

Introduction aux éléments de contrôle / supervision d'une installation industrielle moderne à base de matériel et logiciels ouverts

Jason Chevie^{1,2}, Nicolas Lardenois¹, Laurent Vermeiren^{1,2}
jason.chevie@uphf.fr, nicolas.lardenois@uphf.fr, laurent.vermeiren@uphf.fr

Adresse : ¹ IUT de Valenciennes, département Génie Électrique et Informatique Industrielle
² UPHF, CNRS, UMR 8201 - LAMIH, F-59313 Valenciennes, France

RÉSUMÉ : Cet article présente les éléments d'un dispositif pédagogique et les activités mises en place dans le cadre de séances de SAÉ (Situation d'Apprentissage et d'Évaluation) au 3^e semestre du parcours Automatique et Informatique Industrielle du Bachelor Universitaire de Technologie (BUT) spécialité Génie Électrique et Informatique Industrielle à l'IUT de Valenciennes. Ce dispositif a pour objectif d'introduire de manière simplifiée aux étudiants les différents éléments qu'ils pourraient être amenés à mettre en œuvre dans une situation professionnelle où ils auraient à concevoir la partie contrôle / supervision complète d'une installation industrielle. Le dispositif présente l'avantage d'individualiser le travail de chaque étudiant tout en restant à coût raisonnable en utilisant une base de cartes Raspberry Pi et Arduino et des logiciels gratuits et ouverts à la place de matériel ou logiciels industriels onéreux. Les activités proposées permettent de la part des étudiants une mobilisation de connaissances pluridisciplinaires tout en maintenant un fil directeur autour du dispositif. L'article décrit également des extensions possibles d'activités à mettre en œuvre avec ce dispositif si le temps le permet ou pour donner une coloration plus proche du parcours Électronique et Systèmes Embarqués. Les retours étudiants informels et à travers un questionnaire anonyme sont également présentés et des pistes d'amélioration des activités sont proposées.

Mots clés : dispositif pédagogique, retour d'expérience, Raspberry PI, automatisation, SCADA, automatique.

1 INTRODUCTION

Les activités pédagogiques pratiques se rapprochant du contexte industriel ne sont pas toujours faciles ni rapides à mettre en place, entre autres pour des raisons budgétaires liés aux coûts inévitables du matériel et logiciels industriels et au temps de développement nécessaire pour concevoir ces activités. Elles sont pourtant indispensables dans de nombreuses formations technologiques ayant un lien fort avec le monde professionnel, comme c'est par exemple le cas avec le Bachelor Universitaire de Technologie (BUT), et en particulier dans le cadre de ce document celui de Génie Électrique et Informatique Industrielle (GEII). La réforme du BUT a conduit à un fort besoin en activités prenant la forme de projets, notamment pour la mise en place des Situations d'Apprentissage et d'Évaluation (SAÉ), comme ce fut déjà le cas pour la mise en place de la première année de formation [1]. Ces SAÉ doivent permettre aux étudiants de mobiliser tout un ensemble de ressources acquises depuis le début de leur formation à travers des mises en contexte proches de situations professionnelles leur permettant de développer leurs compétences individuelles et d'en permettre la validation par l'équipe pédagogique.

Ce document présente un projet de SAÉ complet mis en place au troisième semestre pour des étudiants en parcours Automatique et Informatique Industrielle (AII) du BUT GEII à l'Institut Universitaire de Technologie de Valenciennes et dont l'objectif est de permettre aux étudiants d'aborder un ensemble de thématiques rencontrées dans le contexte de l'automatisation des

systèmes industriels, tout en utilisant des solutions ouvertes peu onéreuses à la place du matériel et des logiciels industriels.

Le document est organisé comme suit : la section 2 introduit le contexte général du projet avec les ressources devant être mobilisées par les étudiants ainsi que le matériel utilisé. La section 3 détaille les logiciels utilisés. Les activités pratiques mises en place pour le projet sont présentées dans la section 4 avec d'autres activités supplémentaires également réalisables avec le système. La section 5 fait un bilan du déroulement du projet et présente un premier retour sur la perception des activités par les étudiants.

2 CONTEXTE GÉNÉRAL

2.1 Contexte

Une première contrainte choisie pour le développement de ce projet est de faire en sorte que les étudiants puissent travailler individuellement et tous en parallèle, permettant ainsi à chacun de manipuler pendant un maximum de temps et de réellement participer à chaque étape du projet, tout en facilitant la validation individuelle des compétences. La deuxième contrainte, liée à la première, est de faire en sorte de garder un coût total réduit mais en évitant cependant autant que possible de réduire les coûts par de la mutualisation d'équipements. Cela risquerait en effet d'entraîner soit un embouteillage au niveau de l'accès individuel aux éléments communs à tous les étudiants, soit la nécessité

de faire des groupes, rendant l'individualisation des évaluations des compétences plus difficile.

La SAÉ devant aussi faire intervenir un maximum de notions vues dans différents cours ou ressources du parcours AII, le choix d'activités pour le projet s'est tourné vers la découverte et la réalisation progressive d'une solution complète de pilotage et de supervision d'un système simple. Les solutions retenues pour la mise en œuvre du projet sont inspirées de [2] et ont été complétées et adaptées au matériel disponible. Comme illustré sur la figure 2, l'ensemble est conçu pour représenter une partie opérative à état continu pilotée par un automate par l'intermédiaire d'un premier réseau de terrain, un système de SCADA (Supervisory Control And Data Acquisition) gérant l'acquisition de données depuis l'automate via un autre réseau et un poste de supervision pour fournir une interface Homme-Machine (IHM). Le système permet donc d'introduire dans un ensemble cohérent à la fois des éléments d'automatisme par la programmation de l'automate, de réseaux industriels par la gestion de différentes communications, d'automatique par l'asservissement de la partie opérative et de supervision par la gestion du système SCADA et la réalisation de l'IHM.

2.2 Matériel commun

2.2.1 Partie opérative : Système bille sur table

Comme illustré sur les figures 1 et 2, la partie opérative utilisée est ici une plateforme d'équilibrage de bille sur une table inclinée (Acrome Robotics [3]). Elle constitue le seul élément mis en commun pour tous les étudiants. Cette table possède une surface tactile résistive permettant de mesurer la position 2D de la bille. Une carte d'interface intégrée à la table gère la calibration de la surface et la conversion des coordonnées vers une liaison USB. L'inclinaison de la table suivant les deux directions est contrôlée via un système bielle-manivelle par deux servomoteurs alimentés par une carte de conversion de puissance. À noter que ce système était déjà présent et inutilisé au département GEII de l'IUT de Valenciennes lors de la conception de cette SAÉ. N'étant pas le sujet principal des activités proposées, il pourrait être remplacé par une autre partie opérative

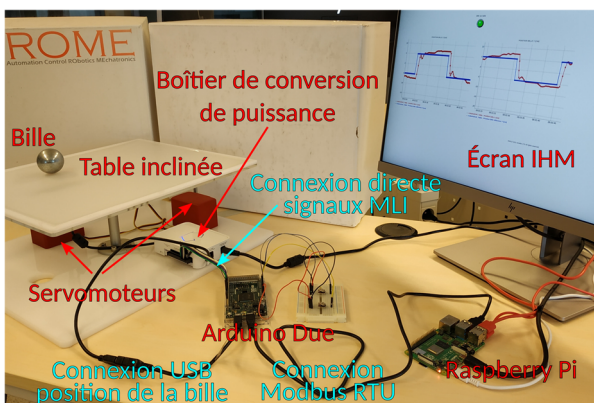


fig 1 : Présentation des éléments du système complet.

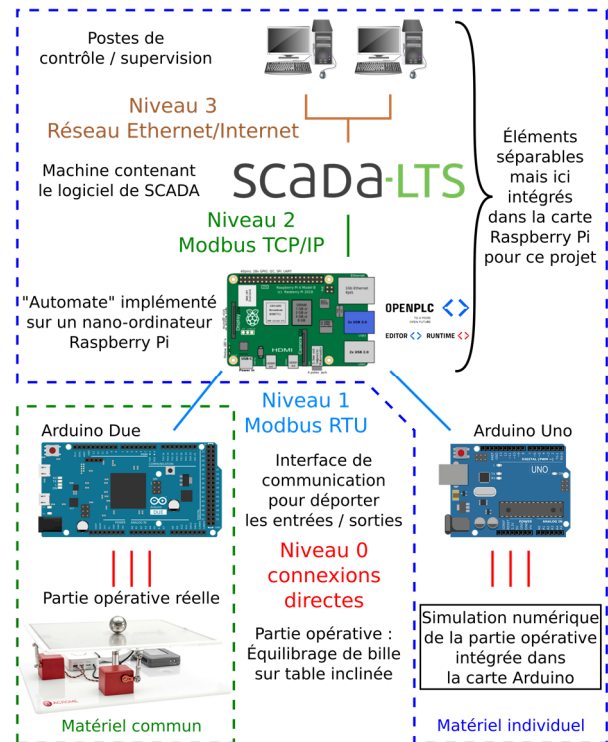


fig 2 : Organisation hiérarchique des éléments du système et des communications suivant le modèle de Purdue.

disponible, voire faire l'objet d'une réalisation par les étudiants dans le cadre d'un projet indépendant.

2.2.2 Interface de communication : Arduino Due

L'Arduino Due [4] est une carte à microcontrôleur de la famille ARM 32 bits. Elle possède un port USB utilisé pour la programmation et pour la communication, un second port USB pouvant jouer le rôle d'hôte USB, ainsi que de nombreuses broches d'entrées / sorties numériques et analogiques. Cette carte est programmée pour jouer le rôle d'une interface Modbus RTU pour la partie opérative afin que l'ensemble se comporte comme un module comportant des entrées / sorties déportées via un bus de terrain. Le protocole Modbus [5] est couramment utilisé dans les applications industrielles d'automatisme pour permettre à différents appareils d'échanger des informations de processus. L'interface Modbus RTU de la carte Arduino donne accès à :

- 5 entrées numériques, 4 sorties numériques, 6 entrées analogiques et 3 sorties analogiques associées chacune à une des broches de la carte ;
- 1 entrée numérique indiquant si la bille est présente sur la table ;
- 2 entrées analogiques sur 16 bits représentant les coordonnées de la bille sur la table ;
- 2 sorties analogiques sur 16 bits contenant les consignes d'angle des servomoteurs.

Le port USB hôte de la carte est utilisé pour recevoir les données de position de la bille depuis la liaison USB de la table grâce à la bibliothèque logicielle USBHost [6], l'autre port USB est utilisé pour faire passer la

communication Modbus RTU avec la partie automate et deux sorties numériques (sur les broches 8 et 9) sont utilisées pour générer les signaux en modulation de largeur d'impulsion (MLI) servant de signaux de commande pour les servomoteurs.

2.3 Matériel individuel

Afin de garantir une progression indépendante de chaque étudiant dans son projet, chacun dispose du matériel individuel détaillé dans la suite.

2.3.1 Poste principal : Raspberry Pi

La carte Raspberry Pi [7] est un nano-ordinateur de petite taille embarquant un système d'exploitation Linux (Raspbian) et possédant des broches d'entrées / sorties numériques en plus des ports informatiques classiques permettant de connecter différents périphériques. Elle est couramment utilisée dans de nombreux projets ouverts grâce à son faible coût. Chaque étudiant dispose durant tout le projet d'une carte Raspberry Pi 4 dédiée avec un clavier, une souris et un écran. Les broches sont toutes reliées sur une carte de prototypage (breadboard) afin de simplifier la réalisation de connexions avec des composants de test. Elle est utilisée dans ce projet pour assurer la quasi-totalité des fonctionnalités nécessaires au contrôle / supervision du système, se substituant donc aux éléments industriels qu'elle émule :

- Poste de programmation, à la fois pour le programme automate, les éléments de SCADA et l'IHM ;
- Exécution de la partie automate, comprenant la loi de commande pour stabiliser la bille sur la table ;
- Exécution de la partie SCADA ;
- Affichage de l'IHM.

L'ensemble des éléments logiciels utilisés sur cette carte est détaillé dans la section 3.

2.3.2 Interface de communication et simulation de la partie opérative : Arduino Uno

L'Arduino Uno [4] est une carte à microcontrôleur de la famille ATmega. Elle possède un port USB utilisé pour la programmation et pour la communication, ainsi que des broches d'entrées / sorties. Chaque étudiant dispose d'une carte programmée avec une simulation en temps réel reproduisant le comportement dynamique du système d'équilibrage de bille sur table. Les entrées (commandes des angles des servomoteurs) et les sorties (positions 2D de la bille) de la simulation sont accessibles depuis l'extérieur via une interface Modbus RTU passant par la liaison USB. Cette interface est programmée pour être exactement similaire à celle de la carte Arduino Due utilisée avec la plateforme réelle dans la section 2.2.2. Cela permet aux étudiants de réaliser et de tester l'ensemble de leur projet comme s'ils étaient connectés au système réel sans risques et en toute autonomie grâce à la simulation avant de simplement venir se brancher sur la carte Arduino Due pour réaliser les tests finaux sur le système réel.

2.3.3 Matériel divers

Du matériel électronique (résistances, LED, boutons poussoir, fils) est également disponible afin de réaliser des petits circuits de test câblés sur breadboard utilisant directement les broches des cartes Raspberry Pi et/ou Arduino Uno pour tester la gestion des entrées / sorties.

3 LOGICIELS

Plusieurs logiciels ouverts fonctionnant sur le système d'exploitation de la carte Raspberry Pi sont utilisés pour réaliser les différentes activités nécessaires à la programmation, à l'exécution et à la vérification de l'ensemble des éléments intervenant dans le système.

3.1 Utilitaire EasyModbus

Une petite interface graphique, illustrée sur la figure 3, codée en langage Python et basée sur la bibliothèque EasyModbus [8] est fournie aux étudiants afin de pouvoir étudier le protocole ModbusRTU et tester la communication entre la carte Raspberry Pi (maître) et la carte Arduino (esclave). Cette interface permet aux étudiants de choisir les paramètres de communication (débit, bit de parité, longueur du bit de stop), de construire des trames de requêtes à envoyer et d'observer les trames de réponse reçues.

3.2 OpenPLC Runtime et OpenPLC Editor

OpenPLC est une solution ouverte permettant de réaliser et d'exécuter des programmes automate sur diverses plateformes, telles que sur des cartes Arduino ou des cartes Raspberry Pi. Il est divisé en deux parties : OpenPLC Runtime pour l'exécution et OpenPLC Editor pour la programmation. Pour ce projet de SAE les deux sont exécutées sur la carte Raspberry Pi, permettant aux étudiants de réaliser et tester leur programme automate.

3.2.1 OpenPLC Editor

L'éditeur de OpenPLC, visible sur la figure 4, permet de créer les programmes automate. Les langages de programmations disponibles sont ceux de la norme IEC 61131-3 classiquement utilisés pour la programmation d'automates industriels, comme les langages graphiques à contacts (LD : Ladder Diagram), par blocs (FBD : Function Block Diagram) ou par Grafcet (SFC : Sequential Function Chart) et les langages textuels en liste d'instructions (IL : Instruction List) ou en texte structuré (ST : Structured Text).

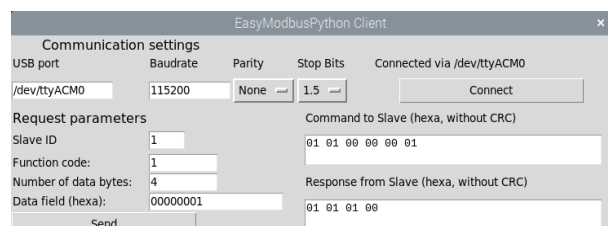


fig 3 : Interface graphique de surcouche sur la bibliothèque EasyModbus permettant aux étudiants d'étudier la communication ModbusRTU.

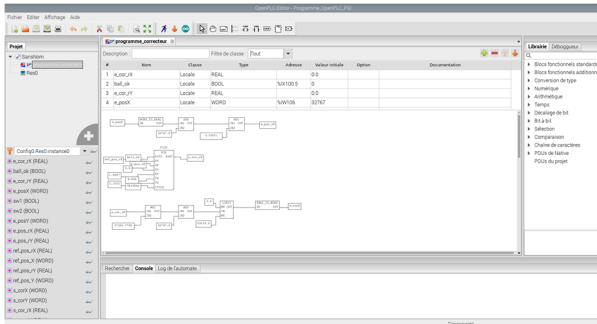


fig 4 : Interface de programmation de l'éditeur OpenPLC Editor, ici en langage Function Block Diagram.

Une simulation du programme est également disponible, permettant de vérifier son bon fonctionnement avant de lancer son exécution sur le système réel. À noter que OpenPLC Editor n'a pas été conçu initialement pour fonctionner sur les cartes Raspberry Pi. L'installation est cependant possible sur Raspbian à partir de la branche de développement du dépôt Git du projet [9]. Dans ce cas la simulation est indisponible et oblige les étudiants à tester leur programme directement, ce qui n'a pas posé de souci particulier dans le cadre de ce projet mais pourrait nécessiter l'utilisation d'ordinateurs standards dans un autre contexte si la simulation est nécessaire.

3.2.2 OpenPLC Runtime

Une fois le programme automate réalisé et compilé dans OpenPLC Editor, il peut être transféré sur la plateforme matérielle choisie et OpenPLC Runtime gère alors l'exécution du programme. Sur les plateformes suffisamment avancées (avec système d'exploitation) une interface Web est disponible avec plusieurs onglets permettant d'accéder à des fonctionnalités utiles pour le développement. Il est par exemple possible de configurer des esclaves communiquant sur un réseau de terrain pour que leur entrées / sorties soient scrutées périodiquement pendant l'exécution du programme automate. Ces entrées / sorties déportées sont alors accessibles directement via un espace d'adressage dédié dans le programme automate. Dans le cadre de ce projet les étudiants sont donc amenés à configurer l'interface ModbusRTU de la partie opérative (simulée avec les cartes Arduino Uno ou réelle avec la carte Arduino Due) afin de pouvoir implémenter le pilotage de la table inclinée grâce aux entrées / sorties déportées.

Une interface de supervision basique est également disponible dans l'interface Web, permettant aux étudiants de visualiser l'état de fonctionnement de leur programme, comme illustré sur la figure 5, avant même de développer leur propre écran de supervision.

Des réglages supplémentaires sont également disponibles pour activer des interfaces de communication supplémentaires utilisant des protocoles réseaux classiques de l'industrie pour ouvrir l'accès aux données d'entrées / sorties de l'automate, par exemple via une interface Modbus TCP/IP. Cela permet de mettre en place facilement une remontée de données depuis un autre appareil du réseau pour l'aspect supervision.

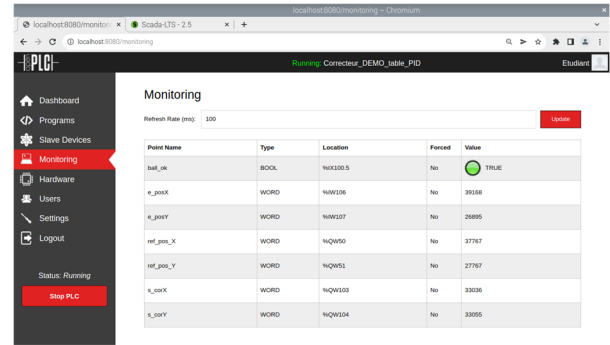


fig 5 : Interface du Runtime OpenPLC, ici sur la vue de supervision durant l'exécution du programme automate.

3.3 Scada-LTS

Scada-LTS [10] est une solution ouverte multi-plateforme permettant de mettre en place un système SCADA pour l'acquisition de données et la supervision de système. Elle fonctionne uniquement via une interface Web permettant de gérer les différentes fonctions nécessaires à un système de SCADA, comme :

- Les accès utilisateurs ;
- Les alarmes ;
- Les bases de données pour le stockage ;
- Le paramétrage de l'acquisition des données provenant de différentes sources via des protocoles spécifiques, comme par exemple Modbus TCP/IP ;
- La programmation et l'affichage d'une IHM graphique simple pour l'aspect contrôle / supervision, comme illustré sur la figure 6.

L'accès à l'interface Web et l'accès aux données de la partie automate étant possibles par le réseau, il est envisageable de faire tourner le serveur SCADA soit directement sur la carte Raspberry Pi soit sur un poste séparé. Les autres postes de supervision nécessitant uniquement un accès à l'IHM via un navigateur Web peuvent aussi être séparés du serveur SCADA.

Dans le cadre du projet, il a été choisi de faire fonctionner toutes les activités directement sur la carte Raspberry Pi afin d'éviter la multiplication des postes informatiques pour les étudiants.



fig 6 : Exemple d'IHM de supervision simple sur Scada-LTS affichant un voyant pour indiquer la présence de la bille sur la table et des courbes de mesures des coordonnées de consigne et réelles de la bille sur la table.

4 ACTIVITÉS

4.1 Activités mises en place pour le projet

Afin d'amener progressivement les étudiants à réaliser le contrôle / supervision du système complet sans se perdre dans la complexité des nombreux éléments à étudier et à paramétrer, 8 séances de 4 heures de projet en présentiel étalées sur 8 semaines avec des activités guidées sont fixées. La quasi-totalité des séances fait l'objet d'un livrable que les étudiants doivent rendre afin de permettre l'évaluation de leurs compétences via des grilles de notation dont ils ont connaissance.

4.1.1 Découverte des logiciels

Deux séances de découverte des logiciels sont d'abord effectuées, avec l'utilisation progressive des différentes fonctionnalités des environnements OpenPLC et Scada-LTS. Les premières séances font uniquement appel aux brochures d'entrées / sorties disponibles sur la carte Raspberry Pi, sans faire de lien avec la partie opérative finale. Les brochures sont reliées à des boutons poussoir pour les entrées et à des LED pour les sorties, et le schéma de câblage de ces éléments sur la plaque de prototypage est laissé à l'initiative des étudiants afin qu'ils mobilisent leurs connaissances en électronique. Les premiers programmes automate à réaliser sont également simples afin de permettre aux étudiants de prendre en main sans difficultés la programmation avec OpenPLC Editor, l'environnement d'exécution du programme avec OpenPLC Runtime, le paramétrage du système de SCADA pour la récupération des données via Modbus TCP/IP et la réalisation d'une petite IHM.

4.1.2 Ajout d'un esclave Modbus RTU

Un fois les outils découverts, la connexion avec des entrées / sorties déportées sur un esclave Modbus RTU est étudiée. L'esclave est ici une carte Arduino Uno reliée à la Raspberry Pi par un câble USB, comme décrit dans la sous-section 2.3.2. Une première séance d'analyse du protocole Modbus RTU est effectuée durant laquelle les étudiants doivent proposer un protocole de test pour déterminer les capacités de communication complètes de l'interface Modbus RTU de l'esclave grâce à l'utilitaire présenté dans la sous-section 3.1 : débits, bits de parité et longueur de bit de stop supportés, codes fonction Modbus supportés, tailles des espaces d'entrées / sorties disponibles pour les différents types de données échangeables.

La séance suivante consiste à configurer les caractéristiques de l'esclave dans OpenPLC Runtime et à réadapter les programmes, SCADA et IHM des séances de découverte pour utiliser les broches d'entrées / sorties de l'esclave Arduino à la place de celles de la carte Raspberry Pi. Une fois cette séance terminée les étudiants sont censés être plutôt familiers avec les différents logiciels et peuvent aborder en autonomie l'implémentation avec le système de table inclinée.

4.1.3 Étude d'un correcteur pour la table inclinée

Afin de stabiliser la bille sur la table, un correcteur adéquat doit être implémenté. Pour simplifier au maximum l'étude de la conception de ce correcteur, compte-tenu du placement assez précoce de cette SAE dans la formation, le modèle continu du système a été simplifié en découplant le contrôle de chaque coordonnée de la bille et en ignorant la dynamique des servomoteurs. Le système se réduit alors à deux doubles intégrateurs indépendants (un pour chaque coordonnée de la bille / servomoteur). Les correcteurs à étudier sont limités à un correcteur proportionnel et un correcteur proportionnel-dérivé, permettant de garder des systèmes bouclés d'ordre 2 dont les performances théoriques peuvent être étudiées assez facilement. Après deux séances de conception du correcteur continu permettant de respecter un cahier des charges imposé aux étudiants, la conversion en correcteur numérique implémentable dans le programme automate est totalement guidée. Finalement les étudiants disposent alors de tous les éléments nécessaires pour programmer l'ensemble de contrôle / supervision, effectuer des tests sur la simulation du système et évaluer les performances sur le système réel lors des deux dernières séances.

4.2 Autres activités envisageables

Dans le cadre de ce projet, les cartes Arduino étaient fournies préprogrammées aux étudiants, cependant la programmation de ces cartes pourrait faire l'objet d'activités supplémentaires pour aborder les aspects de programmation de microcontrôleur, par exemple avec l'implémentation de la simulation de la partie opérative sur les cartes Arduino Uno (activité plutôt orientée informatique générale avec une programmation proche du langage C) ou celle de la génération des signaux MLI de pilotage des servomoteurs sur la carte Arduino Uno ou Due (activité plutôt orientée informatique embarquée (vue et développée au semestre 2) avec l'étude des registres du microcontrôleur).

Comme mentionné précédemment la réalisation d'une partie opérative différente pourrait également être rajoutée en phase préliminaire à ce projet, permettant éventuellement à chaque étudiant de disposer d'une plateforme individuelle.

En fonction du niveau de formation des étudiants, l'étude du correcteur pourrait aussi être plus poussée. L'étude théorique pourrait aussi être remplacée par une étude numérique, en général plus appréciée par les étudiants, par exemple en utilisant les outils présents dans Matlab / Simulink si le logiciel est disponible.

5 BILAN DE LA MISE EN PLACE ET RETOURS ÉTUDIANT

5.1 Limites du dispositif

Étant la première mise en place de cette SAE, plusieurs limitations et points d'amélioration sont apparus

pendant le déroulement du projet. Bien que le projet rassemble tous les éléments d'un système complet de contrôle / supervision, les différentes briques restent de fait très abstraites pour certains étudiants puisqu'elles sont presque toutes implémentées à l'intérieur de la carte Raspberry Pi. Il n'est pas toujours évident pour certains d'extrapoler ce que cela représenterait sur une installation industrielle réelle.

D'un point de vue matériel, il a été observé une certaine lenteur des interfaces Web, en particulier pour la programmation sur Scada-LTS, ce qui peut probablement s'expliquer par la puissance limitée des cartes Raspberry Pi qui se retrouvent entre autres à jouer en simultané les rôles de 2 serveurs, de 2 clients, et d'une base de données. La séparation du serveur Scada-LTS sur une machine séparée pourrait éventuellement arranger ces deux problèmes en même temps, mais nécessiterait donc des machines supplémentaires.

5.2 Retour général des étudiants

Cette SAÉ a globalement été bien reçue par les étudiants qui ont pu découvrir de nouveaux logiciels de programmation en langage automate, d'implémentation d'un système SCADA et d'IHM utilisables chez eux. La partie opérative et le matériel fourni ont également été appréciés. Cependant le temps alloué à la SAÉ ne leur a pas permis de mener à bien l'ensemble des activités prévues pour arriver à l'implémentation finale du correcteur et de la supervision sur le système réel. L'étude théorique du correcteur leur a posé de grandes difficultés malgré sa simplification, notamment parce que les notions et outils théoriques en automatique n'avaient pas eu le temps d'être étudiés entièrement dans le cours au moment de l'activité. Cette partie devra donc être remplacée par une étude empirique qualitative ou le correcteur devra être fourni entièrement si la SAÉ est maintenue comme actuellement dans la seconde moitié du semestre 3. L'activité pourrait cependant être maintenue si la SAÉ est décalée à un moment de la formation où les notions théoriques ont été suffisamment abordées, par exemple au semestre 4 ou 5. À l'opposé, la prise en main des différents logiciels pendant les séances de découverte a été assez rapide et ne semble pas nécessiter autant de séances qu'à l'heure actuelle. Un rééquilibrage en réduisant le nombre d'activités initiales de programmation avec les boutons poussoir et les LED semble donc possible, d'autant plus qu'elles ont moins de lien avec la partie opérative finale.

5.3 Retours étudiant relatifs aux compétences

L'objectif principal d'une SAÉ étant de pouvoir évaluer et valider les compétences des étudiants, un questionnaire leur a été distribué après la fin de la SAÉ pour évaluer leur perception de cette SAÉ en lien avec leur progrès dans les compétences du référentiel du BUT GEII, à savoir les compétences « Concevoir », « Vérifier », « Maintenir » et « Intégrer ». Les étudiants ont majoritairement (90%) trouvé la SAÉ utile pour développer les deux premières compétences, ainsi que la

compétence « Intégrer » dans une moindre mesure (80%) probablement à cause du manque de temps pour intégrer la solution finale. Cependant 55% des étudiants ont estimé que la SAÉ ne les avait « plutôt pas » aidés à développer la compétence « Maintenir », ce qui peut venir du fait que cette compétence est moins présente de base dans la SAÉ (principalement dans le diagnostic des capacités de l'esclave Modbus) et du fait que la frontière entre les compétences « Vérifier » et « Maintenir » n'est pas toujours claires pour les étudiants en général.

6 CONCLUSION

Les éléments d'un dispositif pédagogique et les activités pluridisciplinaires associées ont été présentés pour la mise en place d'une SAÉ sous forme de projet guidé au semestre 3 du parcours AII du BUT GEII à l'IUT de Valenciennes. Ce dispositif permet d'introduire aux étudiants les différents éléments qu'ils pourraient être amenés à mettre en œuvre dans une situation professionnelle de conception d'une partie contrôle et supervision complète d'une installation industrielle. La première mise en œuvre de la SAÉ est encourageante et montre qu'il est possible de mettre en place des activités à coût réduit avec des outils de programmation d'automate et de supervision se rapprochant d'outils industriels tout en individualisant les activités et en faisant appel à de nombreuses thématiques. Le retour des étudiants est plutôt positif et les aspects négatifs reportés devraient pouvoir être corrigés sans trop de difficulté en modifiant le temps alloué à chaque type d'activités et en simplifiant certaines parties.

Bibliographie

- [1] N. Lardenois, J. Chevré et L. Vermeiren, «Feedback on a Robotic Project for Competency-based Learning in French University Bachelor of Technology,» chez *ICERI2022 Proceedings*, Seville, Spain, 2022.
- [2] J. Bernhardt, PLC Programming with the Raspberry Pi and the OpenPLC Project: ModbusRTU and ModbusTCP Examples with the Arduino Uno and ESP8266, Elektor, 2021.
- [3] <https://acrome.net/product/ball-balancing-table>.
- [4] E. Bartmann, Le grand livre d'Arduino, 2e éd., Librairie Eyrolles, 2022.
- [5] <https://modbus.org/>.
- [6] <https://www.arduino.cc/reference/en/libraries/usbhost/>.
- [7] F. Mock, Raspberry Pi 4 : Exploitez tout le potentiel de votre nano-ordinateur, 2e éd., Eni, 2021.
- [8] <http://easymodbus.net/en/>.
- [9] https://github.com/thiagoralves/OpenPLC_Editor/tree/dev-python3.
- [10] <http://scada-lts.com/>.