

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



CDTA

Mémoire de Master

Filière Électronique
Spécialité Instrumentation

Présenté par

ZAHAFI NAOUAL

&

HAOUAM Imène

Réalisation d'une interface d'acquisition et de contrôle d'un régulateur de température via le bus RS-485 Modbus et un PC

Proposé par : Mr A. KEDADRA 1 & Mme N. BRAHIMI

Année Universitaire 2019.2020

Remerciements

Ce travail est l'aboutissement d'un long cheminement au cours duquel on a bénéficié de l'encadrement, des encouragements et du soutien de plusieurs personnes, à qui nous tenons à dire profondément et sincèrement MERCI.

À NOTRE MAITRE, ET RAPPORTEUR DE MÉMOIRE DE FIN D'ÉTUDES MASTER 2. LE MAITRE DE RECHERCHE CDTA Mr. ABDELKRIM KEDADRA et notre Co-promotrice et enseignante Mme. Brahimi Nouzha Nous sommes très honorées de vous avoir comme encadreurs de notre PFE master 2. Nous vous remercions pour la gentillesse et la spontanéité avec lesquelles vous avez bien voulu diriger ce travail. Nous avons eu le grand plaisir de travailler sous votre direction, et avons trouvé auprès de vous le conseiller et le guide qui nous a reçu en toute circonstance avec sympathie, sourire et bienveillance. Votre compétence professionnelle incontestable ainsi que vos qualités humaines vous valent l'admiration et le respect de tous. Vous êtes et vous serez pour nous l'exemple de rigueur et de droiture dans l'exercice de la profession. Veuillez, chères Maîtres, trouvé dans ce modeste travail l'expression de notre haute considération, de notre sincère reconnaissance et de notre profond respect.

AUX JUGES DU PFE MASTER 2 :

C'est pour nous un grand honneur de vous voir siéger dans notre jury. Nous vous sommes très reconnaissantes de la spontanéité et de l'amabilité avec lesquelles vous avez accepté de juger notre travail. Veuillez trouver, chers professeurs, le témoignage de notre grande Reconnaissance et de notre profond respect.

أي تغيير غير مرغوب في درجة الحرارة يمكن أن يسبب بعض السلبيات التي لا ينبغي التغاضي عنها، ولهذا السبب فإن درجة الحرارة عامل مهم للغاية في العملية الصناعية. من المهم أن تعرف قيمتها بدقة، لتوفير الوقت وتحسين الإنتاجية، ما يستلزم التنظيم التلقائي. هذا العمل مساهمة في مشروع CDTA للتحكم في درجة حرارة الزيوت الغذائية المستخدمة وتنظيمها من أجل اتخاذ قرار بشأن تدهورها. للقيام بذلك، يتم تنفيذ العمل النظري على الصناعة التلقائية، واعداد تجريبي. يتضمن ذلك تطوير واجهة اكتساب والتحكم في منظم EW4822 عبر محول USB / RS-485 بواسطة جهاز كمبيوتر تحت واجهة رسومية LabVIEW.

كلمات المفاتيح: التنظيم التلقائي; تنظيم درجة الحرارة; القيادة والاستحواذ والسيطرة.

Résumé :

Tout changement de température indésirable peut engendrer certains inconvénients qu'il ne faut pas négliger, car la température représente un paramètre très important dans le processus industriel. Sa valeur doit être connue avec précision pour gagner du temps et avoir une meilleure productivité, la régulation automatique s'impose. Ce travail consiste à contribuer dans un projet du CDTA pour commander et assurer la régulation de la température des huiles agroalimentaires utilisés afin de décider sur leur dégradation. Pour ce faire, un travail théorique sur l'automatisme industriel, et un montage expérimental sont réalisés. Il s'agit de développer une interface d'acquisition et de contrôle d'un régulateur EW4822 via le convertisseur adaptateur USB/RS-485 par un PC sous l'interface graphique LabVIEW.

Mots clés : la régulation automatique ; régulation de température ; commande, acquisition et contrôle.

Abstract :

An unsolicited temperature change can cause some drawbacks that should not be overlooked. That's why the temperature is a very important parameter in an industrial process. Its value must be known precisely, to save time and have better productivity, automatic regulation is required. This work consists a contribution at a CDTA project to achieve the temperature control of used food oils in order to decide on their degradation. A theoretical work on industrial automation and an experimental set-up are carry out. This includes developing an acquisition interface and controlling an EW4822 regulator via the USB / RS-485 adapter converter by a PC under the LabVIEW graphical interface.

Keywords : automatic regulation ; temperature regulation ; command, acquisition and control.

Listes des acronymes et abréviations :

PO : Partie Opérative.

PC : Partie Commande.

PID : Proportionnel Intégral Dérivé.

PV : Process Valu.

SV : Set Point.

CDTA : Centre de Développement des Technologies Avancées.

LED : Diode Electro-Luminescente.

PTC : Coefficient de Température Positif.

TTL : Transistor-Transistor Logic.

RTU : Remote Terminal Unit (Unité Terminale Distante).

ASCII : American Standard Code for Information Interchange.

TCP : Transmission Control Protocol.

IP : Internet Protocol.

CAN : Controller Area Network.

PLC : Contrôleur Logique Programmable.

DCS : Système de Contrôle Distribué.

CRC : Contrôle de Redondance Cyclique.

LRC : Contrôle de Redondance Longitudinale.

LabVIEW : Laboratory Virtual Instrument Engineering Workbench.

VI : Virtual Instrument.

FTDI : Future Technology Devices International.

Liste des symboles :

$e(t)$: l'erreur à l'instant.

$C(t)$: la commande générée du régulateur.

T_i : la constante du temps d'action intégrale (en seconds).

T_d : La constante du temps d'action dérivée (en seconde).

K_p : le gain proportionnel.

K_i : le gain intégral.

K_d : le gain dérivé.

Table des matières

Introduction générale	01
Chapitre 1 : Généralités sur les systèmes automatisés	
1.1 Introduction.....	03
1.2 Définition d'un système.....	03
1.2.1 Systèmes de production industriels.....	03
1.2.2 Systèmes automatisés.....	04
a. Fonctionnement d'un système automatisé.....	05
b. Echange d'informations dans un système automatisé.....	05
c. Avantages et Inconvénients de l'automatisation.....	06
1.3 Régulation des systèmes automatiques.....	07
1.3.1 Régulation en boucle ouverte.....	07
1.3.2 Régulation en boucle fermée.....	07
1.4. Différents types de régulation.....	08
1.5. Régulation par PID.....	08
1.5.1 Principe de fonctionnement.....	08
a. Régulateur à action proportionnel (P).....	09
b. Régulateur à action intégrale (I).....	09
c. Régulateur à action dérivée (D).....	10
1.5.2 Influence des paramètres d'un PID.....	10
1.5.3 Réglage d'un PID.....	11
1.5.4 Différentes structures d'un PID.....	11
a. Structure parallèle.....	12
b. Structure série.....	12
c. Structure mixte.....	12
1.6. Conclusion.....	13
Chapitre 02 : Etudes des parties matérielle et logicielle	
2.1. Introduction.....	14
2.2. Partie matérielle.....	14
2.2.1 Description du dispositif expérimental.....	14

2.2.2 Régulateur EW4822.....	15
a. Réglage manuel du régulateur EW4822.....	16
b. Réglage manuel du point de consigne.....	17
c. Menus de programmation du régulateur EW4822.....	17
2.2.3 Protocoles de communications.....	18
a. Protocole Modbus.....	19
b. Principe de fonctionnement du Modbus.....	19
c. Types de données Modbus.....	20
d. Modes de transmissions en Modbus.....	21
e. Trame d'échange requête/réponse pour Modbus RTU.....	22
2.2.4 Zone mémoire.....	24
2.2.5 Techniques de transmissions.....	25
a. Types de transmission.....	26
2.2.6 Ports de connexion et quelques standards électrique.....	26
a. Ports série.....	27
2.2.7 Quelques standards électriques.....	27
a. Liaison RS-485.....	28
b. Adaptation de la liaison RS-485 avec un PC.....	28
c. Convertisseur adaptateur USB/RS-485.....	28
2.3 Partie logicielle.....	29
2.3.1 Logiciel LabVIEW.....	29
2.3.2 Environnement LabVIEW.....	29
a. Face-avant.....	30
b. Bloc diagramme.....	31
c. Barres d'outils.....	31
d. Types de données utilisées sous LabVIEW.....	34
3.4 Conclusion.....	35
Chapitre 03 : Implémentation et résultats	
3.1 Introduction.....	36
3.2 Câblage du Dispositif de test.....	36
3.3 Réglages des éléments du dispositif.....	37

3.3.1 Réglage des paramètres du régulateur EW4822.....	37
3.3.2 Réglages du port COM.....	39
3.4 Tests de connectivité et activation du MODBUS.....	40
3.4.1 Interface de test Commix 1.4.....	40
3.4.2 Interface de test Modbus Poll.....	41
3.5 Résultats des interfaces de test.....	42
3.5.1 Tests sur le simulateur Commix1.4.....	43
a. Lecture de la valeur de consigne.....	43
b. Lecture de la valeur du paramètre DEA de l'étiquette ADD.....	44
c. Lecture de la valeur du paramètre FAA de l'étiquette ADD.....	45
3.5.2 Tests sur le simulateur Modbus Poll.....	45
a. Lecture de la valeur du registre du point de consigne.....	46
b. Lecture de la valeur du registre DEA.....	47
c. Lecture du registre FAA.....	48
3.5.3 La différence entre les deux simulateurs de test.....	49
3.6 Implémentation et résultats sur le logiciel LabVIEW.....	49
3.6.1 Programmation avec la bibliothèque Modbus.....	50
a. Lecture du registre de la valeur de consigne.....	52
b. Lecture du registre DEA.....	52
c. Lecture du registre FAA.....	53
3.6.2 Programmation avec MB master serial Query.....	53
a. Lecture du registre de la valeur de consigne.....	55
b. Lecture du registre DEA.....	55
c. Lecture du registre FAA.....	56
3.7 Conclusion.....	56
Conclusion générale	57
Bibliographie	59

Liste des figures

Figure 1.1. Structure générale d'un système de production industriel.	04
Figure 1.2. Échange d'information dans un système automatisé.	06
Figure 1.3. Structure parallèle.	12
Figure 1.4. Structure série.	12
Figure 1.5. Structure mixte.	13
Figure 2.1. Dispositif expérimental du CDTA.	15
Figure 2.2. Schéma électrique et photo du régulateur de température EW4822.	16
Figure 2.3. Touches composant l'écran du régulateur EW4822.	16
Figure 2.4. Menus de programmation utilisateur et installateur d'EW4822.	17
Figure 2.5. Communications entre plusieurs équipements avec le protocole Modbus TCP/IP.	18
Figure 2.6. Principe de fonctionnement du protocole Modbus.	20
Figure 2.7. La trame Modbus RTU.	22
Figure 2.8 Techniques de transmission parallèle et série.	25
Figure 2.9. Port de connexion d'un PC portable.	26
Figure 2.10. Connexion d'un maître et plusieurs esclaves utilisant la liaison RS-485.	28
Figure 2.11. USB to TTL/RS485 adaptateur.	29
Figure 2.12. Composantes d'un VI.	30
Figure 2.13. Face-avant.	30
Figure 2.14. Bloc diagramme.	31
Figure 2.15. Barre d'outils de la face-avant.	32
Figure 2.16. Barre d'outils du diagramme.	32
Figure 2.17. Palette d'outils.	32
Figure 2.18. Palette de commande et de fonction.	34
Figure 2.19. Fenêtre d'aide contextuel.	34
Figure 3.1. Dispositif de test.	36

Figure 3.2. Câblage schématisé du dispositif de test.	37
Figure 3.3. Réglages des paramètres du régulateur EW4822 à partir de l'étiquette ADD.	38
Figure 3.4. Réglages des paramètres du Port COM.	39
Figure 3.5. Interfaces du logiciel Commix1.4	40
Figure 3.6. Interface du Modbus Poll.	41
Figure 3.7. Lecture de la valeur de consigne du régulateur EW4822 sur Commix.	43
Figure 3.8. Lecture de la valeur du paramètre DEA de l'étiquette ADD sur Commix.	44
Figure 3.9. Lecture de la valeur du paramètre FAA de l'étiquette ADD sur Commix.	45
Figure 3.10. Réglages des paramètres pour lire le registre du point de consigne sur Modbus Poll.	46
Figure 3.11. Réglage des paramètres du port COM.	46
Figure 3.12. Résultat de lecture du registre du point de consigne sur Modbus Poll.	47
Figure 3.13. Réglages des paramètres pour la lecture du registre de DEA sur Modbus Poll.	47
Figure 3.14. Résultat de lecture du DEA sur Modbus Poll.	48
Figure 3.15. Réglages des paramètres pour la lecture du registre de FAA sur Modbus Poll.	48
Figure 3.16. Résultat de lecture du registre du FAA sur Modbus Poll.	49
Figure 3.17. Instruments virtuels nécessaire pour l'implémentation sur LabVIEW utilisant Modbus Library.	51
Figure 3.18. Bloc diagramme de la lecture des registres du régulateur EW4822 avec Modbus Library.	51
Figure 3.19. Résultats de la lecture du registre du point de consigne sur la face-avant du logiciel LabVIEW.	52
Figure 3.20. Résultats de la lecture du registre DEA sur la face-avant du logiciel LabVIEW.	52
Figure 3.21. Résultats de la lecture du registre FAA sur la face-avant du logiciel LabVIEW.	53
Figure 3.22. Les VI's utilisés de MB serial Query pour l'implémentation sur LabVIEW.	54
Figure 3.23. Bloc diagramme de la lecture des registres du régulateur EW4822 avec MB Master Serial Query.	54
Figure 3.24. Résultat de la lecture du registre du point de consigne sur la Face-avant.	55
Figure 3.25. Résultat de la lecture du registre DEA sur la Face-avant.	55
Figure 3.26. Résultat de la lecture du registre FAA sur la Face-avant.	56

Liste des tableaux

Tableau 1.1. Influence du régulateur PID.	10
Tableau 1.2. Synthèse sur les actions PID.	11
Tableau 2.1. Accès au menu de programmation « User » et « Inst ».	18
Tableau 2.2. Comparaison RTU/ASCII [30].	22
Tableau 2.3. Codes de fonction Modbus RTU	23
Tableau 2.4. Zones mémoire des appareil Modbus.	24
Tableau 2.5. Différences entre RS232/RS485 [35].	27
Tableau 2.6. Types de données sous LabVIEW.	35
Tableau 3.1. Réglages des paramètres dans les 3 niveaux.	37
Tableau 3.2. Description et réglage nécessaire des paramètres de l'étiquette ADD.	38
Tableau 3.3. Adresses et description des paramètres du régulateur EW4822.	42
Tableau 3.4. Trame de requête de la lecture du point de consigne sur Commix1.4.	43
Tableau 3.5. Trame de réponse de la lecture du point de consigne sur Commix1.4.	43
Tableau 3.6. Trame de requête de la lecture de la valeur de DEA sur Commix1.4.	44
Tableau 3.7. Trame de réponse de la lecture de la valeur de DEA sur Commix1.4.	44
Tableau 3.8. Trame de requête de la lecture de la valeur de FAA sur Commix1.4.	45
Tableau 3.9. Trame de réponse de la lecture de la valeur du FAA sur Commix1.4.	45

Introduction générale

A l'ère de l'automatique, la technique de la régulation revêt une importance primordiale pour les systèmes industriels modernes. Cette technique permet de réaliser des produits innovants et intelligents qui constituent les conditions de base de la compétitivité sur le marché mondial [1].

La plupart des processus industriels nécessitent de contrôler un certain nombre de paramètres (température, pression, niveau, débit, pH, etc...) [2]. La température dans l'industrie est considérée comme une information importante. Elle intervient comme une grandeur principale dont la valeur doit être connue avec précision. Afin d'obtenir un résultat de température exacte, les ingénieurs ont opté pour la mesure automatique pour la contrôler selon une valeur de fonctionnement opérationnel [3].

C'est dans ce contexte qu'il nous est proposé un projet de fin d'étude de Master 2 au niveau du CDTA. Il a pour but de commander et de contrôler la température d'un dispositif de caractérisation des huiles (alimentaire et industrielles). Il s'agit de garder constamment la valeur de température très proche de la valeur de consigne. Cela répond aux besoins de la chaîne de l'automatisation et de la production du groupe Cevital.

Pour réaliser ce travail en complément à la réalisation expérimental au CDTA, qui utilise le régulateur de température Eliwell 4822 de type PID. Notre travail sera focalisé sur la partie commande instrumentale et qui consiste à faire une acquisition et un affichage (passif) des paramètres dudit régulateur. Ce dernier sera connecté à un ordinateur PC à travers un adaptateur convertisseur USB /RS-485 suivant le protocole Modbus RTU. Le logiciel de programmation choisi pour atteindre notre objectif est le logiciel graphique LabVIEW.

Pour ce faire, nous avons organisé notre travail en trois chapitres. Le premier chapitre comprend des généralités sur les systèmes automatisés dans l'industrie. Nous abordons également quelques détails liés à la technique d'automatisation pour la régulation de température.

Dans le deuxième chapitre nous allons étudier les deux parties constituant le système, la partie matérielle et la partie logicielle. Dans la première partie nous décrivons le dispositif expérimental du CDTA, suivi d'une démonstration qui montre des instructions pour régler manuellement la valeur du point de consigne au niveau du régulateur de température EW4822. Ensuite nous présentons les différents protocoles de communication industriels en concentrant sur le protocole Modbus RTU utilisé par notre instrument. La deuxième partie mettra en évidence le logiciel LabVIEW que nous avons choisi pour effectuer le travail.

Le dernier chapitre est consacré à présenter les résultats obtenus lors de l'implémentation sur LabVIEW et les tests de connectivité Modbus qui précède cette étape.

Enfin, nous terminons par une conclusion générale et le travail de perspectif.

Chapitre 1 Généralités sur les systèmes automatisés

1.1 Introduction :

Devant la compétitivité sans cesse croissante et de plus en plus dure, les industries doivent garder leurs outils de production performant et fiable. Dans ce domaine, l'automatisation tient une place très importante et il serait difficile de concevoir un système de production sans avoir recours aux différentes technologies et composants qui forment les systèmes automatisés de production [4].

L'automatisme en industrie s'obtient par la technique de régulation. Selon une consigne désirée et avec des modèles mathématiques à implémenter sous forme d'un software, l'automatisme en industrie nous permet de contrôler et de réaliser une fonction telle que la régulation de température dans notre cas.

1.2 Définition d'un système :

Un système est un ensemble d'instruments qui peut donner une valeur ajoutée dans un environnement donné [4]. Sa matière d'œuvre elle peut être un produit, une énergie, ou bien une information. Sa valeur ajoutée est caractérisée par sa nature, sa quantité et sa qualité. Son environnement est un contexte physique, social, économique, etc.

On distingue différents types de systèmes :

1.2.1 Systèmes de production industriels :

Sont divisés en plusieurs sous-systèmes interconnectés entre eux (**figure 1.1**), chaque sous-système contient plusieurs actionneurs et capteurs qui renseignent à chaque l'état du système [5].

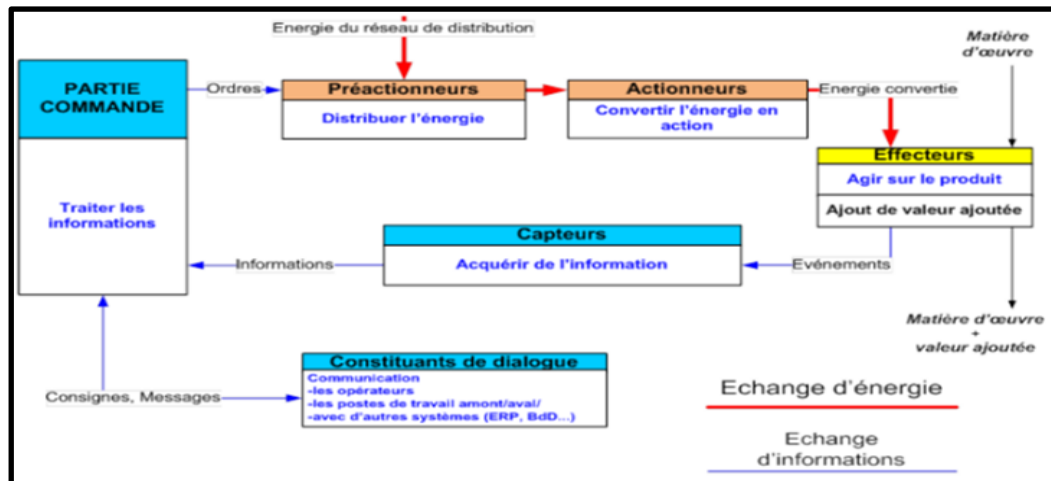


Figure 1.1. Structure générale d'un système de production industriel.

Il peut se résumer par [5] :

- Un ensemble de sous-systèmes destinés à l'exécution d'une fonction globale.
- Un ensemble de système d'actionnements constituant une interface de puissance tels qu'un distributeur, un variateur de vitesse, etc. Reliés à des actionneurs comme des vérins, des moteurs, des résistances, etc...
- Une instrumentation constituée d'un ensemble de chaînes de mesures comprenant des capteurs. Cette chaîne assure en particulier le filtrage et la mise en forme des signaux provenant des capteurs.

1.2.2 Systèmes automatisés :

Sont des systèmes qui effectuent toujours le même cycle après avoir reçu les instructions (consignes) de la part d'un opérateur. Ils sont composés d'un ou plusieurs éléments programmés pour effectuer un ensemble de tâches sans intervention humaine [6].

L'objectifs de l'automatisation est d'apportée à la valeur ajoutée par le système des éléments supplémentaires exprimables en termes d'objectifs [7] :

- Augmentation de la productivité (rentabilité, compétitivité) du système
- Améliorer la flexibilité de production
- Améliorer la qualité du produit

- Réduire les risques produits par l'environnement hostiles à l'homme (milieu toxique, dangereux, nucléaire...) et éliminer les tâches pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées...).
- Augmenter la sécurité.
- Economiser les matières premières et l'énergie.

a Fonctionnement d'un système automatisé :

La partie commande envoie des ordres aux actionneurs et elle reçoit des informations d'état en provenance des capteurs. Il y a donc une chaîne de transmission entre la partie opérative et la partie commande (**figure 1.1**). Chaque partie doit être alimentée en énergie [8].

Un système automatisé est structuré de deux parties fondamentales :

- **Partie opérative (PO) :**

C'est la partie visible du système qui nous permet de transformer la matière d'œuvre entrante. Elle est composée d'éléments mécaniques, d'actionneurs (vérins, moteurs), de pré-actionneurs (distributeurs et contacteurs) et d'éléments de détection (capteurs, détecteurs). Elle agit sur le procédé industriel afin de lui donner sa valeur ajoutée.

Elle contient l'ensemble des capteurs qui fournissent des informations à la partie commande sur l'état du processus et l'ensemble des actionneurs qui reçoivent les commandes produites par la logique de la partie commande [9].

- **Partie commande (pc) :**

C'est la partie qui traite les informations, elle regroupe tous les composants de traitement d'informations, elle est utilisée pour faire fonctionner la partie opérative et elle gère et contrôle le déroulement du cycle.

Une fois les ordres accomplis, la PO va le signaler au PC qui va à son tour le signaler à l'opérateur. Ce dernier pourra donc dire que le travail a bien été transmis et réalisé [9].

b Echange d'informations dans un système automatisé :

L'ensemble des échanges d'informations dans un système automatisé (**figure 1.2**) est contrôlé par le programme de la partie commande [10] :

- L'opérateur donne des consignes à la partie commande.
- La partie commande donne des ordres à la partie opérative.
- Les actionneurs exécutent les ordres reçus (production d'un phénomène physique).
- Les capteurs réagissent à une variation d'état (détection d'un phénomène physique).
- La partie opérative adresse des comptes-rendus à la partie commande.
- La partie commande envoie à l'opérateur des signaux sur l'état du système ou de son environnement.

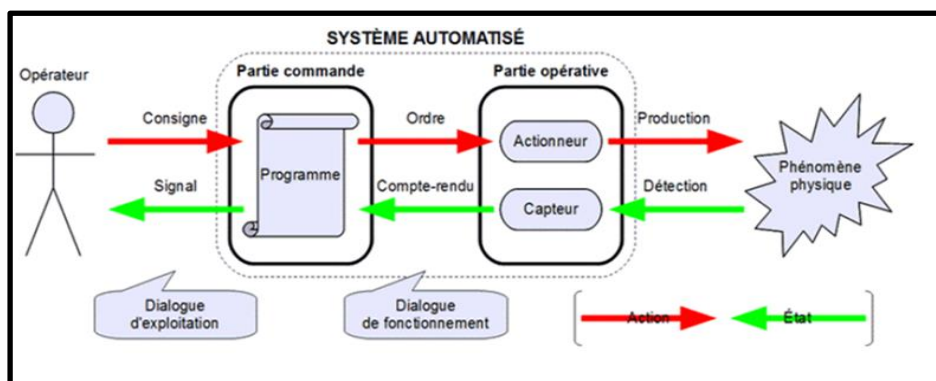


Figure 1.2. Échange d'information dans un système automatisé.

c Avantages et inconvénients de l'automatisation :

L'automatisation dans l'industrie peut améliorer les résultats de l'entreprise en augmentant considérablement sa productivité. Néanmoins, on peut citer quelques avantages et inconvénients [11].

- **Avantages :**
 - Améliorer les conditions de travail (réaliser des tâches complexes ou dangereuses pour l'homme, effectuer des tâches pénibles ou répétitives).
 - Augmenter la Sécurité.
 - Améliorer Précision.
 - Réduire les coûts de fabrications.
 - Augmenter la productivité (réduire le temps de travail nécessaire à la production implique un taux de travail élevé).
 - La Flexibilité (une machine peut s'adapter à plusieurs productions).
 - Confidentialité.

- Un Système automatisé peut travailler 24h sur 24h et sans suspensions [8].
- **Inconvénients :**
 - Incidence sur l'emploi (le chômage).
 - Investissement pour l'achat de machines.
 - Coût de la maintenance élevé.
 - Pannes.
 - Consommation d'énergie.
 - Formation d'un personnel plus qualifié [8].

1.3 Régulation des systèmes automatique :

La régulation automatique est un système de contrôle d'une ou plusieurs grandeurs physiques (température, humidité, pression, débit, etc.) [12]. Ce système permet d'agir sur la grandeur physique par le biais d'un actionneur.

Un système de contrôle est concrétisés en boucle ouverte ou en boucle fermée :

1.3.1 Régulation en boucle ouverte :

Dans une régulation en boucle ouverte, nous n'avons pas besoin de mesurer la grandeur de sortie. L'organe de contrôle n'a aucune réaction sur le processus de fonctionnement. Il faut connaître uniquement la corrélation entre la valeur mesurée et la mesure. Elle permet d'anticiper les phénomènes et d'obtenir des temps de réponse très courts. De plus, il n'y a pas d'oscillation à craindre.

Parmi les inconvénients d'une régulation en boucle ouverte, il est nécessaire de connaître la loi régissant le fonctionnement du processus. Cependant, il est très fréquent que nous ne connaissions pas la loi en question et il n'y a aucun moyen de contrôler, et/ou de compenser les erreurs à l'intérieur de la boucle. Cela signifie qu'il y a moins de précision et surtout la fidélité est moindre en dépendant de la qualité intrinsèque des composants [13].

1.3.2 Régulation en boucle fermée :

La régulation en boucle fermée c'est l'action de la variable de contrôle sur la variable contrôlée à travers un « processus » qui boucle la chaîne. Il s'agit d'exercer une influence

sur la valeur d'entrée en mesurant la valeur de sortie pour la maintenir dans des limites définies.

Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs est automatiquement compensée par la rétroaction à travers le processus. Parmi les inconvénients d'une régulation en boucle fermée est que la précision et la fidélité de la régulation dépendent des valeurs mesurées et de la consigne. Aussi un simple mauvais choix de certains composants peut amener le système d'une boucle fermée à entrer en oscillation [14].

1.4 Différents types de régulation :

Il existe différents types de régulation [15] :

- 1) TOR : Tout Ou Rien (ON-OFF Control).
- 2) MLI : Modulation de la Largeur d'Impulsion (PWM) avec PID.
- 3) FUZZY : Continue PID avec l'introduction de la logique floue.
- 4) MULTI-BOUCLES : Maître Esclave, Cascade, Rapport.
- 5) NUMERIQUE : Modèle de référence.
- 6) PID : Continue Proportionnelle, Intégrateur, Dérivateur [16].

Parmi les types indiqués ci-dessus, on s'intéresse à la régulation par PID, pour cela nous allons discuter quelques informations générales à ce sujet.

1.5 Régulation par PID :

La régulation par PID est la plus utilisée en industrie, elle sert à Contrôler la variable d'un processus industriel avec le maximum de sécurité. Pour atteindre cet objectif, chaque élément du processus doit travailler avec précision, et chaque variable doit être contrôlée afin de rester proche de la consigne [17].

1.5.1 Principe de fonctionnement :

La régulation par PID est une simple implémentation de retour d'information (feedback) [18], elle s'obtient grâce à 3 actions :

- L'action proportionnelle.
- L'action intégrale.

- L'action dérivée.

Dans un processus de régulation avec PID, la loi de commande s'obtient par la modélisation de l'erreur entre la valeur de la consigne proposée et la valeur de la sortie mesurée [19] :

$$e(t) = \text{Consigne} - \text{mesure} \quad (1.1)$$

On distingue trois différents types de régulateurs :

a Régulateur à action proportionnel (P) :

Ce régulateur fonctionne par un paramètre de commande K_p proportionnel à l'erreur $e(t)$ calculée entre la valeur mesurée et la consigne. L'équation d'un régulateur à action proportionnelle simple est présentée par :

$$c(t) = k_p \cdot e(t) \quad (1.2)$$

D'après Laplace on obtient :

$$C(p) = k_p \cdot e(p) \quad (1.3)$$

L'idée étant de perturber le système afin que celui-ci réagisse plus rapidement aux changements de consignes. Plus on augmente l'effet de l'erreur $e(t)$, plus la valeur de K_p est grande [20].

b Régulateur à action intégrale (I) :

La loi de commande d'un régulateur à action intégrale est de la forme suivante [20] :

$$c(t) = k_i \int e(\tau) \cdot d\tau = 1/T_i \int e(\tau) \cdot d\tau \quad (1.4)$$

L'erreur $e(t)$ est intégrée sur un intervalle de temps (de 0 à t), puis multipliée par un gain K_i .

Avec : $k_i = K_p/T_i$, ou T_i est la constante de temps d'actions intégrale (s).

D'après Laplace on obtient :

$$C(p) = k_i \cdot (e(p)/p) = (1/T_i \cdot p) \cdot e(p) \quad (1.5)$$

c Régulateur à action dérivée (D) :

Dans ce cas de régulateur à action dérivé, la loi de commande est donnée par [20] :

$$c(t) = kd \cdot (de(t)/dt) = Td \cdot (de(t)/dt) \quad (1.6)$$

L'erreur $e(t)$ est dérivée par rapport au temps, puis multipliée par un gain Kd .

D'après Laplace on obtient :

$$C(p) = kd \cdot P \cdot e(p) = Td \cdot P \cdot e(p) \quad (1.7)$$

Avec $kd = Kp \cdot Td$, ou Td est la constante de temps d'action dérivée(s).

Le **tableau 1.1** nous indique l'influence sur un régulateur en stabilité, précision et rapidité en faisant varier les constantes de temps par augmentation ou diminution du gain Kp :

Augmentation de	Stabilité	Précision	Rapidité
Kp	Diminue	Augmente	Augmente
Ti	Augmente	Pas d'influence	Diminue
Td	Augmente	Pas d'influence	Diminue

Tableau 1.1. Influence du régulateur PID [20].

1.5.2 Influence des paramètres d'un PID :

La régulation par PID est influencée par les 3 gains (paramètres) suivants :

- **Kp** : Lorsque Kp augmente, on obtient un dépassement plus important avec un temps de montée plus court. Donc l'erreur statique se trouve améliorer avec une petite variation du temps de dépassement.
- **Ki** : Lorsqu'il augmente, On obtient un dépassement plus important avec un temps de montée plus court. Le temps d'établissement au régime stationnaire s'allonge, alors on assure dans ce cas une erreur statique réduite. Donc plus ce paramètre est élevé, moins l'erreur statique est grande, et plus la réponse du système est ralentie.

- **Kd** : Lorsque Kd augmente, le dépassement diminue, et le temps de montée change un peu. Avec un temps d'établissement au régime stationnaire meilleur. Donc il n'y a pas d'influences sur l'erreur statique. Si ce paramètre est trop élevé [19].

1.5.3 Réglage d'un PID :

Le réglage d'un PID consiste à déterminer les trois paramètres (K_p , K_i et K_d), pour avoir un système robuste, rapide et précis :

- La robustesse est le paramètre le plus important et délicat. Si la régulation fonctionne toujours malgré les petits changements de modèle, on peut dire que ce système est robuste.
- La rapidité de système dépend du temps de montée et du temps d'établissement du régime stationnaire.
- Le critère de précision est basé sur l'erreur statique [19].

Chaque action de régulation par PID présente des avantages et des inconvénients (**tableau 1.2**). Une loi de commande mixte peut convenir pour obtenir un meilleur fonctionnement [20] :

Action	Avantage	Désavantage
P	Action instantanée (dynamique).	Ne permet pas d'annuler une erreur statique mais permet de la réduire.
I	Annulation d'erreur statique. Amélioration de la robustesse.	Action lente. Ralentit le système (effet déstabilisant).
D	Action très dynamique. Améliorer la rapidité (effet stabilisant).	Sensibilité aux bruits. Forte oscillation de l'organe de commande.

Tableau 1.2. Synthèse sur les actions PID.

1.5.4 Différentes structures de PID :

Pour associer les trois paramètres d'un PID (P, I, D), on distingue trois différentes structures [21] :

a Structure parallèle :

La loi de commande pour une structure parallèle (**figure 1.3**) est donnée par :

$$c(t) = K_p e(t) + K_i \int e(\tau) . d\tau + K_d . (d(e)/dt) \quad (1.8)$$

D'après la transformé de la place on obtient :

$$C(p) = K_p . e(p) + K_i . (e(p)/p) + K_d . P . e(p) \quad (1.9)$$



Figure 1.3. Structure parallèle.

b Structure série :

La commande d'une structure série (**figure 1.4**) est donnée par :

$$C(p) = e(p) . K_p . [K_i . (1/p)] . [K_d . P] \quad (1.10)$$

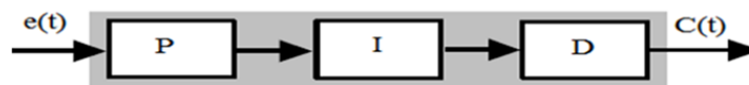


Figure 1.4. Structure série.

c Structure mixte :

D'après la transformé de Laplace on obtient la commande d'une structure mixte (**figure1.5**) :

$$C(p) = K_p . e(p) . [K_i . (1/p) + K_d . P] \quad (1.13)$$

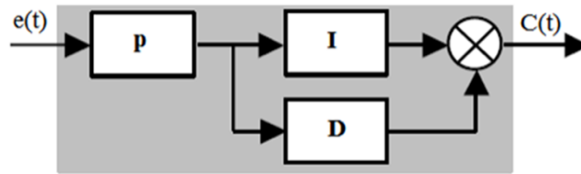


Figure 1.5. Structure mixte.

1.6 Conclusion

Tout système de production industriel fait appel à l'automatisation du processus pour une meilleure performance. L'automatisation est réalisée grâce la technique de la régulation d'une grandeur physique de tels sorte qu'elle reste très proche de la valeur désirée quelques soit les perturbations extérieures.

Dans ce chapitre nous avons étudié les différents types de régulation : la régulation en boucle ouverte, la régulation en boucle fermée et la régulation par PID qui est la plus utilisée pour l'automatisation dans le domaine industriel.

Chapitre 2 Etude des parties matérielle et logicielle

2.1 Introduction :

Ce chapitre sera consacré à l'étude des parties matérielle et logiciel utilisés pour réaliser une maquette complétant le dispositif expérimental développé au CDTA. Nous commençons par décrire le dispositif expérimental et expliquer le réglage manuel de la valeur de consigne concernant le régulateur EW4822. Ensuite nous allons présenter quelques protocoles de communications industriels, en particulier ceux utilisés pour établir la connexion entre le maître et l'esclave. Enfin nous allons détailler l'interface de programmation graphique dans le logiciel LabVIEW.

2.2 Partie matérielle :

Le prototype expérimental développé au CDTA est constitué de :

- Cuve dotée de transducteurs ultrasonores (quelque MHz).
- Sondes de température.
- Résistances chauffantes.
- Système de régulation de température à base du régulateur Eliwell 4822.

Ce dispositif permet la caractérisation de la dégradation des huiles usées.

2.2.1 Description du dispositif expérimental :

Le dispositif est constitué d'un boîtier qui montre de l'extérieur trois LED (deux pour les résistances de chauffage R1 et R2 et l'autre indique la mise en marche du dispositif ON/OFF). Un commutateur rotatif pour le mode MARCHE/ ARRET, et un régulateur de température à deux étages de type Eliwell 4822 (**figure 2.1.b**). Ce dernier utilise une sonde de type PTC Pour prélevé la température d'huile. Cette sonde est un capteur de température qui transforme la grandeur mesurée (en degré Celsius ou en Fahrenheit)

en un signal électrique qui sera traité et affiché sur le premier afficheur. On trouve également deux résistances chauffantes qui servent à augmenter la température de l'huile dans chaque bac lors du refroidissement, et afin de garder la température désirée (**figure 2.1.c**). Le contrôle se fait grâce à un relais (switch Interrupteur output1) permettant d'activer un contacteur électrique de coupure de phase (**figure 2.1.a**). Le dispositif permet d'être contrôlé par un PC à travers une liaison série intégrée au régulateur en TTL Televis, Modbus ou connexion directe RS-485.

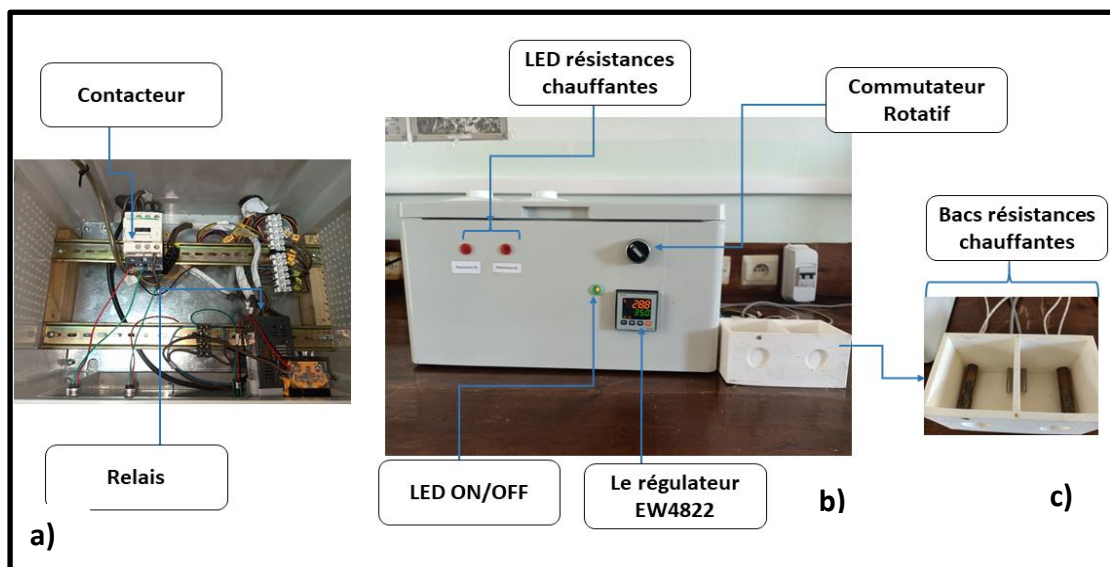


Figure 2.1. Dispositif expérimental du CDTA.

2.2.2 Régulateur EW4822 :

Le régulateur Eliwell 4822 (**figure 2.2.a**) est un type des nouveaux régulateurs de la série EW4800 qui ont été étudiés pour toutes les applications industrielles nécessitant un contrôle de température de haute précision. Ils peuvent être utilisés pour la température, l'humidité et la pression. Selon le modèle, ils sont équipés d'entrées pour sondes thermostatiques (PTC, PT100, NI100, Tcj, Tck), courant de 4 à 20mA pour sondes d'humidité (HS 18, 280, 310) ou sondes de pression (PA007 / 030)). Le modèle EW4822 dispose d'une connexion RS-485 intégrée, qui peut être utilisée pour se connecter à un système Televis ou Modbus (**figure 2.2.b**) [22].

La série EW4800 permet l'utilisation de 2 programmes indépendants personnalisables, chacun pouvant gérer jusqu'à 8 cycles thermiques. Ces programmes permettent de personnaliser le cycle thermique en fonction du point de consigne, du temps de cycle et

de son délai d'activation, ce qui rend l'appareil adapté aux exigences d'applications les plus demandées dans le secteur du contrôle de procédé. La valeur de la sonde est lue par un double affichage (affichage à 4 chiffres). Le premier est utilisé pour la lecture de la sonde et le second est pour le point de consigne. La plage d'alimentation flexible est de 12 ... 24V à 95 ... 240V [22]

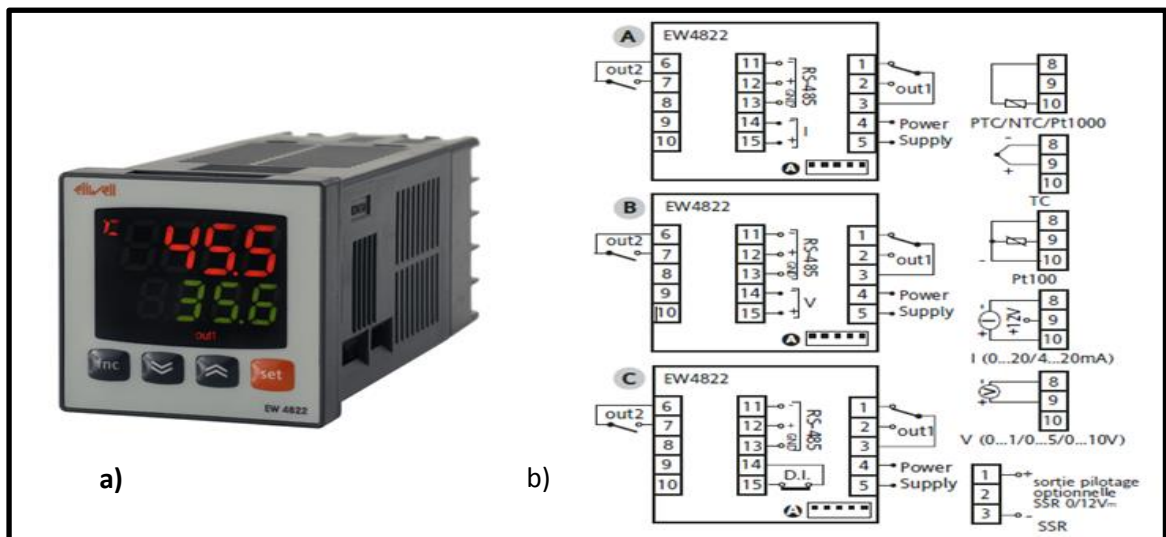


Figure 2.2. Schéma électrique et photo du régulateur de température EW4822.

a Réglage manuel du régulateur EW4822 :

Le régulateur EW4822 contient 4 touches différentes permettant de défiler ses différents paramètres. La **figure (2.3)** expliquera la fonctionnalité de chaque touche [23] :

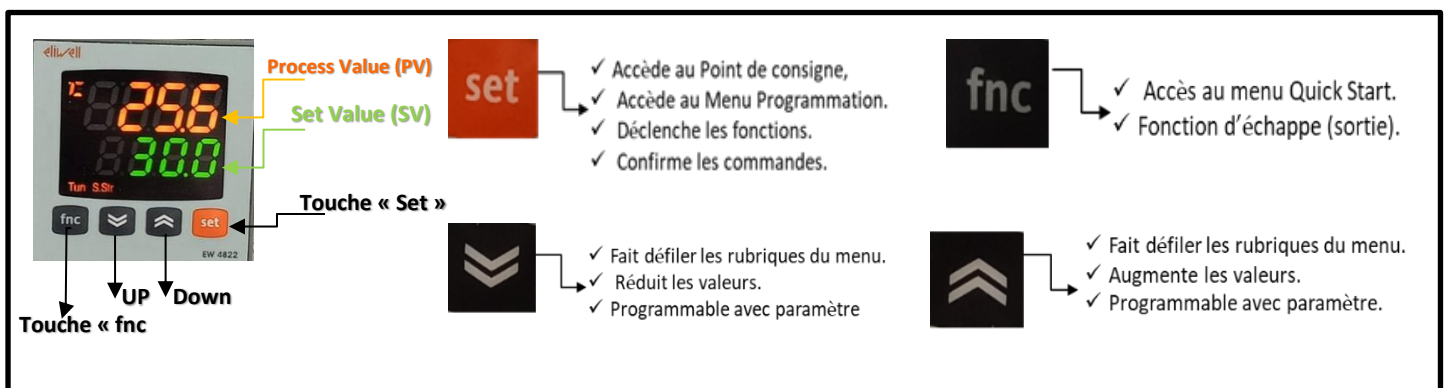


Figure 2.3. Touches composant l'écran du régulateur EW4822.

b Réglage manuel du point de consigne :

Le régulateur EW4822 fonctionne selon deux valeurs de consigne (SET1 et SET2). Pour les programmer, nous devons suivre les étapes suivantes [23] :

- 1) Appuyer et relâcher sur la touche « set » correspondante à la page initiale de l'écran.
- 2) L'afficheur de la valeur mesurée visualise l'étiquette SET1, tandis que l'afficheur de la consigne montre la valeur actuelle du point de consigne (**figure 2.3**). Appuyez de nouveau sur la touche « Set » pour afficher le point de consigne 2 de la même manière.
- 3) Les touches « up » et « down » permettent de modifier la valeur de consigne affichée sur l'écran SV.
- 4) En appuyant sur la touche « set » / « fnc », sinon en attendant 15 secondes pour que la nouvelle valeur sera mémorisée et l'écran affichera à nouveau la page initiale.

c Menus de programmation du régulateur EW4822 :

Le menu de programmation contient tous les paramètres nécessaires à la programmation du fonctionnement de l'instrument. Il est divisé en deux niveaux (L'utilisateur et le fabricant)

La **figure (2.4)**, montre l'affichage des deux niveaux sur le régulateur EW4822 :

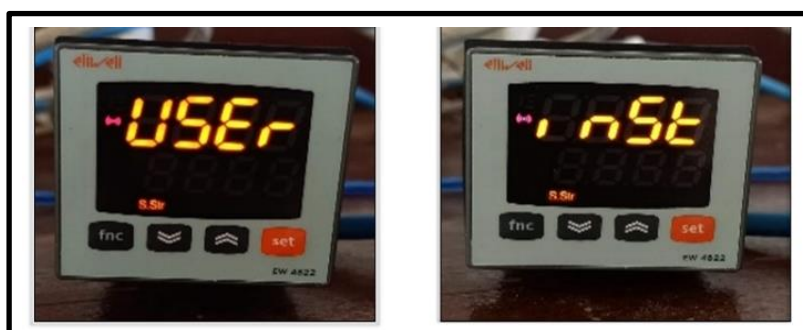


Figure 2.4. Menus de programmation utilisateur et installateur d'EW4822.

Accès au niveau Utilisateur (User)	Accès au niveau installateur (Inst)
<ul style="list-style-type: none"> ✓ Appuyant et relâchant la touche « set » correspondant à l'étiquette User pour accéder aux répertoires contenant les paramètres du niveau utilisateur [25]. 	<ul style="list-style-type: none"> ✓ Agir sur les touches « up » et « down » correspondant à l'étiquette « User », l'étiquette « Inst » sera visualisée. ✓ Appuyant et relâchant la touche « set » correspondant à « Inst » pour afficher les répertoires des paramètres du niveau installateur. [25].

Tableau 2.1. Accès au menu de programmation « User » et « Inst ».

2.2.3 Protocoles de communications :

Dans le monde industriel, il existe de nombreux protocoles de communication qui servent à différents équipements pour communiquer entre eux (**figure 2.5**), mais de quel protocole s'agit-il ? En bref, un protocole est un ensemble de règles préétablies régissant la communication entre un ou plusieurs appareils / services [24].

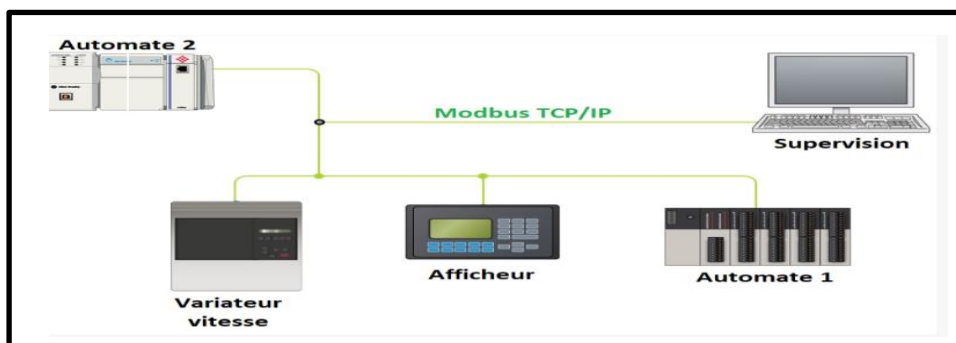


Figure 2.5. Communications entre plusieurs équipements avec le protocole Modbus TCP/IP.

Lorsque deux ou plusieurs appareils industriels communiquent, le protocole spécifie :

- a) Quels types de données peuvent être envoyés.
- b) Comment identifier chaque type de message.
- c) Quelles actions les participants à la conversation peuvent ou doivent entreprendre.

Parmi les protocoles de communication on citera [25] :

- Le Modbus RTU.
- Le Modbus ASCII.
- Le Modbus TCP/IP.
- Le Profibus DP.
- Le CAN.
- Le Profinet.
- Ethernet/IP.
- Ethercat.
- CC-Link.
- Power Link.
- BACnet/IP.
- DeviceNet.
- CANopen.
- ControlNet.
- BACnet MS/TP.

Parmi les protocoles suscités, nous développons une étude du protocole Modbus puisque le régulateur utilisé à savoir le EW4822 intègre le RS-485 Modbus pour communiquer.

a Protocole Modbus :

Modbus est un protocole de communication industriel qui a été développé en 1979 par Modicon (maintenant Schneider Electric) pour communiquer avec plusieurs appareils sur une seule paire torsadée. Le système original fonctionnait sur RS-232, mais a été modifier pour fonctionner avec RS-485 pour des vitesses plus élevées, des distances plus longues et dans un réseau multipoints.

Modbus est devenu l'automatisation industrielle de facto et Modicon le rend accessible au public en tant que protocole gratuit. C'est un système maître-esclave dans lequel le <<maître>> interagit avec un ou plusieurs <<esclaves>>. Le maître est typiquement un PLC (contrôleurs logiques programmables), des ordinateurs, des DCS (système de contrôle distribué) ou des RTU (unité terminale distante). Ce protocole peut être implémenté via des connexions série (RS232, RS422, RS485) ou Ethernet (Modbus TCP/IP). Les protocoles de liaisons série sont Modbus RTU et Modbus ASCII [25].

b Principe de fonctionnement du Modbus :

Dans le protocole Modbus, Les appareils esclaves du réseau Modbus ne peuvent pas initier la communication, ils ne peuvent que répondre. En d'autres termes, ils ne parlent que lorsqu'ils communiquent avec eux, c'est donc le maître qui demande les actions à l'esclave ou à tous les esclaves (**figure 2.6**). L'esclave est identifié par son adresse unique

(des esclaves ne peuvent pas avoir la même adresse qu'un d'autre qui est déjà connecté au bus). Le maître envoie la requête et attend la réponse de l'esclave adressé. Deux esclaves ne peuvent pas communiquer [26].

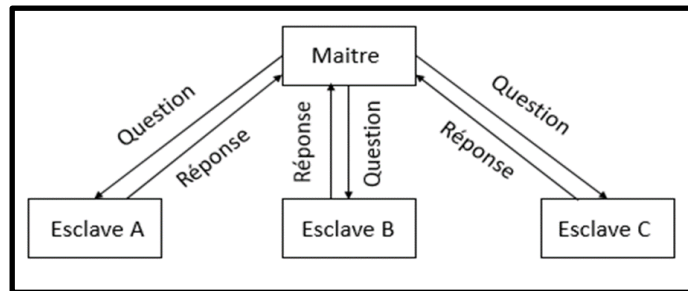


Figure 2.6. Principe de fonctionnement du protocole Modbus.

c Types de données Modbus :

Il existe deux types de données Modbus, on trouve les bobines et les registres [26] :

- **Les bobines** : ce sont des variables booléenne (bits) qui peuvent être activés (1) ou désactivés (0). Certaines bobines représentent des entrées, ce qui signifie qu'elles contiennent l'état de certaines bobines entrées physique discrètes (tout ou rien) ou signifie sortie, c'est-à-dire qu'ils maintiennent l'état du signal de sortie physique.
- **Les registres** : Ce ne sont que des données de registre non signées de 16-bits. La valeur du registre peut être de 0 à 65535 (de 0 à FFFF en hexadécimal). Il n'y a pas de représentation pour les valeurs négatives ni pour les valeurs supérieures à 65535 et c'est pareil pour les données réelles telles que 200.125. Les registres sont divisés en registre d'entrée et registre de stockage. Comme les bobines d'entrée, les registres d'entrée indiquent l'état de l'entrée externe, le format de la valeur est compris entre 0 et 65535.

La valeur de l'entrée analogique est initialement prise en compte par registre d'entrée. C'est une représentation numérique d'un signal analogique, soit une tension ou un courant. La plupart des appareils Modbus actuels ne sont pas des équipement d'E/S et les registres d'entrée fonctionnent exactement de la même manière que les registres de stockage. Les registres de stockage ont été conçus à l'origine comme stockage de programme temporaire pour des appareils tels que les contrôleurs Modbus. Actuellement, les registres de stockage sont utilisés pour le stockage des données de l'appareil.

d Modes de transmissions en Modbus :

Le protocole Modbus a deux modes de transmission différents : RTU (Remote Terminal Unit) et ASCII (American Standard Code for Information Interchange) [26].

Tous les messages Modbus sont envoyés dans le même format, et la seule différence entre les deux types est la méthode du codage [25].

Tous les nœuds d'un même réseau Modbus doivent être du même mode de transmission, ce qui signifie que Modbus ASCII ne peut pas communiquer avec Modbus RTU et vice versa.

- **Modbus ASCII** : Lorsque Le contrôleur est configuré pour les communications ASCII sur un réseau Modbus, chaque octet de 8 bits du message est transmis sous forme de deux caractères ASCII. Le principal avantage de ce mode est qu'il nous permet d'avoir un intervalle de temps jusqu'à 1 seconde entre les caractères sans erreur [27].
- **Modbus RTU** : Lorsque le contrôleur est configuré pour communiquer sur le réseau Modbus en utilisant RTU, chaque octet de 8 bits du message contient deux 4-bits caractères hexadécimaux. Les principaux avantages de ce mode sont : pour le même débit en bauds, la densité de caractères peut fournir un meilleur débit de données que l'ASCII. Chaque message doit être transmis dans un flux continu [27].

Le **tableau (2.2)** montre la différence entre le format de chaque octet dans les deux modes de transmissions [27] :

RTU	ASCII
Système de codage : Binaire 8-bits, hexadécimal 0-9, A-F	Système de codage : Hexadécimal, caractères ASCII 0-9, A-F
Deux caractères hexadécimaux contenus Dans chaque champ de 8-bits du message	Un caractère hexadécimal contenu dans chaque caractère ASCII du message.
Bit/octet : <ul style="list-style-type: none"> - 1bit de Start - 8-bits de données, le bit le moins significatif envoyé en premier. - 1-bit pour la parité : Paire/impaire/pas de bit pour aucune parité. - 1-bit de stop si la parité est utilisée ,2-bit s'il n'y a pas de parité. 	Bit/octet : <ul style="list-style-type: none"> - 1bit de Start. - 7 bits de données, le bit le moins significatif envoyé en Premier. - 1-bit pour la parité : Paire/impaire/pas de bit pour aucune parité.
Des flux constants de message doivent être envoyés.	Autorise des intervalles impliquant les octets d'un message avec une portée maximale de seulement un 2ème.
Champ de contrôle d'erreur : Contrôle de redondance cyclique (CRC)	Champ de contrôle d'erreur : Contrôle de redondance longitudinale (LRC)

Tableau 2.2. Comparaison RTU/ASCII.

Dans notre projet on s'intéresse par le mode de transmission Modbus RT.

e Trame d'échange requête/réponse pour Modbus RTU :

La trame Modbus RTU se compose d'une série de caractères Hexadécimal et contient les informations montrées sur la **figure (2.7)** [28] :

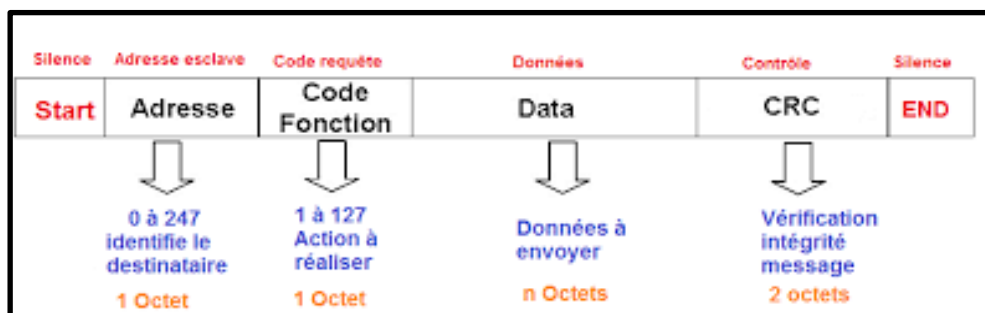


Figure 2.7. La trame Modbus RTU.

Afin de communiquer avec l'appareil esclave, le maitre envoie un message contenant [25] :

- Un bit de Start (silence).
- L'adresse de l'esclave : est un nombre compris entre 0 et 247. Les messages envoyés à l'adresse 0 (message de diffusion) peut être reçus par n'importe quel appareil esclave, mais les numéros 1 à 247 sont des adresses spécifiques. À l'exception des messages diffusés, l'esclave répond toujours aux message Modbus afin que le maitre sache qu'un message a été reçu.
- Le code de fonction : il définit les commandes que l'esclave doit exécuter, telles que la lecture de données, la réception de donnée, l'état du rapport, etc. la plage de codes de fonction va de 1 à 255. Certains codes de fonction ont également d'autres codes de fonction, le tableau suivant montre les codes de fonctions Modbus RTU [25] :

Code de fonction	Description
01	Lecture de n bits de sortie consécutifs
02	Lecture de n bits d'entrée consécutifs
03	Lecture de n mots de sortie consécutifs
04	Lecture de n mots d'entrée consécutifs
05	Ecriture de 1 nit de sortie
06	Ecriture de 1 mot de sortie
07	Lecture du statut d'exception
08	Accès aux compteurs de diagnostique
.	
13	Reset de l'esclave après erreur non recouverte

Tableau 2.3. Codes de fonction Modbus RTU.

- Un champ de données : Les données représentent l'adresse de la carte mémoire de l'appareil de la fonction de lecture et contiennent la valeur des données qui doit être stockée dans la mémoire de l'appareil ou d'autres informations nécessaires pour exécuter la fonction demandée.

- Un champ de contrôle d'erreur : la vérification des erreurs est une valeur numérique de 16-bits qui signifie le contrôle de redondance cyclique (CRC). Si les valeurs CRC ne correspondent pas, le périphérique demande la retransmission du message. Certains systèmes peuvent même effectuer une vérification urgente.

Lorsque l'appareil esclave exécute la fonction demandée, il renvoie un message au maître. Le message de retour contient l'adresse de l'esclave et le code fonction (pour que le maître sache qui répond), les données demandées et une valeur de vérification d'erreur.

2.2.4 Zone mémoire :

Chaque appareil Modbus dispose d'une mémoire pour stocker des données variables. Modbus définit la façon comment récupérer des données et de quels types de données s'agit-il. Cette mémoire est organisée en 4 groupes comme le montre le **tableau 2.4** [25] :

Le type	Plage d'adresse	Accès
Bobine	00001 à 09999	Lire-écrire
Entrée discrète	10001 à 19999	Lire seulement
Registre d'entrée	30001 à 39999	Lire seulement
Registre général	40001 à 49999	Lire-écrire

Tableau 2.4. Zones mémoire des appareil Modbus.

- **Zone des bobines (coils) :** la zone de stockage est organisée en bits (de 00001 à 09999), qui contient la valeur de la sortie TOR discrète.
- **Zone des entrée discrète (discret input) :** la zone est organisée en bits, c'est la valeur de l'entrée TOR, la plage d'adresses est comprise entre 10001 et 19999.
- **Zone des registres d'entrée (input registres) :** Est un registre 16-bits contient la valeur de l'entrée analogique, qui occupe les adresses 30001 à 39999.

- **Zone des registres généraux (Holding registers) :** il est organisé en 16 bits et occupe des adresses comprises entre 40001 et 49999 où se trouve la valeur de sortie.

2.2.5 Techniques de transmissions :

Il existe deux techniques de transmissions, une transmission parallèle et une transmission série [28] :

- **Transmission parallèle :** Cette méthode de transmission de données consiste à envoyer des bits entre l'émetteur et le récepteur sur plusieurs canaux simultanément (**figure 2.8.a**), et nous avons la possibilité d'envoyer des données 8 fois plus vite que progressivement : c'est l'un de ses avantages majeurs.
- **Transmission série :** Le procédé de transmission de données implique l'envoi de bits consécutifs entre un émetteur et un récepteur via un seul connecteur (**figure 2.8.b**) Dans ce type de transmission, chaque caractère est précédé d'un bit de début (appelé bit de Start) et d'un bit de fin (bit de Stop). Par conséquent, ces bits arrivent au récepteur un par un de manière synchrone ou asynchrone, et la différence entre les deux manières se présentes dans le signal de synchronisation (appelé horloge). Ce dernier est nécessaire à l'émetteur et au récepteur dans la méthode synchrone avant la transmission des données, alors que la méthode asynchrone ne nécessite pas d'horloge mais ajoute un bit de parité aux données avant la transmission.

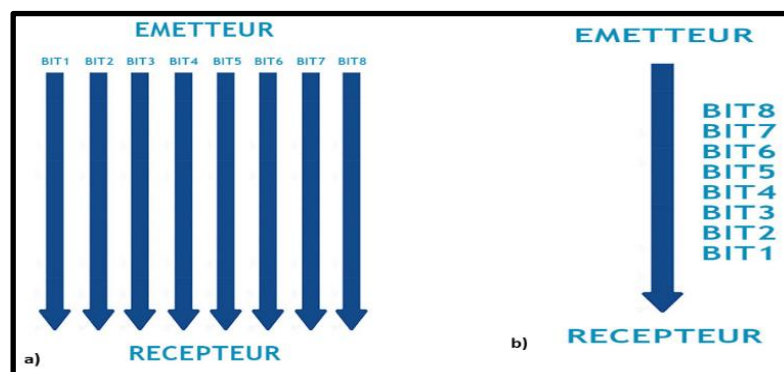


Figure 2.8 Techniques de transmission parallèle et série.

a Types de transmission :

Selon le sens des échanges, on distingue 3 types de transmission [29] :

- **La liaison simplex** : ce type de communication est unidirectionnel et peut être utilisé lorsque l'émetteur n'a pas besoin d'obtenir une réponse du récepteur.
- **La liaison Half-duplex ou alterné** : dans la communication half-duplex, deux systèmes interconnectés sont capables d'émettre et de recevoir chacun leur tour, donc il permet d'avoir une liaison bidirectionnelle utilisant la capacité totale de la ligne.
- **La liaison Full-duplex ou duplex intégral** : ce type représente une liaison dans lequel les données circulent dans les deux sens simultanément. Par conséquent, chaque extrémité de la ligne peut envoyer et recevoir en même temps, ce qui signifie que la bande passante sera divisé par deux dans chaque sens de transmission de données, si le même support de transmission est utilisé dans les deux transmissions.

2.2.6 Ports de connexion et quelque standards électrique :

En regardant l'ordinateur, on peut remarquer qu'il existe différents types de connecteurs appelés <<ports>>. Ceux-ci permettent de connecter de nouveaux périphériques à l'ordinateur comme le montre la **figure (2.9)** [31] :

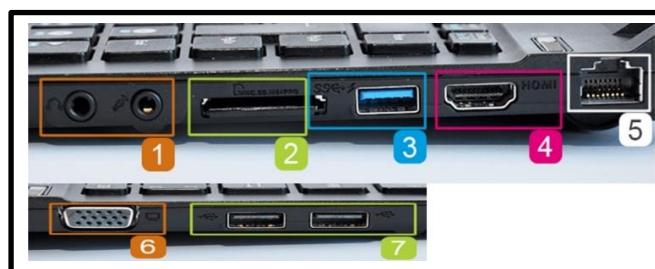


Figure 2.9. Ports de connexion d'un PC portable.

- 1) Une prise pour brancher un casque et une pour un micro.
- 2) Un lecteur de cartes mémoires qui permet par exemple de lire la carte mémoire d'un appareil photo.
- 3) Un port USB 3.0 pour brancher un disque dur externe.
- 4) Un port HDMI qui permet par exemple de relier l'ordinateur à une TV moderne.

- 5) Un port Ethernet pour brancher un câble donnant accès à un réseau et à internet.
- 6) Un port VGA pour connecter l'ordinateur à un projecteur.
- 7) Des ports USB 2.0 pour brancher une souris, un disque dur externe, une clé USB...

a Ports série :

Un port série est une interface de communication pour la configuration série. Sur les ordinateurs compatibles IBM, ils sont généralement appelés <<ports COM>>. Ces ports peuvent être connectés à des périphériques externes qui transmettent des données série à l'ordinateur et permettent une communication bidirectionnelle entre celui-ci et le périphérique [32].

2.2.7 Quelques standards électriques :

Le RS232 et le RS485 sont deux normes pour la transmission électrique qui existaient déjà avant l'avènement des ordinateurs personnels. Elles sont utilisées pour permettre la communication série entre l'ordinateur et les périphériques. Bien qu'ils aient la même fonction (c'est-à-dire permettre la transmission de données en série), le RS232 et le RS485 sont différents [32].

Le **tableau (2.5)** montre la différence entre ces deux Bus :

Paramètres	RS232	RS485
Configuration de ligne	Asymétrique	Différentielle
Mode de fonctionnement	Simplex ou full-duplex	Simplex ou half-duplex
Longueur de câble maximale	15.mètre	1200mètre
Débit de données maximale	20 Kbits/s	10 Kbits/s
Niveaux logiques typiques	±5 à ±15V	±1.5 à ±6V
La valeur maximale d'impédance d'entrée du récepteur.	3 à 7 Ω	12Ω
Sensibilité du récepteur	±3V	±200mV

Tableau 2.5. Différences entre RS232/RS485.

Le régulateur EW4822 intègre un port série RS-485, pour cela nous intéressons à faire la lumière sur ce type de liaison.

a Liaison RS-485 :

C'est l'interface série la plus utilisée dans l'industrie. Par rapport à la liaison RS-232, elle présente des avantages majeurs, une topologie multipoint où on peut connecter plusieurs émetteurs et récepteurs (32 périphériques) (**figure 2.1.**). La transmission des données s'effectue à l'aide de signaux différentiels pour plus de fiabilité [33].

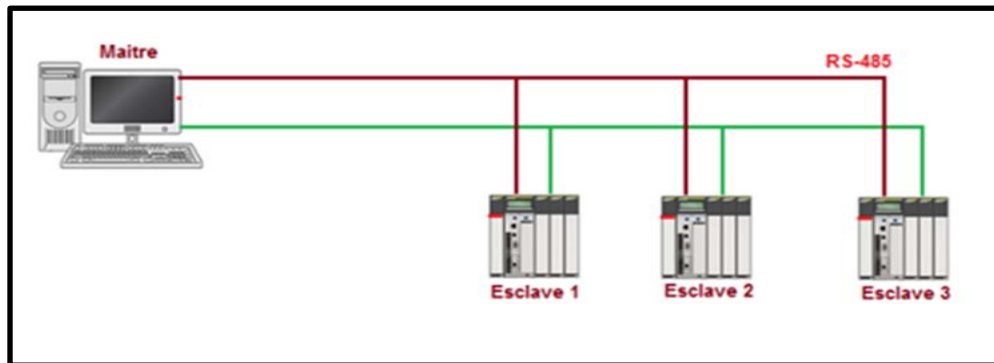


Figure 2.10. Connexion d'un maitre et plusieurs esclaves utilisant la liaison RS-485.

b Adaptation de la liaison RS-485 avec un PC :

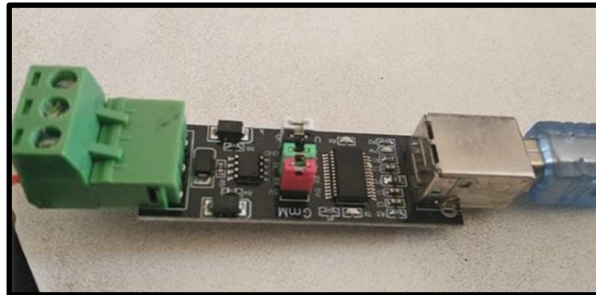
En général les PC grand public ne comportent pas une liaison direct RS-485, pour établir la connexion RS-485 avec le port COM. Il est alors recommandé d'utiliser le Port USB, généralisé actuellement sur tous les ordinateurs, donc il faut utiliser un convertisseur adaptateur USB/RS-485 avec une puce FTDI.

c Convertisseur adaptateur USB/RS-485 :

Le module d'interface adaptateur BUS TTL-RS485 est un modèle simple et pratique qui peut convertir différents formats de port série, son rôle est d'établir la connexion aux systèmes de télégestion télévisé ou Modbus. Parmi ces caractéristiques on citera [24] :

- Adaptateur de série compact. Il permet de convertir différents formats de port série. Modèle simple et pratique, idéal pour les intégrateurs de systèmes, l'électronique, etc.
- Convertisseur USB vers TTL RS485.
- Connecteur USB : Type B-femelle.
- Connecteur série RS485 : bornier à 3 broches.
- Taille : 77 x 19 x 18 mm [32].

Le convertisseur série USB vers RS485 TTL (**figure 2.11**) utilise un port COM série virtuel pour convertir les signaux de communication série bidirectionnelle entre RS485 et le port USB d'un ordinateur personnel. Il est auto-alimenté via USB, utilise FT232RL + 75176 et peut contenir jusqu'à 32 périphériques sur le bus [33].



Figur 2.11. USB to TTL/RS485 adaptateur.

2.3 Partie logicielle :

Parmi les différents langages de programmation utilisés en développement instrumental, on distingue des langages graphiques basés sur la perception visuelle. On citera quelques langages fameux telle que [34] : (Grafcet, Proteus, Simulink (MATLAB), LabVIEW).

Il s'agit de collecter des éléments graphiques et écrire des programmes pour réaliser des fonctions dédiées à des tâches précises.

2.3.1 Logiciel LabVIEW :

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un langage, spécifiquement utilisé pour le contrôle d'instruments et l'analyse de données, créé par National Instruments, sa particularité est basée sur le langage graphique G [34] [35]. Sa programmation consiste à composer certaines phrases caractères graphiques et de texte qui sont disposés spatialement dans des procédures formelles. Le concept de base est d'utiliser des cases et des flèches. Les cases sous forme de boîtes sont traitées comme des unités liées par ces flèches ou des lignes représentant des relations sous forme de câbles pour transmettre le flux de données.

2.3.2 Environnement LabVIEW :

LabVIEW est basé sur le principe d'instrument virtuel (Virtual Instrument où

Encore VI). Il se décompose en trois parties (**figure 2.12**) [34] :

- La face avant.
- Le diagramme.
- Icône/connecteur.

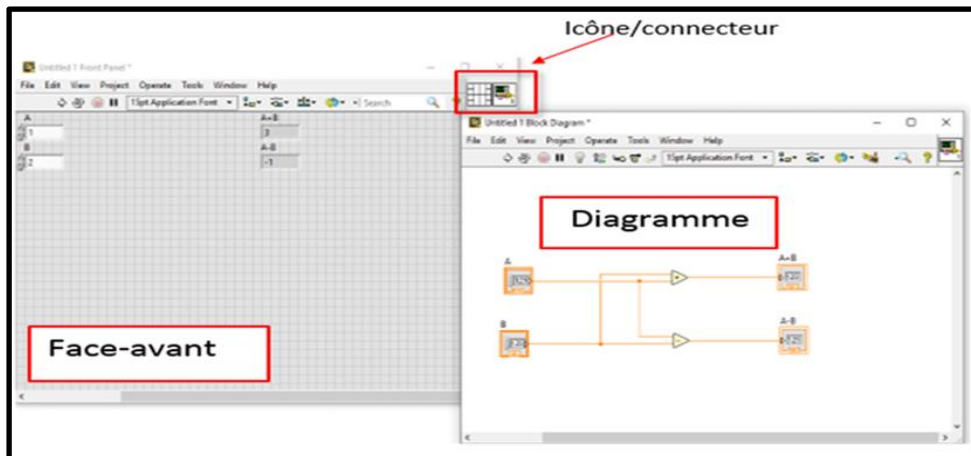


Figure 2.12. Composantes d'un VI.

a Face-avant :

La face avant est l'interface utilisateur, elle comporte : Les contrôles qui simulent les entrées des instruments virtuels et fournissent les données au diagramme. Les indicateurs qui simulent la réponse des instruments et affichent les données acquises engendrées par les Vis (**figure 2.13**) [34].

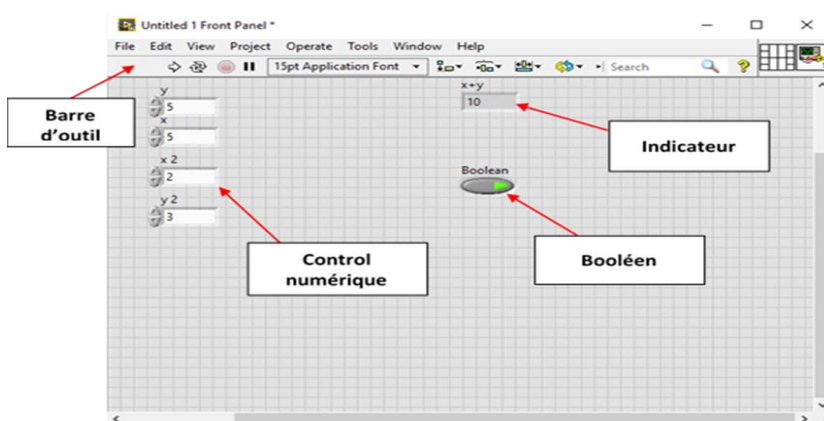


Figure 2.13. Face-avant.

b Bloc diagramme :

Le bloc diagramme (**figure 2.14**) est chargé de mettre en relation les entrées et les sorties et les différents éléments fonctionnels du programme, Il se compose de [34] :

- **Les nœuds** : sont des objets sur le diagramme. Ils possèdent des entrées et/ou des sorties. Ils effectuent des tâches spécifiques lorsqu'un VI fonctionne, ils sont équivalents à des fonctions dans les langages textuels.
- **Les terminaux** : ils sont situés dans la face avant apparaissent comme des terminaux dans le diagramme. Ils représentent des ports de communication entre la face avant et le diagramme. Ils sont équivalents aux paramètres et aux constantes dans les langages textuels.
- **Les fils** : l'ensemble des terminaux et des nœuds sont mis en relation par des fils pour transférer les données dans le diagramme. Ils sont analogues aux variables dans le langage textuel.
- **Les Structures** : plus communément connues sous l'appellation « boucle ». Un programme ne peut être efficace qu'à conditions de disposer de mécanismes d'itération, dans le cas du LabVIEW les plus utilisés les boucles « For et While » ; des structures conditionnelles et de séquence.

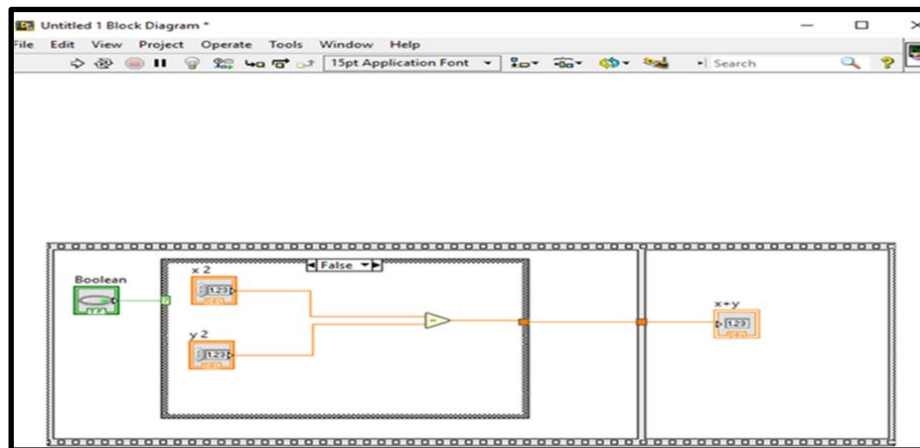


Figure 2.14. Bloc diagramme.

c Barres d'outils :

Dans un langage de programmation graphique tel que LabVIEW, la souris est le moyen principal d'interagir avec l'environnement de programmation. Ceci dit, nous présentons les différents outils LabVIEW :

- **Barres d'outils de la face-avant et du bloc diagramme :**

Les **figures (2.15, 2.16)** représentent la barre d'outils du bloc diagramme et de la face-avant :

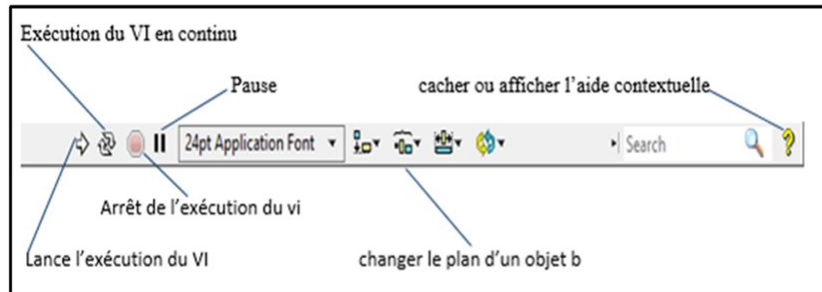


Figure 2.15. Barre d'outils de la face-avant.

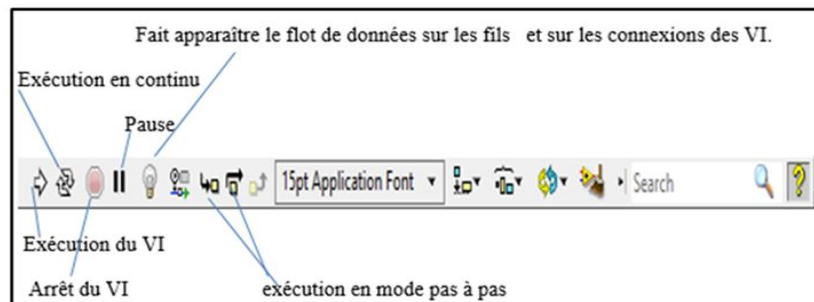


Figure 2.16. Barre d'outils du diagramme.

- **Palettes :**

Les palettes LabVIEW nous fournissent les outils qui nous permettent de créer et d'éditer la face avant ou le diagramme de la face arrière, on trouve trois palettes [34] :

- **Palette d'outils :**

A partir de la palette d'outils (**figure 2.17**) LabVIEW choisit quels outils sélectionnés en fonction de l'emplacement actuel de la souris.



Figure 2.17. Palette d'outils.

- **Palette de Commande :**

La palette Commande (**figure 2.18.a**) est uniquement accessible depuis la face avant Elle permet de créer toutes les commandes et indicateurs quels que soient leurs types. Chaque icone de la palette correspond à une rubrique qui permet d'accéder à une fenêtre secondaire contenant l'ensemble des éléments relatifs à cette rubrique utilisable sur la face avant par exemple [34].

- Tableaux.
- Clusters.
- Graphiques.
- Types spéciaux.
- Nombres.
- Booléens.
- Chaînes de caractères

- **Palette de fonction :**

La palette de Fonction (**figure 2.18.b**) est seulement accessible sur le diagramme, Elle permet de créer toutes les fonctions LabVIEW. Chaque icone de la palette correspond à une rubrique qui permet d'accéder à une fenêtre secondaire contenant l'ensemble des éléments relatifs à cette rubrique utilisable sur le diagramme on cite [34] :

- Structures.
- Opérations numériques.
- Opérateurs logiques.
- Comparaisons
- Fonctions temporelles.
- Entrées/Sorties.
- Acquisition.
- Traitement du signal.



a)



b)

Figure 2.18. Palette de commande et de fonction.

- **Fenêtre d'aide contextuelle :**

Donne des informations relatives à l'objet situé sous le curseur de la souris, peut-être une information sur la nature d'une liaison, le type de données à fournir à un indicateur, le fonctionnement d'un sous VI ou d'une fonction, (**figure 2.19**) [34].



Figure 2.19. Fenetre d'aide contextuel.

d Types de données utilisées sous LabVIEW :

Comme tout langages de programmation, le logiciel LabVIEW a différents types de données et manières pour les organiser (**tableau 2.6**). On les distingue par leur couleur ou décoration, et ils sont tous utilisés pour des raisons précises [36]

Types de données	Scalaire	Tableau 1D	Tableau 2D	Couleur
Numérique – Virgule flottante				Orange
Numérique – Entier				Bleu
Booléen				Vert
Chaîne				Rose

Tableau 2.6. Types de données sous LabVIEW.

2.4 Conclusion :

Dans ce chapitre nous avons présenté l'étude des parties matérielle et logicielle qui nous permettra de réaliser notre travail expérimental et valoriser les résultat objet du chapitre suivant.

L'étude de la partie matérielle a concernée le dispositif expérimental développé au CDTA et les modules à rajouter pour contrôler et réguler la température selon une consigne désirée.

Concernant la partie logicielle elle s'est concentré sur le logiciel graphique LabVIEW afin de maîtriser sa mise en œuvre pour la commande et le contrôle de la température.

Chapitre 3 Implémentation et résultats

3.1 Introduction :

Ce dernier chapitre sera dédié aux résultats obtenus lors de l'implémentation de programmes de contrôle et de commande de notre régulateur EW4822 sous le logiciel LabVIEW via le PC à travers un port COM avec un adaptateur convertisseur USB/RS-485. La réalisation de cette implémentation est concrétisée une fois que nous avons câbler le dispositif de test (le PC, l'adaptateur convertisseur USB/RS-485 et le régulateur EW4822). Ensuite nous confirmons le fonctionnement du port COM au niveau PC par un réglage software et enfin l'ensemble avec l'adaptateur convertisseur USB/RS-485, et le Modbus RS-485 du régulateur.

3.2 Câblage du Dispositif de test :

Avant de passer au réglage on doit s'assurer du câblage et des connexions des E/S du système régulateur de température et le PC via le convertisseur adaptateur RS-485 USB. La **figure (3.1)** présentera l'assemblage des éléments constituant le dispositif de test :

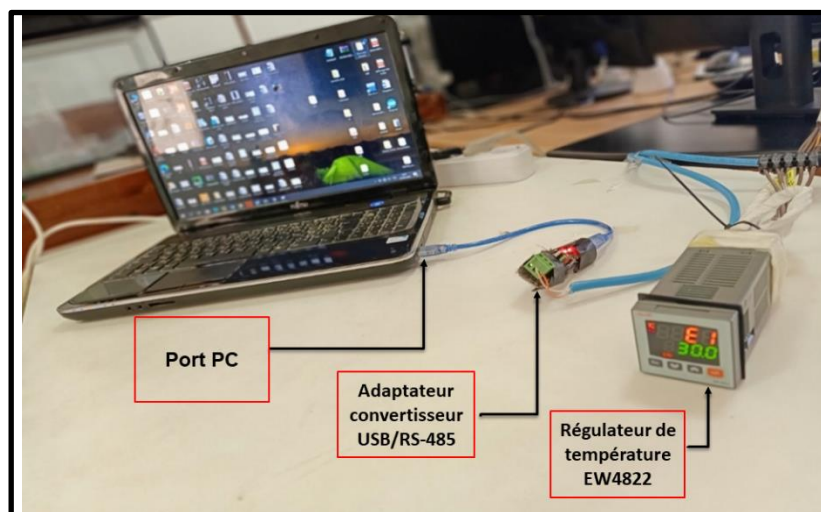


Figure 3.1. Dispositif de test.

- Le standard de transmission du BUS RS-485 se fait par deux fils (Half-duplex). Le régulateur EW4822 et le convertisseur adaptateur RS485/USB sont censés être câblés (+) à (+) et (-) à (-) (**figure 3.2**)

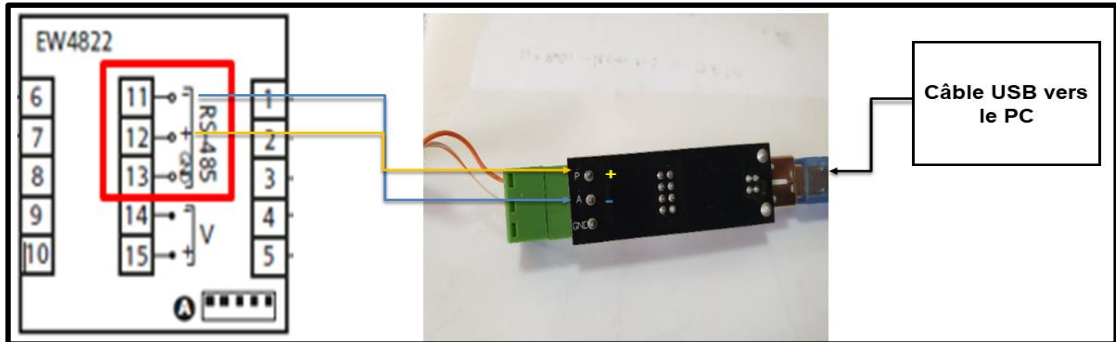


Figure 3.2. Câblage schématisé du dispositif de test.

3.3 Réglages des éléments du dispositif :

Nous allons récapituler tous les paramètres qu'on doit régler selon les trois niveaux permettant d'établir une connexion en lecture/écriture sur le tableau suivant (**tableau 3.1**) :

Paramètres	EW4822	Port COM	LabVIEW
L'adresse de l'esclave	01	Aucun réglage	01
Débit en Baud	Aucun réglage	9600	9600
Protocole	MODBUS	MODBUS RTU	MODBUS RTU
Parité	Aucune	Aucune	Aucune
Bit d'arrêt	1-bit	1-bit	1-bit
Contrôle de flux	Aucun réglage	Aucun	Aucun

Tableau 3.1. Réglages des paramètres dans les 3 niveaux.

3.3.1 Réglage des paramètres du régulateur EW4822 :

Comme nous l'avons décrit dans le deuxième chapitre, notre régulateur de température EW4822 possède deux étiquettes principales <<User>> et <<Inst>>. Ces deux dernières nous permettent d'accéder à différents paramètres et de régler ce qui est nécessaire

pour établir la communication entre l'instrument et le PC. L'entrée des valeurs qui correspond à notre objectif se fera en utilisant le menu <<User>> (figure 3.3).

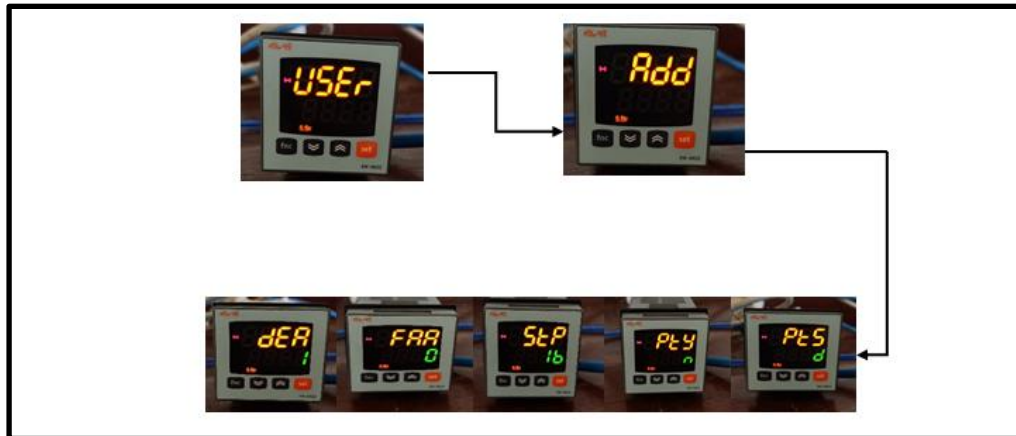


Figure 3.3. Réglages des paramètres du régulateur EW4822 à partir de l'étiquette Add

L'étiquette User :

- Après avoir cliqué sur le bouton **set** pendant 3 secondes, le menu principale <<User>> s'affiche sur l'écran du régulateur.
- Cliquons et relâchons sur le bouton **set** correspondant à l'étiquette <<User>> cela nous permet d'accéder au répertoire contenant tous les paramètres de ce niveau.
- Nous choisissons l'étiquette <<Add>>.

L'étiquette Add :

- Nous cliquons et relâchons à nouveau la touche **set**.
- On obtiendra par suite le dossier correspondant à cette étiquette.
- Pour basculer entre les paramètres, nous continuons à cliquer sur la même touche **set**.
- Pour définir les valeurs que nous voulons entrer dans chaque paramètre, il suffit de cliquer sur les touches **▲** et **▼**.

Le **tableau (3.2)** résume la description des paramètres à choisir et le réglage nécessaire :

Paramètres	Description	Réglage
PtS	Sélection protocole : t= Televis ; d=MODBUS	d
PtY	Bit de parité MODBUS : n=NONE ; E=Even ; o=odd	n
StP	Bit de stop MODBUS: 1b= 1-bit; 2b=2-bits	1b
DEA **	Indice du dispositif à l'intérieur de la famille (0 à 14)	1
FAA**	Famille du dispositif (valeur de 0 à 14)	0

Tableau 3.2. Description et réglage nécessaire des paramètres de l'étiquette ADD.

** : Les deux valeurs FAA et DEA représentent l'adresse du dispositif et sont indiquées au format <<FF : DD>> où FF=FAA ET DD= DEA. Dans notre cas on a pris FAA=0 et DEA=1. Donc l'adresse du régulateur sera 00 :01.

3.3.2 Réglages du port COM :

Après avoir connecté le convertisseur USB/RS-485 avec le régulateur EW4822 et assuré que le câblage est bien fait, ainsi que le branchement du convertisseur USB/RS-485 au PC. Ce dernier le reconnaît comme un nouveau périphérique, nous aurons par suite un USB/RS-485 qui s'installe sur un port COM <<virtuel>> de Windows, ce port COM a besoin d'un numéro (COM°n) qui va être utilisé pour le port COM du maître LabVIEW. Pour savoir sur quel port COM le convertisseur est installé et régler les paramètres qu'on a mentionnés sur le **Tableau 3.1**, nous avons suivi le chemin suivant :

Démarrer → Gestion de l'ordinateur → Gestionnaire de périphérique → Ports (COM et LPT).

Pour accéder au réglage des paramètres du port COM, on fait un clic droit sur <<USB Serial Port (COM8) → propriétés → paramètres de USB serial port → Avancé

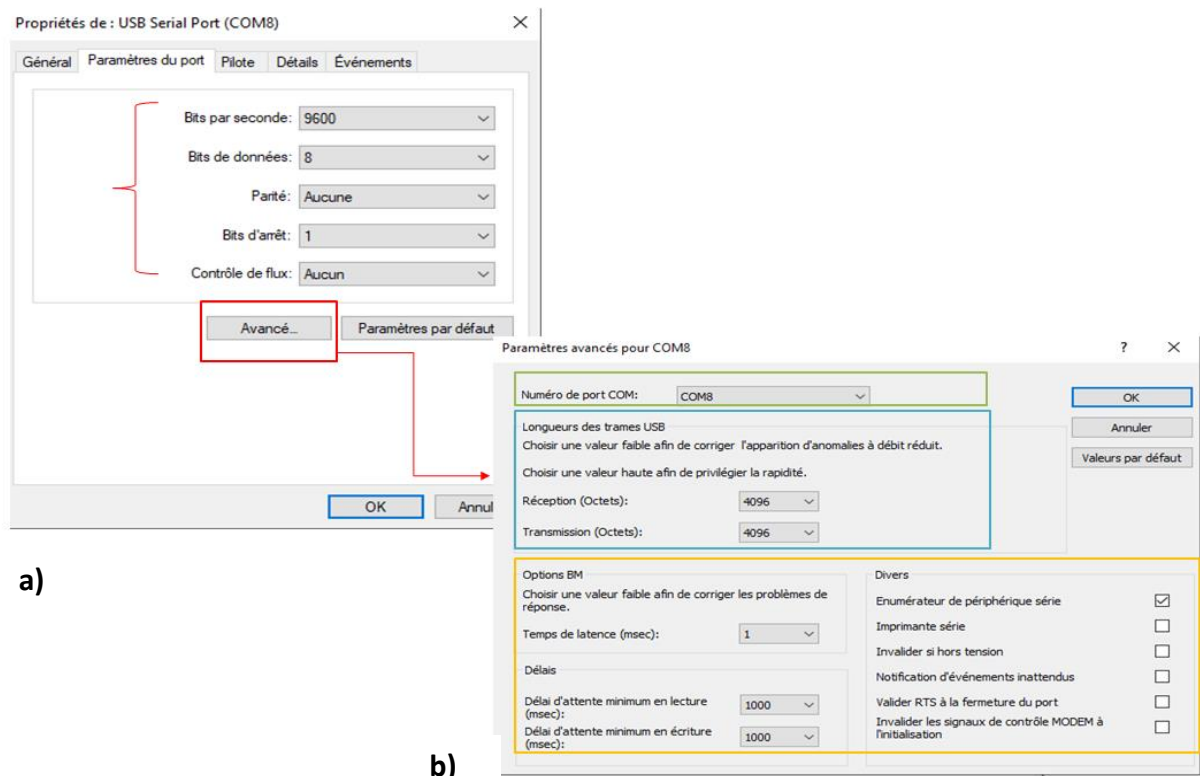


Figure 3.4. Réglages des paramètres du Port COM.

La **figure (3.4.a)** indique des paramètres à régler, à savoir :

- Bits/ seconde : 9600.
- Bits de données : 8.
- Parité : Aucune.
- Bits d'arrêt : 1.
- Contrôle de flux : Aucun.

La prochaine étape consiste à régler les paramètres de la **figure (3.4.b)** :

- Numéro du port COM : COM8
- Longueurs des trames USB : on a pris les valeurs maximales pour la réception et la transmission 4096 Octets pour privilégier la rapidité.
- Temps de latence : on a pris la valeur la plus faible 1 ms
- Le délai d'attente en lecture/écriture : 1000 ms
- Options diverses.

3.4 Tests de connectivité et activation du MODBUS :

Les tests de connectivité peuvent s'effectués grâce à plusieurs logiciels de test qui peuvent être payants ou gratuits. Parmi eux, on peut citer le logiciel Commix 1.4 et le logiciel Modbus Poll qui seront décrits dans les sections suivantes.

3.4.1 Interface de test Commix 1.4 :

Le logiciel Commix 1.4 est un logiciel gratuit qui test toute communication de type Modbus. Il permet de surveiller et d'analyser les trames d'envoi et de réceptions [40]. Son interface, présentée dans la **figure (3.5)**, montre trois panneaux graphiques :



Figure 3.5. Interface du logiciel Commix 1.4.

Description des trois parties :

- **Panneau A** : permet de régler les paramètres de connectivité ; à savoir le numéro du port COM, les bits de données, le débit en baud, la parité, le bit d'arrêt et le champ de contrôle d'erreur (CRC et LRC).
- **Panneau B** : permet de choisir le système de codage (HEX, ASCII) du protocole choisi.
- **Panneau C** : permet d'écrire la trame de requête dans l'espace montré dans la **figure (3.5)**, ensuite nous cliquons sur « open port » → « send » la trame de réponse de l'esclave sera affichée.

3.4.2 Interface de test Modbus Poll :

Modbus Poll est un simulateur (maitre Modbus), principalement utilisé pour aider les développeurs qui souhaitent tester et simuler le protocole Modbus ou d'autres appareils esclaves. Grâce à l'interface multi-documents (plusieurs fenêtres peuvent être ouvertes), on peut surveiller plusieurs esclaves Modbus et/ou Zones de données en même temps [41]. La **figure (3.6)** montre une démonstration de l'interface affichant le résultat de la lecture des registres.

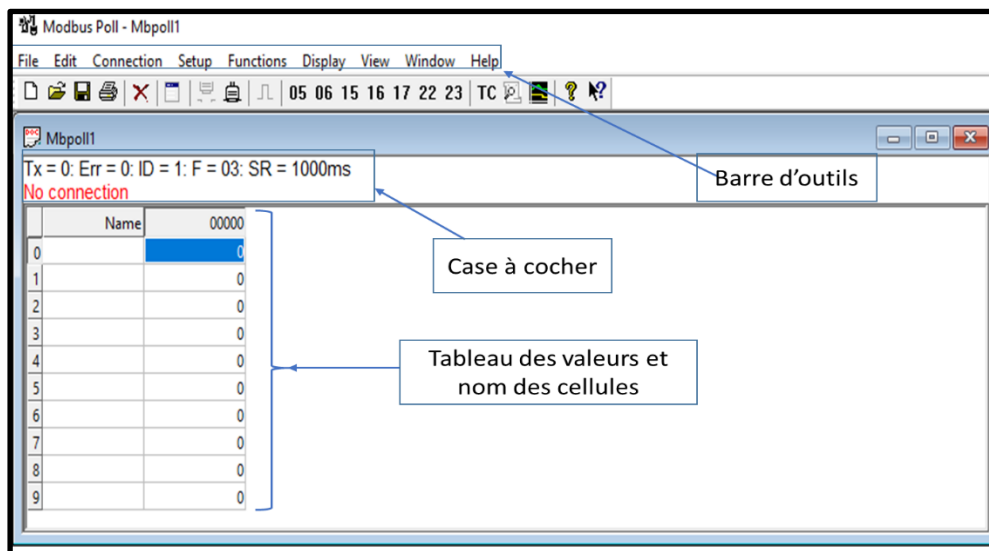


Figure 3.6. Interface Modbus Poll.

3.5 Résultats des interfaces de test :

Dans cette section, nous allons discuter les résultats des tests d'une lecture à partir de l'instrument selon les deux interfaces de test indiqué au paravent.

Parmi les paramètres dont on s'est intéressé sont indiqués dans le **tableau (3.3)** :

Paramètre	Description	Adresse	Action	EXP	Taille des données
SP1	Point de consigne de régulation 1	16392	Lecture/écriture	-1	Mot
bPH	Bande PID proportionnelle H	16714	Lecture/écriture	-1	Mot
tiH	Temps intégral PID H	16716	Lecture/écriture		Mot
tdH	Temps dérivé PID H	16718	Lecture/écriture		Mot
biAH	Polarisation statique PID H	16722	Lecture/écriture	-1	Mot
dEA	Adresse du dispositif	49343	Lecture/écriture		Bit
FAA	Adresse de famille	49344	Lecture/écriture		Bit

Tableau 3.3. Adresses et description des paramètres du régulateur EW4822.

EXP : il s'agit de l'indice multiplicateur qui doit être appliqué pour convertir la valeur lue par le registre en valeurs finales en fonction de l'unité de mesure. Le multiplicateur est calculé en utilisant la fonction exponentielle avec une base de 10 et l'exposant indiqué dans la colonne EXP. Lorsqu'elle n'est pas indiquée, sa valeur est 0.

Les valeurs valides sont les suivantes [42] :

Valeur Multiplicateur correspondant

-2 = 10^{-2} (0.01)

-1= 10^{-1} (0.1)

0= 10^0 (1)

1= 10^1 (10)

2= 10^2 (100)

3.5.1 Tests sur le simulateur commix1.4 :

A titre de test, nous présentons trois situations :

a Lecture de la valeur de consigne :

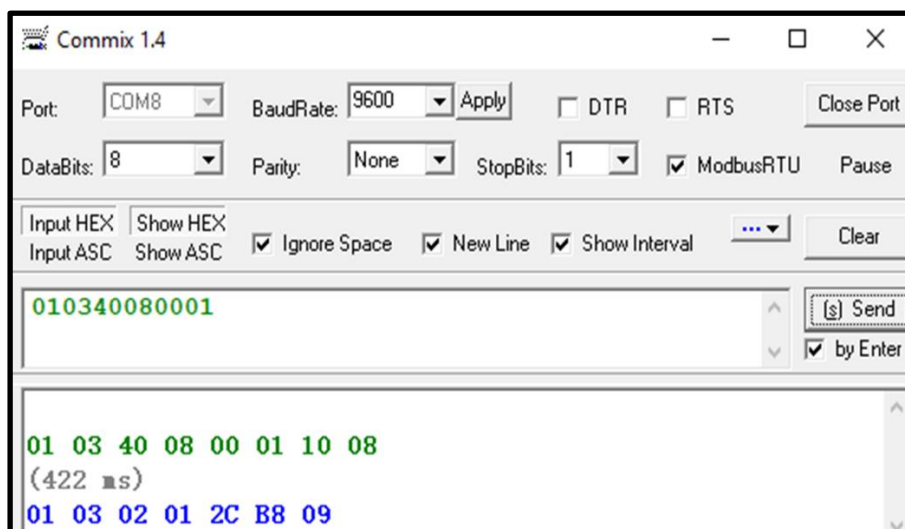


Figure 3.7. Lecture de la valeur de consigne du régulateur EW4822 sur Commix.

- Trame de requête :

01	03	4008	0001	10 08
Adresse du régulateur	Commande de lecture de n mots de sortie consécutifs	Adresse en Hexadécimale du registre point de consigne	Longueur du registre	CRC16 de la trame que le simulateur Commix ajoute automatiquement.

Tableau 3.4. Trame de requête de la lecture du registre du point de consigne.

- Trame de réponse :

01	03	02	01 2C	B8 09
Adresse du régulateur	Commande de lecture de n mots de sortie consécutifs	Nombre de registre qu'il faut lire. 02 donc les 2 prochains blocs de 2 digits.	Valeur en HEX convertie en décimal : 30.0 °C	CRC16 de la trame de réponse.

Tableau 3.5. Trame de réponse de la lecture du registre du point de consigne.

b Lecture de la valeur du paramètre DEA de l'étiquette ADD

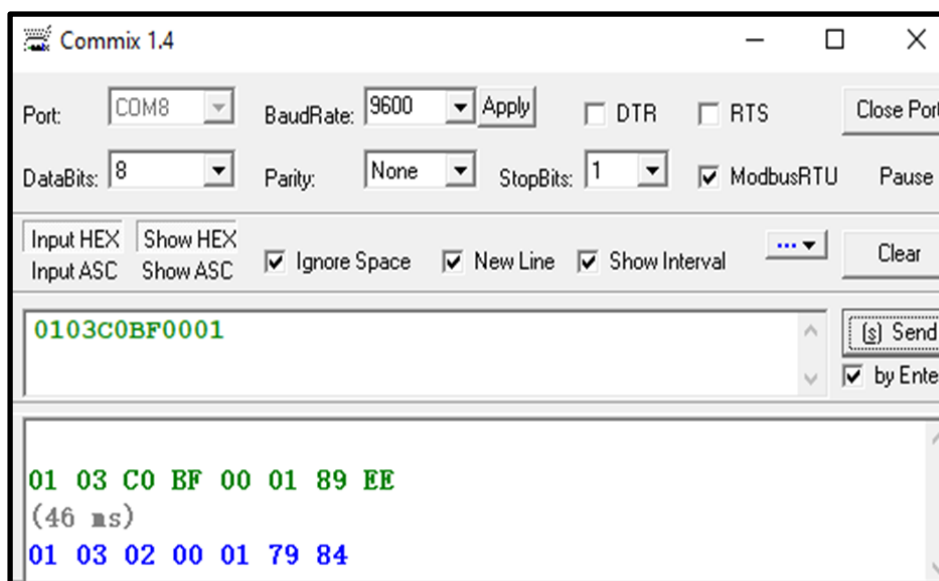


Figure 3.8. Lecture de la valeur du paramètre DEA de l'étiquette ADD sur Commix 1.4.

- Trame de requête :

01	03	C0BF	0001	89 EE
Adresse du régulateur	Commande de lecture de n mots de sortie consécutifs	Adresse en Hexadécimale Du registre DEA	Longueur du registre	CRC16 de la trame que le simulateur Commix ajoute automatiquement.

Tableau 3.6. Trame de requête de la lecture du registre de DEA sur Commix 1.4.

- Trame de réponse :

01	03	02	00 01	79 84
Adresse du régulateur	Commande de lecture de n mots de sortie consécutifs	Nombre de registre qu'il faut lire. 02 donc les 2 prochains blocs de 2 digits.	Valeur en HEX convertie en décimal : 1	CRC16 de la trame de réponse.

Tableau 3.7. Trame de réponse de la lecture du registre de DEA sur Commix 1.4.

c Lecture de la valeur du paramètre FAA de l'étiquette ADD :

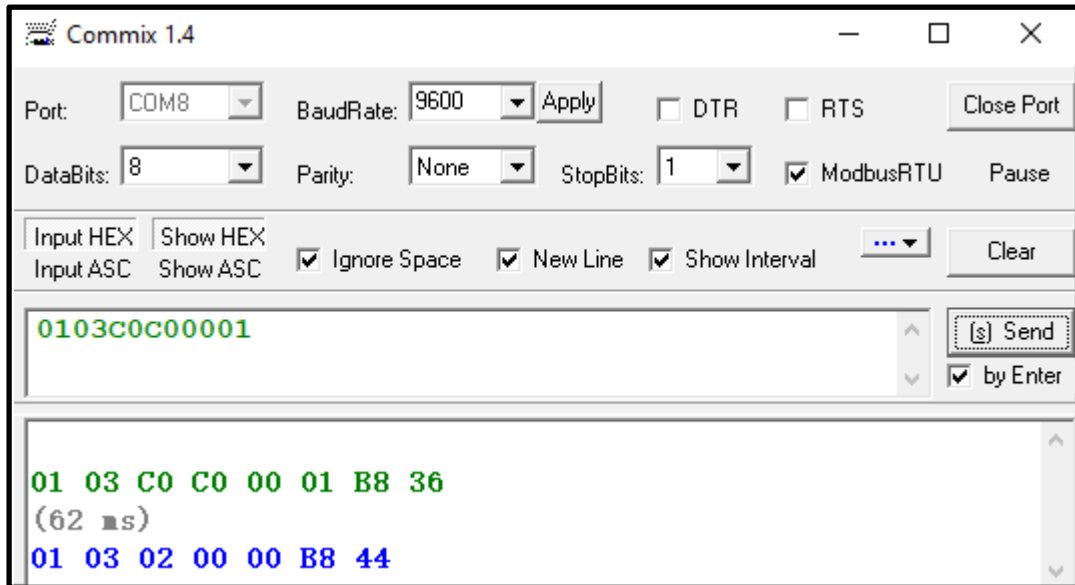


Figure 3.9. Lecture de la valeur du paramètre FAA de l'étiquette ADD sur Commix.

- Trame de requête :

01	03	C0C0	0001	B836
Adresse du régulateur	Commande de lecture de n mots de sortie consécutifs	Adresse en Hexadécimale du registre FAA	Longueur du registre	CRC16 de la trame que le simulateur Commix ajoute automatiquement.

Tableau 3.8. Trame de requête de la lecture du registre de FAA sur Commix 1.4.

- Trame de réponse :

01	03	02	00 00	B8 44
Adresse du régulateur	Commande de lecture de n mots de sortie consécutifs	Nombre de registre qu'il faut lire. 02 donc les 2 prochains blocs de 2 digits.	Valeur en HEX convertie en décimal : 0	CRC16 de la trame de réponse.

Tableau 3.9. Trame de réponse de la lecture du registre de FAA sur Commix 1.4.

3.5.2 Tests sur le simulateur Modbus Poll :

En utilisant le Modbus Poll, nous allons présenter les résultats de la lecture des trois registres choisis précédemment :

a Lecture de la valeur du registre du point de consigne :

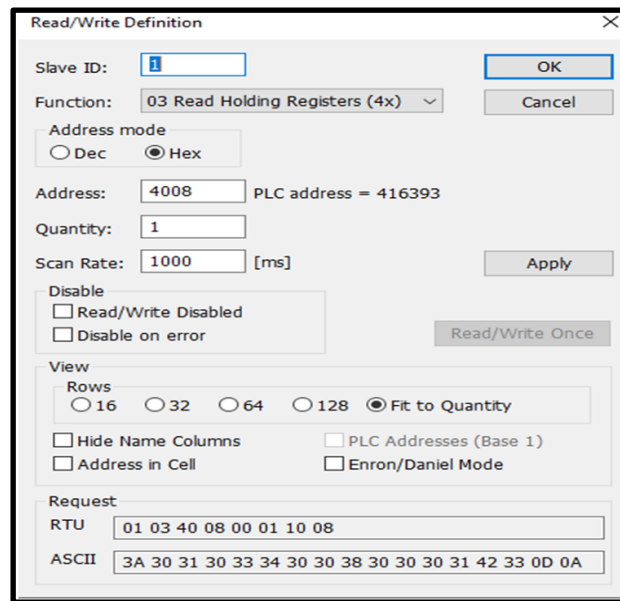


Figure 3.10. Réglages des paramètres pour lire le registre du point de consigne sur Modbus Poll.

- Trame de la requête :

01 **03** **40 08** **00 01** **10 08**

La prochaine étape consiste à régler les paramètres du port COM pour établir la connexion (**figure 3.11**) :

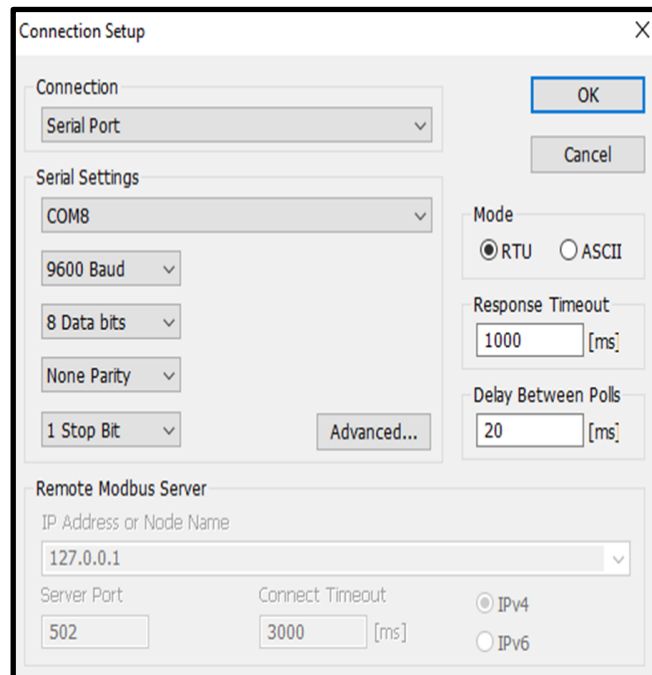


Figure 3.11. Réglage des paramètres du Port COM.

NB : le réglage des paramètres du port COM sera le même pour les deux prochaines lectures.

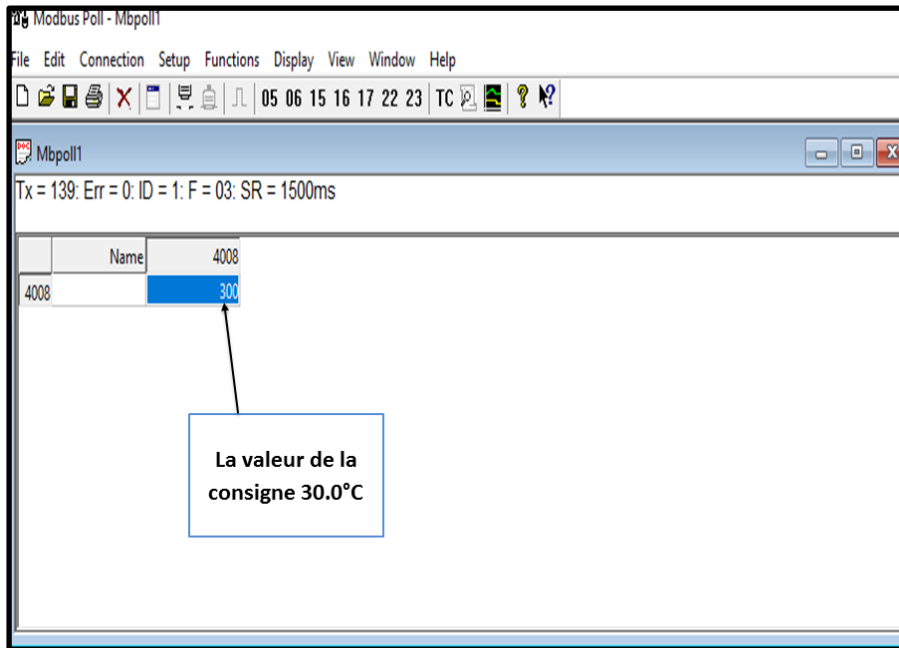


Figure 3.12. Résultat de lecture du registre du point de consigne sur Modbus Poll.

b Lecture de la valeur du registre DEA :

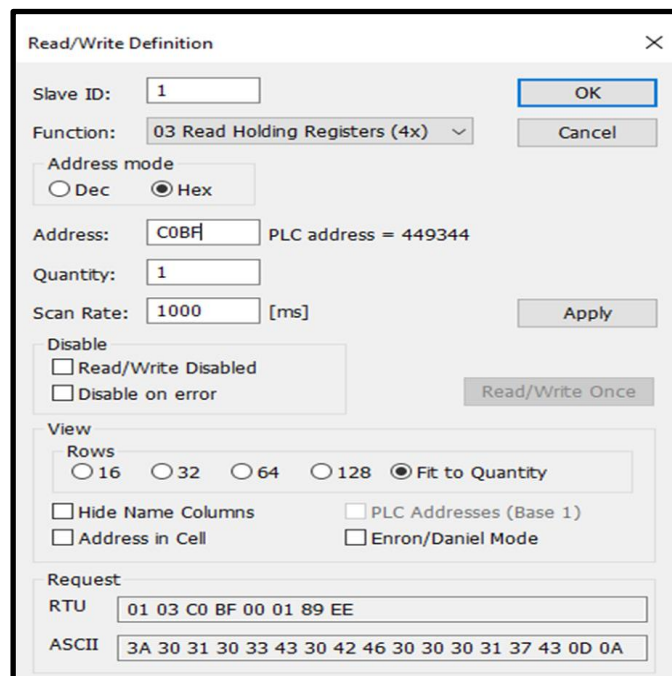


Figure 3.13. Réglages des paramètres pour la lecture du registre de DEA sur Modbus Poll.

- Trame de la requête :

01 03 C0 BF 00 01 89 EE

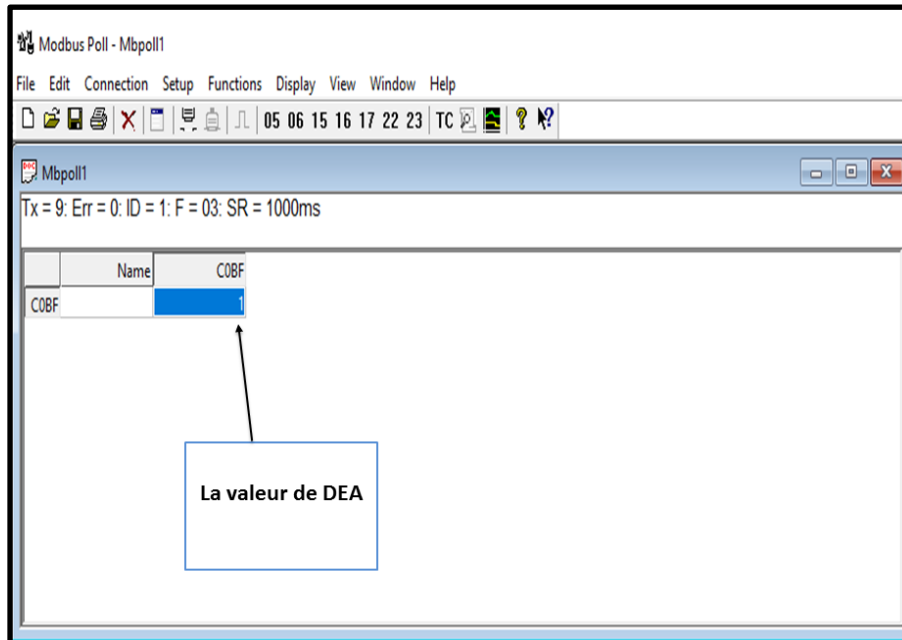


Figure 3.14. Résultat de lecture du DEA sur Modbus Poll.

c Lecture du registre FAA :

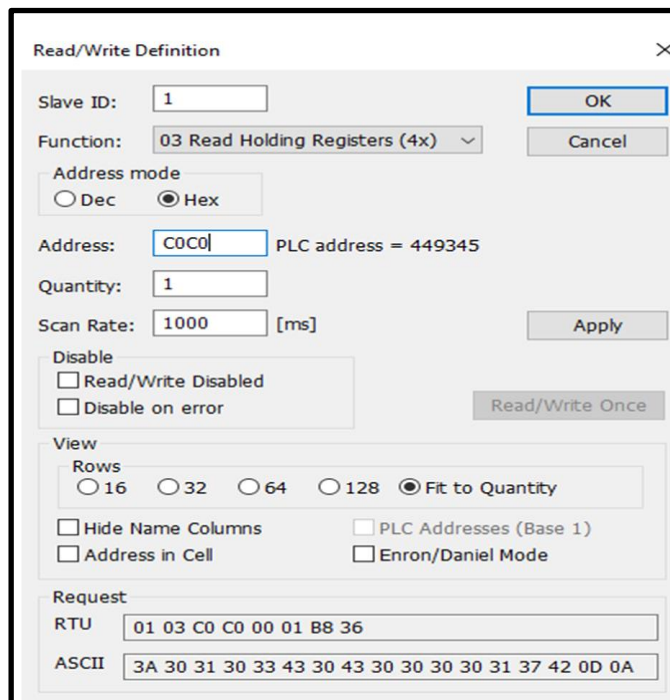


Figure 3.15. Réglages des paramètres pour la lecture du registre de FAA sur Modbus Poll.

- Trame de la requête :

01 03 C0 BF 00 01 89 EE

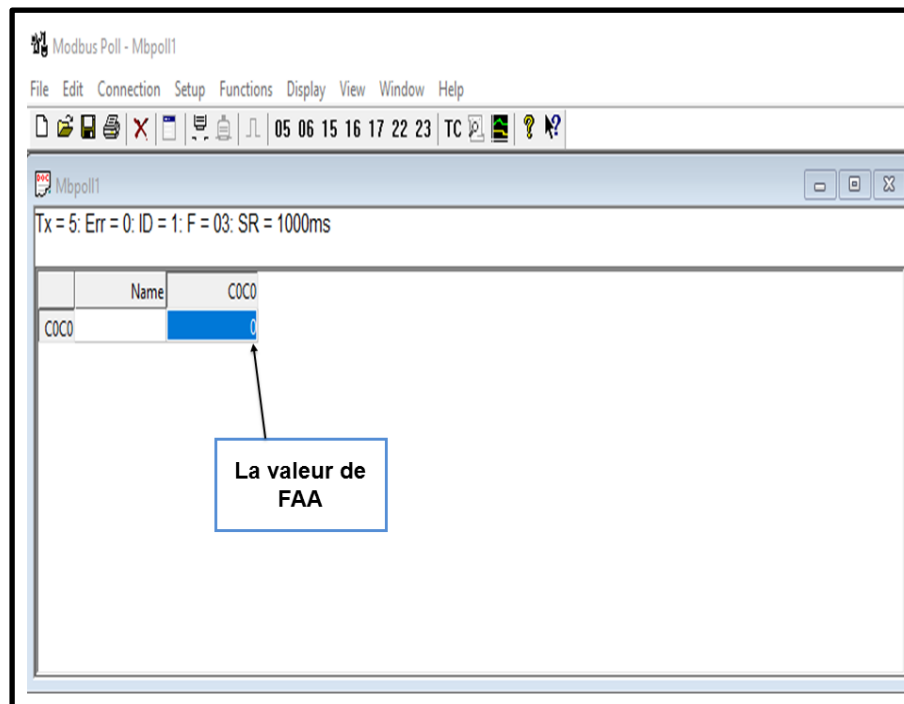


Figure 3.16. Résultat de lecture du registre du FAA sur Modbus Poll.

3.5.3 La différence entre les deux simulateurs de test :

Dans le simulateur de test Commix1.4, on peut définir tous les paramètres dans une fenêtre, ce qui facilite la manipulation pour l'utilisateur par rapport au simulateur Modbus Poll qui fonctionne avec le concept de multi-documents. D'autre part, en cas de problème de communication, le Commix 1.4 indiquera un code d'erreur qui correspond à une description dans le tableau des codes d'erreurs Modbus. Tandis que le simulateur Modbus Poll montre directement la description d'erreur.

3.6 Implémentation et résultats sur le logiciel LabVIEW :

Dans cette section nous allons entamer l'étape de l'implémentation sur LabVIEW objet de la partie pratique de notre projet. Pour cela nous allons réaliser notre instrument virtuel (VI) dans le bloc digramme et nous visualisons le résultat de la lecture de la valeur du registre adressé dans la face-avant.

Pour ce faire, nous avons implémenté deux programmes dans le logiciel, l'un utilise la bibliothèque Modbus [39] et le second est basé sur l'instrument virtuel « requête série du maître Modbus » (MB master serial Query en anglais).

3.6.1 Programmation avec la bibliothèque Modbus :

La bibliothèque MODBUS est un ensemble d'instruments virtuels (VI) téléchargeables gratuitement à partir du NI Package manager, qui peuvent fournir une communication Modbus via un port série standard ou n'importe quel port Ethernet. La bibliothèque LabVIEW implémente le protocole Modbus via son software et fournit des fonctions maître et esclave [39].

- **Algorithme**

La problématique est de faire une lecture des registres mémoire du régulateur.

L'algorithme qui permet à réaliser ceci, est comme suit :

- définir les paramètres nécessaires au niveau de l'instance du maître Modbus pour effectuer des opérations de lecture et d'écriture sur des esclaves Modbus on citera : Le mode de transmission, l'adresse d'esclave (le régulateur EW4822), Le numéro du port COM virtuel, le débit en baud, la parité et le flux de données.
- Les données sont transférées vers un instrument virtuel (LabVIEW) qui va lire les valeurs des registres de maintiens qui coresspondent au code fonction 03.
- Introduction de l'adresse du registre via ce VI et également le nombre des registres que nous souhaitons lire .
- Exécution de la tâche souhaitée :
 - ✓ Test En cas d'erreur , l'exécution du programme s'arrêtera et un message d'erreur s'affichera sur la fenêtre du diagramme .
 - ✓ Sinon l'exécution se poursuivra vers le prochain VI qui permet de fermer le programme grâce à l'instrument virtuel Modbus master shutdown.

La figure (3.17) présente les VI's composant notre block diagramme :

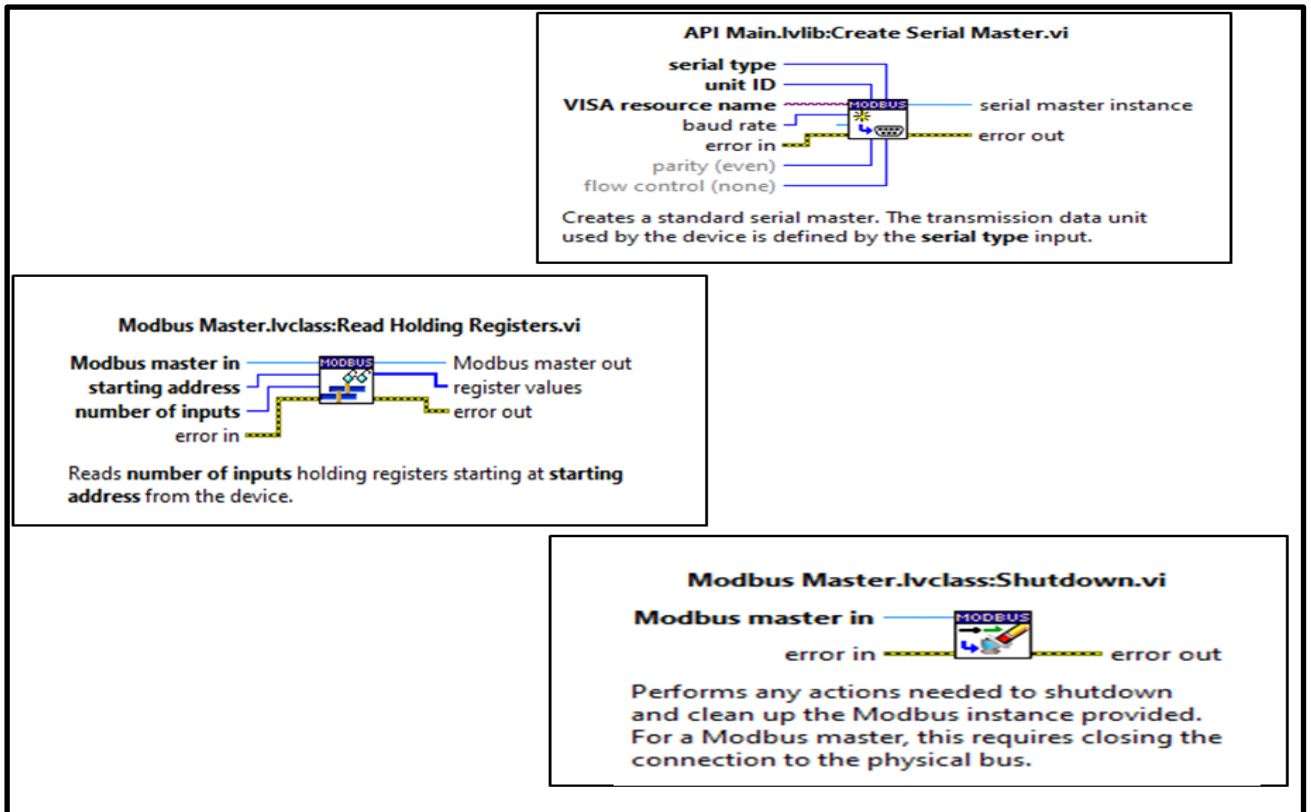


Figure 3.17. Instruments virtuels nécessaire pour l'implémentation sur LabVIEW utilisant Modbus Library.

En connectant l'ensemble des VI's indiqués sur la **figure (3.17)**, nous obtiendrons le VI final (**figure 3.18**) qui nous permet de lire les valeur des trois registres choisi précédemment à partir du **tableau (3.3)**,

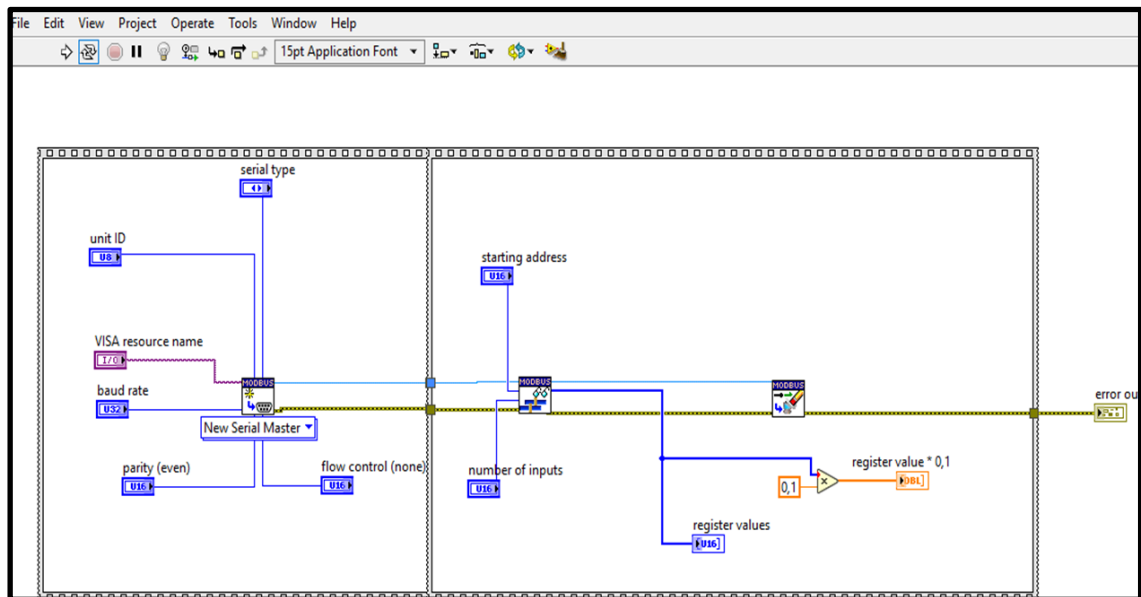


Figure 3.18. bloc diagramme de la lecture des registre du régulateur EW4822 avec Modbus library.

a Lecture du registre de la valeur de consigne :

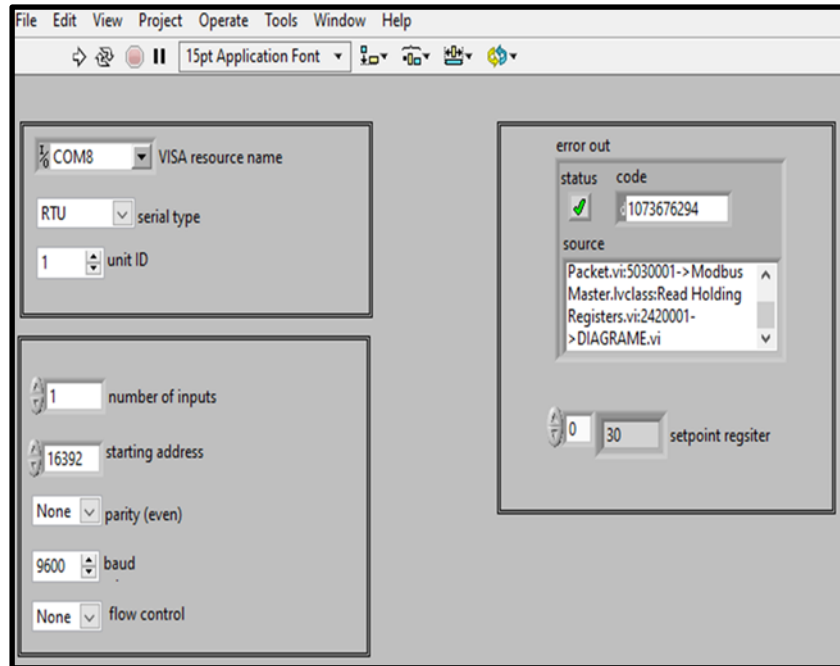


Figure 3.19 résultats de la lecture du registre du point de consigne sur la face-avant du logiciel LabVIEW.

b Lecture du registre DEA :

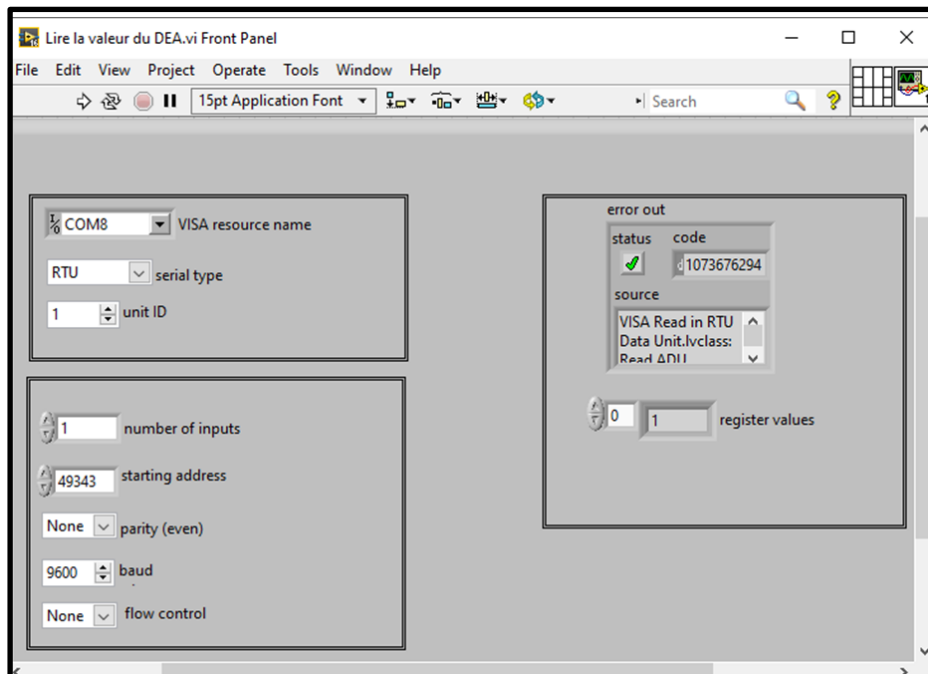


Figure 3.20 résultats de la lecture du registre DEA sur la face-avant du logiciel LabVIEW.

c Lecture du registre FAA :

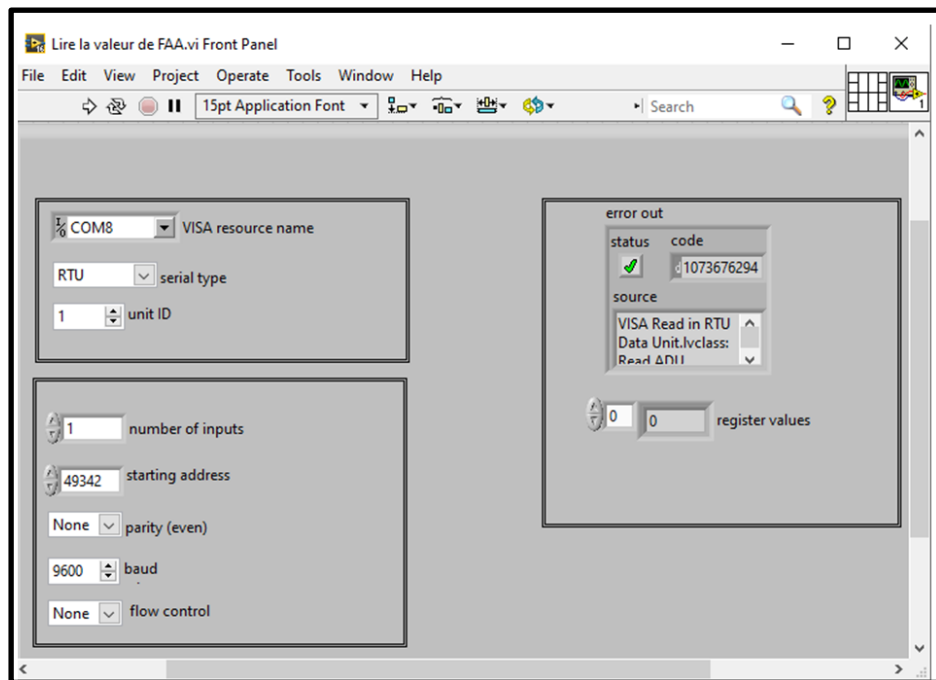


Figure 3.21 résultats de la lecture du registre FAA sur la face-avant du logiciel LabVIEW.

3.6.2 Programmation avec MB master serial Query :

Dans ce cas, le maître utilise ce VI pour lire les registres des esclaves. Cela correspond à un code de fonction publique [40].

- Algorithme :

Idem pour une lecture faite précédemment :

Commençant par configuré Le premier VI (MB Serial Init) utilisé pour choisir les paramètres du port COM tels que : numéro du port COM, le débit en bauds, la parité. Les données venant de MB Serial Init sera destiné vers MB serial master Query, qui permet de définir les paramètres suivants :

- Le code fonction (03 pour lire les registres de maintien).
- L'adresse du registre que nous souhaitons lire.
- La quantité des registres à lire.
- Le mode de transmission.
- L'adresse de l'esclave.
- Le délai d'expiration.

Exécution :

- ✓ En cas d'erreur le programme s'arrêtera et un message d'erreur sera affiché,

- ✓ Sinon le programme va se fermer en passant par l'instrumentent virtuel visa close.

La **figure (3.21)** montrera les VI's indiquées au programme :

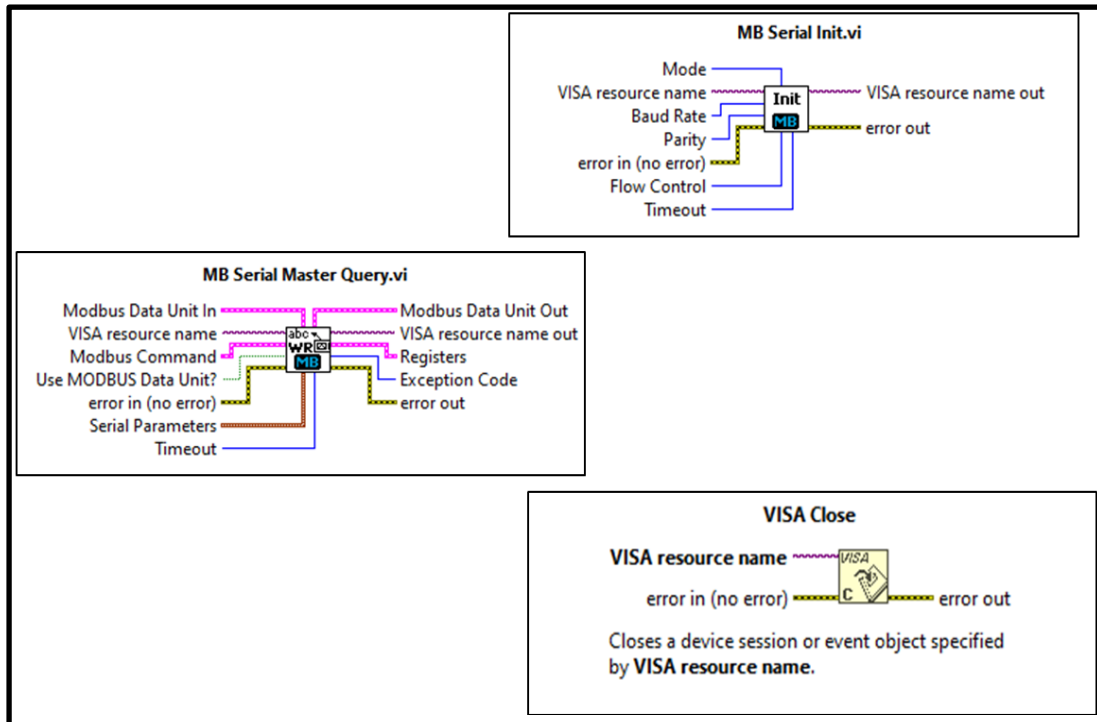


Figure 3.22. Les VI's utilisés de MB serial Query pour l'implémentation sur LabVIEW.

L'implémentation de ces trois VI's indiqués ci-dessus en séquences, donnera le résultat montré sur la **figure (3.22)**, chaque lecture respective des registres adressés sera montrée sur les **figures (3.23 ; 3.24 ; 3.25)**.

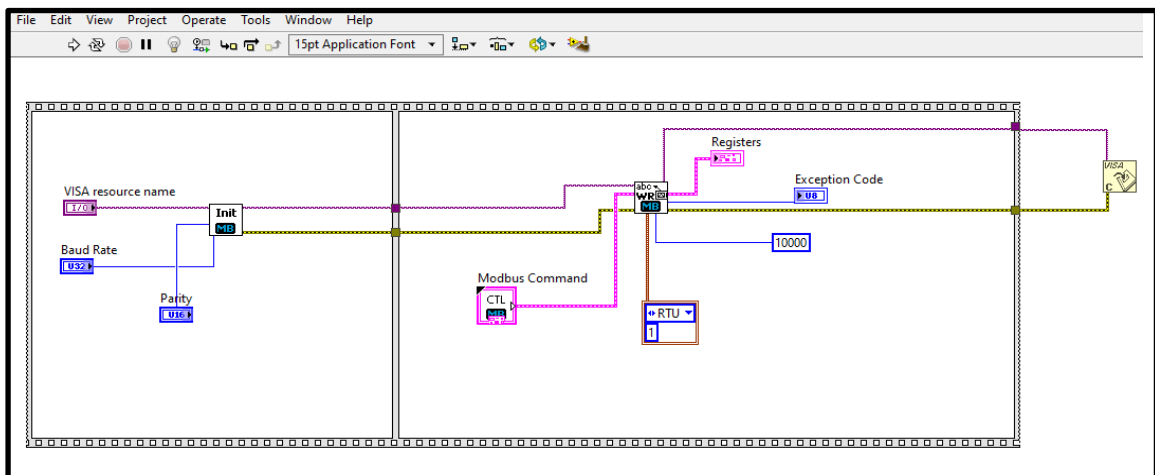


Figure 3.23. Bloc diagramme de la lecture des registre du régulateur EW4822 avec MB Master Serial Query.

a Lecture du registre de la valeur de consigne :

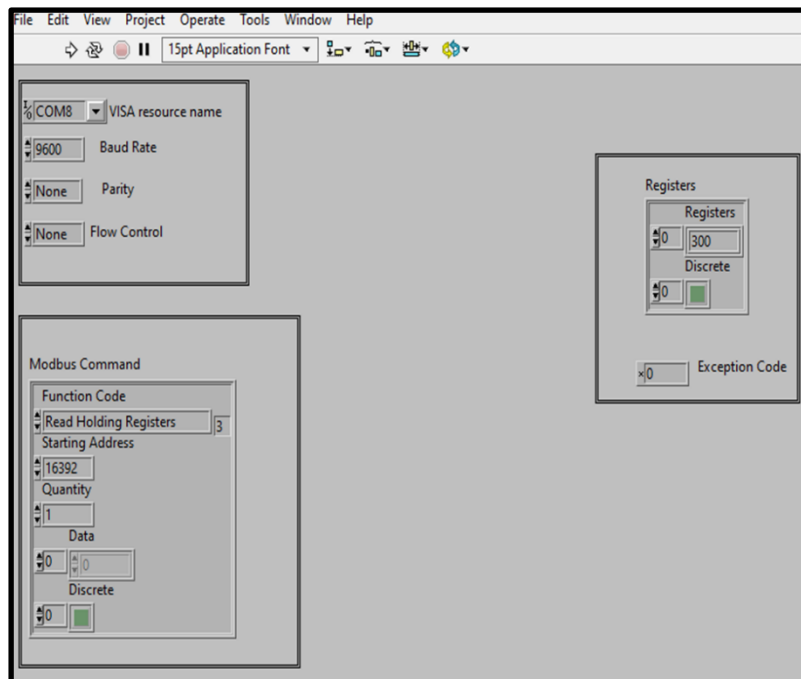


Figure 3.24. Résultat de la lecture du registre du point de consigne sur la Face-avant.

b Lecture du registre DEA :

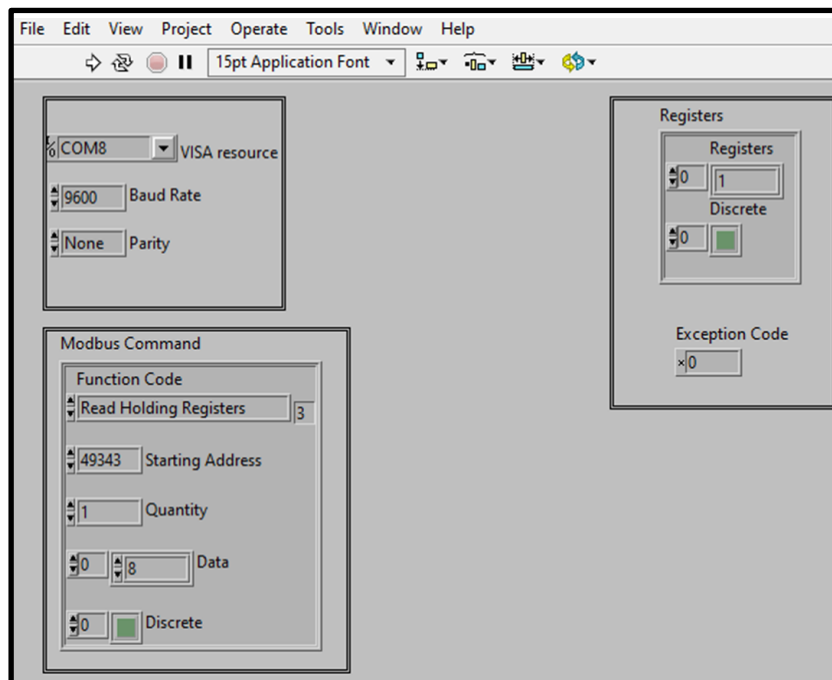


Figure 3.25. Résultat de la lecture du registre DEA sur la Face-avant.

c Lecture du registre FAA :

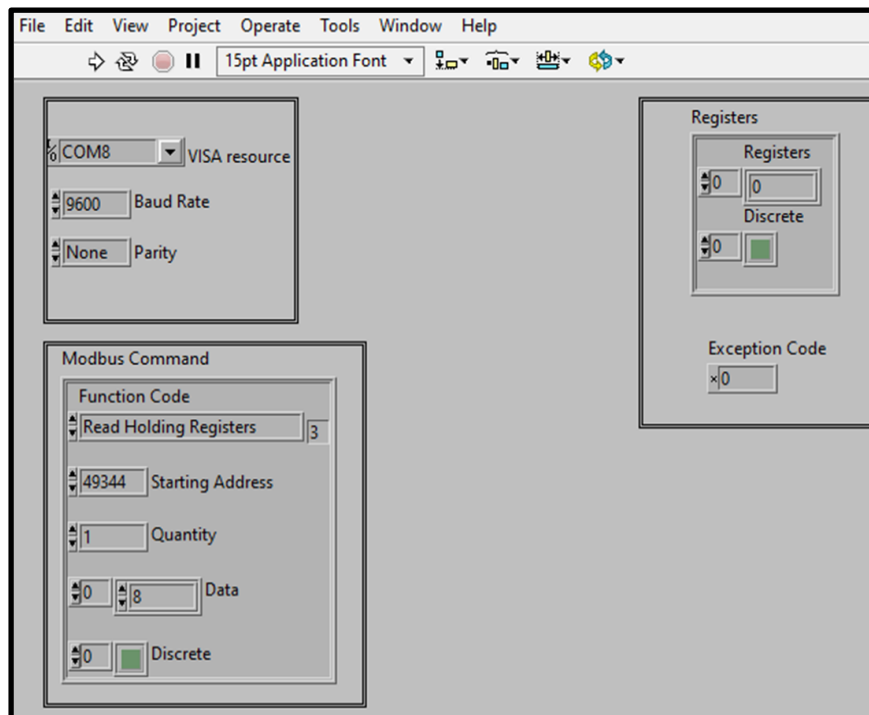


Figure 3.26. Résultat de la lecture du registre FAA sur la Face-avant.

3.7 Conclusion :

Ce dernier chapitre couvre toutes les étapes que nous avons franchies au cours du processus d'acquisition. Nous avons testé et confirmé la connectivité Modbus avec deux simulateurs de test différents pour s'assurer le bon déroulement de la configuration matérielle avant de passer à l'implémentation sur le logiciel LabVIEW où nous avons présenté tous les résultats.

Conclusion générale

En définitive nous pouvons dire que nous avons atteint l'objectif souhaité, qui consiste à contribuer dans un projet du CDTA pour commander et assurer la régulation de la température des huiles agroalimentaires utilisées afin de décider sur leur dégradation.

Notre travail présente une étude de la partie matérielle et la partie logicielle qui nous permettra de réaliser un démonstrateur expérimental et valoriser les résultats. L'étude de la partie matérielle a concerné le dispositif expérimental développé au CDTA et les modules à rajouter pour contrôler et réguler la température selon une consigne désirée.

Le dispositif instrumental fonctionne par l'introduction d'une consigne et faire prélever la température dans le processus par une sonde (capteur) et faire une comparaison pour donner une commande à un actionneur (dans notre cas une résistance chauffante) pour faire atteindre la température de la consigne. La commande est réalisée par un switch interrupteur output1 du régulateur. Ce dernier réalise son automatisme par un circuit électronique tel qu'un microcontrôleur qui régule par un software implémenté dans une mémoire.

Le développement théorique de notre travail s'est concentré sur l'automatisme industriel, et en particulier la régulation de température. La partie logicielle s'est basée sur le logiciel graphique LabVIEW afin de maîtriser sa mise en œuvre pour la commande et le contrôle de la température.

La réalisation pratique est achevée par une acquisition et un affichage passif des paramètres du régulateur de température Eliwell4822 via un adaptateur convertisseur USB/RS-485 et un PC suivant le protocole de communication Modbus RTU. Pour ce faire nous avons passé par :

- Testé la connectivité du Modbus en utilisant deux simulateurs de test Commix 1.4 et Modbus Poll,
- Développé un programme sous LabVIEW pour faire une commande et une lecture à partir des registres mémoire du régulateur EW4822 et d'afficher certains paramètres tel que :la consigne, la famille du dispositif, L'indice du dispositif, etc...

En final, les résultats obtenus répondent au problématique du sujet proposé. Néanmoins, le travail complet demande de contrôler en dynamique ce régulateur de température par imposé la consigne et le processus de fonctionnement (faire un contrôle de température par un PID). Ceci reste comme un travail en perspective. En premier lieu, c'est d'introduire une commande d'écriture sur le registre mémoire du régulateur EW4822 sous LabVIEW juste comme un test pour une éventuelle réalisation de commande contrôle du dispositif dans son fonctionnement réel. Ce travail de perspective fera l'objet éventuellement d'un sujet de master 2 pour l'année universitaire 2020-2021.

Bibliographie

- [1] Technique de régulation, : 'https://www.lucas-nuelle.fr/2431/apg/7472/Technique-de-régulation-appliquée.htm?fbclid=IwAR3HLP2lqxiltIJSZOv3s8vQkOW0ixjXnV_dj3KWzgEnaGKYJNQpBQwvgrE', aout 2020.
- [2] S.Gourari, PDF, : 'La régulation industrielle des échangeurs de chaleur calcul et dimensionnement', p1, aout2020.
<http://www.univ-tebessa.dz/fichiers/skikda/353-1179-1-PB.pdf>
- [3] Boumaza abd Errahmane, mémoire de master : ' Etude et réalisation d'un système de régulation de température', in université Badji Mokhtar Annaba, (page du résumé), 2019.
- [4] Tahar Askri, cours en PDF : 'Notes Cours Automatismes Industriels ', Algérie, p3, p2, p3.
- [5] Mezdour Hala, AYAB Mouna, thèse de master : ' Etude et réalisation d'un système de supervision sous YOKOGAWA CS3000 Application à l'unité de production d'air de l'entreprise nationale SONATRACH', Université 8 Mai 1945 – Guelma, p5, p6, 2019.
- [6] Les systèmes automatisés, PDF,
: 'http://www.kingkongevenement.com/cbruneau/perso/frames/pagetechno/atelecharger/automatisme/4eme-systemes_automatisees.pdf', aout 2020.
- [7] Mebarki Rachid, mémoire de master, : 'Automatisation et supervision d'une section d'huile finie, Université Abderrahmane Mira – Bejaia, p26, 2013-2014.
- [8] Mohamed Lamine Dilmi, mémoire de master : 'Contribution à la modélisation des systèmes automatisés par un outil graphique', université Ferhat Abbas -Setif, p6, p9, 2014.
- [9] Ramtani Jugurtha, Khereddine Lydia, mémoire de master : 'Etude et simulation du chargement des camions citernes au complexe Cevital et programmation de

l'échange de communication entre le S7 400 et Precia Molen', Université Abderrahmane Mira – Bejaia, p20,2017-2018.

- [10] Techno-logique, :'<https://www.techno-logique.com/AUT-systemes-automatiques.shtml>', aout2020.
- [11] Materiel.Industriel :'<http://www.materiel-industriel.com/quelles-sont-les-consequences-de-lautomatisation-de-lindustrie%E2%80%89/?fbclid=IwAR36xHpTbzxDUsbDwhpnl0r9CGrDq7Us2-7RlzsWTCX-Yv24nLb9bzcxa40>', aout2020.
- [12] Mr. Lemdani Rafik Mr. Malouadjmi Nabil, mémoire de master : 'Etude, conception et réalisation d'une plateforme pour l'automatisation et le contrôle à distance des serres agricoles', université M'hamed Bougara-Boumerdes, p42, juin2017.
- [13] Wikipedia, :'https://fr.m.wikipedia.org/wiki/Choix_des_boucles_de_régulation?fbclid=IwAR1FeISTxs_g5PY0Jwc0MkPBmiRBzU2p-QmWB3Nn_kPbJQxyYs0JtC4Kqss#:~:text=Une%20boucle%20de%20régulation%20est,et%20de%20s'y%20maintenir' aout2020.
- [14] Slimani Sofiane, Hachour Ahcen, mémoire de master : ' contribution à l'implémentation d'un régulateur flou simplifié sous un API S7-300', Université Mouloud Mammeri De Tizi-Ouzou, p5, p6, 2018.
- [15] Généralités sur la régulation, : 'http://mechain.lyc.ac-amiens.fr/spip_pead/IMG/pdf/1-generalites-regulation.pdf', P1.
- [16] Hamlaoui Ahmed Abderraouf, mémoire de master : 'Implémentation et test en temps réel d'un bloc régulateur RST sur Le procédé industriel de régulation CE177', Université Oum El Bouaghi, p3,2017.
- [17] Moussaoui Idhir, Bakli Mohand Amine, mémoire de master' Étude du Régulateur Industriel AC STATION (AC20) Au sein de la raffinerie d'huile de CO.G. B La Belle – Bejaia', Algérie', P9,2014-2015.
- [18] Douadi Fadila, mémoire de master : 'méthodologie d'optimisation par les techniques intelligentes d'un contrôleur OID pour un système CSTR, université Ferhat Abbas-Sétif, p4,2014.
- [19] Rahmani Oussama, mémoire de master, : 'La méthode de Zeigler_Nichols pour la détermination des paramètres d'un contrôleur PID pour un système 2^{ème} ordre', université de Bechar, p27, p28, p29,2013,2014.
- [20] Ben kherif Oussama, mémoire de master : 'Etude comparative en simulation entre un régulateur PID classique et un régulateur PID intelligent', UNIVERSITE MOHAMED BOUDIAF - M'SILA, p33, p35, p36 ; p43, 2016.

- [21] Boutana Walid, Ykhelfoune Nedjmeddine, mémoire de master : 'Etude comparative en simulation entre un régulateur PID et un régulateur flou', Université Mohammed Seddik BENYAHIA - Jijel, juillet2019, p15, p16.
- [22] Eliwell France,2020,' <https://www.eliwell.eu/fr/regulateur-deux-etages-pour-la-temperature-ew4822-universel-i-1643.html>', juin2020.
- [23] EW4800 contrôleurs universels 11/2006 F,PDF, : 'régulateurs de température et régulateurs de processus ', p1, p2, p3, p4, p9.
- [24] Qu'est-ce qu'un protocole de communication,septembre2019 : ' <https://www.automation-sense.com/blog/automatisme/qu-est-ce-qu-un-protocole-de-communication-industriel.html>, aout2020.
- [25] Using MODBUS for Process Control and Automation, PDF, : 'The Interface Solution Experts • www.miinet.com', p1, p2.
- [26] Lafioune Sid Ali, mémoire de master : 'Etude et réalisation d'un banc de démonstration pour la Supervision industrielle en utilisant le protocole Modbus RTU', université Mohamed Seddik Ben Yahia, p22, p23, p24 ,2019.
- [27] Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J, p6, p7. (Revu)
- [28] NBS System, transmission série, : ' <https://www.nbs-system.com/blog/transmission-en-serie-vs-transmission-en-parallele/?fbclid=IwAR3KxezWf-vHgLFVj4-8QHt1P-Mbc8HtD6qhgccqTTWAJS7ppE86d7mDpdw>, aout2020.
- [29] Chicoree.fr.'Mai2011 http://www.chicoree.fr/w/Full-duplex,_half-duplex_et_simplex', aout2020.
- [30] ptimic,2013, peripheriques-externes : ' <https://www.pmtic.net/contenu-en-ligne/environnement-numerique/la-machine/peripheriques-externes/recapitulons/ports-d-un-ordinateur?fbclid=IwAR2ZCjrg7V8tWkKG---llWIPAMjmM1Xjf2DqGV2ukH4lDDW7rEZrW07l4gU>', aout2012.
- [31] Elima,2020, virtual-serial-port, : ' https://www.virtual-serial-port.org/fr/article/what-is-serial-port/rs232-vs-rs485.html?fbclid=IwAR2A00lcwPj9E9afqZ9WNBalybJuUrKcsEGJVRh6lDWZaJrqOi1gc-M_V0s#difference, aout2020.
- [32] Your Duino, USB TO TTL/RS-485 Interface, : http://www.yourduino.com/sunshop/index.php?l=product_detail&p=325
- [33] Cablematic, adaptateur-serie, : ' <https://cablematic.com/fr/produits/adaptateur-serie-usb-b-hembra-a-rs485-ttl-3-broches-AV00600/>', aout2020.

- [34] Zaidi Mouloud, Maouche Med Amine, mémoire de master, : ‘Développement d’une interface pour capteur CND à courants de Foucault’, université Mouloud Mammeri, Tizi-Ouzou, p39,2013.
- [35] Ingénieur FREELANCE en Système Industriel et Scientifique LabVIEW, TestStand, LabWindows/CVI, : ‘http://www.ajolly.fr/fr/conseil-et-deloppement-mainmenu-27/19-LabVIEW.html?fbclid=IwAR00fuwTWEFF5PHUih_6cV8egTZvzNMYSdf6i6NNeVvpsU97j5wwi1A6rZg’.
- [36] NI,national instruments, : ‘http://www.ni.com/academic/students/learnlabview/f/datatypes.htm?fbclid=IwAR0GXWkFcaeNPNkIk30vE99WTzYe5E7ug6mRoY_Lqvj0SkPoIE5uESzb9I’.
- [37] POLIE Maîtrise de l'Energie, Logiciel Commix 1.4, : ‘<https://www.polier.fr/pages/question-reponse/modbus/logiciel.html?fbclid=IwAR1D8LS86M7odTNRa4R6kFpGpHBUaRwEBWE4XhITIOXFVu5W-eYO0ob0rvY>’,2020.
- [38] Modbus tools, Modbus Poll, : ‘https://www.modbustools.com/modbus_poll.html?fbclid=IwAR1D8LS86M7odTNRa4R6kFpGpHBUaRwEBWE4XhITIOXFVu5W-eYO0ob0rvY , aout 2020.
- [39] NI, national instrument, : ‘<http://www.ni.com/example/29756/en/> , aout2020.
- [40] POLIE Maîtrise de l'Energie, Logiciel Commix 1.4, : ‘<https://www.polier.fr/pages/question-reponse/modbus/logiciel.html>’
- [41] Modbus LabVIEW VI Documentation, : ‘[https://documentation.help/Modbus-LabVIEW/MBSerialMasterQueryReadHoldingRegisters\(poly\)vi.html](https://documentation.help/Modbus-LabVIEW/MBSerialMasterQueryReadHoldingRegisters(poly)vi.html) , aout 202.
- [42] EW480, ‘Modbus Serial Communication Protocol’ (Modbus Manual), PDF,p5.