestionnaire d'interruption pour recevoir l'interruption de l'accéléromètre.
 - Qu'est-ce qu'une pince « Remappable »? Donnez un exemple de périphérique « remappable »?
 - Est-ce que le périphérique I2C est « remappable »?
 - Qu'est-ce que l'INT4?
- Configurer et faites le pseudo-code de la routine d'interruption adéquatement pour recevoir les données.
 - À quel endroit dans le code on configure l'interruption?
 - À quel endroit dans le code on configure l'accéléromètre?
 - À quel endroit dans le code on configure la routine d'interruption?

Exercice 2 Applications multitâches

- Quelle est le concept de base d'un modèle collaboratif ? Comment construit-on une machine d'états logicielle collaborative?
 - Que 'est-ce qu'une « tâche » du point de vue d'un microcontrôleur/microprocesseur?
 - À quoi fait-on référence quand on parle de "multitâches" dans le contexte de programmation embarquée?
 - Pourquoi réfléchir à l'architecture avant de commencer à programmer?
 - Pourquoi répartir/diviser le programme en plusieurs petites tâches?
 - Comment est-ce qu'un processeur change de tâche ?
- Comment s'assure-t-on que l'architecture ne sera pas « bloquante » lors de la gestion des tâches ou des périphériques?

- c) Faites un tableau comparatif entre : Système d'exploitation préemptif, Système d'exploitation collaboratif et machine d'état collaborative ?

Exercice 3 I2C et Latence

- a) Commentez l'architecture de la librairie I2C fournie par Digilent, en particulier la fonction d'écriture.
- b) La fonction I2C en réception bloque combien de temps?

Exercice 4 Machine à états d'une application

- a) Construisez une machine d'états logicielle qui permette de traiter un paquet du côté serveur dans la problématique.

Exercice 5 PWM

- a) Expliquez succinctement l'architecture du périphérique « OC ». Quel est le rôle du timer dans son fonctionnement?
- b) Expliquez les principes d'une modulation PWM pour varier l'intensité d'un dispositif de puissance.

15 LABORATOIRE

Buts de l'activité

- Être capable d'utiliser l'analyseur logique pour capturer un protocole SPI, et d'interpréter sommairement son contenu ; comprendre comment trouver cette dernière information dans la documentation. Comprendre comment appliquer la théorie des applications multitâches coopératives.
- Faire la rétroingénierie des applications coopératives fournies dans le code de départ.
- Comprendre la structure du code de départ, la séquence des étapes qui sont réalisées et cibler les endroits à modifier pour la problématique.
- Identifier les fonctions utiles, comprendre leurs rôles et les utiliser dans une machine logicielle.

Exercice 1 Code de départ et Client/Serveur UDP

- À partir du code de départ sur MPLab de la problématique pour la carte MX3 fourni sur la page web de l'APP4 et en vous basant sur le guide d'utilisation du code de départ.
 - Trouvez la fonction « main ».
 - Où sont faites les « initialisations » et les « tâches » ? Quelle sont les différentes « tâches » et que font-elles ?
 - Les tâches sont-elles exécutées en parallèle ou en série ? De quel type d'application s'agit-il ?
 - Dessinez le diagramme d'état de la fonction main
- Client/Serveur UDP :
 - Trouvez les fonctions qui gèrent la communication UDP. Il existe une fonction côté Client et une fonction côté Serveur.
 - Construisez le diagramme de séquence et la machine d'états logicielle existante du Client/Serveur UDP, des différents affichages et de la gestion de la console.
- Périphérique et Interruption
 - Trouvez où sont programmés les registres de configuration de l'accéléromètre ? Dans quel mode l'accéléromètre est-il configuré ? Configurez-le dans l'état approprié pour la problématique.
 - Quels sont les routines d'interruption programmées dans le μ Contrôleur ? À quel signal (Vector) sont-ils liés ?
 - Fait le diagramme UML de la routine d'interruption de lien à l'accéléromètre.

Exercice 2 Architecture matérielle et le flot de données

- À partir du code fourni pour le FPGA, construisez l'architecture matérielle et le flot de données existant pour le traitement du signal, en particulier :
 - Comment est structuré le module My_IP ?
 - Comment le traitement est-il déclenché dans ce module ?
 - Comment est générée l'interruption qui signale au logiciel que le traitement est terminé ?
 - Comment sont configurées l'ISR et les interruptions dans le code ?
- À partir du code fourni, construisez le diagramme de séquence et la machine d'états logicielle existante de « l'écho » UDP, de la gestion de la console et du filtrage My_IP.

Exercice 3 Piles

- a) Ciblez, pour les deux piles fournies (PIC32 et Microblaze), les fonctions utiles pour la couche application qui désire utiliser un socket UDP.
- b) Ébauchez une machine logicielle simple qui permette d'utiliser un tel socket.

Exercice 4 DEL RGB

Modifier le code de départ pour contrôler l'état de la DEL RGB en utilisant la SW0.

Si SW0 = 0, afficher la DEL RGB rouge seulement avec des intensités de 0 à 250 par bonds de 50 à chaque seconde et si SW0 = 1, afficher la DEL RGB rouge, verte, bleue et blanche en alternance à chaque seconde.

NOTE : Allez voir la documentation de Digilent et la librairie de code fournie permettant de configurer les interruptions pour le PWM.

Exercice 5 Trame I2C

Capturez une trame I2C avec l'analyseur logique (*Analog Discovery*). Essayer d'en décoder le contenu, en vous servant de la fiche technique de l'accéléromètre.

16 PROCÉDURALE 2

Exercice 1 Modèle OSI, pile Ethernet

Définissez chaque couche de ces modèles de communication et leur utilité.

- a) Couche physique
 - i) Qu'est-ce que MDI vs MDI-X (connexion directe vs crossover)?
 - ii) Qu'est-ce qu'un signal différentiel? Pourquoi et quand utilise-t-on une communication différentielle? Pourquoi ce signal différentiel est-il isolé par un transformateur aux deux bouts?
 - iii) Discutez de la structure du paquet physique Ethernet et de la fonction des différents champs.
- b) Couche de connexion
 - i) Discutez de la structure du Paquet MAC et de la fonction des différents champs.
- c) Couche réseau
 - i) Comment est structuré un réseau Ethernet?
 - ii) Qu'est-ce qu'une adresse IP? IPv4? Un masque réseau?
 - iii) Discutez de la structure d'un Paquet IPv4.
- d) Couche de transport
 - i) Quelle est la différence entre une connexion TCP et UDP?
 - ii) Discutez de la structure d'un paquet UDP.
 - iii) Expliquez la notion de port dans la couche de transport.
 - iv) Qu'est-ce qu'un socket?
 - v) Donnez un exemple du processus de connexion UDP client/serveur.
- e) Application
 - i) Donnez un exemple du comportement des piles TCP/IP respectives du client et du serveur lors d'une communication complète, par exemple l'accès à une page web.
 - ii) Expliquez pourquoi vous pouvez être votre propre destinataire dans un modèle OSI.

Exercice 2 Adressage et réseau

- a) Combien d'adresses IP différentes et valides sont contenues dans un sous-réseau dont le masque de sous-réseau IPv4 est 255.250.255.0?
- b) Lesquelles des adresses IP suivantes sont dans le même sous-réseau, dont le masque de sous-réseau est 255.255.90.0?

192.168.3.78, 192.168.15.201, 192.168.11.47, 192.168.19.2

- c) Donnez un masque de sous-réseau et l'adresse du sous-réseau qui feraient en sorte que les deux adresses IP suivantes soient dans le même sous-réseau. Faites le masque avec le plus de bit à « 1 » possible.

128.152.88.249, 120.69.42.1

Exercice 3 Plan de validation MVP

Comparer les plans de validations des étudiants.