UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

ÉLECTRONIQUE DE COMMANDE POUR SYSTÈMES ÉLECTROMÉCANIQUES EN AGRICULTURE DE PRÉCISION

MÉMOIRE PRÉSENTÉ

COMME EXIGENCE PARTIELLE DE LA

MAÎTRISE GÉNIE ÉLECTRIQUE

PAR

LECTICIA BRINDA KAMENI

JUILLET 2026

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MAITRISE EN GÉNIE ÉLÉCTRIQUE

Direction de recherche :

Prof. Alben Cardenas — Directeur de recherche

Jury d'évaluation

Dr. Mauricio Higuita Cano
Supra Énergie inc. — Évaluateur externe

Prof. Mohsen Kandidayeni
Université du Québec à Trois-Rivières — Évaluateur

Prof. Alben Cardenas
Université du Québec à Trois-Rivières — Directeur de recherche

Résumé

Ce mémoire porte sur la conception et la mise en œuvre de solutions d'électronique de commande appliquées aux systèmes électroniques utilisés en agriculture de précision. L'objectif est de proposer des architectures de commande adaptées aux contraintes des systèmes embarqués agricoles, en tenant compte des performances énergétiques, du traitement temps réel et de la communication entre les différents sous-systèmes.

Une étude théorique des principaux composants d'un système électromécanique est présentée, incluant les actionneurs, les capteurs de courant, de position et de vitesse, les convertisseurs de puissance ainsi que les unités de traitement telles que les microcontrôleurs, les FPGA et les micro-ordinateurs. Les protocoles de communication **SPI, I^2C, UART** sont également analysés afin d'évaluer leur pertinence pour l'échange de données dans un environnement de commande distribuée.

Sur cette base, une architecture de commande hétérogène combinant **FPGA** et **Raspberry Pi** est proposée. Le **FPGA** assure les tâches critiques en temps réel, telles que l'acquisition des signaux et la génération des commandes, tandis que le Raspberry Pi est utilisé pour la supervision, la communication et l'affichage des données. Les protocoles de communication sont intégrés afin de garantir un transfert fiable des informations entre différents modules du système.

Des validations expérimentales sont réalisées sur des bancs intégrant des moteurs électriques, des encodeurs et des capteurs disponibles au laboratoire de l'Université du Québec à Trois-Rivières (UQTR). Les résultats obtenus permettent de valider le bon fonctionnement de l'architecture proposée et démontrent de l'intérêt des solutions d'‘électronique de commande intégrant des protocoles de communication pour les applications d'agriculture de précision.

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à mon directeur de recherche, **Professeur Alben Cardenas**, pour son encadrement, sa disponibilité et la qualité de ses conseils tout au long de la réalisation de ce mémoire. Son expertise, sa rigueur scientifique et ses orientations ont été déterminantes dans l'aboutissement de ce travail.

Je souhaite également remercier chaleureusement **les membres de l'équipe de recherche du professeur Alben Cardenas**, pour leur soutien technique, leurs échanges constructifs et l'esprit de collaboration qui règne au sein du laboratoire. Les discussions scientifiques et l'entraide au sein de l'équipe ont grandement contribué à l'avancement de mes travaux expérimentaux.

Mes remerciements s'adressent également aux enseignants et au personnel du programme de maîtrise en génie électrique de **l'Université du Québec à Trois-Rivières (UQTR)** pour la qualité de la formation reçue et les connaissances acquises tout au long de mon parcours académique.

Je tiens aussi à remercier toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet, notamment celles qui m'ont apporté leur aide technique, leurs conseils ou leur soutien durant les différentes phases de ce travail. Enfin, j'adresse une pensée toute particulière à **ma famille et à mes proches**, pour leur soutien moral, financier, leur patience et leurs encouragements constants tout au long de mes études.

Ce mémoire représente l'aboutissement de nombreuses années d'efforts, de persévérance et de passion pour le domaine du génie électrique. Sa réalisation n'aurait pas été possible sans l'accompagnement, la confiance et le soutien des personnes mentionnées ci-dessus, auxquelles j'exprime toute ma reconnaissance.

Avant-propos

Le choix du sujet de ce mémoire s'inscrit dans un intérêt particulier pour le système électromécanique et ses applications en agriculture de précision. Les enjeux actuels liés à l'automatisation, à l'efficacité énergétique et à l'intégration des technologies numériques dans le domaine agricole ont motivé l'orientation de ce travail vers l'électronique de commande et les protocoles de communication associés.

Ce mémoire a été réalisé dans le cadre du programme de maîtrise en génie électrique de l'Université du Québec à Trois-Rivières (UQTR). Il vise à approfondir les connaissances acquises durant la formation et à les appliquer à une problématique concrète de recherche, en lien avec les systèmes embarqués et la commande des actionneurs électromécaniques. L'auteure souhaite exprimer sa reconnaissance à son directeur de recherche pour son accompagnement, ses conseils et son soutien tout au long de la réalisation de ce travail. Des remerciements sont également adressés aux membres du laboratoire et de l'équipe pour l'aide technique et l'environnement de travail favorable qu'ils ont su offrir.

Table des matières

Résumé iii

Remerciements iv

Avant Propos v

Table des matières vi

Liste des tableaux ix

Liste des figures x

Liste des symboles xiii

Chapitre 1 - Introduction 1

1.1 Contexte et problématique 1

1.2 Objectif de la recherche 3

1.3 Méthodologie de travail 3

1.4 Conclusion 4

Chapitre 2 - Étude théorique 7

2.1 Les éléments de l'agriculture de précision 8

2.2 Structure générale et sous-systèmes fonctionnels d'un robot agricole 14

2.3 Avantages et inconvénients des robots agricoles 16

2.4 Architecture des robots agricoles électriques 19

2.4.1 Alimentation électrique 19

2.4.2 Électronique de commande embarquée 20

2.4.3 Communication et interfaçage de capteurs et actionneurs 28
2.4.4 Actionneurs électromécaniques 37
2.4.5 Convertisseurs de puissance 40
2.4.6 Capteurs et éléments de mesure 42
2.4.7 Modulation par largeur d'impulsion (MLI) 48
2.5 Conclusion 49

Chapitre 3 - Proposition d'architecture de commande 50
3.1 Introduction 50
3.2 Cas d'applications 50
3.3 Architecture globale de commande proposée 51
3.3.1 Éléments et principe de fonctionnement 53
3.3.2 Justification du choix de l'architecture proposée 54
3.3.3 Contraintes et exigences du système 55
3.4 Implémentation en FPGA 56
3.5 Architecture interne de mesure et de commande basée sur FPGA 59
3.6 Intégration de l'architecture du banc de test du laboratoire 60
3.7 Conclusion 62

Chapitre 4 - Validation expérimentale 64
4.1 Organisation des tests expérimentaux 64
4.1.1 Test sur capteur avec convertisseur analogique-numérique en I^2C .. 65
4.1.2 Validation de la commande RC du servomoteur avec les protocoles UART et SPI 67
4.1.3 Test sur servomoteur via protocole SPI 73
4.1.4 Analyse et comparaison globale des protocoles SPI et UART 77

4.1.5 Test sur capteur de courant avec convertisseur analogique-numérique SPI 78
4.1.6 Simulation et expérimentation avec différentes commandes MLI .. 83
4.1.7 Comparaison entre les valeurs théoriques et expérimentales 85
4.1.8 Tests sur le récepteur RC et le générateur MLI 87
4.1.9 Expérimentation sur encodeur 91
4.2 Conclusion 94

Chapitre 5 - Conclusion 96
5.1 Synthèse des travaux réalisés 96
5.2 Difficultés rencontrées et limites du travail 97
5.3 Choix des protocoles de communication 98
5.4 Perspectives et recommandations 99

Annexe A - Standard I^2C protocole entre Maître-esclave 108

Annexe B - Capture bloc design I2C 109

Annexe C - Bloc Design pour le récepteur RC 110

Annexe D - Bloc Design pour Encodeur 111

Annexe E - Capture du Test numérique de la balance I^2C 1953g sur l'ILA 112

Liste des tableaux

Tableau 2-1 Synthèse des principaux avantages et inconvénients des robots agricoles. 18
Tableau 2-2 Exemples de cartes FPGA commerciales et leurs utilisations. 25
Tableau 2-3 Principales versions du Raspberry Pi et leurs utilisations. 27
Tableau 2-4 Ligne de communication SPI. 30
Tableau 2-5 Caractéristiques SPI. 30
Tableau 2-6 Comparaison des protocoles I^2C et SPI. 36
Tableau 2-7 Comparaison des protocoles SPI, I^2C et UART. 36
Tableau 2-8 Exemples d'applications des actionneurs. 37
Tableau 2-9 Types d'encodeurs, rôles et applications agricoles. 45
Tableau 3-1 Valeurs cibles des contraintes du système de commande. 56
Tableau 4-1 Comparaison entre les valeurs théoriques et expérimentales des impulsions du servomoteur (UART). 72
Tableau 4-2 Comparaison entre les valeurs théoriques et expérimentales des impulsions du servomoteur (SPI). 77
Tableau 4-3 Comparaison entre les valeurs théoriques et expérimentales des commandes MLI. 86

Liste des figures

Figure 1.1 Démarche méthodologique du projet. 6

Figure 2.1 Prototype de véhicule agricole semi-autonome - laboratoire du Groupe de recherche en électronique industrielle (GRÉI) - UQTR. . 13

Figure 2.2 Exemple des véhicules agricoles 14

Figure 2.3 Architecture typique des robots agricoles électriques. 19

Figure 2.4 Vue réelle d'une carte Arduino UNO et d'une carte Arduino MEGA. 21

Figure 2.5 Carte FPGA Nexys 4 DDR - Xilinx Artix-7 - Digilent. 25

Figure 2.6 Carte graphique (GPU) de type NVIDIA, utilisée pour le traitement graphique et l'accélération des calculs parallèles. 26

Figure 2.7 Carte Raspberry Pi 3 Model B+ 27

Figure 2.8 Connexion entre les lignes de communications SPI. 30

Figure 2.9 Format du paquet de données UART. 34

Figure 2.10 Vue réelle de l'encodeur utilisé pour l'expérimentation. 45

Figure 2.11 Exemple de convertisseur analogique-numérique utilisé pour l'interface d'un capteur de force basé sur une cellule de charge. 48

Figure 3.1 Architecture CPU+FPGA pour le système de commande embarqué. 52

Figure 3.2 Diagramme bloc et modules. 57

Figure 3.3 Architecture interne de mesure et de commande basée sur FPGA. .. 59

Figure 3.4 Banc de tests développé pour la validation expérimentale au laboratoire du groupe de recherche en électronique industrielle. 62

Figure 4.1 Plateforme expérimentale de capteur de force avec convertisseur analogique-numérique HX711 utilisant le protocole I^2C. 65

Figure 4.2 Résultats obtenus des lignes SCL et SDA du module HX711 avec l'implémentation sur FPGA. 66
Figure 4.3 Chaîne de commande RC du servomoteur via le protocole UART. .. 68
Figure 4.4 MLI du servomoteur simulation protocole UART. 70
Figure 4.5 MLI produit par le FPGA pour des commandes à 0 et à 128 générées par le Raspberry Pi et transmises via le protocole UART. . 71
Figure 4.6 Chaîne de commande RC d'un servomoteur via SPI. 73
Figure 4.7 MLI produit par le FPGA pour des commandes à 0, à 128 et à 255 générées par le Raspberry Pi et transmises via le protocole SPI. 75
Figure 4.8 Architecture matérielle et schéma de connexion du capteur de courant avec convertisseur analogique-numérique SPI. 78
Figure 4.9 Vue réelle du récepteur RC et du capteur de courant avec le convertisseur analogique-numérique LTC1864. 79
Figure 4.10 Courbe de l'évolution du courant mesuré lorsque la commande est générée à l'aide de la manette RC. 80
Figure 4.11 Test sur capteur de courant avec charge résistive de $3.3\,\Omega$. 82
Figure 4.12 Architecture de commande des moteurs CC par FPGA avec génération MLI et contrôleurs MD20A. 83
Figure 4.13 Test expérimental de la génération de signal MLI avec commandes à 1C00, à 4000 et à 6000. 84
Figure 4.14 Architecture matérielle et diagramme RTL du système complet. 87
Figure 4.15 Résultats de la réception RC et de la génération MLI sur les moteurs 1 et 2, en commande de marche avant et arrière. 90
Figure 4.16 Résultats de la réception RC et de la génération MLI sur les moteurs 1 et 2, en commande de virage à gauche et à droite. 90

Figure 4.17 Architecture matérielle et diagramme RTL du système embarqué de commande et mesure par FPGA. 92

Figure 4.18 Résultats expérimentaux du test de l'encodeur - signaux A et B observés à l'oscilloscope. 93

Figure 4.19 Résultats expérimentaux du test de l'encodeur - vitesse obtenue avec et sans facteur de réduction. 93

Figure A.1 Standard I^2C protocole entre Maître-esclave 108

Figure B.1 Bloc design I^2C 109

Figure C.1 Bloc Design pour le récepteur RC 110

Figure D.1 Bloc Design pour l'Encodeur 111

Figure E.1 Capture du Test numérique de la balance I^2C 1953g sur l'ILA 112

Liste des symboles

CA	Courant alternatif
CAN	Convertisseur analogique numérique
CC	Courant continu
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit
I^2C	Inter-Integrated Circuit
IA	Intelligence artificielle
MLI	Modulation de Largeur d'Impulsion
MISO	Master In Slave Out
MOSI	Master Out Slave In
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
WSN	Réseaux de capteurs sans fil

Chapitre 1 - Introduction

1.1 Contexte et problématique

La croissance démographique mondiale et les évolutions socio-économiques actuelles exercent une pression accrue sur le secteur agricole. Celui-ci est confronté à de nombreux défis, notamment la sous-production, une demande alimentaire supérieure à l'offre, l'exode rural [1], le changement climatique, la pénurie de main-d'œuvre et la réduction des surfaces cultivables [2]. Ces contraintes obligent des exploitations agricoles à produire d'avantage sur des parcelles de plus en plus limitées, tout en respectant des exigences de durabilité.

Dans ce contexte de ressources limitées en eau, en terres agricoles et en main-d'œuvre, la production agricole mondiale devra augmenter d'au moins 25% afin de répondre à la demande alimentaire future [1]. Cette situation rend primordiale l'automatisation des systèmes agricoles afin d'améliorer la productivité, de réduire l'effort humain et d'optimiser le temps de production. Les robots agricoles apparaissent ainsi comme une solution prometteuse pour répondre à ces enjeux. Ils permettent d'optimiser l'utilisation des ressources, de réduire l'emploi des fertilisants et des pesticides, ainsi que de limiter le gaspillage de l'eau [1]. De plus, l'intégration de capacités d'intelligence artificielle constitue une réponse pertinente à la pénurie de main-d'œuvre agricole.

Dans cette même voie de modernisation, l'agriculture de précision est vue comme une approche moderne de la gestion qui repose sur l'utilisation de technologies avancées afin d'optimiser les opérations de production, d'augmenter la productivité et de réduire les impacts environnementaux. Elle vise une utilisation plus rationnelle des ressources à travers une meilleure connaissance et un meilleur contrôle des systèmes agricoles [3].

Quel que soit le type d'application considéré, ces systèmes reposent généralement sur une architecture similaire. Celle-ci comprend une source et un convertisseur d'énergie chargés d'alimenter un système électromécanique composé d'actionneurs, de robots ou de véhicules agricoles. Normalement, la commande du système repose sur l'utilisation de techniques de modulation, dont la modulation par largeur d'impulsion (MLI), ce qui permet de contrôler la durée d'activation d'un signal périodique afin de piloter efficacement les actionneurs. Ces éléments permettent de réaliser les tâches mécaniques requises sur le terrain.

Un système de mesure est également associé à cette architecture. Il inclut notamment des capteurs, tels que les capteurs de courant et tension, permettant de mesurer l'intensité électrique circulant et le potentiel dans le système électromécanique. D'autres grandeurs d'état du système électromécanique incluent notamment la position, la vitesse, le couple ou la force, et l'accélération. L'ensemble de ces informations est essentiel pour assurer une commande précise et fiable des systèmes électromécaniques.

Les signaux analogiques issus de ces capteurs sont ensuite convertis en données numériques binaires ou en données à plusieurs bits à l'aide de convertisseurs analogique-numérique. Ces données sont alors transmises à un système d'acquisition intégré au processeur, qui les traite avant de les envoyer au contrôleur. Les échanges d'informations entre les différents blocs sont assurés par des protocoles de communication tels que SPI (pour Serial Peripheral Interface), I²C (pour Inter-Integrated Circuit), UART (pour Universal Asynchronous Receiver-Transmitter) ou Ethernet, choisis en fonction des contraintes de débit, de latence et de robustesse.

Plusieurs types de processeurs numériques peuvent être utilisés pour implémenter ces fonctions de commande, notamment des microcontrôleurs, des FPGA (pour Field Programmable Gate Array), des processeurs graphiques (GPU pour Graphics Processing Unit) et des micro-ordinateurs.

Ce projet porte sur l'étude et proposition de solutions en électronique pour la commande destinées aux systèmes électromécaniques utilisés en agriculture de précision [4, 5]. Plus précisément, ce travail propose des architectures de commande hétérogènes intégrants des processus de type CPU, GPU et FPGA pour la commande des actionneurs, ainsi que des interfaces d'électronique de puissance optimisées [6].

1.2 Objectif de la recherche

Cette recherche a pour objectif de proposer des solutions en électronique de commande de systèmes électromécaniques en contexte d'agriculture de précision. Nous allons nous concentrer sur un actionneur électromécanique et sa commande en utilisant plusieurs types de processeurs. La contribution principale du travail repose sur la proposition et l'implémentation d'une architecture de commande adaptée au contexte spécifique des systèmes électromécaniques en agriculture de précision.

1.3 Méthodologie de travail

La démarche méthodologique adoptée dans ce mémoire est présentée sous forme graphique dans la figure 1.1. Elle démarre par l'analyse de la problématique et la définition des objectifs, passe par une étude théorique et une phase de conception, puis se conclut par une validation expérimentale.

Notre travail est structuré en plusieurs chapitres. Le deuxième chapitre porte sur l'étude théorique, dans laquelle sont analysés les différents types d'électronique de commande.

Les avantages et les inconvénients de chaque approche y sont présentés, suivis du choix de la solution de commande retenue pour la réalisation du projet, en justifiant les raisons de ce choix.

Le troisième chapitre est dédié à la proposition d'une architecture de commande et à la conception du système. Le quatrième chapitre présente la validation expérimentale réalisée sur des robots et des actionneurs développés et disponibles au laboratoire du Groupe de recherche en électronique industrielle (GRÉI) de l'Université du Québec à Trois -Rivières (UQTR). Enfin, une conclusion générale est présentée au dernier chapitre.

1.4 Conclusion

En conclusion, ce premier chapitre a permis de situer le contexte général de l'étude en mettant en évidence les enjeux majeurs de l'agriculture de précision, notamment l'optimisation des ressources, l'amélioration de la productivité et la réduction de l'impact environnemental. Il a également souligné le rôle essentiel des systèmes électromécaniques et de l'électronique de commande dans la modernisation des pratiques agricoles. La problématique abordée met en lumière la nécessité de développer des architectures de commande performantes, capables de répondre aux contraintes de temps réel, d'efficacité énergétique et de communication entre les différents sous-systèmes. Dans ce cadre, l'intégration de technologies telles que les FPGA, les microcontrôleurs et les micro-ordinateurs apparaît comme une solution pertinente pour concevoir des systèmes embarqués intelligents et fiables.

Enfin, la méthodologie présentée structure le déroulement du travail en articulant l'étude théorique, la conception de l'architecture de commande et sa validation expérimentale. Cette approche progressive permet d'assurer la cohérence du mémoire

et de répondre de manière rigoureuse aux objectifs fixés. Le chapitre suivant sera ainsi consacré à l'étude théorique des différents composants et technologies intervenant dans les systèmes de commande pour l'agriculture de précision.

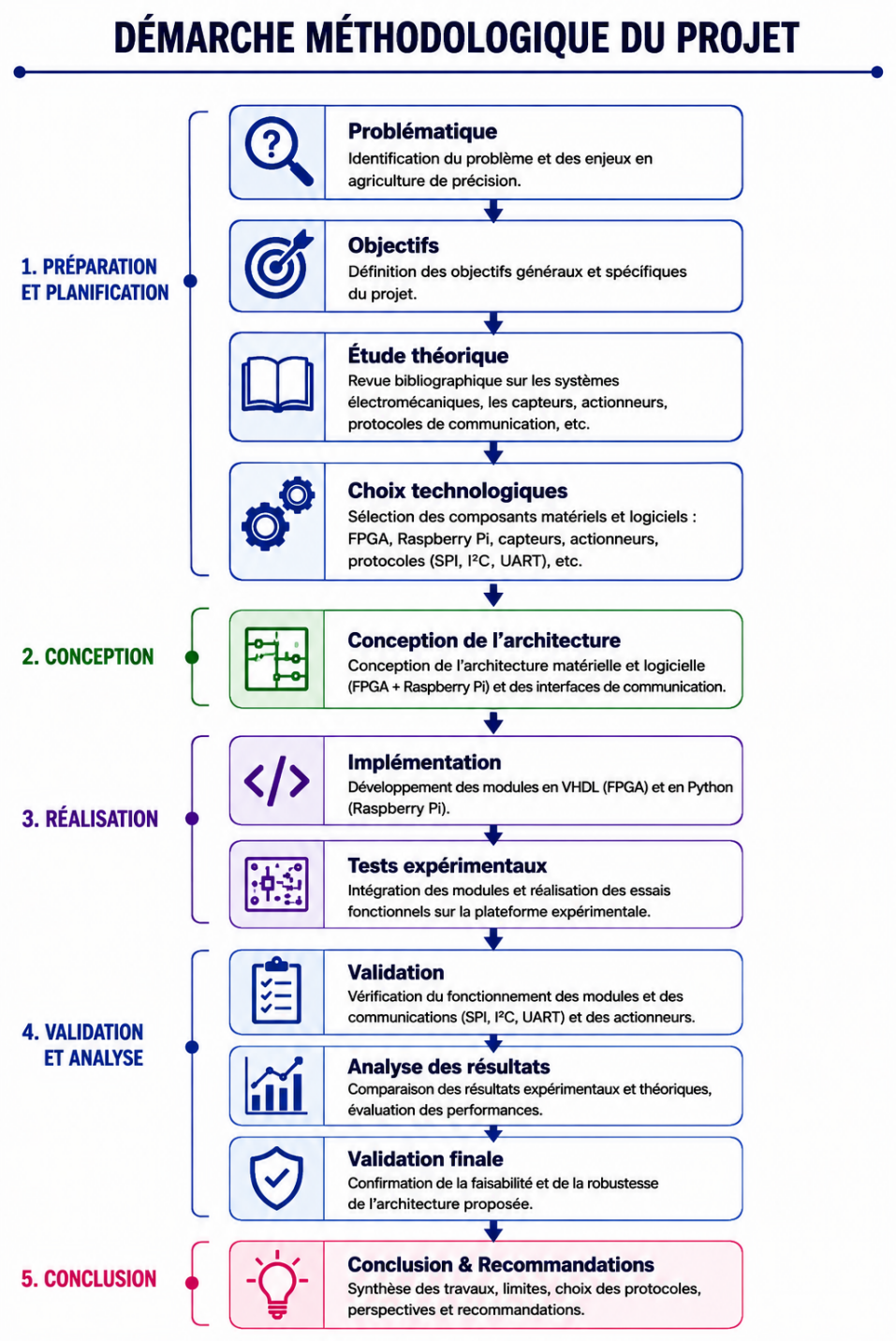


Figure 1.1 Démarche méthodologique du projet.

Chapitre 2 - Étude théorique

L'agriculture est la science, l'art et la pratique de la culture des sols, de la production végétale et de l'élevage du bétail. Elle englobe l'ensemble d'activités liées à la préparation, à la production, à la transformation et à la commercialisation des produits qui en résultent [7]. Comme mentionné au premier chapitre, l'industrie agricole est confrontée à plusieurs défis tels que la croissance et le changement démographique, la pénurie de main-d'œuvre, le risque de pandémies, l'augmentation de la demande énergétique, les émissions des gaz à effet de serre et le réchauffement climatique [8]. L'exode rural, caractérisé par la diminution de la population des villages au profit des centres urbains, constitue également un facteur majeur qui ralentit le développement du secteur agricole. [2].

Le développement de la technologie des robots et la modernisation des infrastructures du secteur agricole, par l'adoption de systèmes électromécaniques plus efficaces et intelligents, dans un contexte d'agriculture de précision, contribueront à résoudre partiellement ces problèmes [9]. Ceci est également une exigence incontournable pour équilibrer l'offre et la demande sur le marché mondial, et ce, de façon durable, afin de relever les défis liés à la pénurie de main-d'œuvre. De plus, les travailleurs agricoles éprouvent de plus en plus de difficultés à accomplir leurs tâches en raison des variations de température. Celles-ci les contraignent à interrompre leur travail ou à réduire progressivement leur effort physique, indépendamment de leur volonté.

Dans ce contexte, l'intégration de systèmes robotisés et automatisés constitue une piste intéressante pour répondre à ces défis. Le remplacement des équipements lourds par des machines légères, télécommandées, semi-autonomes ou autonomes, peut également réduire les problèmes associés à la compression de la couche d'érable en agriculture [1].

Les objectifs de l'agriculture de précision sont orientés vers l'utilisation de systèmes intelligents intégrant des capteurs et des algorithmes de traitement afin d'améliorer la gestion des équipements et des opérations agricoles [10, 11]. Dans ce cadre, la commande correspond à l'ensemble des moyens mis en œuvre pour assurer la régulation et le pilotage d'un dispositif ou d'un circuit électronique. Elle repose sur différentes méthodes ainsi que sur divers composants matériels et logiciels.

2.1 Les éléments de l'agriculture de précision

Après avoir présenté le contexte général et les enjeux liés aux systèmes agricoles modernes, il est désormais essentiel d'examiner les principaux éléments technologiques qui constituent l'agriculture de précision.

Les robots agricoles sont des machines autonomes ou semi autonomes conçues pour améliorer l'efficacité et le rendement des opérations agricoles, réduire la dépendance à la main-d'œuvre et accroître la production globale [1]. Ils peuvent également être définis comme des machines robotisées mobiles destinées à soutenir ou à exécuter des activités de production agricole de manière autonome ou semi-autonome [7]. Initialement développés pour des applications industrielles telles que le traitement des matériaux, le transport et le contrôle de la qualité, les robots sont désormais largement utilisés en agriculture afin d'automatiser les opérations, d'améliorer la productivité et de limiter la pénibilité du travail humain [12].

L'adoption de robots agricoles légers présente un avantage majeur en réduisant la dégradation des sols par rapport aux machines agricoles lourdes traditionnelles [13]. Il existe plusieurs types de robots agricoles, qui se distinguent par leurs applications et leurs modes de fonctionnement [1]. Les robots agricoles au sol dédiés à l'agriculture

de précision se répartissent principalement en deux catégories : **les robots mobiles automoteurs** et **les outils robotisés intelligents** montés sur des véhicules agricoles. Ces systèmes assistent ou remplacent l'intervention humaine afin d'améliorer la précision et d'éviter la répétabilité des opérations [1]. Ils sont utilisés dans les applications industrielles, comme le traitement des matériaux, le transport et l'inspection de contrôle qualité [1].

Les robots légers limitent les dommages au sol causés par les machines lourdes pour les raisons suivantes: la précision accrue, l'accessibilité dans les petites parcelles, la réduction des coûts, l'adaptabilité et l'automatisation et enfin la durabilité. Par ailleurs, les équipements agricoles motorisés traditionnels, tels que les tracteurs, les épandeurs et les récolteuses, ont connu une automatisation progressive au cours des dix dernières années grâce à l'intégration de technologies d'autoguidage basées sur les systèmes de navigation GPS (pour Global Positioning System) ou GNSS (pour Global Navigation Satellite System). Ces équipements, aujourd'hui largement commercialisés, constituent la majorité des solutions robotisées actuellement déployées dans le secteur agricole. Récemment, certaines entreprises ont également lancé des tracteurs autonomes sans cabine, compatibles avec les outils agricoles standards.

La robotique agricole repose fortement sur la navigation autonome, qui regroupe l'ensemble des méthodes permettant aux véhicules agricoles autonomes d'assurer leurs déplacements de manière efficace et sécurisée.

En agriculture de précision [7], la navigation autonome comprend généralement :

- L'organisation de l'agencement du champ;
- La planification des trajectoires;
- La génération des mouvements du véhicule;

- L'autoguidage basé sur des capteurs, des systèmes de perception et des stratégies de contrôle.

Les données collectées sur le terrain jouent un rôle essentiel dans l'optimisation des performances des robots, notamment pour améliorer leur efficacité énergétique et pour réduire l'impact du compactage des sols [1]. Enfin, les robots agricoles agissent directement sur les cultures et leur environnement en appliquant, de manière précise et contrôlée, des matériaux ou de l'énergie. Leur efficacité repose sur la combinaison d'une cadence d'actions élevée et d'un haut niveau de précision [7]. Pour atteindre ces objectifs, ils intègrent diverses technologies modernes, telles que les capteurs et les caméras, les systèmes de commande avancés et l'intelligence artificielle.

Dans le contexte agricole, l'évolution des véhicules est étroitement liée au développement de l'agriculture de précision, qui repose sur des technologies avancées pour optimiser les rendements et l'efficacité des opérations agricoles. D'après la littérature [14], on distingue trois principaux types de véhicules électriques dits propres: les véhicules électriques à batterie, les véhicules hybrides et les véhicules à pile à combustible. Bien que ces solutions présentent des avantages environnementaux notables, les véhicules électriques à batterie demeurent limités par l'autonomie, le temps de recharge, les coûts élevés et la disponibilité de l'infrastructure de recharge. Ces contraintes sont particulièrement marquées dans le cas des véhicules agricoles tout-terrain, tels que les tracteurs, qui requièrent des apports énergétiques importants sur de courtes périodes [15].

Les **véhicules hybrides** combinent généralement un moteur thermique et un ou plusieurs moteurs électriques alimentés par une batterie. Cette architecture permet de tirer parti des avantages des deux sources d'énergie : le moteur électrique fournit un couple élevé à basse vitesse, particulièrement utile lors des phases de démarrage, des manœuvres

et des travaux nécessitant de fortes sollicitations, tandis que le moteur thermique assure l'autonomie lors des travaux de longue durée. Dans le domaine agricole, cette technologie contribue à réduire la consommation de carburant, les émissions polluantes et le bruit de fonctionnement tout en maintenant une puissance adaptée aux travaux exigeants. Les véhicules hybrides constituent ainsi une solution de transition vers l'électrification des machines agricoles, notamment pour les tracteurs et les véhicules utilitaires opérant sur de grandes exploitations [16].

Les **véhicules à pile à combustible** utilisent l'hydrogène comme source d'énergie. La pile à combustible convertit l'énergie chimique de l'hydrogène en énergie électrique afin d'alimenter un ou plusieurs moteurs électriques, le seul sous-produit étant de la vapeur d'eau. Cette technologie présente plusieurs avantages pour les applications agricoles, notamment une autonomie supérieure à celle des véhicules électriques à batterie, un ravitaillement rapide en hydrogène et une capacité à fournir une puissance importante pendant de longues périodes de fonctionnement. Ces caractéristiques en font une solution prometteuse pour les engins agricoles de forte puissance, tels que les tracteurs et les machines destinées aux travaux intensifs. Toutefois, son déploiement demeure limité par le coût élevé des piles à combustible, les défis liés à la production et au stockage de l'hydrogène, ainsi que par le manque d'infrastructures de ravitaillement dans les zones rurales [16]

Les véhicules électriques autonomes et semi-autonomes constituent une évolution majeure des machines agricoles. Il s'agit de systèmes capables de se déplacer, d'exécuter des tâches agricoles et de prendre des décisions de manière autonome, sans intervention humaine directe, tout en étant propulsés par des moteurs électriques [1]. Ces véhicules se caractérisent principalement par leur autonomie de fonctionnement et par leur

alimentation électrique. La capacité de naviguer et de fonctionner de façon autonome reposent sur l'intégration de divers capteurs (caméras, lidars, radars et systèmes GPS), combinés à des algorithmes d'intelligence artificielle et à des systèmes de commande avancés. Ces technologies permettent au véhicule de naviguer de manière autonome dans les champs, de détecter et d'éviter les obstacles, ainsi que d'adapter sa vitesse et sa trajectoire en fonction des conditions opérationnelles.

L'alimentation électrique constitue un autre aspect fondamental de ces véhicules, généralement elle est assurée par des batteries ou dans certains cas, par des systèmes de recharge solaire. Cette motorisation électrique contribue à réduire les émissions de gaz à effet de serre par rapport aux machines agricoles à moteur thermique. Enfin, grâce à l'automatisation des tâches agricoles et aux capacités de traitement des données issues des capteurs, ces véhicules offrent une grande précision d'exécution essentielle en agriculture de précision. La figure 2.1 présente un prototype de véhicule agricole semi-autonome alimenté par l'énergie solaire. La structure est équipée de panneaux photovoltaïques qui assurent la conversion de l'énergie solaire en électricité, permettant ainsi d'alimenter les moteurs, l'électronique de contrôle et l'ensemble des capteurs embarqués.



Figure 2.1 Prototype de véhicule agricole semi-autonome - laboratoire du Groupe de recherche en électronique industrielle (GRÉI) - UQTR.

Ces derniers peuvent inclure des modules de positionnement GPS pour la navigation, des capteurs de vision ou de télédétection (caméra,lidar, ultrasons) pour la détection d'obstacles et l'analyse de l'environnement, ainsi que des capteurs spécifiques pour le suivi des cultures (p.ex., température, humidité, état du sol). L'intégration, dans ce type de véhicule, d'actionneurs électriques reliés au système de commande rend possible l'exécution de tâches agricoles telles que: le transport de charge, le désherbage mécanique et la surveillance automatisée des parcelles. La figure 2.2, tirée de [1], illustre l'évolution des véhicules, de l'agriculture traditionnelle à l'agriculture de précision, où l'autonomie énergétique, la robotisation et l'intelligence embarquée améliorent l'efficacité, réduisent les coûts et limitent l'impact environnemental.

Les véhicules autonomes et semi-autonomes en milieu agricole sont conçus pour effectuer des tâches variées selon le type de culture. Notamment, ils sont appelés à

accomplir des missions telles que le semis, l'irrigation, la fertilisation, la pulvérisation de pesticides, la récolte, la surveillance des cultures, ainsi que le suivi de la qualité du sol, de la précision et de l'efficacité des processus.

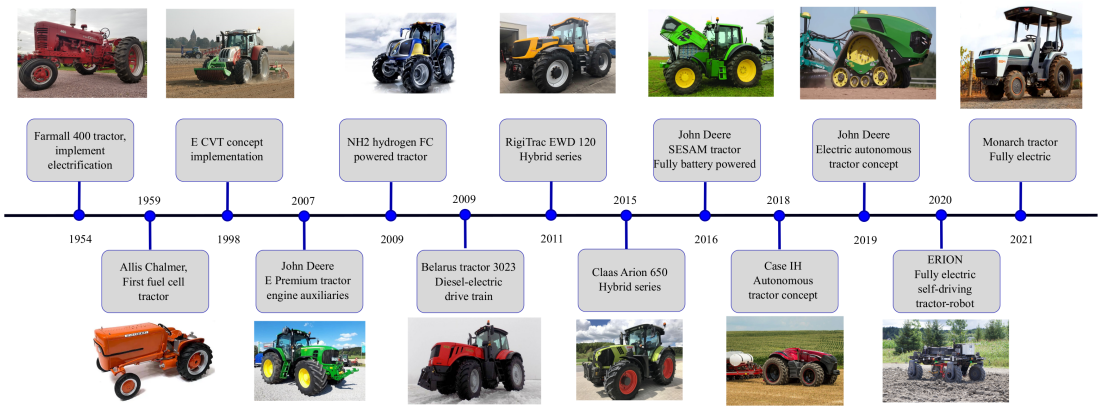


Figure 2.2 Exemple des véhicules agricoles - tirée de [1].

2.2 Structure générale et sous-systèmes fonctionnels d'un robot agricole

Les robots agricoles intègrent généralement des capteurs et des actionneurs destinés à une large variété de tâches [17]. Ces éléments sont agencés dans une structure pouvant être définie par trois couches principales:

- Dans la première couche, nous trouvons **la couche générale**, qui est composée du châssis, des roues ou des chenilles, des actionneurs (moteurs, vérins, servomoteurs). Elle assure les déplacements et les manipulations physiques.
- Dans la deuxième, **la couche de l'électronique et des capteurs** qui regroupe les capteurs de distance (ultrasons, LIDAR, caméras), les capteurs de position (GPS, encodeurs, IMU) et les interfaces de communication (SPI, l'I^2C, UART...).
- Dans la troisième, **la couche de logiciels et de contrôle** gère le traitement des données, la planification des trajets et les algorithmes de décision (souvent basés sur l'intelligence artificielle ou la logique floue).

Selon la nature des fonctions implémentées dans un robot autonome ou semi-autonome, nous pouvons définir plusieurs sous-systèmes :

- **Système de perception:** Il permet au robot de « voir » son environnement grâce à des capteurs optiques (caméra RGB, caméra multispectrale, LIDAR) ou proprioceptifs (gyroscope, accéléromètre, encodeur). Ces données sont utilisées pour détecter les obstacles, identifier les cultures et évaluer les conditions du sol [18, 19].
- **Système de localisation et de navigation:** Il utilise le GPS, la vision artificielle et les encodeurs de roues pour déterminer la position du robot sur le champ et planifier des trajets optimaux [20].
- **Système de décision et de commande:** Implémenté sur une carte embarquée (FPGA, processeur graphique, microprocesseur, ou microcontrôleur), il traite les informations issues des capteurs pour générer des signaux de commande pour des actionneurs et des outils agricoles [10].
- **Systèmes d'actionnement :** Ils convertissent les ordres électroniques en mouvements physiques (rotation du moteur, déplacement du bras robotisé, ouverture d'électrovalves) [21].
- **Systèmes de communication :** Ils assurent les échanges d'informations nécessaires au fonctionnement du robot. La communication interne entre les différents modules électroniques (capteurs, unité de commande et actionneurs) peut être réalisée à l'aide de protocoles tels que le CAN bus, l'I^2C, le SPI ou l'UART.

  La communication externe entre le robot et l'opérateur ou une station de contrôle repose généralement sur des technologies sans fil telles que la radio, le Wi-Fi ou le Bluetooth [22].

2.3 Avantages et inconvénients des robots agricoles

Les robots et tracteurs électriques présentent de nombreux avantages, notamment : le développement de l'agriculture de précision, l'amélioration des conditions de travail, la limitation des impacts environnementaux et la rentabilité des cultures [1]. Le principal avantage des robots agricoles réside dans leur efficacité et leur rentabilité. Bien que l'investissement initial puisse être élevé, certains robots sont conçus pour fonctionner 24 heures sur 24 et 7 jours sur 7 sans interruption. Cette capacité de fonctionnement continu permet un amortissement rapide de l'investissement, car les exploitations agricoles gagnent en efficacité tout en réduisant la main-d'œuvre nécessaire, les pertes de production et les coûts d'exploitation.

Les avantages principaux peuvent être définis comme:

- **Productivité élevée** : les robots sont capables d'exécuter des tâches à un rythme plus élevé que celui des opérateurs humains, sur de longues périodes, tout en maintenant un niveau de performance constant, sans risque de fatigue ni d'interruption.
- **Main-d'œuvre garantie** : dans le cadre des activités saisonnières, notamment la récolte des fruits, le manque de main-d'œuvre disponible au moment opportun constitue un défi majeur. Les robots offrent une solution efficace en assurant la cueillette des produits dès leur maturité, limitant ainsi les pertes liées à leur détérioration sur le terrain.
- **Réduction des déchets** : le manque de main-d'œuvre peut conduire à laisser les récoltes au sol ou à retarder leur conditionnement et leur distribution. Ce qui entraîne des pertes significatives. L'automatisation constitue une solution efficace à cette problématique en garantissant une gestion continue, rapide et optimisée des

récoltes.

- **Précision** : Contrairement aux opérateurs humains, les robots ne sont pas affectés par la fatigue ni par des erreurs involontaires. Ainsi, les tâches répétitives, voire complexes, sont réalisées avec constance, précision et fiabilité tout au long du processus [23].

Les systèmes robotiques permettent aux agriculteurs d'exécuter plusieurs tâches simultanément, voire en temps réel. La robotique agricole améliore ainsi la productivité tout en réduisant la pénibilité du travail manuel, notamment grâce à l'automatisation de la surveillance, du suivi et du traitement des cultures même dans des conditions difficiles. Elle contribue également à réduire les coûts et les délais de production, tandis que les avancées technologiques favorisent le développement de robots agricoles de plus en plus précis, performants et adaptés aux besoins du secteur [24].

Parmi les principaux inconvénients de la machinerie agricole traditionnelle figure l'impact environnemental lié à l'utilisation de moteurs alimentés par des combustibles fossiles. La combustion de ces carburants génère des émissions de gaz polluants, contribuant ainsi à la dégradation de la qualité de l'air et à l'augmentation des émissions de gaz à effet de serre. Cet aspect serait réduit en introduisant des robots électriques ou propulsés par des sources propres.

L'environnement de travail peut être affecté par les niveaux sonores élevés générés par les moteurs des machines agricoles conventionnelles. Afin de réduire ces nuisances tout en améliorant l'efficacité énergétique, la plupart des robots agricoles modernes sont propulsés par des systèmes électriques. Par ailleurs, l'intégration croissante de technologies numériques et de systèmes connectés dans l'agriculture de précision soulève de nouveaux enjeux liés à la gestion des données. Dans ce contexte, la protection de la

confidentialité des informations ainsi que l'intégrité des données constituent des défis majeurs pour les exploitants agricoles. [25]. Nous nous intéressons, dans cette étude, aux robots agricoles électriques et à leur électronique de commande.

Tableau 2-1 Synthèse des principaux avantages et inconvénients des robots agricoles.

Avantages	**Inconvénients**
Fonctionnement continu (24 h/24, 7 j/7).	Coût d'investissement élevé.
Grande précision et répétabilité.	Autonomie limitée des batteries.
Réduction de la dépendance à la main-d'œuvre.	Maintenance spécialisée.
Amélioration des conditions de travail.	Sensibilité aux conditions environnementales.
Réduction des pertes de récolte.	Investissements en infrastructures numériques.
Faibles émissions polluantes et faible niveau sonore.	Défis de cybersécurité et de gestion des données.
Surveillance des cultures en temps réel.	Intégration complexe aux équipements existants.

2.4 Architecture des robots agricoles électriques

L'architecture d'un robot agricole, comme le montre la figure 2.3, repose sur la combinaison de plusieurs sous-systèmes complémentaires, permettant la perception de l'environnement, la prise de décision et l'action sur le terrain. Ces sous-systèmes interagissent afin d'automatiser les tâches agricoles telles que: la plantation, la récolte, la pulvérisation et/ou la surveillance des cultures.

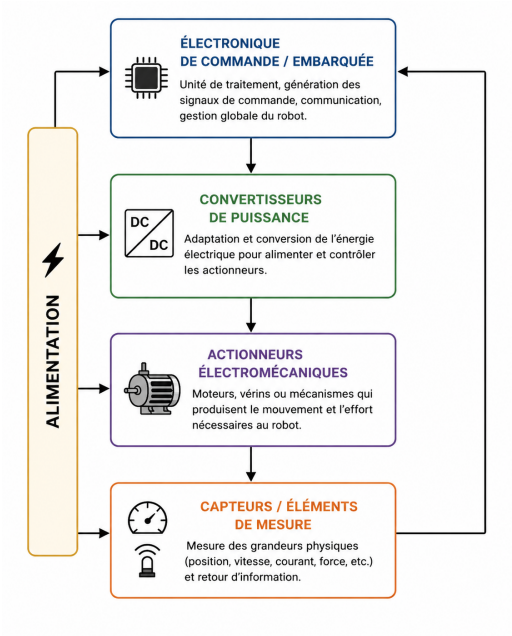


Figure 2.3 Architecture typique des robots agricoles électriques.

2.4.1 Alimentation électrique

L'alimentation électrique constitue un élément fondamental des systèmes électromécaniques en agriculture de précision. Elle assure l'apport énergétique nécessaire au fonctionnement (capteurs, unités de traitement, interfaces de communication et actionneurs), principalement à partir de batteries et de sources d'énergie renouvelables. L'intégration de systèmes de stockage à batteries et d'architectures électromécaniques

embarquées favorise une gestion optimisée de l'énergie et assure une commande robuste et fiable des machines agricoles électrifiées [26]. L'énergie fournie est adaptée aux besoins des différents sous-systèmes grâce à des convertisseurs de puissance, ce qui garantit la fiabilité et l'efficacité énergétique des systèmes embarqués.

2.4.2 Électronique de commande embarquée

L'électronique de commande embarquée joue un rôle central dans les systèmes électromécaniques, notamment dans les véhicules et les robots utilisés en agriculture de précision. Elle regroupe les composants matériels et logiciels intégrés permettant l'acquisition des données des capteurs, leur traitement en temps réel et la génération des commandes vers les actionneurs [27].

Selon les besoins de l'application, différents types d'électronique embarquée peuvent être utilisés, on distingue principalement:

- Les **microcontrôleurs**, généralement utilisés pour des tâches simples de surveillance et de commande.
- Les **micro-ordinateurs**, comme le Raspberry Pi, sont utilisés pour la supervision, la gestion des communications et l'interface utilisateur.
- Les **FPGA**, grâce à leur parallélisme et à leur comportement déterministe, sont privilégiés pour les applications en temps réel exigeantes, telles que la commande des actionneurs et le traitement rapide des signaux.
- Les **GPU** trouvent leur application dans le traitement intensif des données, notamment pour la vision et l'intelligence artificielle embarquées.

Ainsi, l'électronique de commande embarquée assure le lien entre les capteurs, les stratégies de commande et les actionneurs. Elle constitue un élément clé pour garantir la performance, la précision et la fiabilité des systèmes agricoles électrifiés et automatisés.

Dans le cas particulier de cette étude, nous nous concentrons sur trois types de processeur afin de réaliser l'implémentation des fonctions de commande : les microcontrôleurs, les micro-ordinateurs et les FPGA.

On utilise les microcontrôleurs pour plusieurs applications, comme permettre à un robot d'éviter les obstacles. À titre d'exemple, la carte **Arduino Uno** est une carte électronique basée sur le microcontrôleur **ATmega328P** [28]. Elle dispose de 14 broches d'entrées/sorties numériques, dont 6 peuvent être configurées pour générer des signaux MLI (Modulation de Largeur d'Impulsion), ainsi que 6 entrées analogiques. Selon [28], grâce à ses caractéristiques matérielles et à sa flexibilité, cette carte constitue une solution pertinente pour diverses applications. Toutefois, dans cette étude, nous nous focaliserons spécifiquement sur son rôle dans la commande des moteurs. Cette carte utilise un résonateur céramique de 16 MHz, des connecteurs USB, d'alimentation, d'ICSP (In Circuit Serial Programming) et un bouton de réinitialisation [28]. Elle est suffisamment équipée pour permettre une programmation efficace. La figure 2.4 présente une vue réelle d'une carte Arduino Uno et d'une carte Arduino Mega.

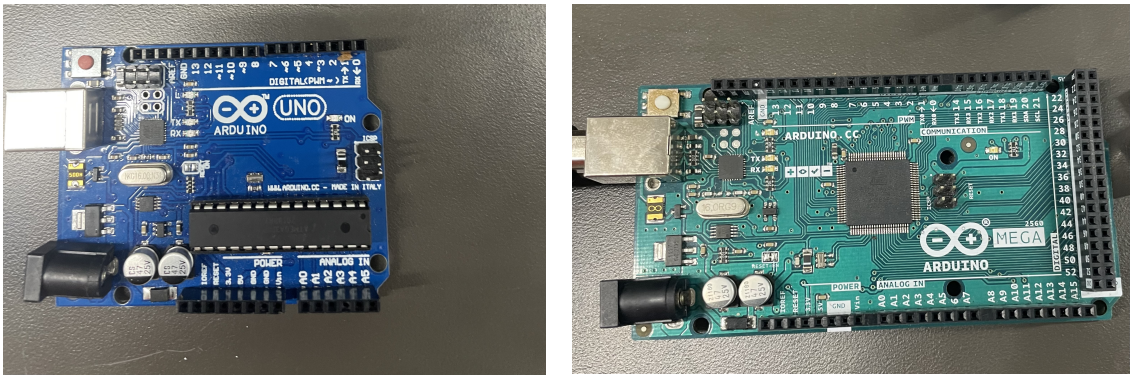


Figure 2.4 Vue réelle d'une carte Arduino UNO et d'une carte Arduino MEGA.

Les FPGA, ou réseaux prédiffusés programmables par l'utilisateur, sont des dispositifs reconfigurables basés sur des technologies d'intégration à très grande échelle. Ils sont largement utilisés en électronique et en informatique pour concevoir des circuits logiques et des systèmes numériques. Essentiellement, un FPGA est une puce en silicium conçue pour être programmée électroniquement par l'utilisateur, ce qui lui permet de fonctionner comme n'importe quel circuit numérique ou système embarqué [29, 30]. Les FPGA sont mis en avant comme des dispositifs permettant de développer des capteurs plus précis, de résolution supérieure, tout en réduisant la consommation énergétique grâce à leur capacité de reprogrammation et à leur architecture parallèle [31].

Ce qui distingue les FPGA, c'est leur incroyable puissance de calcul, rendue possible par le traitement parallèle des données [29]. Bien que les FPGA offrent de nombreux avantages, ils ne disposent pas d'une interface de communication numérique pour l'acquisition en temps réel de données provenant de capteurs. Pour collecter ces données, il est nécessaire d'utiliser un convertisseur analogique-numérique (CAN) dont la fonction principale est de convertir le signal analogique détecté en signal numérique, comme son nom l'indique. Ce composant constitue un élément essentiel de l'architecture développée dans le cadre de ce travail.

Les FPGA représentent une solution technologique performante et adaptée aux applications de l'agriculture de précision, tout en contribuant aux objectifs de développement durable. Lorsqu'on parle de l'agriculture 4.0 (agriculture de précision), on évoque les FPGA, car c'est l'une des technologies les plus utilisées ces dernières années, grâce à sa faible consommation énergétique, sa polyvalence et ses performances élevées [10]. La configuration des FPGA est réalisée à partir de la description matérielle.

La compréhension du fonctionnement d'un FPGA est essentielle pour maîtriser la conception numérique. Un FPGA est constitué en interne d'une multitude de blocs logiques configurables (CLB), reliés entre eux par des ressources d'interconnexion. Lorsqu'un utilisateur souhaite réaliser une fonction ou implémenter un algorithme, il emploie un langage de description matérielle (HDL), tel que VHDL ou Verilog, afin d'exprimer les comportements logiques attendus ou l'architecture de traitement souhaitée. Cette opération de configuration du FPGA se déroule en plusieurs phases:

- **Saisie de conception** : l'utilisateur décrit la logique ou l'architecture souhaitée à l'aide du code HDL;
- **Synthèse** : l'outil de développement convertit ce code en un réseau de circuits logiques;
- **Implantation** : les circuits obtenus sont associés aux ressources physiques des puces FPGA;
- **Configuration**: la conception finale est chargée dans le FPGA, qui adopte alors le comportement prévu.

Les FPGA sont utilisés dans divers domaines et applications. En vision par ordinateur, les FPGA accélèrent des tâches telles que la capture d'images, le traitement parallèle, la segmentation, l'extraction de caractéristiques et la reconnaissance de motifs [31]. Ces applications comprennent également l'intégration de caméras intelligentes et le traitement par intelligence artificielle en sécurité, en robotique et en véhicules autonomes [31]. Dans les systèmes de contrôle industriel, la robotique et l'automatisation, les FPGA jouent un rôle essentiel dans l'amélioration des performances en temps réel. Leur architecture parallèle leur permet d'assurer efficacement l'acquisition, le traitement et la transmission des données dans des systèmes distribués, ce qui les rend particulièrement adaptés aux applications en agriculture de précision, en santé et dans les environnements industriels

[31]. Certaines applications incluent le filtrage, la compression et l'analyse en temps réel des signaux audio, biomédicaux et de télécommunications. Dans les applications de télécommunications, le traitement du signal en temps réel, le chiffrement et la conversion de protocoles sont utilisés pour rendre les réseaux plus rapides.

Dans les domaines de l'aéronautique et de l'automobile, les FPGA sont largement utilisés pour l'implémentation de systèmes radar, de navigation, de communications sécurisées et de contre-mesures électroniques en raison de leur flexibilité, de leur parallélisme et de leur capacité à fonctionner dans des environnements contraignants. Ils sont également employés dans les systèmes avancés d'aide à la conduite (ADAS), la fusion de capteurs et les applications multimédias embarquées nécessitant un traitement rapide et fiable des données [32].

Dans le domaine de l'intelligence artificielle, les FPGA constituent une plateforme efficace pour l'accélération matérielle des réseaux neuronaux et du traitement de grandes quantités de données en temps réel [33, 34]. Cette approche est notamment utilisée en robotique, en inspection intelligente et dans les systèmes de contrôle automatisé, où elle contribue à améliorer les performances de calcul, à réduire la latence et à accroître l'autonomie des systèmes [35, 36].

La figure 2.5 montre la vue réelle de la carte Nexys-4-DDR fabriquée par Digilent. Cette carte, utilisée dans nos travaux, est équipée d'un FPGA Xilinx Artix-7 (XC7A100T-1CSG324C) avec suffisamment de ressources logiques, d'entrées/sorties numériques et de blocs de mémoire RAM (DDR2 SDRAM) intégrée pour l'implémentation des protocoles de communication ainsi que pour la génération de signaux de commande. Le tableau 2-2 présente quelques exemples de cartes FPGA disponibles commercialement et leurs utilisations.

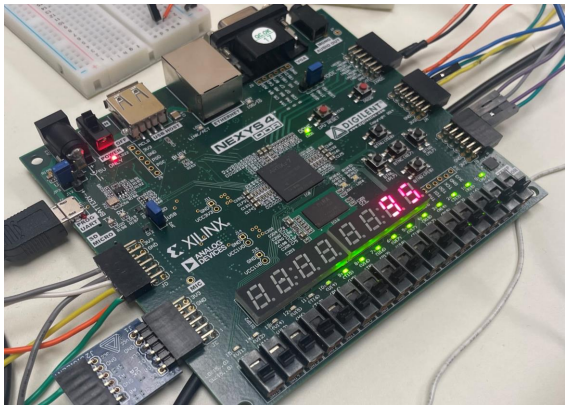


Figure 2.5 Carte FPGA Nexys 4 DDR - Xilinx Artix-7 - Digilent.

Tableau 2-2 Exemples de cartes FPGA commerciales et leurs utilisations.

Carte FPGA	**Fabricant**	**FPGA utilisé**	**Utilisation principale**
Nexys 4 DDR	Digilent / Xilinx	Artix-7 XC7A100T	Utilisée dans ce projet. Idéale pour la robotique, les systèmes embarqués et la génération de signaux (PWM, SPI, I^2C, UART).
Basys 3	Digilent / Xilinx	Artix-7 XC7A35T	Carte éducative populaire pour l'apprentissage des logiques combinatoires, séquentielles et de la conception numérique de base.
Nexys A7	Digilent / Xilinx	Artix-7 XC7A100T	Version modernisée de la Nexys 4, adaptée aux projets de commande, de traitement du signal et d'interfaçage avec microcontrôleurs ou Raspberry Pi.
DE10-Lite	Intel / Terasic	MAX 10 FPGA	Très utilisée pour l'initiation au FPGA et les projets légers. Compatible avec Quartus Prime.
DE10-Nano	Intel / Terasic	Cyclone V SoC	Combine un FPGA et un processeur ARM double cœur. Utilisée pour l'embarqué avancé, la vision et les systèmes Linux + FPGA.
ZedBoard	Avnet / Xilinx	Zynq-7000 SoC	Intègre un processeur ARM Cortex-A9 et un FPGA. Employée pour les systèmes complexes, le traitement temps réel et la robotique avancée.

Les GPU (graphics processing units) sont des processeurs spécialisés conçus principalement pour accélérer le rendu des images et des vidéos. Bien que les architectures GPU présentent une capacité de calcul élevée, particulièrement adaptée aux algorithmes d'apprentissage automatique et au traitement massif de données, leur consommation énergétique élevée constitue une contrainte majeure. Cette limitation entrave leur intégration dans des systèmes embarqués à faible consommation [37]. La figure 2.6 présente à titre d'exemple une vue réelle d'une carte GPU Nvidia.

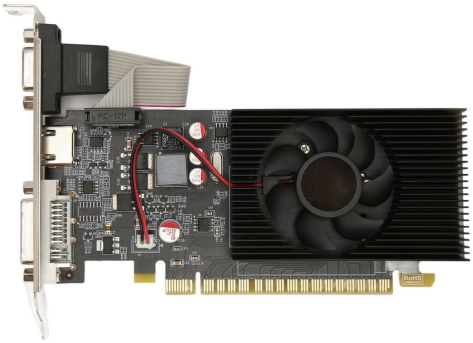


Figure 2.6 Carte graphique (GPU) de type NVIDIA, utilisée pour le traitement graphique et l'accélération des calculs parallèles.

Un exemple de micro-ordinateur est le Raspberry Pi, un ordinateur monocarte (SBC - single-board computer) développé par la fondation Raspberry Pi au Royaume-Uni, qui utilise, dans la plupart des cas, le langage Python. Dans le cadre de ce projet, le micro-ordinateur joue un rôle fondamental dans la collecte, le traitement et la transmission des données provenant des différents capteurs intégrés au robot. La figure 2.7 montre une vue réelle d'une carte Raspberry Pi 3 Model B+. Le tableau 2-3 présente un résumé des versions commerciales du micro-ordinateur monocarte Raspberry Pi.

Le tableau 2-3 présente les principales versions du Raspberry Pi ainsi que leurs caractéristiques matérielles, d'après la documentation officielle de la Raspberry Pi Foundation [38].

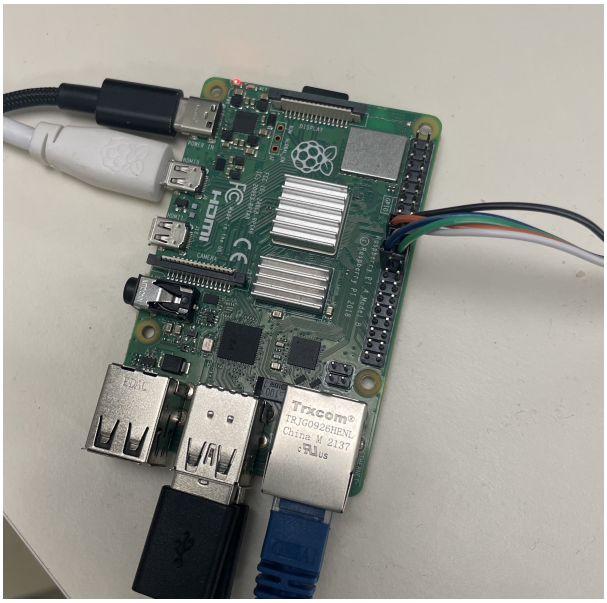


Figure 2.7 Carte Raspberry Pi 3 Model B+ .

Tableau 2-3 Principales versions du Raspberry Pi et leurs utilisations.

Modèle	**Année**	**Processeur**	**Utilisation principale**
Pi 1 (A/B)	2012–2014	ARM11 @ 700 MHz	Modèle éducatif pour l'apprentissage et les petits montages.
Pi 2 (B)	2015	Quad Cortex-A7 @ 900 MHz	Projets embarqués simples et automatisation de base.
Pi 3 (B/B+)	2016–2018	Quad Cortex-A53 @ 1.2 GHz	Utilisé en robotique et IoT. Dans ce projet : communication via SPI, I^2C, UART et commande du servomoteur.
Pi 4 (B)	2019	Quad Cortex-A72 @ 1.5 GHz	Vision, traitement d'images et intelligence artificielle embarquée.
Pi Zero / 2 W	2015–2021	Cortex-A53 @ 1 GHz	Format compact, faible consommation, adapté à l'IoT et aux mini-robots.
Pi 5	2023	Cortex-A76 @ 2.4 GHz	Robotique avancée et applications agricoles autonomes.

2.4.3 Communication et interfaçage de capteurs et actionneurs

Dans les systèmes de robotique agricole, la communication et l'acquisition de données constituent des éléments essentiels au bon fonctionnement des architectures de commande. Les robots agricoles s'appuient sur une grande diversité de capteurs pour mesurer et assurer l'acquisition de données environnementales (p.ex., l'humidité et la température) et de données relatives à l'opération du système (p.ex., position, vitesse, accélération, couple) [39, 40].

Dans un système robotique de complexité moyenne, il est possible de trouver plusieurs technologies de capteurs qui utilisent différents protocoles de communication. Certains capteurs sont branchés directement aux processeurs sous forme d'entrées binaires, et d'autres via des systèmes de conversion analogique-numérique (ADC pour analog-to-digital converter) à plusieurs bits. Les protocoles SPI (Serial Peripheral Interface), I^2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver/Transmitter) et Ethernet sont couramment utilisés pour l'intégration des capteurs via des ADC, pour envoyer des informations de commande et pour réaliser la supervision des opérations.

Le protocole **SPI (Serial Peripheral Interface)** permet un transfert rapide et synchrone entre les processeurs numériques (p.ex: Raspberry Pi, FPGA) et entre le processeur et les ADC solidaires des capteurs. Le protocole SPI est un standard de communication synchrone largement utilisé pour les échanges de données à haute vitesse entre microcontrôleurs et périphériques dans les systèmes embarqués [41]. En effet, un seul appareil est utilisé pour initier toutes les communications avec les esclaves. Il est utilisé pour la communication entre les circuits intégrés mais présente un inconvénient majeur qui est l'utilisation de plusieurs broches [42], c'est un protocole qui fonctionne sur

quatre lignes de signaux. Selon la littérature scientifique, les avantages du protocole SPI reposent sur sa simplicité, son débit élevé et sa fiabilité pour les applications embarquées et en temps réel [43, 44].

Selon la littérature scientifique, le protocole SPI se distingue par sa simplicité, son débit élevé et sa fiabilité pour les applications embarquées et en temps réel. Par ailleurs, la communication RC (Radio Control), basée sur des signaux MLI, est largement utilisée dans les drones, les robots mobiles et les véhicules télécommandés pour transmettre les commandes de pilotage grâce à sa simplicité, sa robustesse et sa faible latence. Le protocole SPI utilise quatre lignes de communication principales [43]. Le signal d'horloge (SCLK), généré par le maître du bus, est envoyé à l'ensemble des périphériques esclaves afin de synchroniser les échanges de données. Le signal de sélection d'esclave (SS ou SSn) dédié à chaque périphérique permet au maître d'initier et de contrôler la communication avec l'esclave sélectionné. La ligne MOSI (Master Out – Slave In) assure le transfert des données du maître vers l'esclave, tandis que la ligne MISO (Master In – Slave Out) permet à l'esclave de transmettre des données au maître [45].

Le fonctionnement est le suivant : lorsque le maître souhaite envoyer ou recevoir des données, il sélectionne l'esclave concerné en mettant la ligne SS correspondante à l'état bas (0). Il génère ensuite un signal d'horloge sur la ligne SCLK, à une fréquence compatible avec les deux dispositifs. À chaque front d'horloge, un échange simultané de données s'effectue : le maître envoie un bit à l'esclave via la ligne MOSI, tandis que l'esclave peut renvoyer un bit au maître via la ligne MISO. Ce mécanisme de communication full-duplex se poursuit bit par bit, généralement par blocs de 8 bits. Une fois la transmission terminée, le maître désactive l'esclave en remettant la ligne SS à l'état actif (1). Durant tout l'échange, le maître génère les données sur la ligne MOSI tout en

échantillonnant simultanément la ligne MISO [43]. Les Tabelaux 2-4 et 2-5 présentent un résumé des lignes utilisées et les principales caractéristiques du protocole SPI.

Tableau 2-4 Ligne de communication SPI.

Nom de la ligne	Rôle	Direction
MOSI (Master out Slave In)	Données du maître vers l'esclave	Maître-esclave
MISO(Master in Slave Out)	Données de l'esclave vers maître	Esclave- Maître
SCLK(serial clock)	Horloge générée par le maître	Maître-esclave
SS/CS (Slave/chip select)	Sélection de l'esclave actif	Maître-esclave

Tableau 2-5 Caractéristiques SPI.

Avantage	Inconvénient
Rapide (plus qu'I^2C)	Utilise plus de fils
Envoie et reçoit en même temps	Nécessite une ligne SS par esclave
Simple à implémenter en matériel	Distance limitée

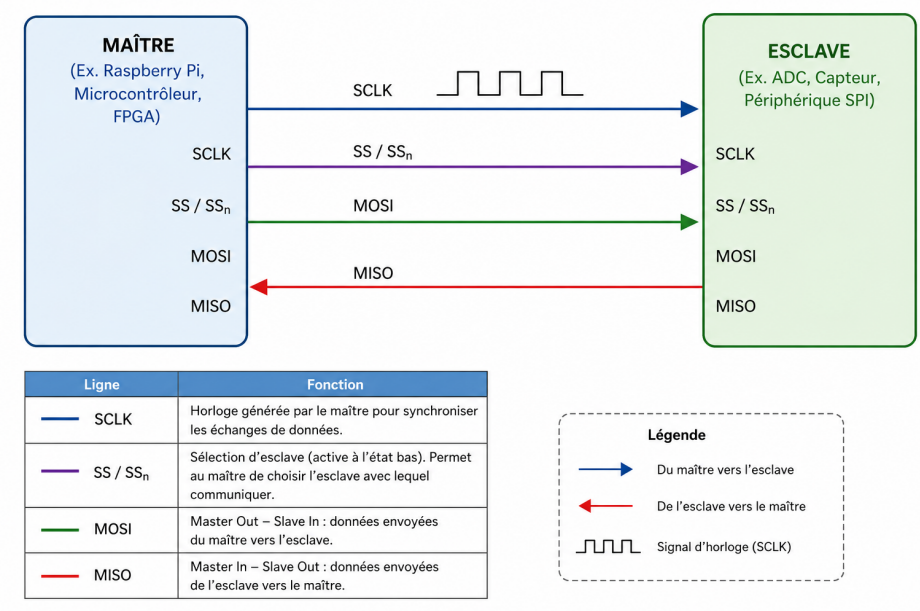


Figure 2.8 Connexion entre les lignes de communications SPI.

Le protocole **I^2C (Inter-Integrated Circuit)** est particulièrement adapté à la communication avec des capteurs à faible taux de transfert de données, tels que le convertisseur analogique-numérique HX711 et d'autres modules ADC. Il permet de configurer le processeur numérique en maître pour superviser et interroger simultanément plusieurs dispositifs esclaves connectés au même bus. L'I^2C figure parmi les protocoles de communication numériques les plus couramment utilisés. C'est un standard largement adopté pour la communication entre différents composants électroniques [29].

Ce protocole permet à plusieurs périphériques comme des capteurs ou des convertisseurs, de communiquer avec un microcontroleur ou un FPGA en utilisant seulement deux fils (ce qui est l'une de ses particularités par rapport aux autres protocoles) facilitant ainsi l'intégration et la réduction du câblage nécessaire dans les systèmes embarqués [46]. C'est un protocole de communication série synchrone à courte distance, conçu et développé il y a enivrions trente ans par Philips Semicondutors [29]. Il a été spécifiquement pensé pour permettre une communication simple et efficace entre différents composants sur une même carte ou dans un système embarqué avec un faible nombre de fils nécessaires pour le transfert de données.

Il permet d'interfacer des circuits intégrés lents avec des microcontrôleurs et des microprocesseurs, ou avec des FPGA rapides. L'I^2C simplifie l'adressage des composants grâce à l'adressage de puce, éliminant la sélection manuelle. Il intègre également un mécanisme de détection et de correction des erreurs, avec un accusé de réception (ACK/NACK), afin d'assurer une transmission fiable des données [29]. Étant donné le fait qu'il utilise juste deux fils bidirectionnels pour la transmission des données, cela rend la conception du contrôleur plus complexe, car les données doivent être synchronisées avec une horloge. Malgré cette complexité, I^2C offre un protocole de communication

robuste, extensible, flexible et à faible coût. C'est un protocole de transfert de données série populaire. Il a plusieurs avantages, notamment la contrôlabilité, et ne demande pas beaucoup de connexions filaires [44]. Il possède plusieurs modes de fonctionnement, mais le choix dépend de l'application [42]. De plus, l'interface et la stratégie de contrôle de chaque module sont spécifiquement adaptées aux exigences propres à chaque application [47].

Le protocole **UART (Universal Asynchronous Receiver/Transmitter)** est généralement utilisé pour l'échange d'informations entre processeurs numériques. Par exemple pour la transmission des commandes depuis le processeur de supervision vers le processeur chargé de générer les signaux de commande. Ce canal asynchrone permet également d'échanger des données ou de contrôler une interface de supervision sur un ordinateur. Il est plus utilisé pour l'échange de données à petite distance, à faible vitesse et à un coût réduit entre un ordinateur et ses périphériques [22, 48]. L'article de Yang (2024) propose une approche de communication série fiable pour les systèmes embarqués en combinant DMA et interruptions IDLE, et compare cette solution aux communications série traditionnelles [49].

Ce protocole permet à deux appareils de communiquer directement. D'un côté, l'appareil qui envoie les données (l'émetteur) reçoit les informations envoyées en parallèle par le processeur (plusieurs bits à la fois), les transforme en une seule suite de bits envoyés un par un (en série), puis les transmet. De l'autre côté, l'appareil qui reçoit (le récepteur) prend cette série de bits et les convertit en données utilisables en parallèle, comme si elles provenaient directement d'un processeur [50]. Concernant le mode de communication, les UART communiquent sans utiliser de signal d'horloge partagé. Cela veut dire que l'émetteur et le récepteur ne sont pas parfaitement synchronisés. Pour que le récepteur

sache quand une transmission commence et finit, l'émetteur ajoute un bit de début (start bit) avant les données et un bit de fin (stop bit) après.

Grâce à ces marqueurs, le récepteur peut déterminer où commencent et se terminent chaque paquet de données, même sans être parfaitement synchronisé avec l'émetteur [50]. Selon l'application, nous pouvons parfois ne pas avoir besoin d'utiliser toutes les fonctionnalités de l'UART, par exemple dans une application industrielle. Dans cette application, on utilise uniquement la partie centrale de l'UART [22]. Elle est constituée des éléments suivants : le générateur de débit en bauds, le récepteur et l'émetteur [22]. Pour réaliser une communication de données en duplex, il faudrait juste deux lignes de signal: le **TXD**, comme le mot transmission, qui est le côté émission ou la sortie de l'UART, et le **RXD**, le côté récepteur ou encore l'entrée de l'UART. Lorsque l'émetteur est inactif, la ligne de données est au niveau logique actif **1** [22].

Le protocole utilise trois bits de contrôle : le bit de démarrage, le bit de parité et le bit d'arrêt. Le bit de démarrage: il est utilisé pour alerter le récepteur qu'un mot de données est sur le point d'être envoyé. Il est également utilisé pour forcer la synchronisation des horloges du récepteur et de l'émetteur afin que pendant la transmission des autres bits du , il n'y ait pas une dérive de fréquence de plus de 10%. C'est pourquoi les signaux d'horloge doivent être suffisamment précis pour garantir une synchronisation fiable et un échange correct des données.

Après le bit de démarrage, les bits de données sont envoyés du bit le plus faible (LSB) au plus fort (MSB). La transmission de tous les bits se fait avec la même durée. Le récepteur détermine si le bit est **1** ou **0** en observant le fil à environ la moitié de la période de chaque bit. Puis, il attendra quelques il attend ensuite une période de bit pour passer à l'examen du prochain bit, ainsi de suite.

Après la transmission des bits de données, l'émetteur insère un bit de parité destiné à la détection d'erreurs par le récepteur. La trame se termine ensuite par un ou plusieurs bits d'arrêt signalant la fin de la communication. Si le bit d'arrêt n'apparaît pas au moment demandé, l'UART considère que le mot entier est brouillé et signale une erreur de cadrage au processeur hôte. Dès que le bit d'arrêt précédent a été envoyé et qu'un autre mot est prêt à être transmis, le bit de démarrage du nouveau mot peut être envoyé [49].

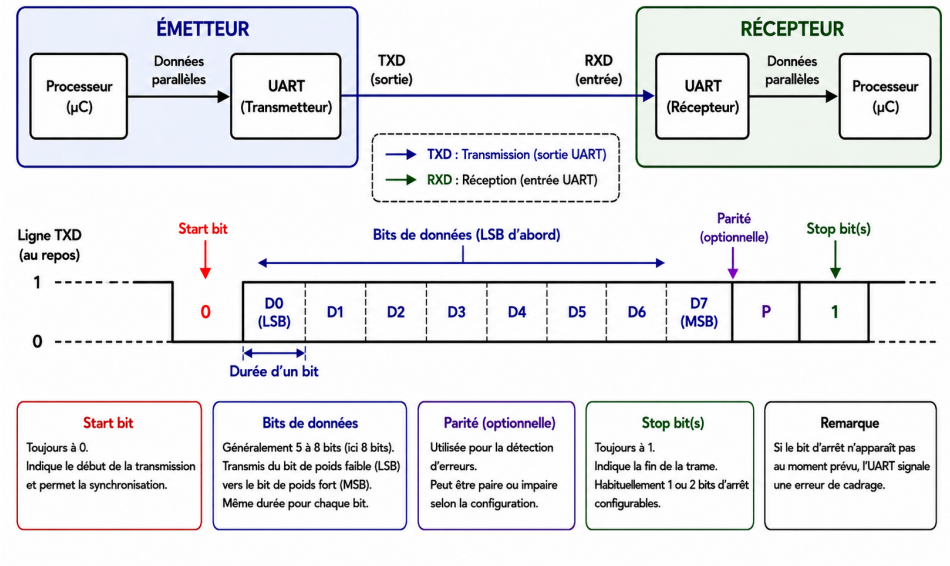


Figure 2.9 Format du paquet de données UART.

Le protocole **Ethernet** est un protocole de communication filaire largement utilisé dans les réseaux informatiques pour connecter différents périphériques au sein d'un réseau local (LAN). Dans le contexte des systèmes embarqués, Ethernet est particulièrement adapté aux applications nécessitant une grande bande passante, une communication fiable et des distances de transmission longues [51].

Le mode de transmission Ethernet repose sur un réseau câblé et s'appuie généralement sur la pile de protocoles TCP/IP pour établir les communications [52]. Les données sont échangées sous forme de paquets, ce qui permet une gestion efficace du trafic réseau

et une communication simultanée entre plusieurs équipements [53, 54]. Le protocole Ethernet se caractérise par des débits élevés, une communication robuste et la possibilité de connecter plusieurs équipements au sein d'un même réseau à l'aide de dispositifs tels que les commutateurs ou les routeurs. Il permet une transmission rapide des données sur de longues distances et facilite l'interconnexion avec des réseaux étendus grâce au protocole IP. Néanmoins, la mise en œuvre de ce protocole implique une architecture réseau plus complexe ainsi qu'une consommation énergétique accrue par rapport aux protocoles de communication élémentaires, ce qui peut limiter son intégration dans des systèmes embarqués à ressources contraintes [55].

Pour déterminer la technologie la plus appropriée pour un système de commande, plusieurs critères doivent être pris en considération, notamment les objectifs du projet, le mode de fonctionnement, le coût d'implémentation, l'impact environnemental ainsi que la consommation énergétique. À la lumière des différentes analyses effectuées dans la littérature, certains travaux, tels que l'étude comparative présentée dans [56] portant sur les protocoles **UART, I^{2}C** et **SPI**, mettent en évidence les performances et l'efficacité du protocole SPI pour les applications de mesure et de commande à haut débit. Les tableaux 2-6 et 2-7 présentent les principales caractéristiques et une comparaison des protocoles I^{2}C, SPI et UART.

Tableau 2-6 Comparaison des protocoles I^2C et SPI.

Critère	**I^2C**	**SPI**
Année / Auteur	Philips (1982)	Motorola (1979)
Interface	Série (2 fils)	Série (3–4 fils)
Débit	Faible à moyen	Élevé
Mode de transfert	Demi-duplex	Duplex intégral
Complexité	100 kbit/s à 3,4 Mbit/s	1 à 50 Mbit/s
Multi-esclave	Oui	Oui (CS requis)
Multi-maître	Oui	Non
Adressage	Logiciel	Matériel (CS)
Consommation	Faible	Faible
Application	Capteurs	Acquisition rapide, contrôle temps réel, FPGA

Tableau 2-7 Comparaison des protocoles SPI, I^2C et UART.

Protocole	**Fils**	**Débit**	**Caractéristiques principales**
SPI	4	10–50 Mbit/s	Full-duplex, rapide, nécessite davantage de fils.
I^2C	2	Jusqu'à 3,4 Mbit/s	Peu de fils, multi-esclaves.
UART	2	115,2 kbit/s à quelques Mbit/s	Simple, communication longue distance, point à point.

2.4.4 Actionneurs électromécaniques

Un actionneur est un dispositif qui convertit une forme d'énergie, qu'elle soit électrique, hydraulique, thermique ou pneumatique, en mouvement ou en action mécanique. Les actionneurs servent à entraîner, à déplacer ou à contrôler un mécanisme dans un système automatisé. Ils trouvent des applications dans différents domaines. Le tableau 2-8 présente quelques exemples d'applications des actionneurs.

Tableau 2-8 Exemples d'applications des actionneurs.

Domaine	**Exemples d'application**
Industrie	Robots manipulateurs (moteurs électriques + vérins)
Médecine	Pompes à insuline (micro-actionneurs), robots chirurgicaux
Aéronautique	Volets d'avions (actionneurs hydrauliques)
Robotique moderne	Robots souples, pinces pneumatiques

On distingue plusieurs catégories d'actionneurs selon la source d'énergie utilisée :

- **Les actionneurs électriques et électromécaniques**, qui transforment l'énergie électrique en mouvement mécanique contrôlé. Ils sont largement utilisés dans les systèmes automatisés, notamment pour la commande des moteurs.
- **Les actionneurs hydrauliques**, qui exploitent la pression d'un fluide (huile ou eau) pour produire un mouvement, généralement dans des applications nécessitant une grande force.
- **Les actionneurs pneumatiques**, qui utilisent l'air comprimé pour générer un mouvement rapide et efficace.
- **Les actionneurs thermiques**, qui fonctionnent grâce aux variations de température pour produire un déplacement mécanique [57].

Les **actionneurs électriques et électromécaniques** sont les éléments responsables de la transformation de l'énergie électrique en énergie mécanique dans les systèmes

agricoles électrifiés. Ils assurent l'exécution d'actions physiques telles que le déplacement, la rotation, le positionnement et l'orientation des équipements [1]. Dans les systèmes modernes d'agriculture de précision, leur fonctionnement est étroitement lié à l'électronique embarquée ainsi qu'aux chaînes de commande et de puissance [58].

Les actionneurs électriques et électromécaniques sont largement utilisés dans les systèmes embarqués en raison de leur précision, de leur fiabilité et de leur facilité de commande. Parmi les principaux types d'actionneurs, on distingue les moteurs et servomoteurs à courant continu (DC), les moteurs à courant continu sans balais (BLDC), les moteurs pas à pas et les moteurs à courant alternatif (AC). D'autres dispositifs, tels que les solénoïdes, permettent de produire des déplacements linéaires, tandis que les actionneurs piézoélectriques sont utilisés pour des mouvements de très faible amplitude nécessitant une grande précision.

Les **actionneurs pneumatiques** utilisent de l'air comprimé pour produire un mouvement mécanique, notamment à l'aide de vérins. Ils sont rapides, légers et propres. Toutefois, ils présentent une précision limitée et nécessitent des équipements auxiliaires, dont un compresseur et des régulateurs de pression.

Les **actionneurs hydrauliques** fonctionnent à l'aide d'un fluide sous pression. Ils offrent une forte puissance et une grande robustesse. Ils ont un encombrement important, des risques de fuites, un entretien régulier requis, un coût d'installation élevé et une précision de commande parfois inférieure à celle des actionneurs électriques [57].

Les **actionneurs thermiques, à mémoire de forme (SMA) et souples** reposent sur la déformation d'un matériau sous l'effet de la température ou d'un courant électrique. Ils sont utilisés en robotique souple et en micromécanique. Les actionneurs souples sont

des dispositifs innovants utilisés en robotique bio-inspirée. Ils utilisent des technologies de polymères électroactifs et des systèmes pneumatiques souples. On les utilise dans des applications de robots médicaux, de prothèses intelligentes et de robots inspirés du vivant [57].

Les **moteurs électriques** sont largement utilisés dans les machines agricoles électrifiées en raison de leur rendement élevé, de leur fiabilité et de leur facilité de commande. Ils interviennent notamment dans la traction des véhicules, l'entraînement des outils agricoles ainsi que dans la réalisation de mouvements continus. Parmi les technologies les plus couramment employées figurent les moteurs à courant continu, les moteurs asynchrones, les moteurs synchrones et les moteurs brushless. Le choix du type de moteur dépend principalement des exigences en matière de puissance, de couple et de vitesse propres à l'application considérée [58]. Dans le domaine de l'agriculture de précision, les moteurs électriques offrent un avantage significatif en matière de contrôle précis et d'intégration avec les systèmes automatisés et autonomes [1].

Les **servomoteurs** sont des actionneurs électromécaniques utilisés lorsque des mouvements précis et contrôlés sont requis. Ils intègrent généralement un moteur électrique, un capteur de position ainsi qu'une électronique de commande interne permettant d'assurer un asservissement précis. Leur fonctionnement repose le plus souvent sur la modulation de largeur d'impulsion (MLI), qui permet de contrôler avec précision la position angulaire ou la vitesse selon l'application [59]. En robotique agricole, les servomoteurs sont utilisés pour le positionnement d'organes mécaniques, l'orientation d'outils ou la direction de véhicules autonomes, où la précision et la répétabilité sont essentielles [60].

2.4.5 Convertisseurs de puissance

Les **convertisseurs de puissance** assurent l'interface entre les unités de commande et les actionneurs. Ils adaptent les niveaux de tension et de courant nécessaires au fonctionnement des moteurs et des servomoteurs, tout en intégrant des fonctions de protection [61]. La conception des convertisseurs repose sur les principes fondamentaux de l'électronique de puissance, garantissant un fonctionnement sûr et efficace des actionneurs [62]. La gestion de la puissance est essentielle pour assurer le bon fonctionnement des actionneurs, notamment lors des phases de démarrage ou sous charge élevée. Elle influence directement les performances et la fiabilité du système [63, 64]. Les éléments de puissance, tels que les MOSFET, les IGBT et les diodes, sont disposés en structures simples, telles que le pont en H, ou en topologies plus complexes, telles que les convertisseurs multiniveaux [65].

Ces circuits permettent de réguler l'apport d'énergie électrique aux actionneurs. Leur choix conditionne la sécurité et l'efficacité énergétique des systèmes électromécaniques [66]. Selon le type d'actionneur, la puissance doit être fournie soit par modulation d'une tension continue, soit par des tensions alternatives. Comme la source principale embarquée dans le robot est souvent de type continu, il peut être nécessaire d'utiliser des convertisseurs pour fournir des tensions de sortie de type continu (hacheurs) ou de type alternatif (onduleurs) toujours à partir de la même source continue.

Les **hacheurs** sont des dispositifs électroniques permettant de convertir une tension continue en une autre tension continue de valeur différente, soit en la réduisant, soit en l'augmentant [67, 68]. Il existe plusieurs types de hacheurs utilisés dans les systèmes de conversion d'énergie:

- **Hacheur élévateur (Step-up ou Boost Converter)**: Ce type de convertisseur permet d'augmenter la tension de sortie par rapport à la tension d'entrée [67].
- **Hacheur abaisseur (Step-down ou Buck Converter)**: Il réduit la tension de sortie par rapport à la tension d'entrée tout en maintenant un rendement énergétique élevé [67].
- **Hacheur bidirectionnel**: Ce convertisseur permet à la fois d'augmenter ou de diminuer la tension selon les exigences de l'application. Il est particulièrement utilisé dans les systèmes de stockage d'énergie, notamment dans les applications à batteries [69].

Le choix du type de hacheur dépend des exigences en matière de conversion d'énergie et de régulation de la tension [68]. Ils sont utilisés dans les alimentations à découpage, le contrôle de moteurs électriques et les systèmes de conversion d'énergie. Ils assurent une régulation efficace de l'énergie tout en limitant les pertes et en améliorant le rendement du système.

Les **onduleurs** sont des dispositifs électroniques qui convertissent une tension continue en tension alternative selon une stratégie de commutation appropriée [70, 71]. Après modulation et filtrage, le signal obtenu se rapproche d'une onde sinusoïdale adaptée aux applications résidentielles ou industrielles [72]. Il existe plusieurs types d'onduleurs, dont la classification dépend principalement de leur architecture, de leur mode de commande et de l'application visée [72].

- **Onduleur à onde carrée**: Comme son nom l'indique, il produit une onde carrée. Ce type d'onduleur est simple à mettre en œuvre, mais génère un signal présentant un taux d'harmoniques élevé [72].
- **Onduleur à onde sinusoïdale**: il génère une approximation de l'onde sinusoïdale,

généralement sous forme d'onde en escalier. Il constitue un compromis entre la simplicité de conception et la qualité du signal [72].

- **Onduleur à onde sinusoïdale pure**: il produit une onde sinusoïdale de haute qualité, similaire à celle du réseau électrique domestique, ce qui le rend adapté aux équipements sensibles [71].

2.4.6 Capteurs et éléments de mesure

Les **capteurs et éléments de mesure** jouent un rôle essentiel dans les systèmes d'agriculture de précision en permettant la collecte de données fiables sur l'état des cultures, du sol et de l'environnement. Ils constituent la base des systèmes de surveillance et de contrôle, en fournissant des informations nécessaires à la prise de décision et à l'optimisation des pratiques agricoles. Grâce à ces dispositifs, il est possible de mesurer différentes grandeurs physiques telles que l'humidité, la température, la position, la vitesse ou encore le courant, ce qui contribue à une gestion plus efficace et automatisée des systèmes agricoles.

Un **encodeur** est un dispositif électromécanique qui convertit une information physique (telle que le mouvement, la position ou la vitesse) en un signal électrique ou numérique exploitable par un système électronique. Il remplit ainsi une fonction de codage, qui consiste à transformer une grandeur mécanique en une représentation numérique, généralement sous la forme d'un signal binaire [73]. Nous pouvons dire qu'un encodeur est un capteur qui convertit un mouvement mécanique (généralement une rotation ou un déplacement linéaire) en un signal électrique utilisable par un système électronique (comme un microcontrôleur ou un FPGA). Les encodeurs peuvent être optiques (à haute précision), linéaires (mesure du déplacement dans des environnements industriels) ou magnétiques (robustes et adaptés aux systèmes automatisés).

Dans le cadre de l'électronique de commande pour les systèmes électromécaniques en agriculture de précision, les encodeurs jouent un rôle essentiel en convertissant les mouvements mécaniques (rotation ou translation) des actionneurs en signaux électriques exploitables par les systèmes de commande [74]. Ces informations permettent de connaître avec précision la position, la vitesse et le sens de déplacement des équipements agricoles automatisés. Selon leur type et leur mode de fonctionnement, les encodeurs peuvent fournir une mesure de position absolue ou relative. L'analyse de la fréquence des signaux générés permet également d'estimer la vitesse de rotation ou de déplacement. De plus, les encodeurs à quadrature utilisent deux signaux déphasés pour déterminer le sens du mouvement. Grâce à ces informations, ils contribuent au contrôle précis et à la synchronisation des différents actionneurs au sein des systèmes agricoles automatisés.

Ils peuvent également être utilisés pour générer des signaux ou des impulsions de référence afin de contrôler d'autres composants. Ce qui permet au système de savoir quand déclencher les actions spécifiques. Un encodeur incrémental peut être utilisé pour compter des événements répétitifs, tels que la rotation d'un arbre moteur ou le passage d'un objet sur la ligne de production. Ils peuvent fournir des informations pour la surveillance de l'état d'un système et signaler les anomalies (mouvements inattendus, arrêts brusques, vitesses ou positions hors des limites définies). Dans certains cas, les encodeurs sont utilisés comme dispositifs d'entrée pour les systèmes d'interface utilisateur (en anglais, Human-Machine Interface, HMI). Il existe plusieurs types d'encodeurs, qui se distinguent par leur principe de fonctionnement et leur domaine d'application dans les systèmes de mesure et de contrôle.

L'**encodeur incrémental** fournit une **position relative** et une estimation de la **vitesse** à partir d'une série d'impulsions successives.

L'**encodeur absolu** fournit une valeur de **position unique** pour chaque position de l'axe, ce qui permet de retrouver immédiatement l'état du système après une coupure d'alimentation.

L'**encodeur magnétique** fonctionne par détection de champs magnétiques et se distingue par sa robustesse dans des environnements exposés à la poussière, à l'humidité et aux vibrations.

L'**encodeur linéaire** mesure avec une grande précision le déplacement d'un objet dans un système mécanique.

Le tableau 2-9 présente les principaux types d'encodeurs utilisés en agriculture de précision, ainsi que leurs rôles et leurs applications. La figure 2.10 présente à titre d'exemple une vue réelle d'un encodeur rotatif optique fixé à l'arbre d'un moteur à courant continu utilisé dans le projet. Le choix de l'encodeur dépend de la grandeur à mesurer, telle que la position, la vitesse ou le déplacement, ainsi que des contraintes de robustesse et de précision propres à l'environnement agricole. Notons que la vitesse est une variable essentielle pour la commande de moteurs et de robots agricoles. Elle permet d'assurer une exécution précise des tâches et est généralement mesurée à l'aide d'encodeur puis traitée par les microcontrôleurs, FPGA ou unités de contrôle embarquées chargés de piloter les actionneurs [75, 76]. La position est essentielle pour le guidage, le suivi des trajectoires et l'automatisation des opérations en agriculture de précision et peut être estimée à partir des impulsions fournies par un encodeur incrémental [77, 78].

Dans le cadre de ce travail, différentes grandeurs physiques sont analysées, notamment la vitesse, la position, le courant et la force, afin de garantir un contrôle précis et une meilleure performance du système électromécanique.

Tableau 2-9 Types d'encodeurs, rôles et applications agricoles.

Type d'encodeur	**Rôle**	**Application agricole**
Incrémental	Vitesse / déplacement	Roues d'un robot agricole
Absolu	Position précise	Bras robotisé / vérin électrique
Magnétique	Vitesse + robustesse	Tracteur
Linéaire	Déplacement linéaire	Semoir / doseur d'engrais
Optique (rotatif)	Angle de rotation	Guidage moteur

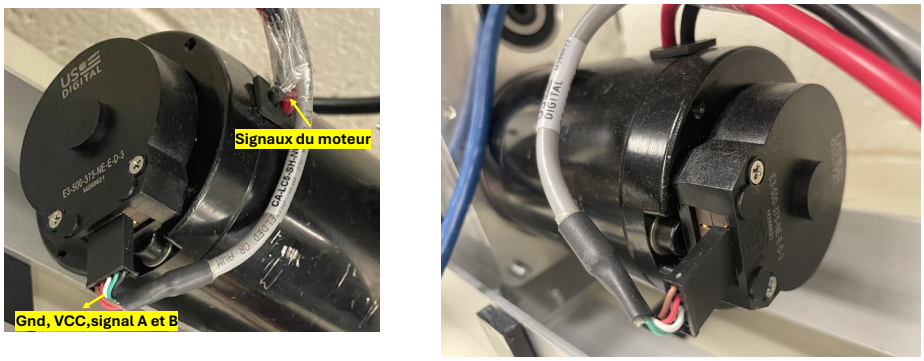


Figure 2.10 Vue réelle de l'encodeur utilisé pour l'expérimentation.

Le **courant électrique** constitue une grandeur fondamentale dans la commande des systèmes électromécaniques, car il est directement lié au couple produit par le moteur [63, 79]. Sa mesure permet de surveiller l'état du système, d'optimiser la consommation d'énergie et de détecter les surcharges dans les applications agricoles [21].

Un **capteur de courant** est un dispositif permettant de mesurer l'intensité électrique circulant dans un circuit et de convertir cette information en un signal exploitable, généralement sous forme de tension ou de données numériques.

La **mesure de la force** permet d'adapter l'action mécanique du robot à son environnement. Elle contribue à améliorer la sécurité, la précision des interventions et la protection des cultures et des équipements [80]. Complémentairement, les capteurs et systèmes de télédétection permettent, quant à eux, de collecter des données essentielles sur l'état du sol, la santé des cultures, l'humidité, la température et les conditions environnementales [81].

L'agriculture de précision repose sur l'utilisation combinée de plusieurs technologies permettant une gestion optimisée des opérations agricoles. Parmi les éléments clés figurent les **systèmes de localisation et de navigation** utilisés pour la cartographie des champs, le guidage des machines agricoles et l'amélioration de la précision des interventions, ainsi que les **systèmes d'information géographique (SIG)**, qui jouent un rôle important dans la visualisation et l'analyse des données spatiales collectées, facilitant ainsi la prise de décision [82]. De plus, les drones et les images satellitaires offrent une vue aérienne des parcelles, permettant de surveiller la croissance des cultures, de détecter les maladies et d'identifier les besoins en eau et en nutriments. Les modèles prédictifs et l'analyse de données contribuent à la prévision des rendements, à l'optimisation des calendriers de plantation et à la gestion des risques climatiques [83].

Dans ce contexte, le contrôleur ou récepteur constitue l'élément central assurant la réception, le traitement et la transmission des informations provenant de ces différentes sources [84]. Il permet de piloter les systèmes électromécaniques complexes, notamment des moteurs électriques, des relais, et des dispositifs de commande à distance. En assurant la coordination entre capteurs, unités de traitement et actionneurs, le contrôleur garantit le fonctionnement cohérent, automatisé et fiable des systèmes agricoles modernes [85, 86].

Un **système d'acquisition de données** un module et logiciel matériel dont la mission principale est de collecter des données physiques via des capteurs [87]. Puis les convertir en signaux numériques afin qu'ils soient traités par un microcontrôleur, un FPGA ou un ordinateur [88]. Ainsi, un système d'acquisition de données permet de mesurer une grandeur physique et de la transformer en information numérique exploitable par un système de traitement. Il comprend généralement le capteur, une ou plusieurs étapes de conditionnement (filtrage, adaptation de niveaux, isolations), une conversion analogique-numérique, une unité de traitement, un moyen de stockage et une interface de communication et d'affichage permettant aux utilisateurs d'accéder aux informations.

Comme son nom l'indique, un **convertisseur analogique-numérique** transforme un signal analogique en une valeur numérique exploitable par un microcontrôleur, un FPGA ou un ordinateur. La performance d'un convertisseur analogique-numérique (CAN) dépend essentiellement de sa fréquence d'échantillonnage. L'augmentation du taux d'échantillonnage améliore la précision et la qualité de la conversion analogique–numérique, mais elle implique également une complexité accrue du système ainsi qu'un coût plus élevé [29].

Les CAN intègrent des protocoles de communication numérique qui sont utilisés pour assurer le transfert et la synchronisation des données entre le convertisseur et l'unité de traitement (P.ex., FPGA ou microcontrôleur), garantissant aussi une interaction fluide et efficace.

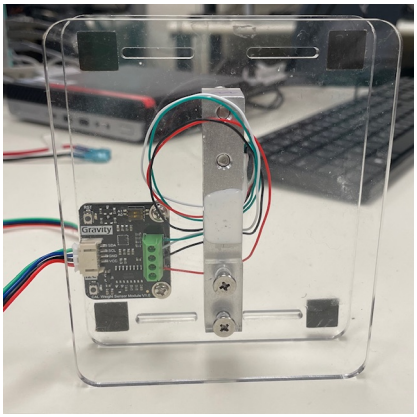


Figure 2.11 Exemple de convertisseur analogique-numérique utilisé pour l'interface d'un capteur de force basé sur une cellule de charge.

2.4.7 Modulation par largeur d'impulsion (MLI)

La modulation de largeur d'impulsion (MLI, pour Pulse Width Modulation), permet de représenter des informations numériques sous une forme temporelle ou quasi-analogique. Elle est largement utilisée pour piloter des actionneurs tels que des moteurs, des servomoteurs ou d'autres dispositifs dont la puissance peut être modulée. Ce procédé assure la conversion d'une grandeur numérique en un signal physique directement exploitable par les dispositifs de commande.

Dans le cadre de cette étude, une attention particulière sera portée à la modulation de largeur d'impulsion (MLI), en raison de son utilisation répandue dans les systèmes embarqués de commande d'actionneurs. Le principe de fonctionnement peut être défini comme suit. On génère un signal carré de période fixe ; la largeur de l'impulsion haute

est ajustée en fonction d'une valeur numérique (position, commande moteur, etc.). Ce signal est ensuite envoyé à un actionneur, comme un servomoteur, qui interprète cette largeur pour ajuster son point de fonctionnement. Dans ce projet, la modulation de largeur d'impulsion (MLI) est utilisée pour piloter les moteurs et les servomoteurs, en fonction des signaux provenant du récepteur et du générateur RC.

2.5 Conclusion

Ce chapitre a permis de présenter les bases théoriques nécessaires à la compréhension des systèmes de commande en agriculture de précision. Les différents éléments constitutifs, tels que les capteurs, les actionneurs, les systèmes d'acquisition et les unités de traitement, ont été analysés, ainsi que leur rôle au sein d'une architecture embarquée. Par ailleurs, une attention particulière a été portée aux protocoles de communication, notamment SPI, I^2C et UART, en mettant en évidence leurs caractéristiques, leurs avantages et leurs limites selon les applications.

Cette étude théorique constitue ainsi une base essentielle pour la conception de l'architecture de commande proposée. Le chapitre suivant sera consacré à la proposition et à la mise en œuvre de cette architecture de commande embarquée en FPGA.

Chapitre 3 - Proposition d'architecture de commande

3.1 Introduction

Dans le chapitre précédent, une étude théorique des systèmes électromécaniques en agriculture de précision a été présentée. Cette étude a permis de décrire les principaux éléments et composants impliqués tels que les processeurs, les capteurs, les actionneurs et les protocoles de communication.

L'objectif du présent chapitre est de mettre en œuvre ces notions théoriques en proposant une architecture de commande fonctionnelle et réaliste, adaptée aux contraintes des systèmes électromécaniques en milieu agricole. Cette architecture est conçue pour être implémentée et validée expérimentalement sur les équipements disponibles au laboratoire du Groupe de recherche en électronique industrielle de l'Université du Québec à Trois-Rivières (UQTR).

3.2 Cas d'applications

Afin de valider les concepts présentés au chapitre précédent, une plateforme expérimentale représentative des systèmes électromécaniques utilisés en agriculture de précision a été retenue. Cette plateforme est constituée d'un robot mobile à entraînement différentiel équipé de moteurs à courant continu, d'un servomoteur, d'un encodeur incrémental, d'une balance instrumentée et de différents modules d'acquisition de données.

L'architecture matérielle repose sur l'utilisation d'un FPGA assurant les fonctions de commande et de traitement en temps réel, ainsi que d'un Raspberry Pi servant d'unité de supervision et d'interface avec l'utilisateur. Les différents capteurs et actionneurs communiquent avec ces unités de traitement à travers plusieurs protocoles

de communication tels que SPI, I^2C, UART et RC.

Cette plateforme permet de reproduire plusieurs fonctions couramment rencontrées dans les équipements agricoles modernes, notamment le contrôle de la vitesse et de la position des actionneurs, l'acquisition de données issues de capteurs ainsi que la transmission d'informations entre les différents modules du système. Elle constitue ainsi un environnement pertinent pour la conception, l'implantation et la validation expérimentale d'une architecture de commande destinée aux systèmes électromécaniques en agriculture de précision.

3.3 Architecture globale de commande proposée

L'architecture de commande proposée repose sur une organisation modulaire qui sépare clairement les fonctions de mesure, de traitement et de commande. La figure 3.2 présente l'architecture générale du système proposé. Dans cette architecture, les capteurs mesurent les grandeurs physiques nécessaires à la commande du système électromécanique, telles que le courant, la force, la vitesse et la position. Ces informations sont transmises à une unité centrale de traitement, principalement composée d'un FPGA, qui constitue le cœur du système de commande.

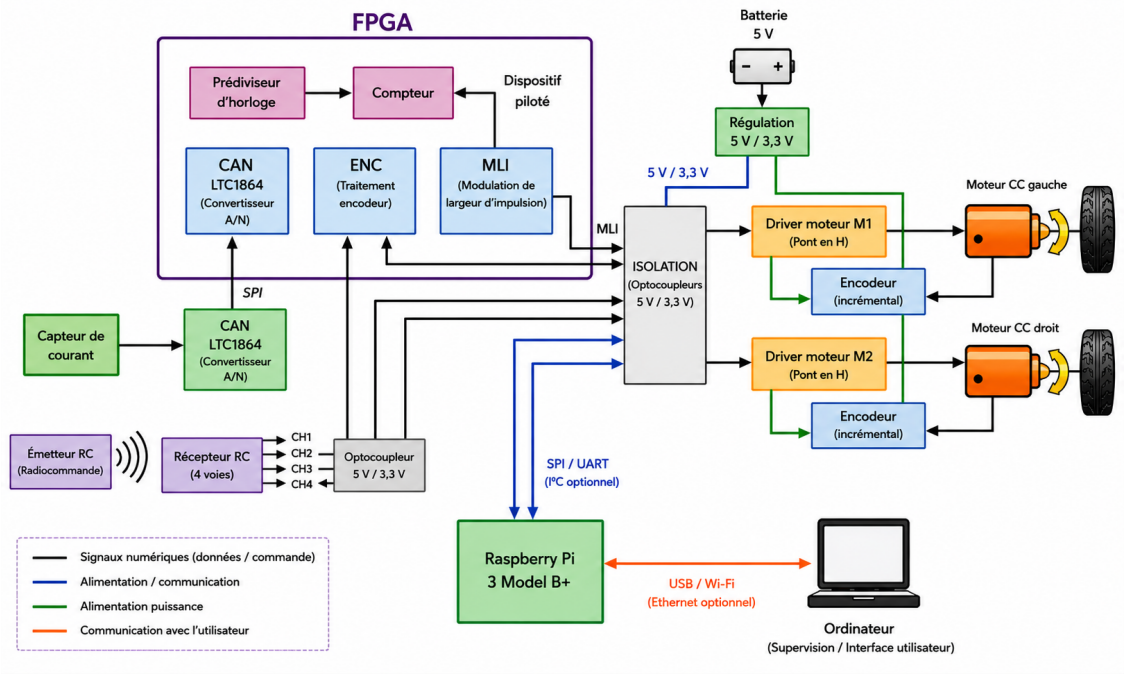


Figure 3.1 Architecture CPU+FPGA pour le système de commande embarqué.

Dans le système développé, le FPGA est associé à un micro-ordinateur Raspberry Pi chargé de la supervision, des échanges de données et de l'interface avec l'utilisateur. Les actionneurs, notamment les moteurs à courant continu et les servomoteurs, sont pilotés à partir des signaux générés par le FPGA afin de réaliser les mouvements mécaniques souhaités. L'alimentation électrique est assurée par une batterie reliée à un convertisseur de puissance, lequel fournit l'énergie nécessaire aux moteurs. Ceux-ci entraînent les roues du robot par l'intermédiaire de réducteurs de vitesse.

Cette organisation permet une répartition efficace des fonctions entre les différents composants, tout en garantissant la modularité, l'évolutivité et une intégration aisée dans un environnement expérimental. Il convient également de noter que, dans les applications de robots mobiles destinés à l'agriculture de précision, la vitesse de déplacement est généralement limitée à quelques mètres par seconde.

3.3.1 Éléments et principe de fonctionnement

Le FPGA constitue le cœur de l'architecture de commande. Il assure l'acquisition des informations provenant des différents capteurs, notamment celles issues du convertisseur analogique–numérique (CAN) utilisé pour la mesure du courant, ainsi que des signaux délivrés par les encodeurs (ENC) permettant d'estimer la vitesse de rotation des moteurs. Les données acquises sont ensuite traitées afin d'élaborer les signaux de commande nécessaires au pilotage des actionneurs.

À partir de ces informations, le FPGA génère les signaux de modulation de largeur d'impulsion (MLI) destinés à commander individuellement chaque moteur. Avant d'être appliqués à l'étage de puissance, ces signaux traversent un circuit d'isolation qui protège la logique numérique alimentée en 3,3 V des perturbations électriques provenant de la partie puissance fonctionnant sous des tensions supérieures à 12 V. Le convertisseur de puissance adapte ensuite ces commandes aux exigences des moteurs, lesquels entraînent le système mécanique.

Le fonctionnement du système est surveillé en continu grâce aux encodeurs, qui fournissent une mesure de la vitesse de rotation, ainsi qu'aux capteurs de courant, qui renseignent sur la consommation électrique de chaque moteur. Ces informations sont réinjectées dans le FPGA afin d'assurer le suivi des performances, la surveillance du fonctionnement et, le cas échéant, l'activation des mécanismes de protection du système.

Comme mentionné précédemment, le système intègre également un Raspberry Pi 3 Model B+, utilisé pour la supervision, le traitement complémentaire des données et la communication avec un ordinateur. Les échanges entre le Raspberry Pi et le FPGA sont réalisés via des protocoles de communication tels que SPI, I^2C et UART. Enfin, un module

de radiocommande (RC) permet de transmettre des consignes au système et offre une interface utilisateur supplémentaire pour piloter le robot manuellement.

Cette architecture permet ainsi une distribution efficace des tâches entre les différents éléments du système, tout en assurant une commande précise, robuste et adaptée aux contraintes des applications en agriculture de précision.

3.3.2 Justification du choix de l'architecture proposée

Dans ce travail, le choix de l'architecture repose sur une approche combinant un FPGA et un micro-ordinateur, afin de répondre aux exigences des systèmes électromécaniques en agriculture de précision, notamment en termes de temps réel, de flexibilité et de communication.

Le **FPGA** est utilisé comme élément central pour assurer les tâches critiques grâce à sa capacité de traitement parallèle et son comportement déterministe. Il permet d'acquérir, traiter les données et générer des signaux de commande précis tels que le MLI pour piloter les actionneurs. Le **micro-ordinateur**, quant à lui, est dédié aux fonctions de supervision, de communication et de traitement non critique, facilitant l'échange de données, l'enregistrement des mesures et l'exécution d'algorithmes de haut niveau.

L'architecture repose ainsi sur une répartition claire des tâches entre le FPGA (temps réel) et le micro-ordinateur (haut niveau), ce qui améliore les performances, la modularité et l'évolutivité du système. L'intégration des protocoles SPI, I^2C et UART permet d'assurer une communication flexible avec les différents composants. Cette architecture répond ainsi aux contraintes de robustesse, de précision et d'adaptabilité, et constitue une solution efficace pour les systèmes embarqués en agriculture de précision. L'analyse du flux de données dans le système permet de décrire le cheminement des informations depuis

les capteurs jusqu'aux actionneurs. Les données sont d'abord acquises à l'aide d'interfaces de communication telles que SPI, I^2C et UART, puis transmises au FPGA pour être traitées en temps réel. Le FPGA assure le traitement des signaux et leur conversion en commandes adaptées, notamment sous forme de signaux MLI destinés au pilotage des actionneurs. Ce flux de données garantit une transmission fiable, une bonne synchronisation entre les différents modules ainsi qu'un contrôle précis et réactif du système.

3.3.3 Contraintes et exigences du système

Les contraintes et exigences du système concernent principalement le fonctionnement en temps réel, la précision des mesures, la latence de communication et la robustesse face aux perturbations. Pour garantir un pilotage fiable des actionneurs, le système de commande doit respecter des valeurs cibles adaptées à la dynamique des équipements étudiés. Ainsi, l'acquisition de la vitesse par encodeur est réalisée avec une période d'échantillonnage de 10 ms, soit une fréquence de 100 Hz.

La génération des signaux MLI destinés aux moteurs et au servomoteur doit être suffisamment stable pour assurer une commande régulière, avec une fréquence typique de 50 Hz pour le servomoteur. La communication SPI utilisée pour l'acquisition rapide des données fonctionne à une fréquence de l'ordre de 4 MHz, tandis que la communication UART destinée à l'envoi de commandes fonctionne à 9600 bauds. Ces valeurs permettent de limiter la latence globale de la chaîne de commande et d'assurer une réponse compatible avec les exigences d'un système électromécanique embarqué.

Tableau 3-1 Valeurs cibles des contraintes du système de commande.

Paramètre	**Valeur cible**
Période d'échantillonnage de l'encodeur	10 ms
Fréquence d'échantillonnage de la vitesse	100 Hz
Fréquence MLI du servomoteur	50 Hz
Fréquence SPI	4 MHz
Débit UART	9600 bauds

3.4 Implémentation en FPGA

Comme mentionné précédemment, le FPGA joue un rôle central dans l'architecture de commande proposée et son choix est motivé par sa capacité à assurer un fonctionnement en temps réel, un traitement parallèle des données et une grande flexibilité de conception. Dans cette section, les différents modules implémentés sur le FPGA sont détaillés ainsi que leur fonctionnement.

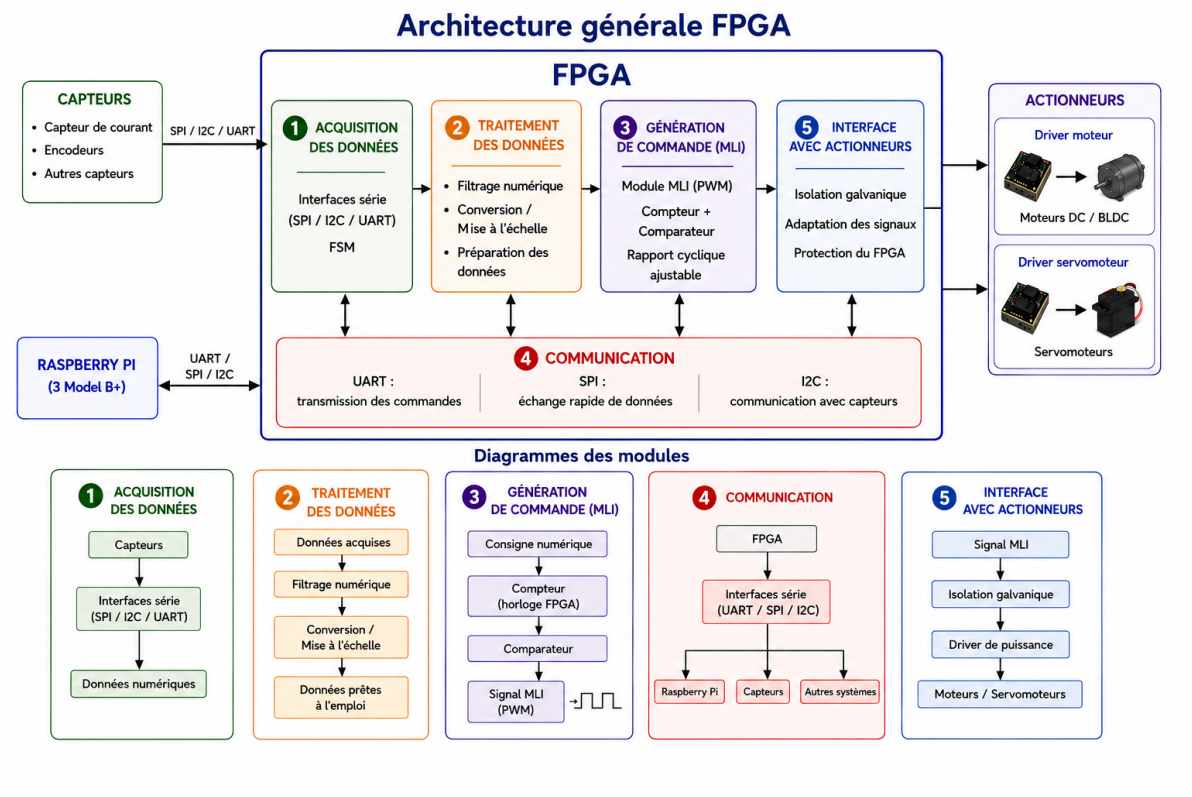


Figure 3.2 Diagramme bloc et modules.

Le **module d'acquisition des données** permet de récupérer les données provenant des capteurs de courant et des encodeurs. Ce module est implémenté en FPGA à l'aide d'interfaces de communication série telles que SPI, I^2C et UART. Pour le capteur de courant, un module SPI est utilisé pour lire les données du convertisseur analogique-numérique (LTC1864). Ce dernier offre une résolution de 16 bits et un taux de conversion de 250 ksps, soit 250 000 échantillons par seconde. Pour les autres capteurs, des interfaces adaptées (I^2C ou UART) sont implémentées sous forme de machines à états finis (FSM). Le **module de traitement des données** assure la mise en forme et le filtrage des données acquises. Il est implémenté sous forme de blocs logiques permettant les fonctions suivantes:

- Le filtrage numérique (réduction du bruit);
- La conversion et la mise à l'échelle des données;
- La préparation des données pour leur exploitation.

Grâce à l'architecture parallèle du FPGA, ces opérations sont réalisées en pipeline pour une meilleure performance en temps réel. Le **module de génération de commande** produit les signaux de commande des actionneurs. Cette fonction est réalisée à l'aide d'un module (MLI) implémenté en VHDL. Les éléments fondamentaux du module sont un compteur cadencé par l'horloge du FPGA et une fonction de comparaison entre la valeur du compteur et la consigne reçue. Le rapport cyclique du signal MLI est ajusté en fonction de la commande numérique provenant soit d'un contrôleur embarqué dans le processeur numérique (FPGA ou Raspberry Pi), soit d'une information reçue via le récepteur de radiofréquence (RC), ce qui permet de piloter les moteurs et servomoteurs avec précision.

Le **module de communication** assure le transfert de données en temps réel entre les processeurs numériques. De cette manière, le FPGA communique avec le Raspberry Pi via les protocoles UART, SPI et I^2C. Le protocole UART est utilisé pour la transmission des commandes, le protocole SPI permet un échange rapide de données, et le protocole I^2C est utilisé pour certains capteurs. Ces fonctions sont également implémentées sous la forme de machines à états finis, ce qui garantit une communication fiable.

Le **module d'interface avec les actionneurs** permet de relier le FPGA aux convertisseurs de puissance qui assurent le contrôle des moteurs et servomoteurs. Les signaux MLI générés sont transmis aux drivers via des circuits d'isolation afin de protéger la partie logique du FPGA. Ce module garantit une commande sécurisée et adaptée aux contraintes du système électromécanique. En conclusion, l'implémentation

de ces différents modules sur le FPGA permet d'assurer une architecture de commande robuste, flexible et en temps réel. Le comportement déterministe du FPGA garantit une synchronisation précise des opérations, essentielle pour les applications en agriculture de précision.

3.5 Architecture interne de mesure et de commande basée sur FPGA

La figure 3.3 illustre l'architecture interne du système de mesure et de commande basée sur le FPGA. Cette architecture est organisée en plusieurs blocs fonctionnels assurant le traitement des données, de leur acquisition jusqu'à la commande des actionneurs.

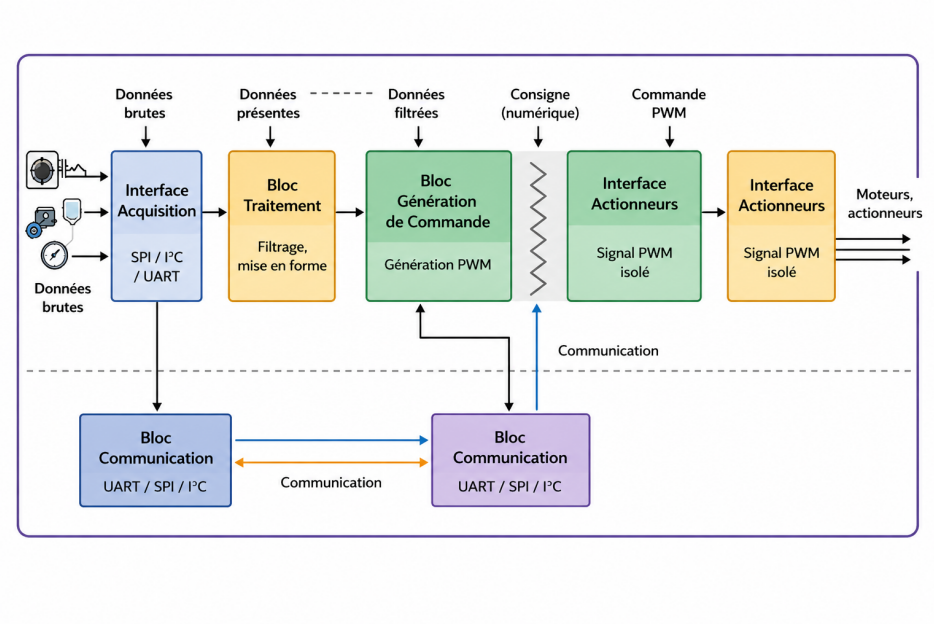


Figure 3.3 Architecture interne de mesure et de commande basée sur FPGA.

Dans un premier temps, les données brutes issues des capteurs sont acquises via une interface d'acquisition utilisant des protocoles de communication série tels que SPI, I^2C et UART. Ces données peuvent provenir de différents capteurs mesurant des grandeurs physiques telles que le courant, la position, la vitesse ou la force. Les données acquises

sont ensuite transmises au bloc de traitement, où elles subissent diverses opérations de filtrage et de mise en forme afin d'éliminer le bruit et de les rendre exploitables. Ce traitement permet d'obtenir des données fiables, appelées données filtrées, prêtes à être utilisées dans le système de commande.

Par la suite, ces données sont utilisées par le bloc de génération de commande, qui élabore une consigne numérique adaptée au fonctionnement du système. Ce bloc génère notamment un signal de modulation de largeur d'impulsion (MLI/), dont le rapport cyclique dépend des informations traitées. Le signal MLI généré est ensuite envoyé au bloc d'interface des actionneurs, où il est adapté et isolé afin de protéger la partie logique du FPGA. Cette étape est essentielle pour assurer la compatibilité entre les niveaux de tension et garantir la sécurité du système.

Enfin, le signal MLI est appliqué aux actionneurs, tels que les moteurs ou les servomoteurs, ce qui permet d'effectuer les actions mécaniques souhaitées. Par ailleurs, un bloc de communication assure l'échange de données entre le FPGA et les autres éléments du système, notamment le Raspberry Pi. Cette communication, réalisée via les protocoles UART, SPI et I^2C, permet la transmission des consignes et la supervision du système en temps réel.

3.6 Intégration de l'architecture du banc de test du laboratoire

L'architecture de commande proposée a été conçue en tenant compte des équipements disponibles au laboratoire de l'UQTR. Les essais expérimentaux sont réalisés sur le banc de test comprenant un mini-robot, des moteurs électriques, des servomoteurs ainsi que les différents capteurs utilisés pour mesurer les grandeurs physiques. La figure 3.2 présente l'architecture générale de commande, tandis que la figure 3.4 illustre le banc de

tests développé et utilisé pour les validations dans le cadre du projet. Cette intégration permet de valider le fonctionnement global de l'architecture de commande, tant au niveau matériel que logiciel dans des conditions proches de celles rencontrées en agriculture de précision. Cette étape constitue une transition vers le chapitre suivant, consacré à la validation expérimentale de l'architecture proposée. Cette plateforme regroupe l'ensemble des composants nécessaires à l'acquisition, au traitement et à la commande du système électromécanique. Au cœur du dispositif se trouve la carte FPGA, qui assure le traitement des données et la génération des signaux de commande, notamment les signaux de modulation de largeur d'impulsion (MLI) utilisés pour piloter les moteurs. Elle est associée à un Raspberry Pi, qui joue un rôle de supervision, d'interface utilisateur et de communication avec un ordinateur externe.

Le système intègre également un convertisseur analogique-numérique LTC1864, utilisé pour mesurer le courant à l'aide d'un capteur à effet Hall dédié. Les données acquises sont transmises au FPGA pour traitement. Deux encodeurs sont également présents pour mesurer la vitesse du moteur.

Le pilotage des moteurs est réalisé à l'aide convertisseur statique recevant les signaux MLI issus du FPGA. Les consignes sont envoyées par un système de radiocommande (RC) composé d' 'un émetteur et d'un récepteur permettant le contrôle à distance. L'ensemble est alimenté par une batterie couplée à un convertisseur DC/DC assurant la distribution des tensions adaptées aux différents sous-systèmes.

Par ailleurs, un module de radiocommande (RC), composé d'un émetteur et d'un récepteur, permet d'envoyer des consignes au système. Un optocoupleur est utilisé pour assurer l'isolation électrique entre les différents niveaux de tension et protéger les circuits sensibles.

Ce montage expérimental permet ainsi de valider l'architecture proposée dans des conditions proches de celles d'une application réelle, en intégrant les fonctions de mesure, de traitement, de communication et de commande.

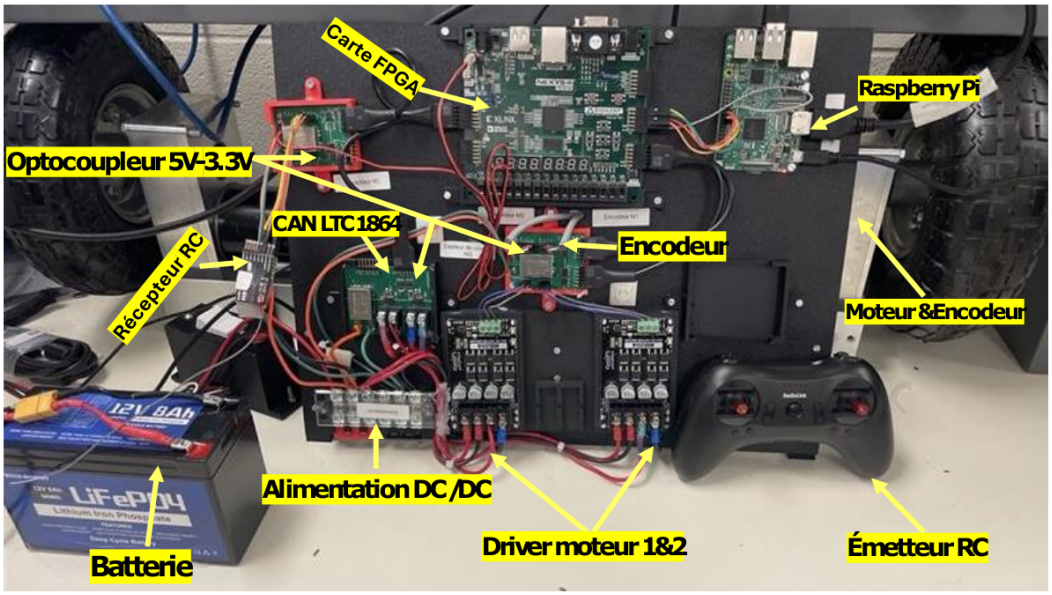


Figure 3.4 Banc de tests développé pour la validation expérimentale au laboratoire du groupe de recherche en électronique industrielle.

3.7 Conclusion

Ce chapitre a présenté la proposition d'une architecture de commande pour les systèmes électromécaniques en agriculture de précision. L'architecture développée repose sur l'association d'un FPGA et d'un micro-ordinateur, permettant une acquisition fiable des données des capteurs, un traitement efficace des informations et une commande précise des actionneurs. Comparativement aux architectures reposant uniquement sur un microcontrôleur ou un micro-ordinateur, l'association d'un FPGA et d'un Raspberry Pi permet de tirer parti des avantages de chaque technologie. Le FPGA assure l'acquisition et le traitement des données en temps réel grâce à son architecture parallèle et à son

comportement déterministe, ce qui réduit les latences et améliore la réactivité du système.

Le Raspberry Pi fournit quant à lui des capacités de supervision, de communication et de traitement de plus haut niveau. Cette répartition des tâches permet d'améliorer les performances globales de l'architecture, notamment en termes de rapidité d'exécution, de précision de commande, de flexibilité et de fiabilité pour les applications de l'agriculture de précision. Conçue pour être implémentée sur les équipements disponibles au laboratoire de l'UQTR, cette architecture constitue une base solide pour la validation expérimentale présentée dans le chapitre suivant.

Chapitre 4 - Validation expérimentale

Ce chapitre est dédié à la description de la démarche de validation et à la présentation des résultats expérimentaux obtenus avec l'architecture de mesure et de commande proposée au Chapitre 3. La validation a été réalisée sur le banc de tests décrit précédemment.

4.1 Organisation des tests expérimentaux

Afin d'assurer une cohérence avec les modules décrits au Chapitre 3, les tests expérimentaux sont réalisés suivant une démarche progressive, allant de l'acquisition des données jusqu'à la commande des actionneurs. L'ordre de présentation des résultats est structuré comme suit :

- **Test du convertisseur analogique-numérique (CAN) en I^2C** : Ce test permet de valider le module d'acquisition des données en vérifiant la lecture des valeurs issues d'un capteur de force via le protocole I^2C.
- **Test du servomoteur avec générateur radio-commande (RC)** : Ce test valide le module de génération de commande, en analysant le comportement d'un servomoteur en fonction du signal MLI généré par le FPGA compatible avec une radiocommande.
- **Tests du récepteur radio-commande (RC)** : Ce test permet de vérifier le bon fonctionnement du module de communication RC, ainsi que la transmission des commandes vers le FPGA.
- **Tests expérimentaux sur le capteur de courant** : Ce test permet de valider l'acquisition et le traitement des données d'un capteur de courant, ainsi que leur cohérence avec le fonctionnement réel du moteur.
- **Simulation et expérimentation avec différentes commandes MLI** : Cette étape

consiste à analyser la réponse du système en fonction de différentes valeurs de rapport cyclique, afin de valider la précision du module MLI.

- **Expérimentation sur l'encodeur** : Cette étape valide le module de mesure de vitesse et de position, en analysant les signaux issus de l'encodeur et en évaluant leur exploitation par le FPGA.

4.1.1 Test sur capteur avec convertisseur analogique-numérique en I^2C

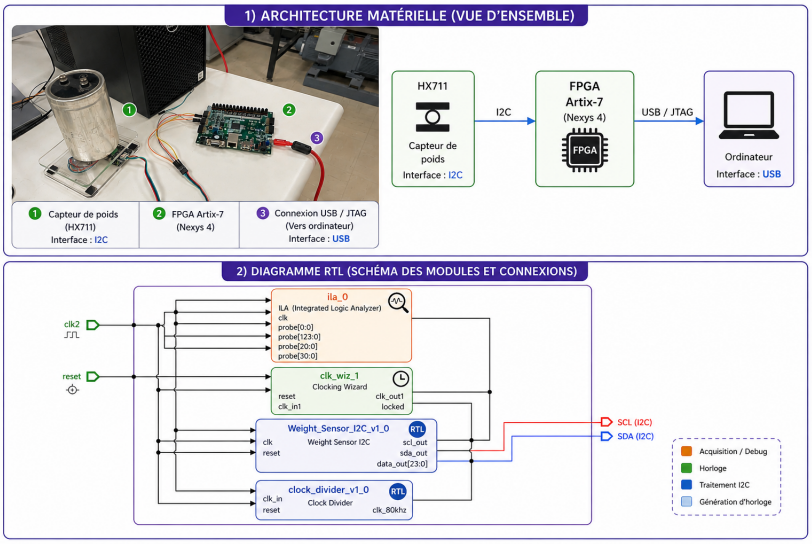


Figure 4.1 Plateforme expérimentale de capteur de force avec convertisseur analogique-numérique HX711 utilisant le protocole I^2C.

La figure 4.1 présente le montage expérimental du capteur de poids HX711 connecté à la carte FPGA Nexys 4, ainsi que le schéma RTL correspondant. Cette architecture permet l'acquisition et le traitement en temps réel des données de pesée, puis leur visualisation sur ordinateur.

Cette carte, programmée pour assurer la gestion de la communication et le traitement des données, est reliée à un ordinateur via une interface USB afin de permettre la configuration du système et la récupération des données. Les câbles visibles permettent

l'interconnexion entre le module de mesure et l'interface FPGA (alimentation et signaux I^2C). Ce montage assure une acquisition fiable des valeurs de force et permet d'envisager l'intégration future dans des applications robotiques en environnement industriel. La figure 4.2 présente les résultats des signaux SCL et SDA obtenus et enregistrés à l'aide d'un oscilloscope.

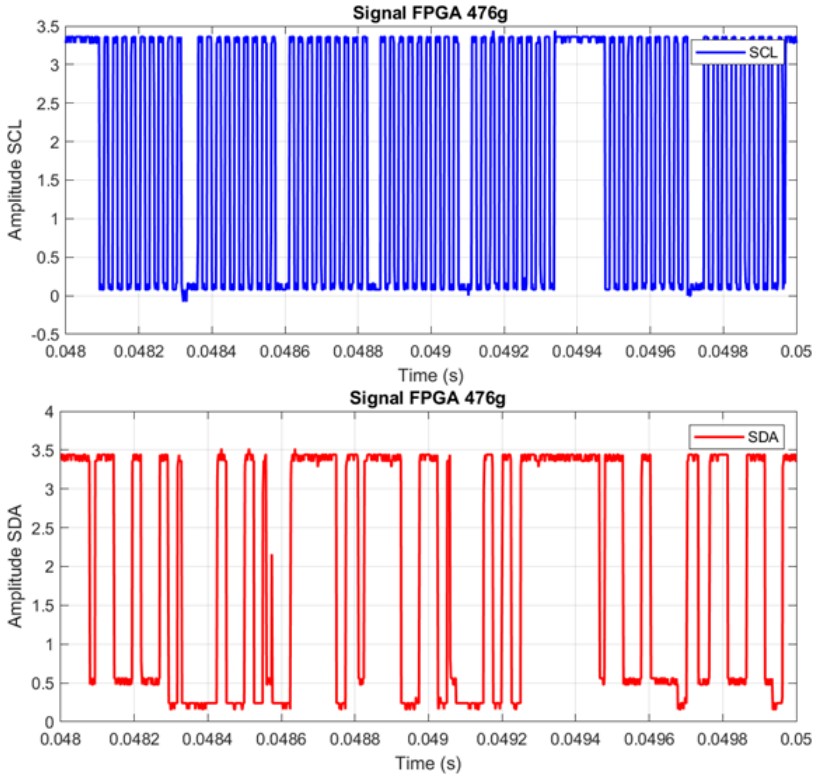


Figure 4.2 Résultats obtenus des lignes SCL et SDA du module HX711 avec l'implémentation sur FPGA.

Une étape clé dans l'exploitation des données du module HX711 sur FPGA est la conversion des 24 bits reçus en information de mesure réelle, exprimée en unités techniques (p.ex: en grammes).

Le capteur transmet une valeur brute sous la forme de trois octets, qui nécessite une calibration pour être correctement interprétée. Le facteur de calibration est déterminé

expérimentalement à partir de la valeur brute mesurée à vide (tare) et d'une masse de référence connue. Cette approche, couramment utilisée pour l'étalonnage des capteurs HX711, permet d'établir une relation linéaire entre la valeur numérique délivrée par le convertisseur et la masse appliquée. Pour déterminer le facteur de conversion, plusieurs mesures ont été effectuées avec des masses connues, permettant d'associer des valeurs hexadécimales aux poids réels, comme : 127301 pour 161 g, 127070 pour 86 g, et 126D8A pour 0 g.

$$\text{Calibration} = \frac{\text{Valeur brute} - \text{Tare}}{\text{Poids réel (g)}} \tag{4.1}$$

$$\text{Calibration} = \frac{0x127301 - 0x126D8A}{161} \approx 8.76 \tag{4.2}$$

Ce facteur a ensuite été intégré dans le code FPGA afin de permettre la conversion automatique des mesures en poids réels.

$$\text{Poids (g)} = \frac{(\text{Valeur brute} - \text{Tare}) \times 1000}{876} \tag{4.3}$$

4.1.2 Validation de la commande RC du servomoteur avec les protocoles UART et SPI

Dans ce projet, un servomoteur est utilisé comme actionneur électromécanique afin d'assurer un contrôle précis de la position angulaire du système. Il intervient principalement dans les fonctions de direction et de positionnement, notamment lors des manœuvres de virage ou du positionnement de mécanismes robotisés. Les commandes issues du récepteur de radiocommande (RC) sont transmises au FPGA à l'aide des protocoles UART et SPI, qui ont tous deux été implémentés et validés expérimentalement.

Dans les deux configurations, le FPGA exploite les informations reçues pour générer les signaux MLI nécessaires au pilotage du servomoteur. L'utilisation du FPGA permet

d'obtenir une commande rapide, stable et reproductible tout en améliorant la précision et la fiabilité du système. Les essais présentés dans cette section visent à valider le fonctionnement de la chaîne de commande RC avec les protocoles UART et SPI. La chaîne de commande RC du servomoteur utilisant le protocole UART est présentée à la figure 4.3.

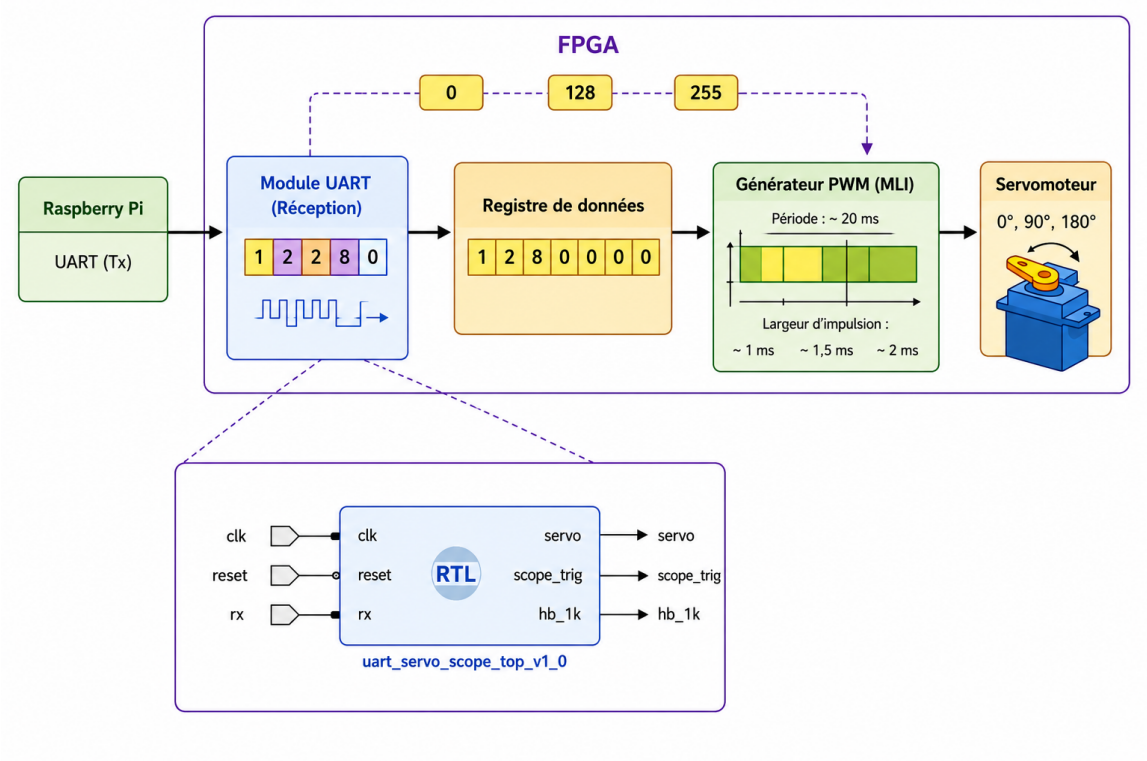


Figure 4.3 Chaîne de commande RC du servomoteur via le protocole UART.

La figure 4.3 illustre la chaîne de commande du servomoteur utilisant le protocole UART entre le Raspberry Pi et le FPGA. Dans cette architecture, le Raspberry Pi agit comme unité de supervision et transmet des consignes numériques au FPGA via la liaison UART. Les valeurs de commande utilisées pour les essais sont 0, 128 et 255, correspondant respectivement aux positions minimale, intermédiaire et maximale du servomoteur. À la réception des données, le module UART implanté dans le FPGA décode les informations reçues et les transfère vers un registre de données. Ce registre assure le

stockage temporaire de la consigne et sa mise à disposition pour les traitements ultérieurs.

La valeur numérique reçue est ensuite exploitée par le générateur MLI afin de produire un signal dont la largeur d'impulsion est proportionnelle à la consigne. Pour le servomoteur utilisé, la période du signal MLI est de ($\approx$ 20ms) (50 Hz), tandis que la largeur d'impulsion varie généralement entre ($\approx$ 1ms) et ($\approx$ 2ms). Une impulsion d'environ ($\approx$ 1ms) correspond à la position minimale du servomoteur, une impulsion de ($\approx$ 1,5 ms) à la position centrale et une impulsion proche de ($\approx$ 2ms) à la position maximale. Le signal MLI généré est ensuite appliqué directement au servomoteur, qui convertit cette information électrique en mouvement mécanique. Cette chaîne de traitement permet ainsi de transformer une consigne numérique transmise par le Raspberry Pi en une position angulaire précise du servomoteur.

La partie inférieure de la figure présente le module VHDL `uart_servo_scope_top_v1_0` implanté dans le FPGA. Ce module reçoit les signaux d'entrée `clk`, `reset` et `rx`, puis génère les sorties `servo`, `scope_trig` et `hb_1k`. La sortie `servo` correspond au signal MLI destiné au servomoteur, tandis que les signaux `scope_trig` et `hb_1k` sont utilisés pour la validation expérimentale et l'observation du fonctionnement du système à l'oscilloscope.

Ainsi, cette entité assure à la fois la réception de la commande UART, la génération du signal MLI de commande du servomoteur, ainsi que des signaux auxiliaires utiles pour la validation expérimentale et l'observation temporelle du système. Comme l'illustre la figure 4.4, nous avons réalisé des simulations préliminaires dans l'environnement Vivado avant de les tester sur le matériel.

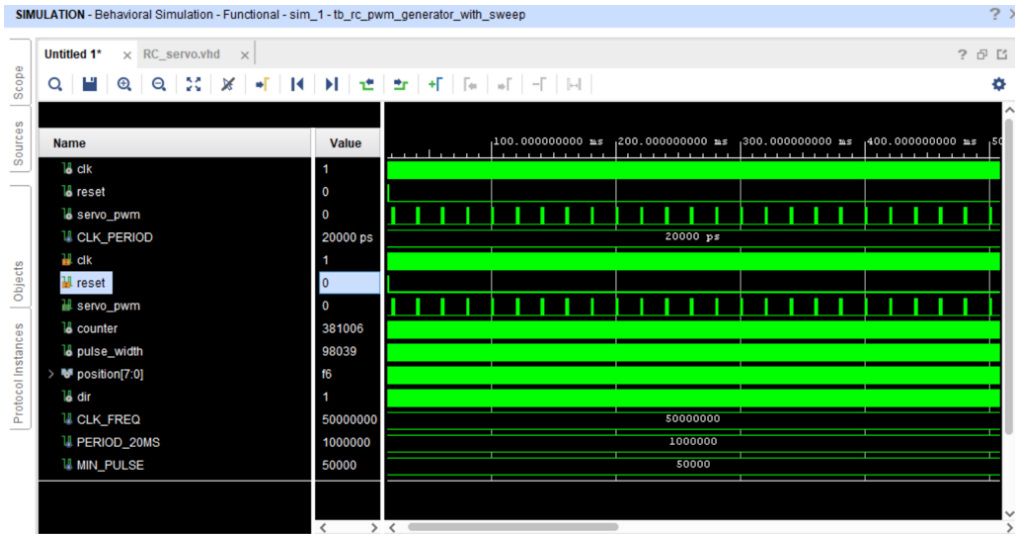


Figure 4.4 MLI du servomoteur simulation protocole UART.

La figure 4.5 présente le signal de modulation de largeur (MLI) généré par le module RC du FPGA, en réponse à la commande numérique transmise par le Raspberry Pi via le protocole UART. Dans l'exemple de la figure, on observe la réponse pour deux valeurs de commande différentes, soit 0 et 128, afin de produire des positions différentes de l'actionneur.

Lorsque la commande 128 est envoyée depuis le Raspberry Pi via le protocole UART. Le FPGA génère alors un signal MLI dont la largeur d'impulsion est d'environ ($\approx$ 1,5 ms), correspondant à la position médiane du servomoteur (90°). La commande 0 génère une impulsion courte ($\approx$ 1 ms), correspondant à la position minimale, soit 0°. Cette configuration garantit une coordination temporelle précise entre les deux dispositifs et une réception fiable des données de commande. Ce résultat confirme le bon fonctionnement de la chaîne de communication utilisant le protocole UART entre le Raspberry Pi et le FPGA, ainsi que du module de génération du signal MLI, puisque la variation du rapport cyclique correspond fidèlement à la commande reçue.

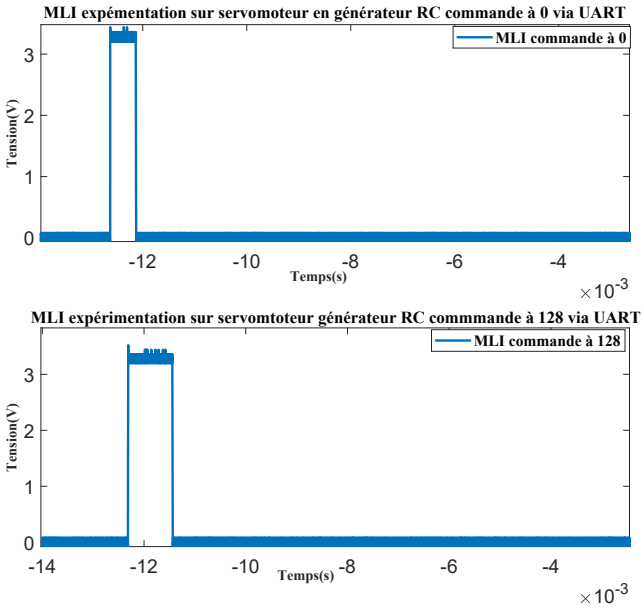


Figure 4.5 MLI produit par le FPGA pour des commandes à 0 et à 128 générées par le Raspberry Pi et transmises via le protocole UART.

Analyse des erreurs absolues des essais UART

De la même manière que pour les essais réalisés avec le protocole SPI, l'erreur absolue entre les largeurs d'impulsion théoriques et les valeurs mesurées expérimentalement est calculée à l'aide de l'équation (4.10). Les applications numériques correspondant aux trois commandes UART sont présentées ci-dessous.

Pour la commande 0 :

$$E_{\text{abs}} = |1,00000 - 0,99996| = 0,00004 \text{ ms} = 40 \text{ ns} \tag{4.4}$$

Pour la commande 128 :

$$E_{\text{abs}} = |1,50000 - 1,50192| = 0,00192 \text{ ms} = 1\,920 \text{ ns} \tag{4.5}$$

Pour la commande 255 :

$$E_{\text{abs}} = |2,00000 - 1,98820| = 0,01180 \text{ ms} = 11\,800 \text{ ns} \tag{4.6}$$

Tableau 4-1 Comparaison entre les valeurs théoriques et expérimentales des impulsions du servomoteur (UART).

Commande UART	**Valeur théorique (ms)**	**Valeur mesurée (ms)**	**Erreur absolue (ns)**
0	1,00000	0,99996	40
128	1,50000	1,50192	1 920
255	2,00000	1,98820	11 800

Le tableau 4-1 présente la comparaison entre les largeurs d'impulsion théoriques et les valeurs mesurées obtenues lors des essais de commande du servomoteur via le protocole UART. Les résultats présentés dans le tableau 4-1 montrent une bonne concordance entre les largeurs d'impulsion théoriques et les valeurs mesurées expérimentalement.

Les erreurs absolues obtenues sont respectivement de 40 ns, 1 920 ns et 11 800 ns pour les commandes 0, 128 et 255. Ces écarts demeurent faibles par rapport aux largeurs d'impulsion nominales de 1 ms, 1,5 ms et 2 ms, ce qui confirme la précision de la génération des signaux MLI par le FPGA. Les différences observées sont principalement attribuables à la résolution temporelle du système d'acquisition ainsi

qu'aux incertitudes de mesure. Ces résultats valident ainsi le bon fonctionnement de la chaîne de communication UART et du module `servo_pwm` implémentés dans le FPGA.

4.1.3 Test sur servomoteur via protocole SPI

La figure 4.6 montre la chaîne de commande RC d'un servomoteur via le protocole SPI. Le module `spi_servo_controller` assure la commande du servomoteur via le protocole SPI. Cette entité VHDL est définie avec des ports d'entrée et de sortie de type `std_logic`, permettant la manipulation de signaux numériques binaires au sein du FPGA.

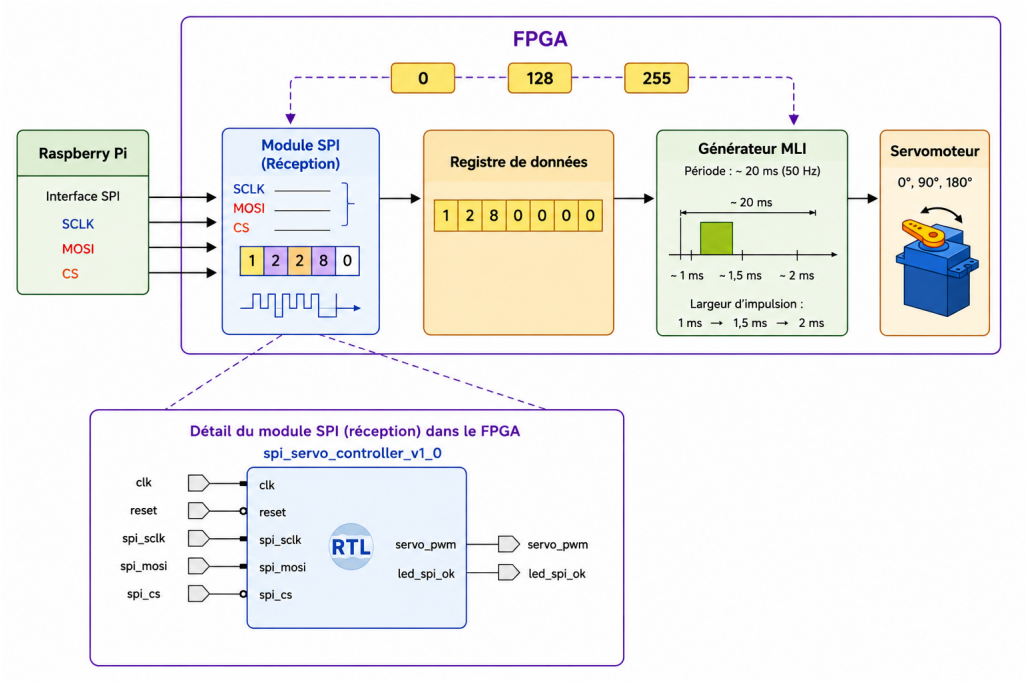


Figure 4.6 Chaîne de commande RC d'un servomoteur via SPI.

La figure 4.6 présente la chaîne de commande du servomoteur utilisant le protocole SPI entre le Raspberry Pi et le FPGA. Dans cette architecture, le Raspberry Pi agit comme maître SPI et transmet des consignes numériques au FPGA à travers les signaux `SCLK`, `MOSI` et `CS`. Les valeurs de commande utilisées lors des essais sont 0, 128 et 255, correspondant respectivement aux positions minimale, intermédiaire et maximale du servomoteur.

Les données reçues sont décodées par le module SPI implanté dans le FPGA puis stockées dans un registre de données. Cette étape permet de conserver la valeur de commande reçue et de la rendre disponible pour le traitement ultérieur. La consigne numérique est ensuite transmise au générateur MLI, qui produit un signal de commande adapté au servomoteur.

Le générateur MLI fonctionne avec une période de ($\approx$ 20 ms) (50 Hz), conformément aux spécifications du servomoteur utilisé. La largeur d'impulsion varie généralement entre ($\approx$ 1 ms) et ($\approx$ 2 ms) selon la valeur de la consigne reçue. Une impulsion d'environ ($\approx$ 1 ms) correspond à la position minimale, une impulsion de ($\approx$ 1,5 ms) à la position centrale et une impulsion proche de ($\approx$ 2 ms) à la position maximale du servomoteur. Cette conversion permet de transformer une commande numérique reçue via SPI en une position angulaire précise de l'actionneur.

La partie inférieure de la figure présente le module VHDL `spi_servo_controller_v1_0` implanté dans le FPGA. Ce module reçoit les signaux `clk`, `reset`, `spi_sclk`, `spi_mosi` et `spi_cs`. Après réception et décodage des données SPI, il génère la sortie `servo_pwm`, correspondant au signal MLI appliqué au servomoteur. La sortie `led_spi_ok` est utilisée comme indicateur de bon fonctionnement de la communication SPI durant les essais expérimentaux. La figure 4.7 présente les résultats

des signaux RC générés par le FPGA lorsque le Raspberry Pi transmet des signaux de commande à 0, à 128 et à 255.

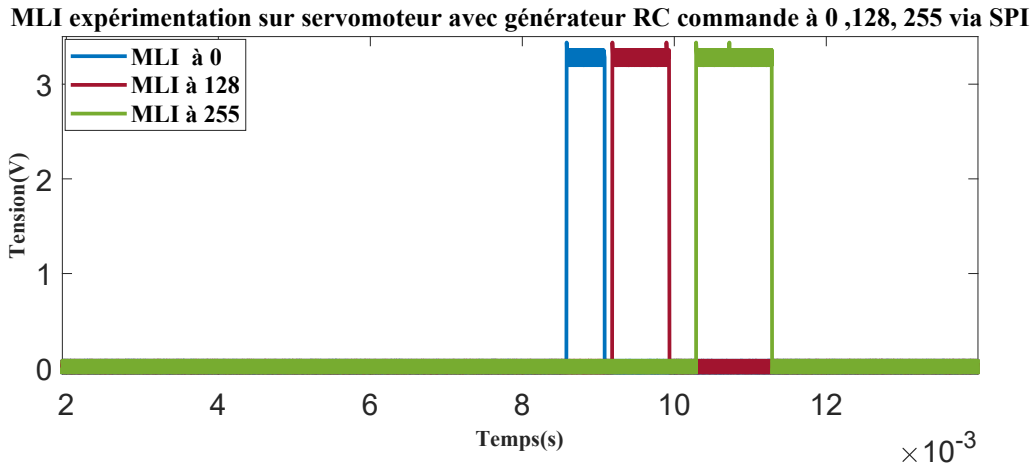


Figure 4.7 MLI produit par le FPGA pour des commandes à 0, à 128 et à 255 générées par le Raspberry Pi et transmises via le protocole SPI.

Les résultats obtenus mettent clairement en évidence le bon fonctionnement du lien de communication SPI pour l'envoi de commandes aux servomoteurs, en passant par le FPGA: chaque consigne transmise par le Raspberry Pi entraîne une position stable et reproductible du servomoteur, démontrant la fiabilité et la précision du générateur RC implémenté sur le FPGA.

On observe clairement que la largeur d'impulsion augmente proportionnellement à la valeur numérique de la commande:

- La commande **0** (en bleu) produit une impulsion très étroite ($\approx$ 1ms) correspondant à la position minimale du servomoteur (0°)
- La commande **128** (rouge) génère une impulsion de largeur moyenne ($\approx$ 1,5 ms) positionnant le servomoteur à **mi-course** (90°)
- Enfin, la commande **255** (en vert) entraîne une impulsion large ($\approx$ 2ms),

correspondant à la position maximale du servomoteur (180°)

L'absence de distorsion ou d'instabilité de la forme d'onde confirme la stabilité du signal MLI et la synchronisation fiable entre les deux dispositifs. Ce résultat valide ainsi la chaîne complète de commande SPI-FPGA-servomoteur, démontrant la capacité du système à générer un signal de position précis et stable à partir d'une consigne numérique issue du Raspberry Pi. De la même manière, l'application numérique de l'équation (4.10) pour les essais réalisés avec le protocole UART est donnée ci-dessous. L'application numérique de l'équation (4.10) pour les trois commandes SPI est donnée ci-dessous.

Pour la commande 0 :

$$E_{\text{abs}} = |1,00000 - 0,99992| = 0,00008 \text{ ms} = 80 \text{ ns} \tag{4.7}$$

Pour la commande 128 :

$$E_{\text{abs}} = |1,50000 - 1,50192| = 0,00192 \text{ ms} = 1\,920 \text{ ns} \tag{4.8}$$

Pour la commande 255 :

$$E_{\text{abs}} = |2,00000 - 1,99996| = 0,00004 \text{ ms} = 40 \text{ ns} \tag{4.9}$$

Tableau 4-2 Comparaison entre les valeurs théoriques et expérimentales des impulsions du servomoteur (SPI).

Commande SPI	**Valeur théorique (ms)**	**Valeur mesurée (ms)**	**Erreur absolue (ns)**
0	1,00000	0,99992	80
128	1,50000	1,50192	1 920
255	2,00000	1,99996	40

4.1.4 Analyse et comparaison globale des protocoles SPI et UART

Les essais expérimentaux réalisés sur la commande du servomoteur montrent que les protocoles SPI et UART permettent tous deux de générer correctement les signaux MLI nécessaires au positionnement du servomoteur. Les mesures expérimentales mettent en évidence une bonne concordance entre les valeurs théoriques et les valeurs mesurées pour les deux protocoles.

Les erreurs absolues calculées montrent que le protocole SPI présente les écarts les plus faibles, avec des erreurs comprises entre 40 ns et 1 920 ns, tandis que les essais réalisés avec le protocole UART conduisent à des erreurs comprises entre 40 ns et 11 800 ns. Ces résultats montrent que les deux protocoles assurent une commande satisfaisante du servomoteur, le protocole SPI offrant toutefois une précision légèrement supérieure dans les conditions expérimentales étudiées.

Sur le plan matériel, le protocole UART présente une architecture de communication plus simple puisqu'il utilise uniquement la ligne de réception **RX** pour transmettre les commandes vers le FPGA, sans nécessiter de signal d'horloge partagé. À l'inverse, le protocole SPI repose sur une communication synchrone nécessitant plusieurs lignes de signaux (**SCLK**, **MOSI**, **MISO** et **CS**), ce qui augmente la complexité du câblage mais

permet des échanges de données plus rapides.

Ainsi, le choix entre les protocoles SPI et UART dépend principalement des exigences de l'application. Le protocole SPI est particulièrement adapté aux applications nécessitant un débit élevé et une grande précision temporelle, tandis que le protocole UART constitue une solution simple et efficace lorsque les contraintes de vitesse sont moins importantes. Comme la commande du servomoteur ne requiert pas de vitesse de commande élevée, le **UART** assure une réponse **régulière et constante** parfaitement adaptée au **contrôle MLI**. Les broches du Raspberry Pi ont été sélectionnées en se basant sur les informations fournies dans sa fiche technique (`https://datasheets.raspberrypi.com/`).

4.1.5 Test sur capteur de courant avec convertisseur analogique-numérique SPI

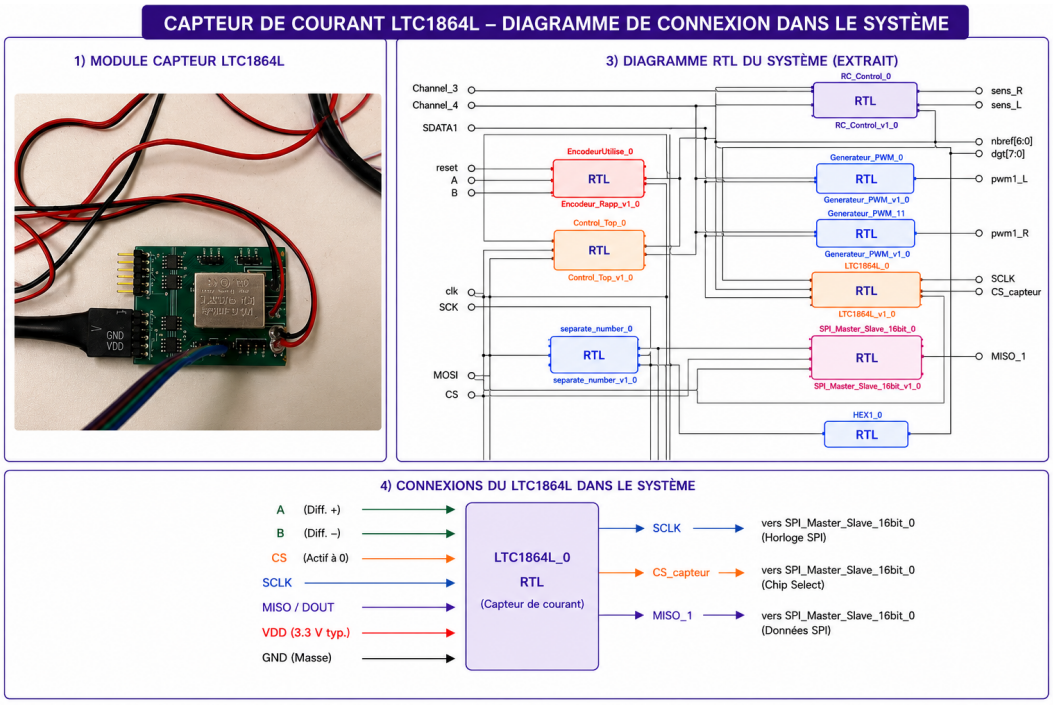


Figure 4.8 Architecture matérielle et schéma de connexion du capteur de courant avec convertisseur analogique-numérique SPI.

La figure 4.8 présente l'intégration du capteur de courant LTC1864L dans le système

embarqué basé sur FPGA. Elle met en évidence le brochage du module, les principales connexions SPI utilisées pour l'acquisition des données ainsi que son interaction avec les blocs RTL internes. Cette architecture permet la mesure la tension de batterie et du courant en temps réel afin d'assurer la supervision et le contrôle des actionneurs du système.

Nous présentons dans cette section les essais expérimentaux réalisés sur le capteur de courant interfacé au convertisseur analogique-numérique LTC1864 intégré au banc de test. L'objectif est de valider son fonctionnement en conditions réelles, d'évaluer la qualité des mesures obtenues et de vérifier son intégration par la communication SPI. Nous évaluons simultanément la réception des signaux de commande RC. La figure 4.9 montre une vue réelle des cartes électroniques permettant l'intégration du récepteur RC et du capteur de courant via le convertisseur analogique-numérique LTC1864.

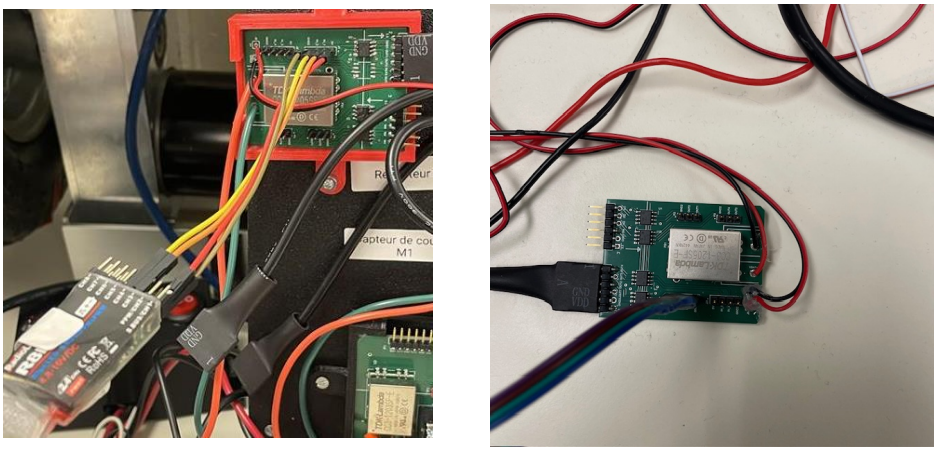


Figure 4.9 Vue réelle du récepteur RC et du capteur de courant avec le convertisseur analogique-numérique LTC1864.

La figure 4.10 présente l'évolution du courant mesuré par le capteur et transmis par le convertisseur analogique-numérique LTC1864 pendant le fonctionnement du moteur, lorsque le moteur est piloté à l'aide de la manette RC.

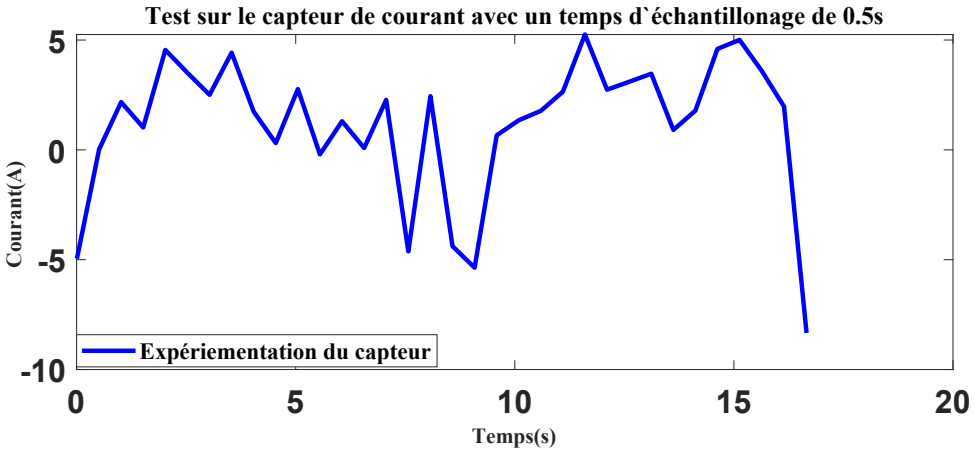


Figure 4.10 Courbe de l'évolution du courant mesuré lorsque la commande est générée à l'aide de la manette RC.

Dans cette configuration, chaque composant de la chaîne joue un rôle précis,permettant une acquisition complète, fiable et synchronisée des grandeurs électriques du système. Tout d'abord, le **récepteur RC** transmet, sous forme d'impulsions, les commandes de déplacement provenant de la radiocommande. Ces impulsions sont envoyée à la carte FPGA, qui se charge d'en extraire les informations essentielles: direction du mouvement (avance, recul, virage gauche/droite) et type de manœuvre (accélération, freinage, virage combiné avant ou arrière). L'unité logique interne du FPGA convertit ensuite ces ordres en signaux MLI destinés aux moteurs, assurant ainsi un contrôle précis et réactif de leur vitesse.

En parallèle, la mesure du courant est assurée par le capteur à sortie analogique, qui nécessite un convertisseur analogique-numérique LTC1864 pour convertir le courant mesuré en une représentation numérique exploitable. Il faut changer la formulation, ce n'est pas le FPGA, ce sont les blocs logiques implémentés dans le FPGA qui permettent de réaliser ces fonctions de communication SPI avec le convertisseur analogique numérique pour récupérer les informations de mesure et les rendre exploitables aux fins de contrôle

et de supervision.

Les données converties par le LTC1864L sont récupérées par les blocs logiques de communication SPI implémentés dans le FPGA. Ces blocs utilisent les signaux **SCLK**, **CS** et **DOUT** pour assurer le transfert synchrone des données numériques issues du convertisseur analogique-numérique. Les valeurs acquises sont ensuite stockées et traitées par les modules internes du FPGA afin d'être rendues exploitables pour les fonctions de contrôle, de supervision et d'acquisition réalisées par le système.:

$$I = \frac{(\text{Raw} - \text{MID_RAW}) \times I_{\text{MAX}}}{\text{MID_RAW}}$$

La constante **MID_RAW** correspond à la valeur brute **32767.5**, représentant la moitié de **65535**, qui est la valeur maximale d'un capteur codé sur 16 bits $(2^{16} - 1)$.

Le **Imax** correspond à **30 A**, la valeur maximale de courant que peut mesurer le capteur. La courbe obtenue montre que le courant démarre à **0 A**, augmente lors des accélérations et présente des fluctuations dues aux variations de charge, du moteur et aux changements rapides de commande RC. Les pointes observées correspondent aux transitions de mouvement (démarrage, inversion, virage). La courbe confirme finalement la cohérence de toute la chaîne d'acquisition et la fiabilité du calcul du courant effectué sur le Raspberry Pi.

$$0\text{ A} \iff 1.65\text{ V} \iff 32767.5$$

$$30\text{ A} \iff 3.3\text{ V} \iff 65535$$

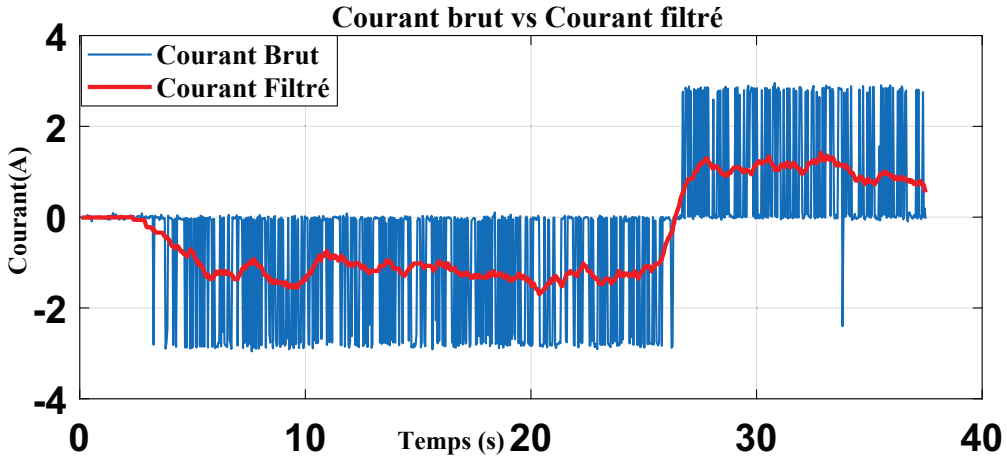


Figure 4.11 Test sur capteur de courant avec charge résistive de 3.3 Ω.

La figure 4.11 présente le signal de courant brut mesuré par le capteur et le signal filtré lorsqu'une charge résistive de **3.3** Ω est utilisée à la place du moteur. On observe que le signal brut reflète toutes les variations de courant imposées par le signal MLI envoyé au convertisseur. Il est également possible d'observer l'inversion de polarité liée aux changements de sens de rotation imposés par la commande RC envoyée grâce à la manette et reçue par le FPGA. On observe une phase initiale de courant négatif, suivie d'une augmentation progressive, puis d'un passage vers des valeurs positives autour de **27 s**, ce qui traduit un changement de régime de fonctionnement. Ainsi, le filtrage améliore considérablement la lisibilité des mesures tout en confirmant la tendance générale observée précédemment.

4.1.6 Simulation et expérimentation avec différentes commandes MLI

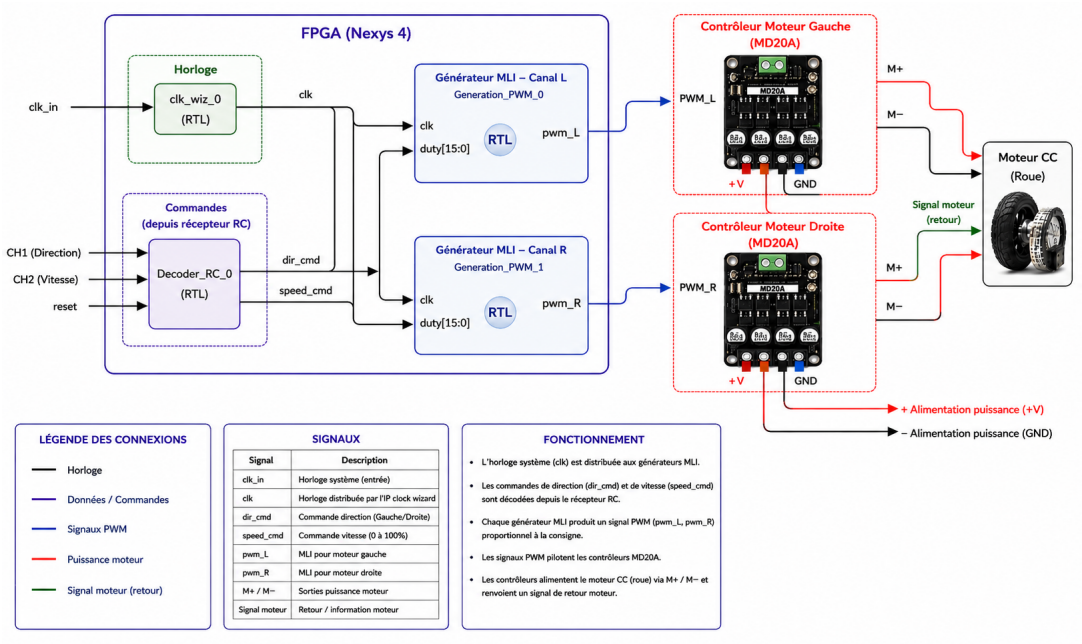


Figure 4.12 Architecture de commande des moteurs CC par FPGA avec génération MLI et contrôleurs MD20A.

La figure 4.12 présente la commande des moteurs CC par le FPGA, où les signaux MLI générés pilotent les contrôleurs MD20A afin de régler la vitesse et le sens de rotation des moteurs gauche et droit. Pour les essais présentés dans cette section, les valeurs de commande MLI ont été fixées à 1000, 4000 et 6000 à partir du Raspberry Pi puis transmises au FPGA. Cette approche permet de générer des rapports cycliques précis et reproductibles afin de comparer les résultats expérimentaux aux résultats de simulation.

La figure 4.13 présente les résultats des simulations et essais expérimentaux réalisés avec différentes commandes MLI afin d'évaluer leur influence sur le comportement du système et de valider l'architecture de commande proposée.

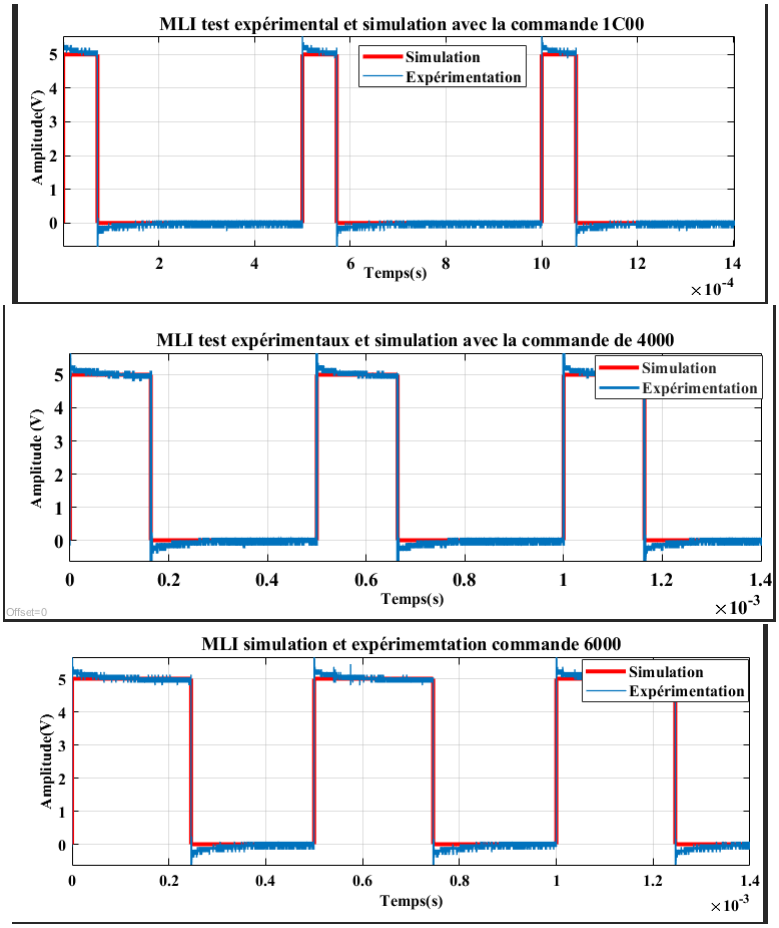


Figure 4.13 Test expérimental de la génération de signal MLI avec commandes à 1C00, à 4000 et à 6000.

Les résultats démontrent une très bonne concordance entre la simulation du signal et la réponse expérimentale du système pour les valeurs de commande reçues par le FPGA (1C00, 4000, 6000). La largeur d'impulsion mesurée correspond exactement à celle attendue en simulation, ce qui valide la justesse du module implémenté sur le FPGA chargé de générer les signaux MLI. Malgré un bruit notable dans les mesures expérimentales, dû principalement aux interférences générées, la forme de la durée des impulsions reste fidèle à la référence simulée. Ces résultats confirment la fiabilité de l'algorithme de commande ainsi que la stabilité de la communication SPI et de la chaîne d'acquisition.

4.1.7 Comparaison entre les valeurs théoriques et expérimentales

Afin d'évaluer la précision du générateur de signaux MLI implémenté dans le FPGA, les largeurs d'impulsion obtenues expérimentalement ont été comparées aux valeurs théoriques attendues pour différentes commandes numériques exprimées en hexadécimal. Les valeurs théoriques sont calculées à partir de la valeur du rapport cyclique chargé dans le registre du générateur MLI et de la fréquence d'horloge du FPGA fixée à 100 MHz. Les valeurs expérimentales sont quant à elles extraites des mesures réalisées sous MATLAB à partir des signaux générés.

L'erreur absolue est calculée selon la relation suivante :

$$E_{\text{abs}} = \left| V_{\text{théorique}} - V_{\text{mesurée}} \right| \tag{4.10}$$

où E_{abs} représente l'erreur absolue, $V_{\text{théorique}}$ la largeur d'impulsion théorique et $V_{\text{mesurée}}$ la largeur d'impulsion obtenue expérimentalement.

L'application numérique de l'équation (4.10) pour les trois commandes MLI est donnée ci-dessous.

Pour la commande `1C00` :

$$E_{\text{abs}} = |71,68 - 71,67| = 0,01\ \mu s = 10\ \text{ns} \tag{4.11}$$

Pour la commande `4000` :

$$E_{\text{abs}} = |163,84 - 163,83| = 0,01\ \mu s = 10\ \text{ns} \tag{4.12}$$

Pour la commande `6000` :

$$E_{\text{abs}} = |245,76 - 245,75| = 0,01\ \mu s = 10\ \text{ns} \tag{4.13}$$

Le tableau 4-3 présente la comparaison entre les valeurs théoriques et expérimentales des commandes MLI.

Tableau 4-3 Comparaison entre les valeurs théoriques et expérimentales des commandes MLI.

Commande	**Valeur théorique (μs)**	**Valeur mesurée (μs)**	**Erreur absolue (ns)**
1C00	71,68	71,67	10
4000	163,84	163,83	10
6000	245,76	245,75	10

Les résultats présentés dans le tableau 4-3 montrent une excellente concordance entre les valeurs théoriques et les valeurs mesurées. Pour les trois commandes testées, l'écart observé est de seulement $0,01\ \mu s$, soit une erreur absolue de 10 ns. Ces faibles écarts sont principalement attribuables à la résolution temporelle de l'acquisition ainsi qu'aux approximations inhérentes au processus de mesure numérique. Les résultats obtenus confirment le bon fonctionnement du générateur MLI implémenté dans le FPGA ainsi que la précision de la conversion des commandes numériques hexadécimales en largeurs d'impulsion conformes aux valeurs attendues.

4.1.8 Tests sur le récepteur RC et le générateur MLI

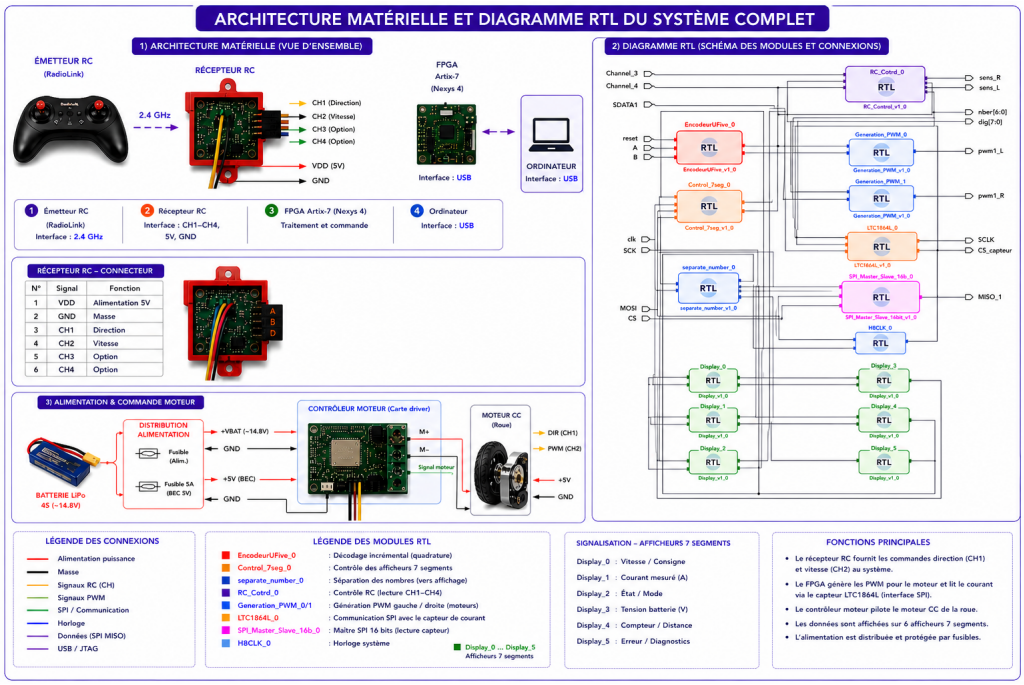


Figure 4.14 Architecture matérielle et diagramme RTL du système complet.

La figure 4.14 présente l'architecture matérielle globale ainsi que le diagramme RTL du système embarqué basé sur FPGA. Le récepteur RC transmet les consignes de direction et de vitesse vers la carte Nexys 4, où elles sont traitées par différents modules internes

dédiés à la commande, à l'affichage et à la génération des signaux MLI. Ces signaux pilotent ensuite le contrôleur moteur afin d'assurer le déplacement du système. Cette structure modulaire facilite l'intégration, le test et l'évolution de l'ensemble. Les essais expérimentaux sont réalisés à l'aide d'une télécommande radio (RC), dont les signaux de commande sont traités par un module dédié implémenté sur le FPGA. Les canaux RC sont caractérisés par des impulsions dont la largeur varie entre 1000 ms et 2000 ms, correspondant respectivement aux commandes de vitesse et de virage. Le FPGA mesure la durée des impulsions des canaux RC à partir d'un comptage en cycle d'horloge, puis applique un filtrage par moyenne glissante afin de réduire le bruit et les variations indésirables.

Le canal 3 (CH3) est utilisé pour déterminer la vitesse et le sens de rotation des moteurs. Une zone neutre centrée autour de 1500 ms permet de définir l'arrêt du système. Lorsque la largeur d'impulsion dépasse cette zone vers les valeurs élevées, le mouvement est interprété comme **marche avant**, tandis que toute diminution en dessous du neutre correspond à une **marche arrière**. La vitesse est calculée proportionnellement à l'écart par rapport à la position neutre. Le canal 4 (CH4) est dédié au contrôle du virage. Selon la valeur mesurée, une correction différentielle est appliquée aux moteurs: lors d'un virage à droite, la vitesse du moteur droit est réduite, tandis que, lors d'un virage à gauche, la réduction s'applique au moteur gauche. Les signaux de commande finaux sont générés sous forme de rapports cycliques (MLI) distincts pour les moteurs gauche et droit, après saturation afin d'éviter toute valeur négative. Ainsi, le système génère simultanément les signaux de vitesse, de direction et de modulation MLI, permettant un contrôle robuste, intuitif et en temps réel des moteurs à partir des commandes RC traitées par le FPGA.

- T_{RC} : durée de l'impulsion RC mesurée en millisecondes ou en nombre de cycles d'horloge;

- $T_{\text{mid}} = 1500$ ms;
- $\Delta =$ NEUTRAL_BAND.

$$DP = \begin{cases} 0, & \text{si } |T_{RC} - T_{\text{mid}}| \leq \Delta \\ T_{RC} - T_{\text{mid}}, & \text{si } T_{RC} > T_{\text{mid}} + \Delta \quad \text{(marche avant)} \\ T_{\text{mid}} - T_{RC}, & \text{si } T_{RC} < T_{\text{mid}} - \Delta \quad \text{(marche arrière)} \end{cases}$$

$$\text{sens} = \begin{cases} 0 & \text{si } T_{RC} \geq T_{\text{mid}} \quad \text{(avant)} \\ 1 & \text{si } T_{RC} < T_{\text{mid}} \quad \text{(arrière)} \end{cases}$$

L'analyse de l'ensemble des essais réalisés sur le récepteur RC montre une bonne cohérence entre les consignes issues de la télécommande et les signaux de commande générés par le FPGA. Les figures 4.15 et 4.16 montrent respectivement les résultats expérimentaux lorsque la commande impose un déplacement en ligne droite en sens avant et en sens arrière, et lorsque des virages à gauche et à droite sont demandés.

Les largeurs d'impulsion mesurées sur les canaux RC sont correctement interprétées par le module de commande, qui permet d'identifier avec précision les états de marche avant et arrière ainsi que les commandes de virage. Le calcul du sens de rotation est conforme à la logique implémentée, avec un signal de direction égal à **0** pour la marche avant et égal à **1** pour la marche arrière.

Les signaux MLI appliqués aux moteurs gauche et droit traduisent correctement les différentes situations de conduite.

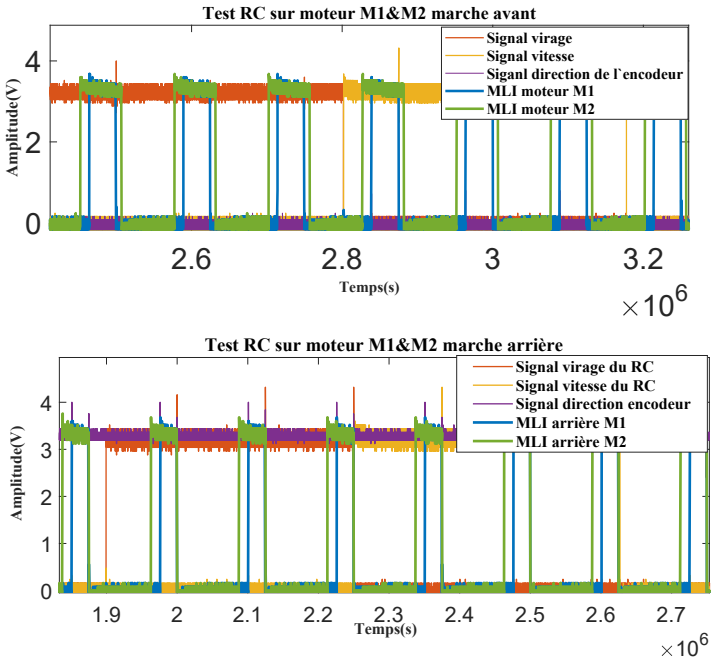


Figure 4.15 Résultats de la réception RC et de la génération MLI sur les moteurs 1 et 2, en commande de marche avant et arrière.

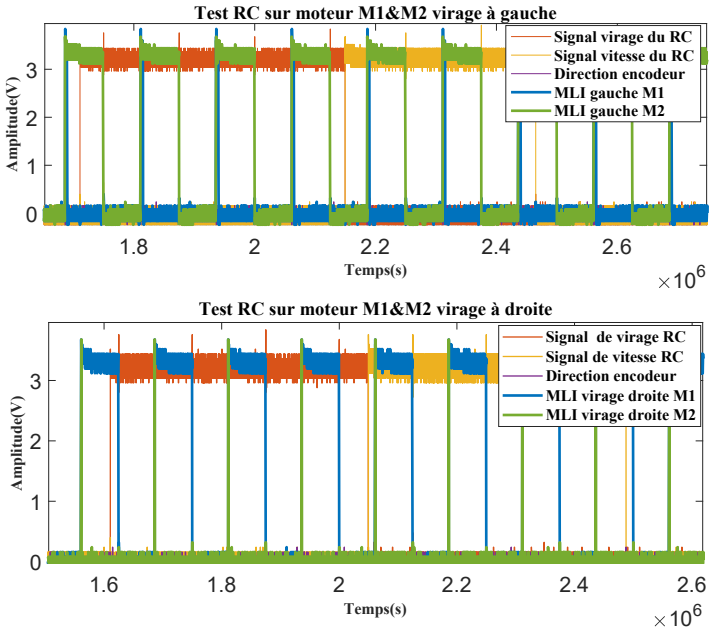


Figure 4.16 Résultats de la réception RC et de la génération MLI sur les moteurs 1 et 2, en commande de virage à gauche et à droite.

4.1.9 Expérimentation sur encodeur

La figure 4.17 présente l'architecture matérielle simplifiée ainsi que le diagramme de blocs des principaux modules implémentés dans le FPGA. L'architecture matérielle, illustrée dans la partie gauche de la figure, est constituée d'un émetteur et d'un récepteur de radiocommande (RC), d'un FPGA Nexys 4, d'un capteur de courant LTC1864L, de deux contrôleurs de moteurs MD20A, d'un moteur à courant continu ainsi que d'un afficheur à sept segments. Les commandes de direction et de vitesse sont transmises par l'émetteur RC puis reçues par le récepteur RC. Les signaux correspondants (CH1 et CH2) sont ensuite envoyés au FPGA, qui assure le traitement des informations et la génération des signaux de commande. Le capteur de courant LTC1864L est connecté au FPGA via une liaison SPI permettant l'acquisition des mesures de courant nécessaires à la supervision du système. Les données traitées peuvent également être visualisées à l'aide des afficheurs à sept segments.

Les signaux MLI générés par le FPGA sont appliqués aux contrôleurs de moteurs MD20A, qui adaptent la puissance électrique fournie au moteur à courant continu. Cette chaîne de commande permet de convertir les consignes de vitesse et de direction en actions mécaniques tout en assurant le suivi des grandeurs électriques du système.

La partie droite de la figure présente une vue simplifiée des principaux blocs logiques implantés dans le FPGA. Le module `RC_Control` assure le décodage des commandes provenant du récepteur RC et génère les signaux de direction (`dir_cmd`) et de vitesse (`speed_cmd`). Le module `LTC1864L_SPI` gère la communication SPI avec le convertisseur analogique-numérique associé au capteur de courant. Ces informations sont centralisées par le module `Control_Top`, qui constitue l'unité principale de contrôle du système. Les modules `PWM_Gen`, `Encoder` et `Display_Driver` assurent respectivement la génération

des signaux MLI, le traitement des informations provenant de l'encodeur et la commande des afficheurs à sept segments.

Cette organisation permet de séparer clairement les fonctions d'acquisition, de traitement, de commande et d'affichage tout en exploitant les capacités de traitement parallèle du FPGA pour assurer un fonctionnement en temps réel. Les résultats présentés dans la figure 4.18 montrent des signaux cohérents et correctement déphasés, confirmant le bon fonctionnement de l'encodeur.

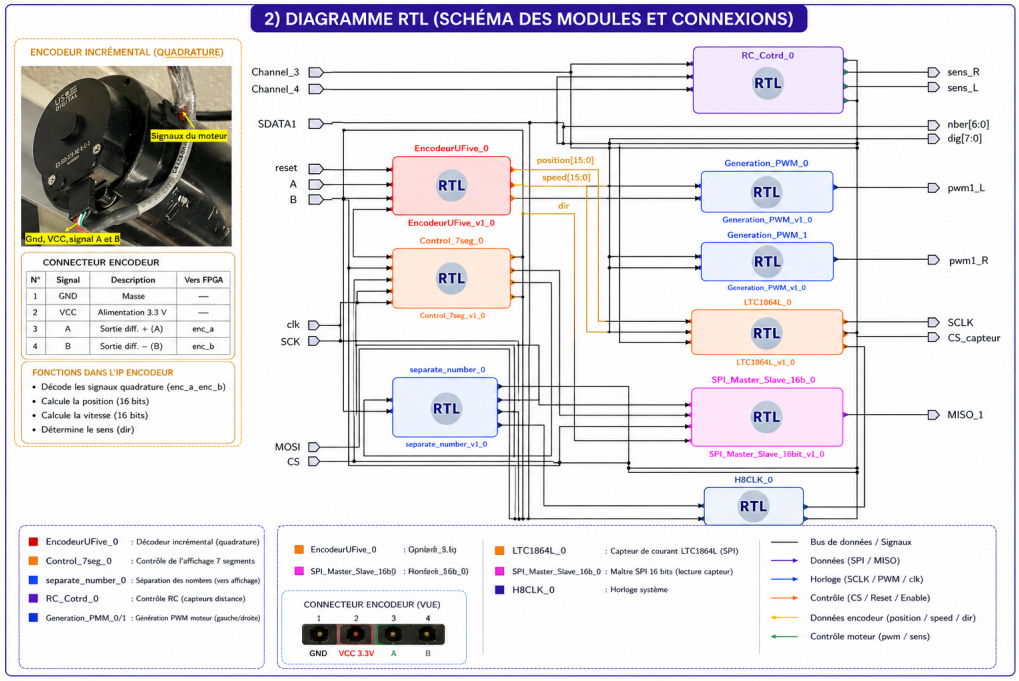


Figure 4.17 Architecture matérielle et diagramme RTL du système embarqué de commande et mesure par FPGA.

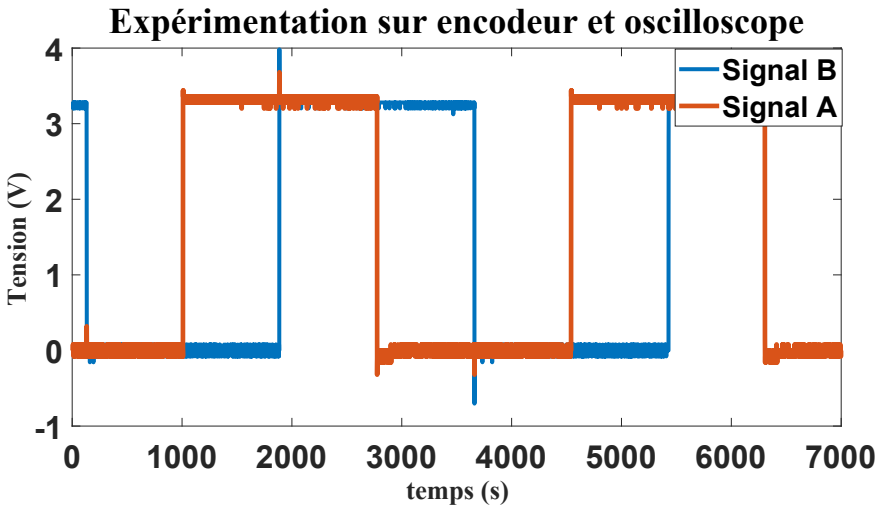


Figure 4.18 Résultats expérimentaux du test de l'encodeur - signaux A et B observés à l'oscilloscope.

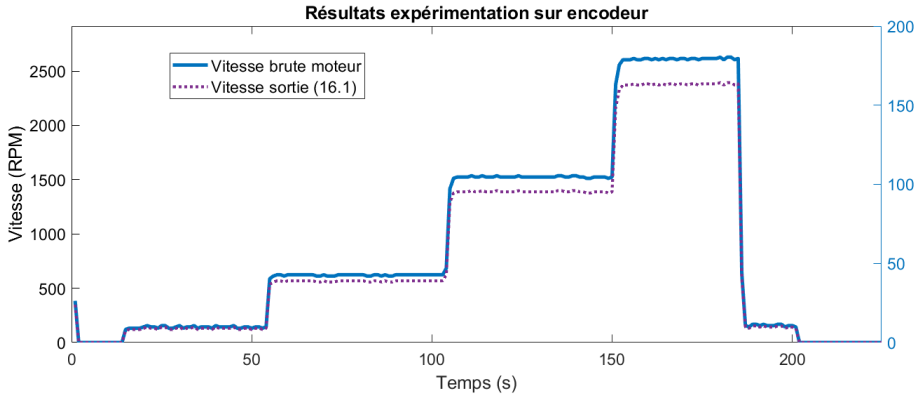


Figure 4.19 Résultats expérimentaux du test de l'encodeur - vitesse obtenue avec et sans facteur de réduction.

La figure 4.19 illustre l'évolution temporelle de la vitesse mesurée par l'encodeur sur l'arbre moteur ainsi que la vitesse brute en sortie du réducteur. La vitesse du moteur présente des paliers stables correspondant aux différentes consignes appliquées, tandis que la vitesse en sortie est réduite proportionnellement tout en conservant la même dynamique. La réduction de la vitesse des roues par rapport à celle de l'arbre du moteur est due à l'utilisation d'un réducteur de rapport de 16:1 (boîte à engrenages). L'ajout du réducteur permet de réduire la vitesse de rotation tout en augmentant le couple disponible en sortie, ce qui est particulièrement adapté aux applications en robotique agricole. La vitesse brute en sortie du réducteur est obtenue à partir de la relation suivante :

$$RAW_{\text{sortie}} = \frac{RAW_{\text{moteur}}}{16}$$

Les valeurs de vitesse sont calculées directement sur le FPGA, puis transmises au Raspberry Pi via le protocole SPI.

4.2 Conclusion

Ce chapitre a présenté les résultats de la validation expérimentale de l'architecture de commande proposée, à travers une série de tests effectués sur la plateforme développée. Les différents modules, notamment l'acquisition des données, la communication (SPI, I^2C, UART) et la génération des signaux de commande (RC ou MLI), ont été testés et analysés progressivement.

Les résultats obtenus montrent une bonne cohérence entre les simulations et les mesures expérimentales, confirmant le bon fonctionnement des différents blocs du système. La génération des signaux MLI, la transmission des données ainsi que la réponse des actionneurs ont démontré la fiabilité de l'architecture mise en œuvre.

Ainsi, l'ensemble des expérimentations valide l'efficacité du système proposé, tout en mettant en évidence sa capacité à répondre aux exigences des applications d'agriculture de précision. Ce travail constitue une base solide pour des améliorations futures, notamment en matière d'optimisation des performances et d'intégration de nouvelles fonctionnalités.

Chapitre 5 - Conclusion

5.1 Synthèse des travaux réalisés

Ce travail de recherche s'est inscrit dans le contexte de **l'électronique de commande appliquée aux systèmes électromécaniques en agriculture de précision**, avec pour objectif principal la conception, l'implémentation et la validation expérimentale d'une architecture de commande reposant sur un FPGA et un Raspberry Pi.

Les contributions majeures de ce mémoire portent sur le développement de plusieurs modules matériels et logiciels permettant la commande d'actionneurs électromécaniques, l'acquisition de données issues de différents capteurs ainsi que la mise en œuvre de plusieurs protocoles de communication, notamment **SPI**, **I^2C** et **UART**. L'architecture proposée exploite les capacités de traitement temps réel du FPGA pour la génération des signaux MLI, la gestion des interfaces de communication et l'acquisition des données, tandis que le Raspberry Pi assure la supervision, le traitement des commandes et les échanges avec les différents modules du système.

Plusieurs validations expérimentales ont été réalisées afin de vérifier le bon fonctionnement des modules développés. Les essais ont permis de valider la commande des moteurs à courant continu, la commande d'un servomoteur via les protocoles SPI et UART, la réception des commandes issues d'un récepteur RC, l'acquisition des mesures du convertisseur analogique-numérique LTC1864, la lecture des données du module HX711 ainsi que la mesure de vitesse à l'aide d'un encodeur incrémental.

Les comparaisons entre les valeurs théoriques et les mesures expérimentales montrent une excellente concordance des résultats. Les faibles erreurs absolues obtenues lors des essais sur la génération des signaux MLI et sur la commande du servomoteur confirment

la précision des modules développés et valident le bon fonctionnement de l'architecture proposée pour des applications de commande embarquée en agriculture de précision.

5.2 Difficultés rencontrées et limites du travail

La réalisation expérimentale de ce projet a mis en évidence plusieurs difficultés techniques, notamment lors de l'implémentation des protocoles de communication et de l'intégration des différents capteurs au sein de l'architecture matérielle.

La mise en œuvre du protocole **I²C** avec le module **HX711** a constitué l'une des principales difficultés rencontrées. La documentation technique disponible étant parfois incomplète ou peu explicite, plusieurs phases de validation ont été nécessaires afin d'assurer une acquisition fiable des données et leur conversion en mesures exploitables. Des difficultés similaires ont également été rencontrées lors de l'intégration de l'encodeur incrémental, notamment pour la détermination de la vitesse de rotation et la validation expérimentale des mesures.

À l'inverse, les protocoles **SPI** et **UART** se sont révélés plus simples à implémenter et à déboguer. Les essais réalisés ont montré qu'ils permettent des échanges rapides, fiables et déterministes entre le FPGA et le Raspberry Pi, tout en assurant une commande précise des actionneurs.

Il convient également de souligner qu'au début de cette recherche, l'auteure ne possédait aucune expérience préalable en programmation VHDL ni en conception sur FPGA. Ce projet a ainsi constitué une importante phase d'apprentissage ayant permis d'acquérir des compétences solides en logique numérique, en conception matérielle et en co-conception matériel–logiciel.

Un autre défi important concerne les perturbations observées sur les signaux issus du récepteur RC lors de l'intégration complète de la plateforme expérimentale. La présence d'interférences électriques générées par les moteurs et les différents équipements a nécessité le développement de filtres numériques implantés dans le FPGA afin de stabiliser les impulsions reçues et de garantir une commande fiable des moteurs et du servomoteur. Les résultats expérimentaux obtenus montrent que ces traitements permettent d'améliorer significativement la robustesse de la commande.

Enfin, les validations ont été réalisées sur une plateforme expérimentale de laboratoire. Les performances de l'architecture proposée devront être confirmées sur un robot agricole évoluant dans un environnement réel, où les contraintes mécaniques, électriques et environnementales sont plus importantes.

5.3 Choix des protocoles de communication

Les expérimentations réalisées au cours de ce travail ont permis de comparer les protocoles **SPI**, **I²C** et **UART** dans différentes applications de commande et d'acquisition.

Le protocole **SPI** s'est montré particulièrement adapté aux applications nécessitant des échanges rapides de données, notamment pour la communication avec le convertisseur analogique–numérique LTC1864 et pour la commande du servomoteur. Les mesures expérimentales ont mis en évidence de très faibles erreurs absolues, confirmant la précision et la rapidité de cette interface de communication.

Le protocole **I²C** a permis l'acquisition des mesures issues du module HX711 tout en limitant le nombre de lignes de communication nécessaires. Toutefois, sa mise en œuvre s'est révélée plus sensible aux problèmes de synchronisation ainsi qu'à la qualité de la documentation technique du composant.

Le protocole **UART** s'est distingué par sa simplicité d'implémentation et sa facilité d'intégration avec le Raspberry Pi. Les essais réalisés sur la commande du servomoteur montrent qu'il constitue une solution fiable lorsque les contraintes de débit sont moins élevées.

Les résultats obtenus montrent ainsi que ces trois protocoles sont complémentaires. Leur utilisation conjointe permet de tirer parti des avantages propres à chacun d'eux selon les fonctions à réaliser, qu'il s'agisse de la commande des actionneurs, de l'acquisition des données ou de la supervision du système.

5.4 Perspectives et recommandations

À la lumière des résultats obtenus, plusieurs perspectives de recherche peuvent être envisagées. Une première perspective consiste à intégrer des lois de commande avancées, telles que des commandes adaptatives, prédictives ou basées sur l'intelligence artificielle, afin d'améliorer les performances dynamiques des systèmes électromécaniques utilisés en agriculture de précision.

L'intégration de protocoles de communication industriels, tels que le bus CAN ou Ethernet industriel, permettrait d'étendre les possibilités d'interconnexion avec des plateformes robotiques de plus grande envergure et de renforcer la robustesse des échanges de données. Par ailleurs, l'ajout de nouveaux capteurs environnementaux, de systèmes de vision et d'algorithmes de traitement intelligent des données contribuerait à accroître l'autonomie décisionnelle des robots agricoles.

Enfin, une validation de l'architecture développée sur un robot agricole autonome évoluant dans des conditions réelles permettrait d'évaluer sa robustesse, sa fiabilité et ses performances dans un environnement représentatif des applications d'agriculture

de précision. Une étude comparative de la consommation énergétique des différentes architectures de commande constituerait également une perspective intéressante en vue de développer des systèmes plus efficaces et mieux adaptés aux exigences de l'agriculture durable.

RÉFÉRENCES

[1] A. Ghobadpour, G. Monsalve, A. Cardenas, and H. Mousazadeh, "Off-road electric vehicles and autonomous robots in agricultural sector: Trends, challenges, and opportunities," *Vehicles*, vol. 4, no. 3, pp. 843–864, 2022.

[2] Y. Miyake, S. Kimoto, Y. Uchiyama, and R. Kohsaka, "Income change and inter-farmer relations through conservation agriculture in ishikawa prefecture, japan: Empirical analysis of economic and behavioral factors," *Land*, vol. 11, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2073-445X/11/2/245

[3] G. Papadopoulos, S. Arduini, H. Uyar, V. Psiroukis, A. Kasimati, and S. Fountas, "Economic and environmental benefits of digital agricultural technologies in crop production: A review," *Smart Agricultural Technology*, vol. 8, p. 100441, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772375524000467

[4] M. Sanyaolu and A. Sadowski, "The role of precision agriculture technologies in enhancing sustainable agriculture," *Sustainability*, vol. 16, no. 15, 2024. [Online]. Available: https://www.mdpi.com/2071-1050/16/15/6668

[5] C. M. Onyango, J. M. Nyaga, J. Wetterlind, M. Söderström, and K. Piikki, "Precision agriculture for resource use efficiency in smallholder farming systems in sub-saharan africa: A systematic review," *Sustainability*, vol. 13, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/3/1158

[6] S. D. Brown, R. J. Francis, J. Rose, Z. G. Vranesic, S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, "Introduction to fpgas," *Field-Programmable Gate Arrays*, pp. 1–11, 1992.

[7] S. G. Vougioukas, "Agricultural robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. Volume 2, 2019, pp. 365–392, 2019. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-control-053018-023617

[8] A. Ghobadpour, A. Cardenas, G. Monsalve, and H. Mousazadeh, "Optimal design of energy sources for a photovoltaic/fuel cell extended-range agricultural mobile robot," *Robotics*, vol. 12, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2218-6581/12/1/13

[9] L. Christiaensen, Z. Rutledge, and J. E. Taylor, "Viewpoint: The future of work in agri-food," *Food Policy*, vol. 99, no. August 2020, p. 101963, 2021. [Online]. Available: https://doi.org/10.1016/j.foodpol.2020.101963

[10] A. Muharemović, D. Jokić, M. Simeunović, and H. Hanjalić, "Fpga technologies for smart and sustainable agriculture: A comprehensive overview," in *2023 12th Mediterranean Conference on Embedded Computing (MECO)*, 2023, pp. 1–5.

[11] M. X. Bui, D. Xiao, and M. F. Rahman, "Sensorless control and inductances estimation of ipmsms using fpga and high bandwidth current measurement," *IEEE Access*, vol. 11, pp. 67 322–67 329, 2023.

[12] S. G. Vougioukas, "Agricultural robotics," *Annual review of control, robotics, and autonomous systems*, vol. 2, no. 1, pp. 365–392, 2019.

[13] R. Sparrow and M. Howard, "Robots in agriculture: prospects, impacts, ethics, and policy," *precision agriculture*, vol. 22, no. 3, pp. 818–833, 2021.

[14] T. Liu, W. Tan, X. Tang, J. Zhang, Y. Xing, and D. Cao, "Driving conditions-driven energy management for hybrid electric vehicles: A review," 2020. [Online]. Available: https://arxiv.org/abs/2007.10880

[15] A. Ghobadpour, H. Mousazadeh, S. Kelouwani, A. S. Malvajerdi, and S. Rafiee, "Design, development, and evaluation of a pv_bio-gen range extender for an off-road electric vehicle," *International Journal of Renewable Energy Research*, vol. 10, no. 1, pp. 388–399, 2020.

[16] Y. Li, M. Liu, Y. Wang, L. Xu, and S. Lei, "Energy management optimization and validation of a hydrogen fuel cell-powered agricultural tractor based on hierarchical dynamic programming," *IEEE Access*, pp. 1–16, 2024.

[17] A. Milella, G. Reina, and M. Nielsen, "A multi-sensor robotic platform for ground mapping and estimation beyond the visible spectrum," *Precision agriculture*, vol. 20, no. 2, pp. 423–444, 2019.

[18] W. B. N.R, P. M., W. H. SK, S. Palarimath, P. S., B. C., S. Kumari V., S. Subitha G., and S. C. S., "Leveraging internet of things (iot) sensors and deep learning techniques for precision agriculture," in *2024 1st International Conference on Innovative Engineering Sciences and Technological Research (ICIESTR)*, 2024, pp. 1–6.

[19] K. V. Sai Vineeth, Y. R. Vara Prasad, S. R. Dubey, and H. Venkataraman, "Ledcom: A novel and efficient led based communication for precision agriculture," in *2019 IEEE Conference on Information and Communication Technology*, 2019, pp. 1–5.

[20] T. Abbas, M. Arif, and W. Ahmed, "Measurement and correction of systematic odometry errors caused by kinematics imperfections in mobile robots," in *2006 SICE-ICASE International Joint Conference*, 2006, pp. 2073–2078.

[21] B. K. Bose, "Power electronics and motor drives recent progress and perspective," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 581–588, 2009.

[22] Y.-y. Fang and X.-j. Chen, "Design and simulation of uart serial communication module based on vhdl," in *2011 3rd International Workshop on Intelligent Systems and Applications*, 2011, pp. 1–4.

[23] Y. Hua, N. Zhang, X. Yuan, L. Quan, J. Yang, K. Nagasaka, and X.-G. Zhou, "Recent advances in intelligent automated fruit harvesting robots," *The Open Agriculture Journal*, vol. 13, no. 1, 2019.

[24] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, "From industry 4.0 to agriculture 4.0: Current status, enabling technologies, and research challenges," *IEEE transactions on industrial informatics*, vol. 17, no. 6, pp. 4322–4334, 2020.

[25] A. Salam, "Internet of things for sustainability: perspectives in privacy, cybersecurity, and future trends," in *Internet of things for sustainable community development: wireless communications, sensing, and systems*. Springer, 2024, pp. 299–326.

[26] J. Stakens, A. Mutule, and R. Lazdins, "Agriculture electrification, emerging technologies, trends and barriers: A comprehensive literature review," *Latvian Journal of Physics and Technical Sciences*, vol. 60, pp. 18–32, 05 2023.

[27] Y. Deng, "Design theory, method, and control of intelligent and safe vehicles," *World Electric Vehicle Journal*, vol. 16, no. 12, 2025. [Online]. Available: https://www.mdpi.com/2032-6653/16/12/652

[28] M. Molina, A. Vera, C. Molina, and P. Garzon, "Design and construction of an obstacle avoiding robot based on arduino platform and programming too," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 2018, pp. 788–791.

[29] P. Bagdalkar and L. Ali, "Hardware implementation of i2c controller on fpga and validation through interfacing with low-cost adc," in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, 2020, pp. 887–891.

[30] Y. Xie, Z. Gu, B. Zhu, L. Wang, W. Han, and L. Yin, "Adversarial examples for chinese text classification," in *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, 2020, pp. 238–245.

[31] G. J. García, C. A. Jara, J. Pomares, A. Alabdo, L. M. Poggi, and F. Torres, "A survey on fpga-based sensor systems: Towards intelligent and reconfigurable low-power sensors for computer vision, control and signal processing," *Sensors*, vol. 14, no. 4, pp. 6247–6278, 2014. [Online]. Available: https://www.mdpi.com/1424-8220/14/4/6247

[32] E. Monmasson and M. Cirstea, "Fpga design methodology for industrial control systems," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1824–1842, 2007.

[33] C. Zhang, P. Li, G. Sun, and Y. Guan, "Optimizing fpga-based accelerator design for deep convolutional neural networks," in *FPGA*, 2015, pp. 161–170.

[34] E. Nurvitadhi *et al.*, "Can fpgas beat gpus in accelerating next-generation deep neural networks ?" *FPGA*, pp. 5–14, 2017.

[35] H. Li, X. Fan, and L. Jiao, "A high performance fpga-based accelerator for large-scale convolutional neural networks," *IEEE Access*, vol. 7, pp. 51 896–51 907, 2019.

[36] Y. Chen, T. Luo, and S. Liu, "A survey of fpga-based neural network accelerator," *IEEE Access*, vol. 7, pp. 122 030–122 048, 2019.

[37] A. Saddik, R. Latif, and A. El Ouardi, "Low-power fpga architecture based monitoring applications in precision agriculture," *Journal of Low Power Electronics and Applications*, vol. 11, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2079-9268/11/4/39

[38] Raspberry Pi Foundation, "Raspberry pi documentation," 2024, consulté le 25 juin 2026. [Online]. Available: https://www.raspberrypi.com/documentation/

[39] C. T. Nautiyal, P. Nautiyal, G. Papnai, H. Mittal, K. Agrawal, and R. Nandini, "Importance of smart agriculture and use of artificial intelligence in shaping the

future of agriculture," *Journal of Scientific Research and Reports*, vol. 30, no. 3, pp. 129–138, 2024.

[40] S. A. Lokhande, "Effective use of big data in precision agriculture," in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE, 2021, pp. 312–316.

[41] M. Sufyan, "Spi protocol: Revolutionizing data communication in embedded systems," https://www.wevolver.com/article/spi-protocol, Jun. 2024, wevolver, accessed 2025.

[42] T. Yu, M. Liya, L. Xiaohui, and J. Yunzhong, "Construction of water supply pipe network based on gis and epanet model in fangcun district of guangzhou," in *2010 Second IITA International Conference on Geoscience and Remote Sensing*, vol. 2, 2010, pp. 268–271.

[43] F. Leens, "An introduction to i2c and spi protocols," *IEEE Instrumentation & Measurement Magazine*, vol. 12, no. 1, pp. 8–13, 2009.

[44] A. Name and S. A. Name, "Vlsi implementation of spi and i2c communication protocols," *International Journal of Advanced Research in AI and IT*, vol. 6, no. 4, p. —, 2020, accessed: 2025-01-xx. [Online]. Available: https://www.ijariit.com/manuscripts/v6i4/V6I4-1312.pdf

[45] C.-W. Liao, H.-C. Yu, and Y.-C. Liao, "Verification of spi protocol using universal verification methodology for modern iot and wearable devices," *Electronics*, vol. 14, no. 5, 2025. [Online]. Available: https://www.mdpi.com/2079-9292/14/5/837

[46] NXP Semiconductors, *UM10204: I2C-bus specification and user manual*, NXP Semiconductors, Eindhoven, Netherlands, 2014, rev. 6. [Online]. Available: https://www.nxp.com/docs/en/user-guide/UM10204.pdf

[47] Z.-w. Hu, "I2c protocol design for reusability," in *2010 Third International Symposium on Information Processing*, 2010, pp. 83–86.

[48] Y. Zhou, P. Yan, Y. Sun, D. Zhang, M. Li, M. Li, Y. Zhou, Y. Zhou, and M. Li, "Analysis on efficiency improvement with a distributed energy store railgun," in *2010 IEEE International Power Modulator and High Voltage Conference*, 2010, pp. 587–589.

[49] R. Yang, "Analysis and research on reliable serial communication based on embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 4, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3675249.3675266

[50] J. Kopparapu and S. Rambabu, "An effective fpga based real time system for measuring temperature and light intensity using uart protocol," in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, 2025, pp. 439–443.

[51] P. Brooks, "Ethernet/ip-industrial protocol," in *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597)*, vol. 2, 2001, pp. 505–514 vol.2.

[52] P. Pedreiras, L. Almeida, and P. Gai, "The ftt-ethernet protocol: Merging flexibility, timeliness and efficiency," in *Proceedings 14th Euromicro Conference on Real-Time Systems. Euromicro RTS 2002*. IEEE Computer Society, 2002, pp. 152–152.

[53] Z. Lin, S. Pearson *et al.*, "An inside look at industrial ethernet communication protocols," *Texas Instruments, White Paper*, 2013.

[54] C. Venkatramani and T.-c. Chiueh, "Design, implementation, and evaluation of a software-based real-time ethernet protocol," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, 1995, pp. 27–37.

[55] J. Jasperneite, M. Schumacher, and K. Weber, "Limits of increasing the performance of industrial ethernet protocols," in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*. IEEE, 2007, pp. 17–24.

[56] C. Zhuang, "Comparison and selection of commonly used communication protocols in measurement and control instruments," *Highlights in Science, Engineering and Technology*, vol. 81, pp. 540–546, 2024.

[57] Z. Ma and D. Sameoto, "A review of electrically driven soft actuators for soft robotics," *Micromachines*, vol. 13, no. 11, p. 1881, 2022. [Online]. Available: https://www.mdpi.com/2072-666X/13/11/1881

[58] S. Chapman, *Electric Machinery Fundamentals*, ser. McGraw-Hill series in electrical and computer engineering. McGraw-Hill, 2012. [Online]. Available: https://books.google.ca/books?id=wyHAuQAACAAJ

[59] W. Bolton, *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, 6th ed. Pearson Education, 2015.

[60] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2010.

[61] Texas Instruments, "Designing motor driver systems," Texas Instruments, Tech. Rep., 2018.

[62] R. Erickson, "Optimal single resistors damping of input filters," in *APEC '99. Fourteenth Annual Applied Power Electronics Conference and Exposition. 1999 Conference Proceedings (Cat. No.99CH36285)*, vol. 2, 1999, pp. 1073–1079 vol.2.

[63] R. Krishnan, *Electric Motor Drives: Modeling, Analysis, and Control*. Prentice Hall, 2001.

[64] F. Blaabjerg and R. Teodorescu, "Overview of control and grid synchronization for distributed power generation systems," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1398–1409, 2006.

[65] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd ed. John Wiley & Sons, 2003.

[66] B. J. Baliga, *Fundamentals of Power Semiconductor Devices*. Springer, 2008.

[67] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*. Springer, 2001.

[68] B. K. Bose, "Power electronics and motor drives—recent progress and perspective," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 581–588, 2009.

[69] M. K. Kazimierczuk, *Pulse-Width Modulated DC-DC Power Converters*. Wiley, 2008.

[70] D. G. Holmes and T. A. Lipo, *Pulse Width Modulation for Power Converters: Principles and Practice*. IEEE Press, 2003.

[71] F. Blaabjerg, Z. Chen, and S. B. Kjaer, "Power electronics as efficient interface in dispersed power generation systems," *IEEE Transactions on Power Electronics*, vol. 19, no. 5, pp. 1184–1194, 2004.

[72] J. Rodriguez, J.-S. Lai, and F. Z. Peng, "Multilevel inverters: A survey of topologies, controls, and applications," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 4, pp. 724–738, 2002.

[73] F. Paredes, C. Herrojo, and F. Martín, "Position sensors for industrial applications based on electromagnetic encoders," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2738

[74] J. K. Mohan and N. Johnson, "Devising simulink optical encoder pulse manipulation and its evaluation," in *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2012, pp. 640–644.

[75] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[76] N. S. Nise, *Control systems engineering*. John Wiley & Sons, 2019.

[77] P. Corke, "Mobile robot vehicles," in *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition*. Springer, 2017, pp. 99–124.

[78] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer, 2009.

[79] B. K. Bose, "Power electronics and motor drives—recent progress and perspective," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 581–588, 2009.

[80] J. J. Craig, *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009.

[81] S. Marios and J. Georgiou, "Precision agriculture: Challenges in sensors and electronics for real-time soil and plant monitoring," in *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2017, pp. 1–4.

[82] M. Perez-Ruiz and S. K. Upadhyaya, "Gnss in precision agricultural operations," *New approach of indoor and outdoor localization systems*, p. 3e26, 2012.

[83] D. J. Mulla, "Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps," *Biosystems engineering*, vol. 114, no. 4, pp. 358–371, 2013.

[84] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision agriculture*, vol. 13, no. 6, pp. 693–712, 2012.

[85] O. Chávez-Martínez, S. A. Monjardin-Armenta, J. G. Rangel-Peraza, W. Plata-Rocha, E. Avila-Aceves, and Y. G. Zambrano-Medina, "Remote sensing in agriculture: Advances in satellite sensors and unmanned aerial vehicles," *Agri-informatics and Eco-friendly Innovations for a Secure Food Future*, pp. 39–57, 2026.

[86] A. Bechar and C. Vigneault, "Agricultural robots for field operations: Concepts and components," *Biosystems Engineering*, vol. 149, pp. 94–111, Sep. 2016, publisher Copyright: © 2016.

[87] J. G. Webster, *The physiological measurement handbook*. CRC Press, 2014.

[88] J. G. Webster and H. Eren, "Measurement, instrumentation, and sensors handbook: Two-volume set (electrical engineering handbook)," 2014.

Annexe A - Standard I²C protocole entre Maître-esclave

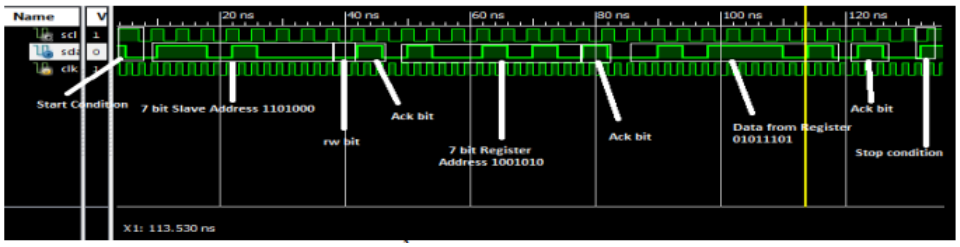


FIGURE A.1 – Standard I²C protocole entre Maître-esclave
[44]

Annexe B - Capture bloc design I2C

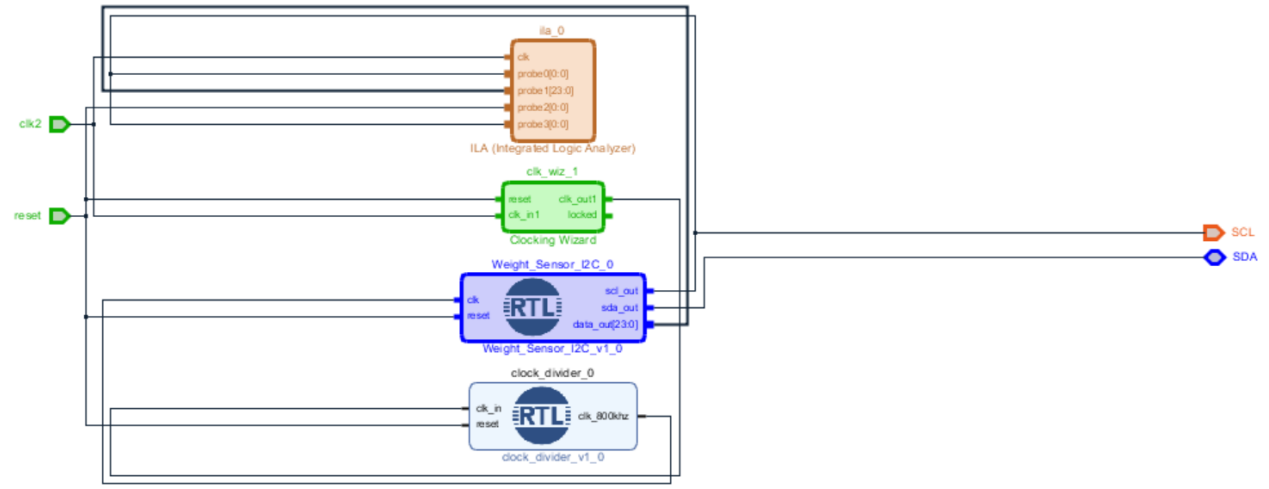


FIGURE B.1 – Bloc design I^2C

Annexe C - Bloc Design pour le récepteur RC

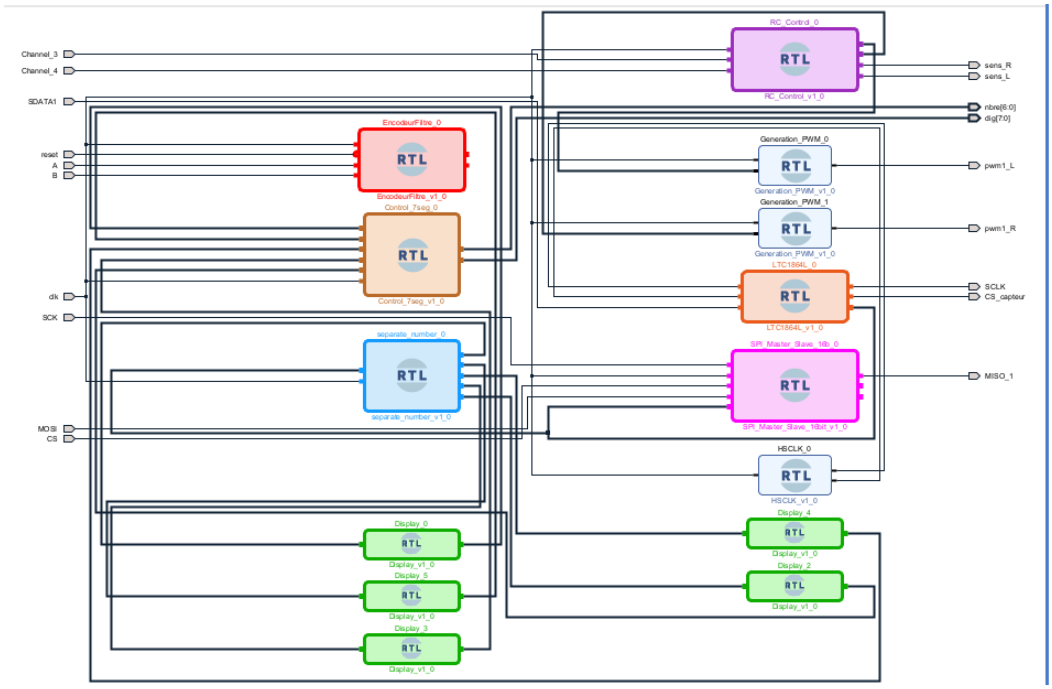


FIGURE C.1 – Bloc Design pour le récepteur RC

Annexe D - Bloc Design pour Encodeur

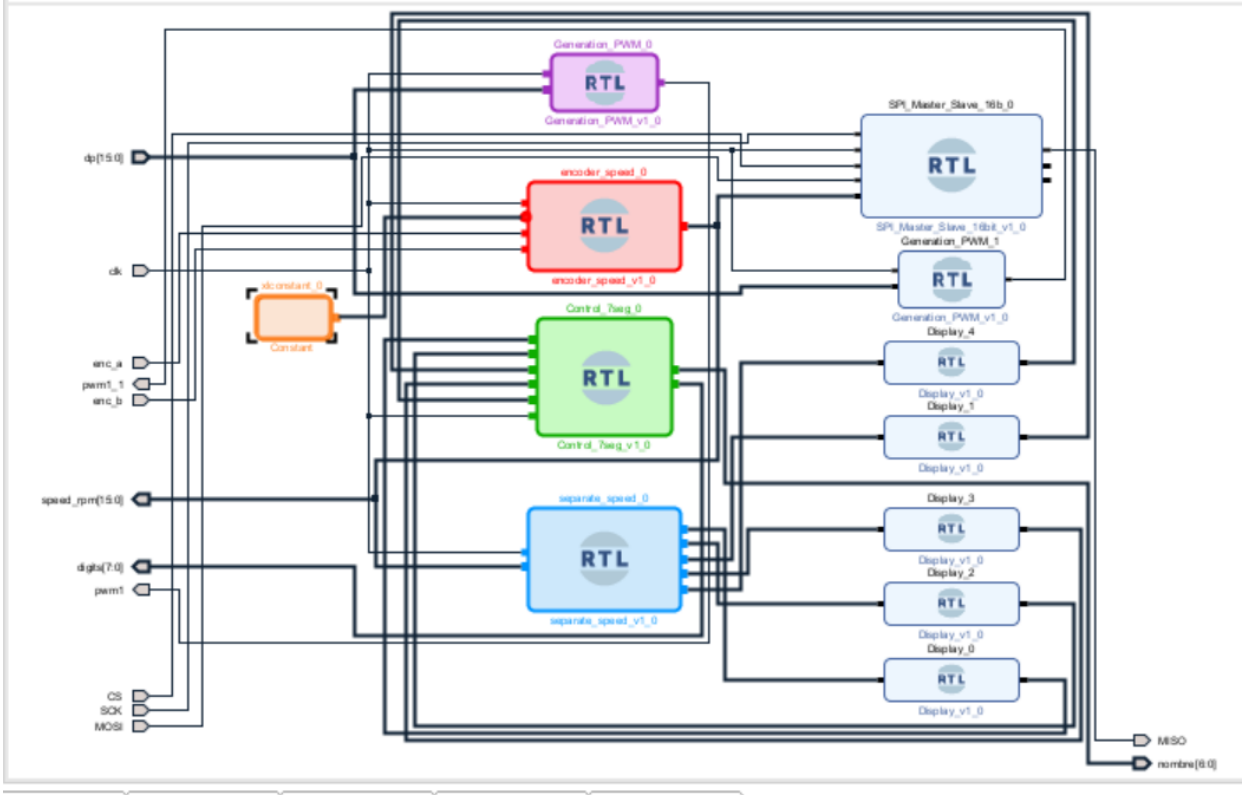


FIGURE D.1 – Bloc Design pour l'Encodeur

Annexe E - Capture du Test numérique de la balance I²C 1953g sur l'ILA

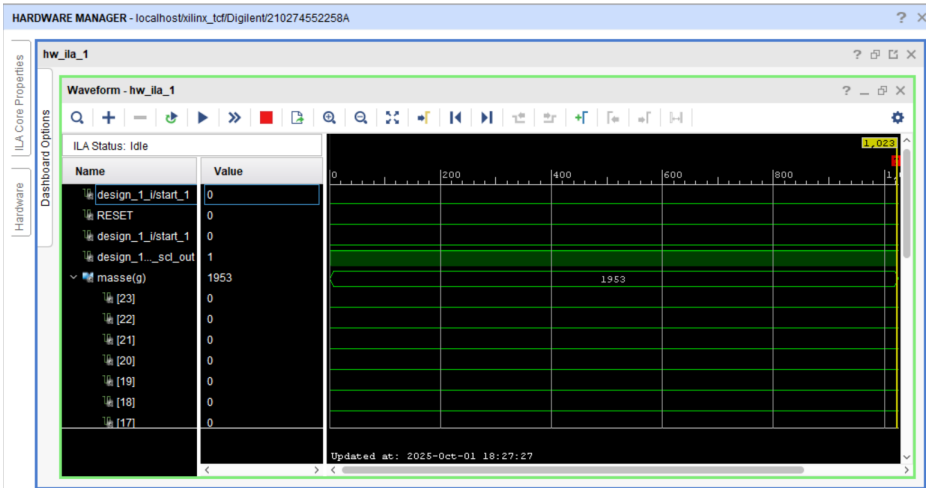


FIGURE E.1 – Capture du Test numérique de la balance I²C 1953g sur l'ILA