



UN I VERS ITÉ
LAVAL
FACULTÉ DES SCIENCES ET GÉNIE

Stage en génie mécanique II
GMC-2580 (NRC: 51133)
Été 2022
Baccalauréat en génie mécanique

Stagiaire en Robotique

Rapport de stage

Destinataire

Département des stages en milieu pratiques

Remise : Septembre 2022

Nicolas Sicard
111 246 676

Superviseur de stage : Alexis Christen

Résumé

Ce rapport présente le travail effectué par Nicolas Sicard, stagiaire en robotique, dans le cadre du stage de formation en entreprise à la compagnie Rupico pendant l'été 2022. Le stage consiste à concevoir une cellule robotique, ce qui comprend la conception du système, la sélection des composants, la programmation de la communication entre appareils et l'implémentation d'un logiciel de vision artificiel. Des projets secondaires réalisés par le stagiaire sont aussi survolés dans ce rapport, dont la conception d'un cric motorisé pour milieu clinique et un générateur de gouttelettes pour cuisine moléculaire.

Table des matières

Résumé.....	ii
Figures.....	iv
Introduction.....	1
Description du stage.....	2
Contexte et problématique.....	2
Tâches et travaux effectués	3
Rôle et contribution du stagiaire.....	5
Mandat : Logiciel de vision artificiel.....	5
Mandat : Communication entre appareils.....	6
Mandat : sélection de composants	8
Méthodologie.....	10
Développement et renforcement des compétences	11
Compétences communication.....	11
Compétences pratiques.....	11
Compétences théoriques.....	12
Mieux cibler le type de carrière envisagé.....	13
Conclusion	13
Bibliographie.....	A

Figures

Figure 1 Cellule robotique de petite taille conçu, assemblé et programmé par Nicolas Sicard.	3
Figure 2 Boîte électrique de la cellule robotique conçu assemblé et programmé par Nicolas Sicard.	3
Figure 1 Modèle numérique résultant de la numérisation du pied d'un patient à la clinique podiatre plus, utilisé comme exemple pour la programmation d'un logiciel de générateur de carte de disparité. L'illustration provient du logiciel Meshmixer d'Autodesk.[2]	4
Figure 2 Carte de disparité générée par le logiciel conçu et programmé par Nicolas Sicard au début de son stage chez Rupico à l'été 2022.	4
Figure 3 Modèle numérique 3D du cric motorisé conçu par Nicolas Sicard en 2022 chez Rupico.	4
Figure 4 Prototype conçu et fabriqué par Nicolas Sicard en 2022 chez Rupico d'un distributeur de gouttes pour cuisine moléculaire.	4
Figure 5 Prototype conçu et fabriqué par Nicolas Sicard en 2022 chez Rupico d'une valve à pincement motorisé.	4
Figure 8 Affichage rendu par le moniteur d'une image comportant des cibles repérées par le logiciel de vision artificiel programmé par Nicolas Sicard en 2022 chez Rupico.	6
Figure 9 Interface physique composée de 6 boutons.	7
Figure 10 Boîte électrique de la cellule robotique conçue et assemblée par Nicolas Sicard en 2022 chez Rupico. À gauche : mini-PC. À droite : PLC.	7
Figure 11 Diagramme de la configuration du logiciel Python de vision artificiel conçu et programmé par Nicolas Sicard en 2022 chez Rupico.	8
Figure 12 Diagramme de connexion pour la cellule robotique conçu, dessiné et assemblé par Nicolas Sicard en 2022 chez Rupico.	10

Introduction

Réalisé dans le cadre d'une formation en génie mécanique; parcours en robotique, ce second stage en milieu pratique vise l'intégration du stagiaire à une firme d'ingénierie mécanique afin de faire l'expérience de la réalisation de projets robotique dans un contexte professionnel. Le stagiaire Nicolas Sicard actuellement en dernière année du baccalauréat de génie mécanique, ayant préalablement fait un stage comme assistant de recherche à l'INRS, cherche à découvrir le monde de la conception et du développement de solutions.

Le stagiaire en robotique sera appelé à travailler sur le développement d'une nouvelle offre de service, la robotique légère. Il sera donc un pilier important pour développer de nouvelles fonctionnalités comme l'intégration de la vision et d'autres applications. Il est souhaité que le stagiaire acquière une excellente compréhension du robot, de son contrôle et des implications de son implantation en environnement contrôlé et manufacturier.

Parmi les activités prévues pour le stagiaire, celles qui occupent une portion importante de son stage sont la programmation du robot Dobot MG400 en utilisant le logiciel du fournisseur et d'autres logiciels, l'implémentation d'un système de vision artificiel pour reconnaître la présence d'objets, leur position, leur orientation et leurs dimensions. Une autre tâche est de faire l'incorporation de préhenseurs robotisés pour différentes applications comme la manipulation de produits alimentaires, domestiques ou électroniques. Le stagiaire est aussi appelé à concevoir et fabriquer des environnements de travail de manipulation, souvent nommé « Pick and place » selon différents scénarios.

Le stage est effectué sous la supervision de Alexis Christen ingénieur et cofondateur de Rupico et il s'étend du 2 mai au 12 août 2022. La mission de Rupico est de réduire la barrière d'accès à l'automatisation pour les petites entreprises manufacturières en utilisant des technologies existantes et de plus en plus abordables. L'objectif est de combler le manque grandissant de main-d'œuvre par l'automatisation de tâches simples et répétitives à l'aide de petits robots.

Tout d'abord, le rapport présente certains projets et tâches effectuées lors du stage, puis l'implication et les fonctions du stagiaire ainsi que la méthodologie utilisée et les compétences et connaissances acquises.

Description du stage

Contexte et problématique

Le stage s'effectue dans le contexte d'un démarrage de service, visant à permettre l'automatisation de petites entreprises manufacturières. Plusieurs scénarios où ce service est nécessaire ont été préalablement identifiés et servent de modèle pour la conception et le développement du système.

L'automatisation de procédés manufacturiers a plusieurs obstacles principaux. D'abord, le coût des installations amène une nécessité pour un volume important de production pour justifier l'investissement. La complexité des procédés nécessite des solutions personnalisées dont les entrées et sorties ont des limitations spécifiques à chaque système. Les coûts élevés et la spécificité des requis sont deux éléments qui rendent l'automatisation d'une chaîne de production dans sa totalité préférable à une automatisation partielle ou une implémentation progressive. Pour cette raison, les compagnies d'automatisations cherchent souvent à prendre en charge une chaîne manufacturière dans sa totalité, ce qui mène les coûts de base à la hausse. Les cellules robotisées les plus simples coûtent actuellement souvent bien au-delà de 100k\$ et les appareils qui composent ces cellules ont des performances excessives pour les besoins des petites entreprises. Pour ces raisons, la conception d'une cellule robotisée de petite taille ayant des performances modestes et une excellente accessibilité est la prochaine étape dans le monde de l'automatisation. Pour ces raisons, le développement de ce service est le mandat principal du stagiaire.

D'une autre part, le stage permet au stagiaire de s'impliquer dans plusieurs projets de plus petite taille pour répondre aux besoins spécifiques de divers clients. Cette approche permet au stagiaire d'atteindre régulièrement des objectifs tangibles tout en progressant continuellement le mandat principal. Ces différents projets amènent le stagiaire à travailler avec une variété de technologies, notamment, la programmation de traitement de modèle 3D de type stéréolithographique, des bibliothèques open source de traitements de données et d'images en python[1], la programmation de microcontrôleur, la conception de valves motorisée et le développement d'interfaces de contrôles.

Tâches et travaux effectués

Le projet principal du stagiaire consiste à concevoir une cellule robotique destinée à des entreprises manufacturières de petite taille. La cellule robotique servant d'exemple pour le développement consiste en un bras robotisé, un système de vision artificielle et un convoyeur. La tâche que la cellule doit accomplir est de repérer un objet sur convoyeur, le prendre et le déposer à un autre endroit. La simplicité de la tâche accentue l'importance d'une installation simple, d'un entretien minimal et d'une implémentation facile. Les tâches consistent d'abord à sélectionner des composants idéaux dont les fonctionnalités sont spécifiques et limitées aux besoins, qui sont abordables, remplaçables et vendus, et ensuite, à éliminer les dépendances d'écosystèmes en programmant un algorithme basé sur des bibliothèques code ouvert et des langages de programmation accessibles. Additionnellement, il faut éliminer le superflu en réduisant les opérations à l'essentiel, en segmentant et regroupant les fonctions du système et programmer des séquences d'opérations résilientes et autonomes en réduisant au minimum les interventions nécessaires au fonctionnement. La cellule robotique conçue, assemblée et programmée par le stagiaire est présentée à la figure 1 et la boîte électrique comprenant la plupart des systèmes est présentée à la figure 2.

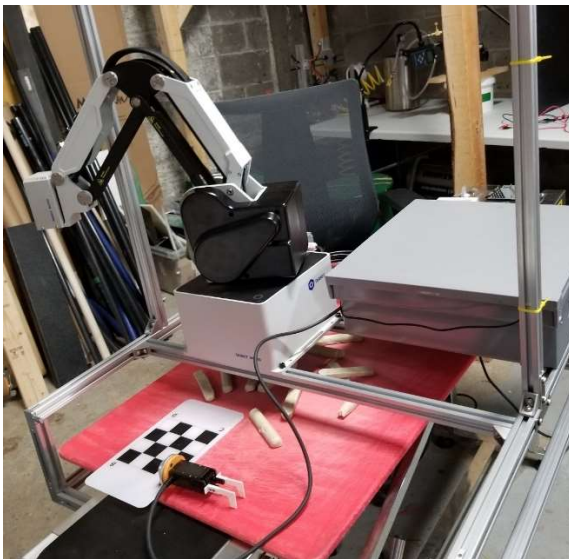


Figure 1 Cellule robotique de petite taille conçue, assemblée et programmée par Nicolas Sicard.

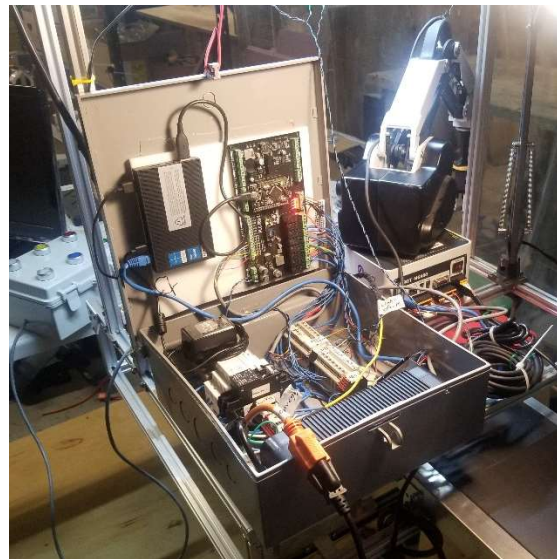


Figure 2 Boîte électrique de la cellule robotique conçue assemblée et programmée par Nicolas Sicard.

Les projets secondaires constituent une portion importante du stage. Ceux-ci incluent la programmation d'un logiciel produisant une carte de disparité comme présentée à la figure 4 à partir de modèles 3D comme montre la figure 3,

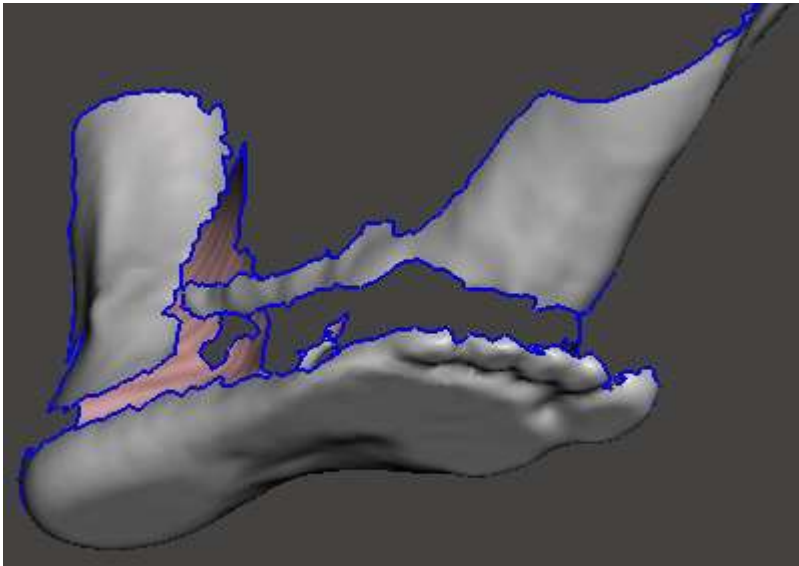


Figure 3 Modèle numérique résultant de la numérisation du pied d'un patient à la clinique podiatre plus, utilisé comme exemple pour la programmation d'un logiciel de générateur de carte de disparité. L'illustration provient du logiciel Meshmixer d'Autodesk.[2]



Figure 4 Carte de disparité générée par le logiciel conçu et programmé par Nicolas Sicard au début de son stage chez Rupico à l'été 2022.

la conception d'un cric motorisé pour milieux cliniques, illustré à la figure 5 et la conception d'une plateforme d'expérimentation pour génération de gouttelettes pour cuisine moléculaire, présentée à la figure 6. Ce prototype a également nécessité la conception et fabrication d'une valve à pincement motorisé, présenté à la figure 7.

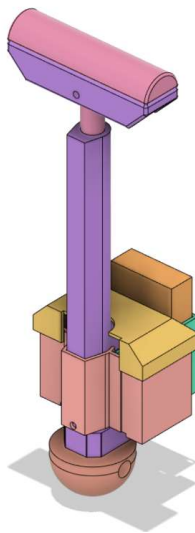


Figure 5 Modèle numérique 3D du cric motorisé conçu par Nicolas Sicard en 2022 chez Rupico.



Figure 6 Prototype conçu et fabriqué par Nicolas Sicard en 2022 chez Rupico d'un distributeur de gouttes pour cuisine moléculaire.



Figure 7 Prototype conçu et fabriqué par Nicolas Sicard en 2022 chez Rupico d'une valve à pincement motorisé.

Rôle et contribution du stagiaire

Le stagiaire agit au meilleur de ses connaissances et de ses capacités en tant que stagiaire en robotique. Il est attendu qu'il progresse au meilleur de ses compétences sur le projet principal, en demandant de l'assistance et de l'aide au besoin, en produisant des rapports de progression régulièrement et en demandant davantage de responsabilité à son gré. Le stagiaire est en mesure de gérer son propre temps, son environnement de travail et est en mesure de s'impliquer dans d'autres projets qui l'intéressent. Sa créativité et ses connaissances dans une grande variété de sujets sont régulièrement sollicitées pour aider l'avancement d'autres projets.

Mandat : Logiciel de vision artificiel

Puisque les logiciels de vision artificielle qui sont commercialement disponibles sont souvent inclus dans une suite de logiciels dispendieux et qu'ils sont restreints à certains appareils approuvés par le fournisseur, choisir un logiciel en particulier limite grandement les options pour le reste de la cellule. Certains logiciels ne fonctionnent qu'avec certaines caméras ou certains robots. D'autres viennent comme supplément à une panoplie d'outils de reconnaissance d'objet, d'identifications de défauts, de modélisations 3D et une multitude de fonctionnalités qui sont inutiles pour une cellule de base. Une différente option à ces produits est l'utilisation de bibliothèques à code ouvert de vision artificielles. Celles-ci permettent d'implémenter uniquement les fonctionnalités qui sont nécessaires et sont disponibles gratuitement.

Le travail effectué par le stagiaire est composé de plusieurs aspects. D'abord, la sélection et la configuration des bibliothèques de reconnaissances d'objet comme Scikit-Image et la programmation du traitement d'image et de la reconnaissance de contours avec python-opencv2, comme illustré à la figure 8. Un autre aspect est la programmation de la transformation de coordonnées et des algorithmes de calibration alignant les coordonnées des objets repérés par la caméra aux coordonnées de l'espace de travail et du robot. L'utilisation de la bibliothèque Matplotlib est utilisée pour illustrer le transfert de coordonnée, qui est composé d'une translation et d'une transformation matricielle.

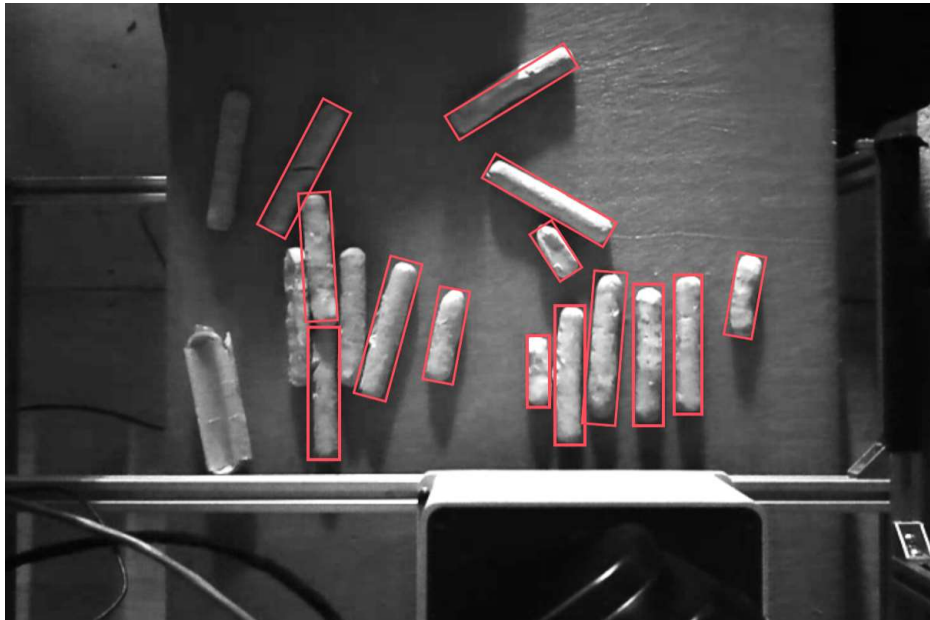


Figure 8 Affichage rendu par le moniteur d'une image comportant des cibles repérées par le logiciel de vision artificiel programmé par Nicolas Sicard en 2022 chez Rupico.

Une attention importante est portée à la documentation liée à l'investigation menée tout au long de la programmation, concernant les fonctionnalités et limites des différents modules et librairies disponibles, les décisions et avantages des modules sélectionnés ainsi que le potentiel de celles pouvant être intégrées ultérieurement.

Mandat : Communication entre appareils.

Afin de synchroniser les actions des différents appareils qui ont des limitations différentes, la communication est programmée de différentes façons. La communication des paramètres des cibles qui sont repérés par le logiciel de vision artificielle est envoyée au robot par communication TCP/IP. La communication entre Le PLC et l'ordinateur roulant la vision artificielle se fait par communication en série. Les commandes manuelles provenant de l'interface physique montrée à la figure 9 sont reçues par le PLC, visible à la figure 10 et transféré par signaux logiques au robot. De cette même façon, les signaux d'état du robot sont envoyés au PLC et transférés au logiciel de vision artificiel qui roule sur le mini-pc, également visible sur la figure 10.



Figure 9 Interface physique composée de 6 boutons.

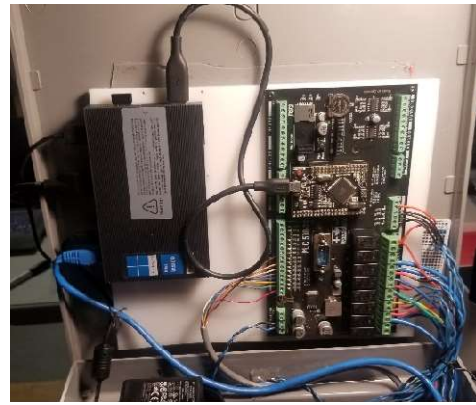


Figure 10 Boîte électrique de la cellule robotique conçue et assemblée par Nicolas Sicard en 2022 chez Rupico. À gauche : mini-PC. À droite : PLC.

Puisque la vision artificielle est au cœur de la majorité des besoins d'automatisation ciblé par ce produit, le logiciel Python est à la tête du système. Le stagiaire a programmé le logiciel comme plusieurs instances roulant en parallèle de façon à séparer les tâches dont le délai est important. Le diagramme hiérarchique des instances est visible à la figure 11. L'instance faisant la gestion de la caméra assure une bonne connexion et un traitement d'image en continu, alors que l'instance de communication attend une demande de la part du robot et transfère la demande à l'instance de reconnaissance. Cette dernière récupère la dernière image traitée par l'instance caméra et renvoie les coordonnées de la cible à l'instance de communication. Parallèlement, l'instance de communication peut recevoir des messages du PLC et gérer les erreurs de connexions sans interrompre le reste du logiciel. Une autre fonctionnalité est celle du moniteur web et du moniteur local, qui affiche en continu les images fournies par l'instance caméra et les objets reconnus par l'instance de reconnaissance d'image.

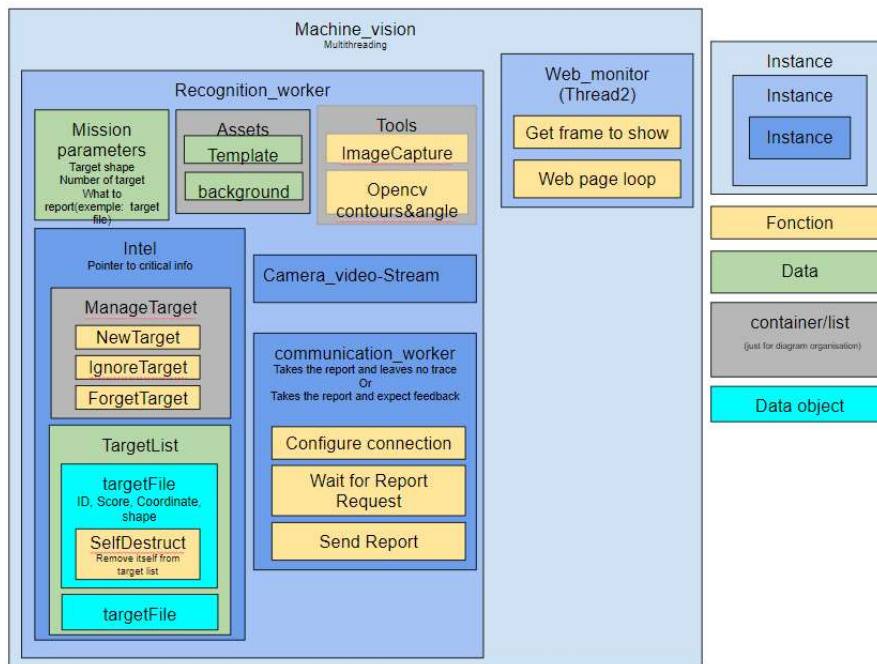


Figure 11
Diagramme de la configuration du logiciel Python de vision artificiel conçu et programmé par Nicolas Sicard en 2022 chez Rupico.

Le travail effectué consiste en la programmation des logiciels fonctionnant sur les différentes plateformes en utilisant des librairies open source. Le logiciel de vision artificielle est programmé en python avec les librairies python-opencv2[3], Flask[4], Numpy[5], Scipy[6] et Matplotlib[7]. Il communique au bras robotisé Dobot[8], dont le scripte est programmé en Lua[9] et avec le PLC, dont le microcontrôleur est un ATmega2560[10] programmé en Arduino. Le code du microcontrôleur utilise la librairie Blynkkk [11] publiée par Volodymyr Shymanskyi.

Mandat : sélection de composants

Pour favoriser l'implémentation de cellule robotique au sein de petites entreprises manufacturières, les composantes sélectionnées doivent être spécifiques à leur rôle. Toute fonctionnalité inutilisée nécessite plus d'espace, plus de coûts et parfois une gestion supplémentaire de ses fonctionnalités.

Tout au long de la conception de la cellule, la configuration entière du système était remise en question de façon à favoriser une configuration idéale et spécifique plutôt que de chercher à modulariser les segments ou à respecter une structure rigide arbitraire. Par exemple, l'ordinateur gérant la vision artificielle pouvait aussi bien être interne à l'ordinateur du bras robotisé, qu'être embarqué à une caméra intelligente, que chargé sur un ordinateur externe. Cette décision dépendait des performances nécessaires pour le logiciel, qui s'avère pouvoir fonctionner sur un simple Raspberry pi[12]. Vu

l'indisponibilité des Raspberry pi, n'importe quel ordinateur plus performant est adéquat et ainsi, un mini-ordinateur de 300\$ fut sélectionné[13].

Pour le contrôle des différents appareils, de leurs alimentations, de l'interface de commande et des signaux lumineux, un contrôleur logique programmable "PLC" est sélectionné. Un PLC fait essentiellement office de commutateurs électroniques, muni de logique très basique. Ce sont généralement des microcontrôleurs aux fonctionnalités limitées ayant une conception et une fabrication très robuste, idéal pour les milieux industriels. Les marques plus reconnues sont dispendieuses et nécessitent d'être programmé avec un logiciel propriétaire. Après réflexion, la conclusion fût qu'un microcontrôleur embarqué sur un module d'expansion de type PLC est fonctionnellement la même chose qu'un PLC, mais peut aussi bien être programmé comme un PLC que comme un microcontrôleur avec un langage Arduino[14], c++ et circuit python[15]. Cette option réduit significativement les coûts et facilite grandement l'implémentation, la maintenance et la modification d'une cellule d'abord parce que les ressources en lignes pour ces langages de programmation sont incroyablement plus nombreuses et accessibles et parce que les composantes ne sont pas barrées par propriété intellectuelle. Les microcontrôleurs populaires sont disponibles sous forme de modules de développement et prototypage. La grande communauté d'amateur et enthousiastes en automatisation amène une compétition importante entre les manufacturiers, qui mène à une réduction des prix significative.

Le travail exécuté par le stagiaire consiste à sélectionner les appareils nécessaires pour l'assemblage d'une cellule robotisé. La sélection se fait progressivement, de façon à tester les différentes limites des différentes configurations. Les éléments qui ont moins de risques d'être remplacés si un changement de structure survient sont obtenus et testés en premier. Ces éléments sont les boutons de l'interface physique, les lumières, le bras robotisé et l'outil préhenseur. Une fois le logiciel de vision artificiel suffisamment avancé pour évaluer les requis en performances, les autres éléments peuvent être sélectionnés et la structure du système peut être testée. La documentation de la structure et le diagramme de connexion sont aussi faits par le stagiaire. Le diagramme suivant est celui des connexions pour le système lors de la fin du stage.

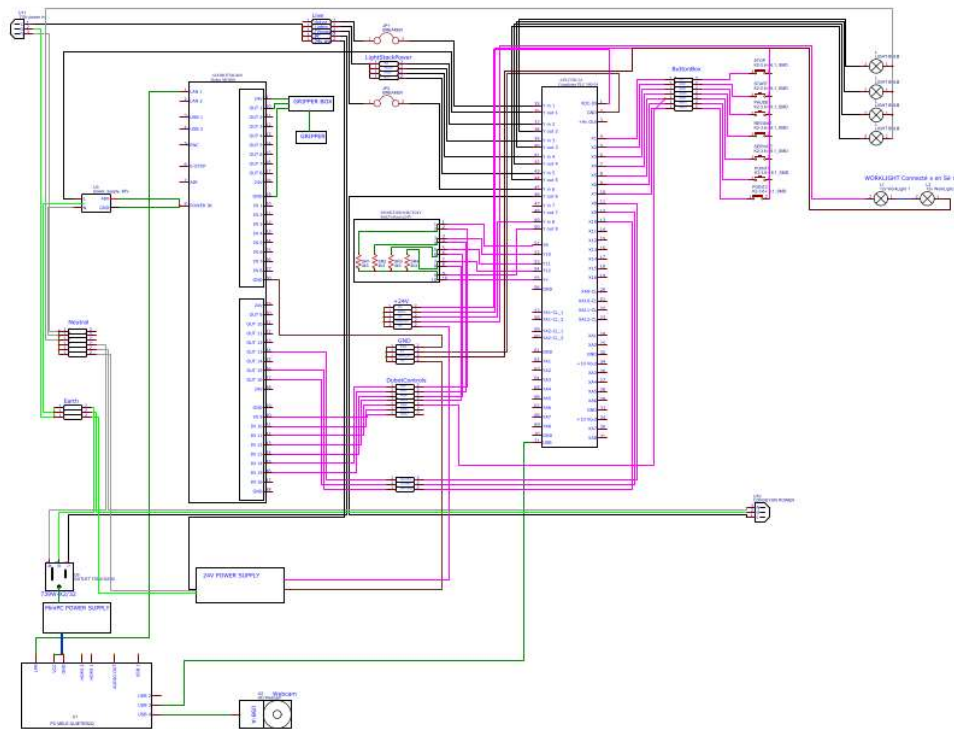


Figure 12 Diagramme de connexion pour la cellule robotique conçu, dessiné et assemblé par Nicolas Sicard en 2022 chez Rupico.

Méthodologie

La méthodologie utilisée pour la programmation du système est la méthode agile. Les différentes fonctionnalités sont ajoutées l'une à la suite de l'autre et à chaque étape, la structure du système est vérifiée et ajustée pour qu'elle soit idéale. En faisant ainsi, les différents modules ne sont pas considérés comme étant terminés tant que toutes les fonctionnalités ne sont pas implémentées. La flexibilité de cette méthode permet d'avancer rapidement en terrain inconnu en gardant les objectifs en vue et en réduisant le syndrome de dette technique[16], qui est le phénomène selon lequel attendre avant de faire une modification simple au code risque d'augmenter drastiquement la complexité des modifications à effectuer plus tard. Puisque le projet est essentiellement exécuté par le stagiaire, autant pour la programmation, la sélection de composantes, la conception électrique, l'assemblage et la configuration des appareils, la documentation de la progression et la communication des décisions sont abrégés. Ainsi l'investigation, les tests et la conception sont exécutés en séquence rapide avec une très grande fluidité.

En approchant la fin du stage, la structure globale du système est plus rigide et le stagiaire peut documenter et mettre en plan de façon plus détaillé, notamment la structure

du logiciel de vision artificielle et la structure électronique. L'attention aux détails et à l'élaboration des informations est réservée pour les composantes sélectionnées.

Développement et renforcement des compétences

Le stage a été une occasion pour le stagiaire de mettre à profit ses connaissances et compétences en prototypage et d'ajuster ses méthodes et habitudes de travail à un environnement professionnel. Les multiples projets lui ont également permis de mieux repérer ses forces et faiblesses ainsi que ses préférences en termes de tâches et responsabilités.

Compétences communication

Puisque le projet principal du stage en est un de développement et de conception, le stagiaire a dû adapter ses méthodes de travail. La nature autonome de l'environnement a permis de révéler certaines nécessités en termes de compétences théoriques et professionnelles. Il est essentiel que toute information, réflexion, idée, problème, option et solution soit systématiquement documentée par le stagiaire tout au long du développement. La documentation se fait principalement sur la plateforme en ligne confluence[17], qui permet une organisation efficace des documents et de l'information. Additionnellement, pour optimiser l'efficacité des rencontres et des discussions, le stagiaire doit préparer ses questions pour que le contexte soit clair et que les solutions ou options envisagées soient partagées. De mêmes manières, la progression du projet doit être présentée de façon ordonnée et concise pour en communiquer un portrait clair.

Compétences pratiques

Un des aspects de la conception qui est significativement différente dans le contexte de ce stage en comparaison aux projets personnels est la gestion des priorités. Le stagiaire a pu apprendre à mieux identifier à quelles étapes il est préférable de ralentir et prendre le temps nécessaire pour envisager toutes les options et éliminer au maximum tous les risques, ainsi que d'identifier quand il est préférable de maintenir la cadence et de se fier aux estimations pour simplement terminer une étape et tester le résultat. Par exemple, pendant la conception de valves motorisées, le prototype initial est conçu en s'inspirant d'un modèle récupéré en ligne[18] d'un projet à licence créative commune dont les dimensions ont d'abord été modifiées et adaptées puis éventuellement dont le modèle entier a été reconstruit en fonctions des nécessités. Tester le prototype initial a permis de repérer les

différents problèmes qui sont liés à ce type de mécanisme et des ajustements manuels sont faits pour valider les solutions potentielles. Les différentes variables sont ainsi découvertes et testées rapidement plutôt que de devoir faire une modélisation et analyse détaillée du mécanisme. Cette approche est préférable par sa rapidité et par le fait que les risques qui y sont liés, dans certaines circonstances, sont de faibles conséquences. Similairement, lors de la conclusion de certains projets, les cours échéanciers ont obligé le stagiaire à se concentrer sur terminer le projet et mettre de côté les détails et l'amélioration.

Compétences théoriques

Les compétences théoriques développées et améliorées par le stagiaire au cours du stage sont multiples. D'abord pour ce qui est de la programmation de système, il a pu développer et appliquer la logique et le rationnel en ce qui concerne l'organisation d'un système comprenant plusieurs séquences de logiciels distincts. L'orchestration des multiples tâches de façon harmonieuse et robuste est significativement plus complexe que les projets de programmation qu'il a pu réaliser précédemment dans ses projets personnels et au cours de son stage à l'INRS. Ces projets comportaient généralement que quelques séquences distinctes et n'interagissaient que très peu avec d'autres systèmes.

Du même coup, les différentes plateformes composant la cellule robotique sont programmées en différents langages, dont le langage Lua, qui est le langage des scripts pour le bras robotisé. De ce fait, le stagiaire a pu apprendre ce langage et l'utiliser. La communication entre appareil par TCP/IP et série était également nouvelle pour lui et il a pu apprendre à les programmer sur Lua, Arduino et Python. En supplément, le logiciel Python qu'il a programmé nécessite des processus parallèles et il a également appris à utiliser la librairie Threading[19].

Une autre compétence que le stagiaire a pu développer et appliquée au cours du projet de cellule robotisé est en lien avec les algorithmes de vision artificielle. Il a développé des connaissances en programmation de reconnaissance d'image avec Scikit-Image[20] qui permettent d'identifier des objets similaires à un patron à l'aide d'intelligence artificielle. Il a développé des compétences en programmation pour la localisation d'objets avec python-opencv2[3] qui permettent de déterminer avec précision la position et l'orientation d'objets et il a aussi élargi ses connaissances en programmation de transformation d'espace de coordonnées. Ces dernières lui ont permis de programmer des algorithmes, principalement avec numpy[5], qui calculent la position et l'orientation

réelle d'un objet à partir de sa position dans l'image observé par la caméra. Cet algorithme permet de faire la calibration entre la caméra et un bras robotisé.

Mieux cibler le type de carrière envisagé

L'opportunité de faire un stage chez Rupico a permis au stagiaire de confirmer le type d'environnement de travail idéal pour lui. Le haut niveau d'autonomie dans l'exécution des projets et la grande confiance qui lui est attribuée, en combinaison à une assignation importante, mais raisonnable de responsabilités, lui permettent d'épanouir sa créativité et son savoir-faire. L'environnement est excellemment propice à l'amélioration continue et naturelle en n'ayant pratiquement aucun recours aux pressions typiques liées à la gestion de projets comme les échéanciers et le manque sévère de budget ou de ressources.

L'une des caractéristiques les plus remarquables de ce stage est l'ouverture à un profil spécialement généraliste comme celui du stagiaire. La carrière idéale pour lui est sans doute l'une où il est constamment sollicité pour ses connaissances et compétences variées et où sa grande curiosité est grandement valorisée. Rupico est une manifestation exemplaire du type d'entreprise permettant une telle carrière.

Conclusion

En bref, ce stage en robotique chez Rupico a permis au stagiaire de lancer la conception d'une cellule robotique abordable et accessible. Il a pu y développer ses compétences en communication, en programmation et en conception. Plus spécifiquement, il a conçu une cellule robotique, allant de la sélection des composants et appareils, leurs configurations, la programmation des logiciels qui y sont chargés et la communication qui les synchronisent. Ce projet est, à la fin du stage, dans un état fonctionnel qui nécessite encore des modifications et optimisations, ainsi que l'ajout de quelques fonctionnalités. Dans cette condition, la cellule est en mesure de repérer les objets qui sont sur un convoyeur, les saisir et les déposer à un autre endroit. Parmi les étapes à réaliser pour que la cellule soit opérationnelle, il faut implémenter l'alimentation du convoyeur lorsque le robot a terminé d'une section et implémenter un préhenseur plus approprié que celui utilisé pour le développement. Le stagiaire a aussi eu l'opportunité de concevoir deux autres appareils, un cric motorisé et un générateur de gouttelettes, ainsi qu'un logiciel générant des cartes de déplacements à partir de modèles 3D. Ce stage a permis au stagiaire de confirmer l'environnement de travail idéal pour son caractère, ses objectifs et ses préférences.

Bibliographie

[1] “Python,” Python.org [Online]. Available: <https://www.python.org/>. [Accessed: 21-Aug-2022].

[2] “Meshmixer” [Online]. Available: <https://help.autodesk.com/view/MSHMXR/2019/ENU/?guid=GUID-9B48D29A-C62F-4E13-8AB4-54E19F1F9875>. [Accessed: 21-Aug-2022].

[3] Bradski, G., 2000, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools.

[4] Grinberg, M., 2018, *Flask Web Development: Developing Web Applications with Python*, O’Reilly Media, Inc.

[5] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E., 2020, “Array Programming with NumPy,” *Nature*, **585**(7825), pp. 357–362.

[6] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik, D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G.-L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J. L., de Miranda Cardoso, J. V., Reimer, J., Harrington, J., Rodríguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M.,

Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, P., McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T. J., Robitaille, T. P., Spura, T., Jones, T. R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y. O., and Vázquez-Baeza, Y., 2020, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nat Methods*, **17**(3), pp. 261–272.

[7] Hunter, J. D., 2007, “Matplotlib: A 2D Graphics Environment,” *Comput. Sci. Eng.*, **9**(3), pp. 90–95.

[8] “DOBOT MG400: Lightweight Desktop Collaborative Robot,” Dobot Store [Online]. Available: <https://store.dobot.cc/products/dobot-mg400-robotic-arm>. [Accessed: 21-Aug-2022].

[9] Ierusalimsky, R., 2006, *Programming in Lua*, Roberto Ierusalimsky.

[10] “Atmel-2549-8-Bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.Pdf.”

[11] Shymansky, V., 2015, “Blynkkk/Blynk-Library” [Online]. Available: <https://github.com/blynkkk/blynk-library>. [Accessed: 20-Aug-2022].

[12] Ltd, R. P., “Raspberry Pi 4 Model B,” Raspberry Pi [Online]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. [Accessed: 21-Aug-2022].

[13] “MeLE Mini PC Quieter2Q Fanless Celeron J4125 Windows 11 Pro Micro Computer 8GB 128GB Small Desktop Servers with Ethernet PXE Dual HDMI 4K Unlock BIOS on Industrial Astrophotography Media Office : Amazon.ca: Electronics” [Online]. Available: <https://www.amazon.ca/Quieter2-Computer-Supports-Dual-Screen-Ethernet/dp/B08ZXXJL6N>. [Accessed: 21-Aug-2022].

[14] Banzi, M., “Arduino” [Online]. Available: <https://www.arduino.cc/>. [Accessed: 21-Aug-2022].

[15] “CircuitPython” [Online]. Available: <https://circuitpython.org/>. [Accessed: 21-Aug-2022].

[16] Trnka, D., 2018, “How to Manage Technical Debt,” Ministry of Programming—Technology.

[17] Atlassian, “Confluence - Team Collaboration Software,” Atlassian [Online]. Available: <https://www.atlassian.com/software/confluence>. [Accessed: 21-Aug-2022].

[18] MichStvs, “Pinch Valve Powered by Servo,” thingiverse [Online]. Available: <https://www.thingiverse.com/thing:5367793>. [Accessed: 21-Aug-2022].

[19] “Threading — Thread-Based Parallelism — Python 3.10.6 Documentation” [Online]. Available: <https://docs.python.org/3/library/threading.html>. [Accessed: 21-Aug-2022].

[20] van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T., 2014, “Scikit-Image: Image Processing in Python,” *PeerJ*, **2**, p. e453.