

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahim - Bordj Bou Arreridj

Faculté des Sciences et de la technologie

Département ELECTROMECHANIQUE

Mémoire

Présenté pour obtenir

LE DIPLOME DE MASTER

FILIERE : automatique

Spécialité : AUTOMATIQUE ET INFORMATIQUE INDUSTRIEL

Par

- **BOUTIH KARIMA**
- **HADJADJ HADJER**

Intitulé

***SUPERVISION DES PANNEAUX PHOTOVOLTAIQUE A PARTIR
D'UNE CARTE ARDUINO ET HMI SCHNEIDER MAGELIS***

Soutenu le : 25/06/2022

DEVANT L'ENCADREUR : RIADH KHENFER

Année Universitaire 2021/2022



Dédicaces

Au nom du tout puissant Je dédie ce travail
A mes très chers et précieux parents qui m'ont
toujours soutenue, et à l'intérêt qu'ils
Ont toujours porté pour mes études, je ne les remercierai
jamais assez, pour tout ce qu'ils font.

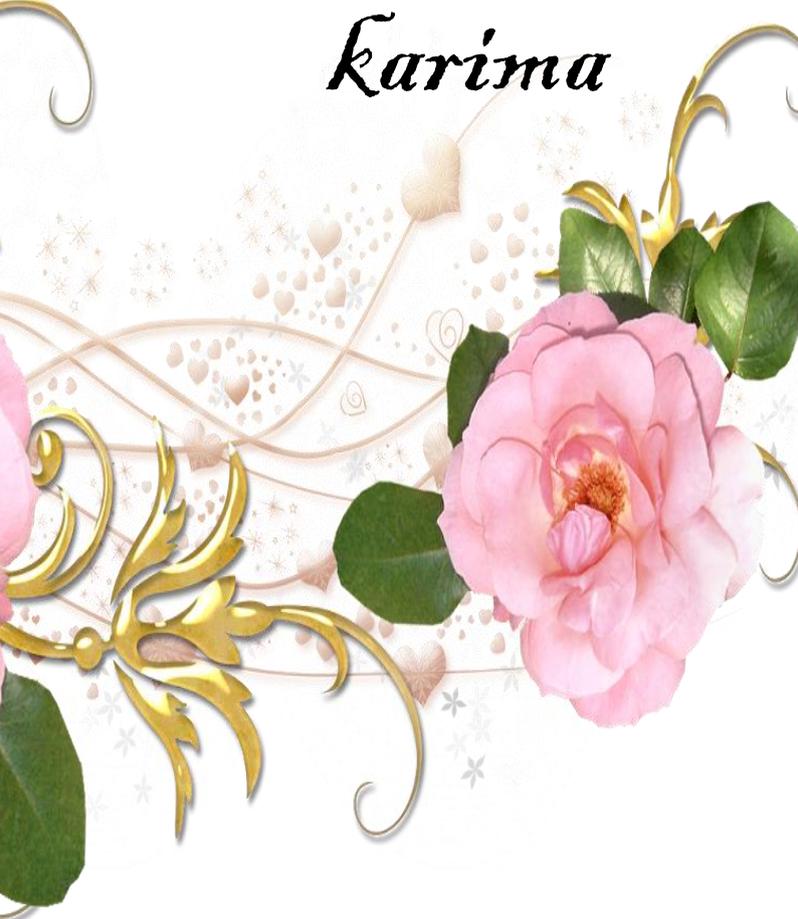
Que dieu me les protèges.

A mes cher frères : Abd Al Malek et Mourad
et Kamel et Amer.

A mes cher sœurs : Hassina et Sabrina.

A tous mes amis de la promotion 2021/2022
Automatique et Informatique Industrielle.

Karima





Dédicaces

Tout d'abord, je tiens à remercier DIEU
De m'avoir donné la force et le courage de mener
à bien ce modeste travail.

Je tiens à dédier cet humble travail à :

A ma tendre mère Khadija et mon très cher
père Mohamed

A mes sœur: Nabila, Nassima, Fatima, Samia,
Et Iman

A mes cher frères : Hicham, Youcef

Tous ceux qui m'aiment

A tous mes amis de la promotion 2021/2022
Automatique et Informatique Industrielle.

HADJER





Remerciements

En premier lieu je tiens à remercier le grand dieu, pour la force, le courage, et la volonté qui m'a donné durant toutes mes années d'étude.

*Je tiens à remercier Mon encadreur, Monsieur **Riadh khenfer Salim** d'avoir accepté de m'encadrer*

Je lui suis sincèrement reconnaissant pour son encadrement, ses orientations et sa disponibilité, Je le remercie également de m'avoir toujours poussé vers l'avant et aussi pour toute la confiance qu'il a portée en moi.

Mes remerciements s'adressent aussi aux membres de jury qui me font l'honneur qu'ils me font de juger mon travail.

Je dédie ce travail, avant tout, A ma Grand-Mère que j'estime beaucoup. À mes très chers parents, et mes très chères sœurs.

A toute ma grande famille, oncles, cousins et cousines Et A mes chers amis.

Souhaitons enfin que ce projet soit au niveau de vos attentes.



TABLE DE MATIERE

CHAPITRE 01: Interface homme machine

Introduction générale	1
1.1 Introduction:	4
1.2 Définition :	4
1.3 Utilisation d’HMI :	5
1.4 Conception d'interface homme-machine :	5
1.5 Description de la partie matérielle :	5
1.5.1 Mode édition :	6
1.5.2 Description Panels MAGELIS :	6
1.5.3 Fonctionnement du terminal :	6
1.5.4 Les interfaces de l’HMI	8
1.7 Conclusion :	14

CHAPITRE02: Protocol De communication Modbus

2 . 1 Introduction	16
2.2 Historique de modbus	16
2.3 Principe de fonctionnement du modbus	17
2.3.1 La requête	17
2.3.2 La réponse	17
2.4 Architecture du réseau Modbus	18
2.4.1 La liaison RS 232	18
2.4.2 La liaison RS 485	19
2.5 Les types de données Modbus	20
2.5.1 Les bobines :	20
2.5.2 Les registres :	20
2. 6 Zones mémoires du Modbus	21
2.6.1 Zone des bobines (coils) :	21
2.6.2 Zone de mémoire d’entrée (entrées discrètes) :	21
2.6.3 Zone des registres d’entrées :	21
2.6.4 Zone des registres à usage général (holding registers) :	21
2.7 Quelques Caractéristique Du Modbus	22

2.8	Modes de transmissions et format de trames	22
2.8.1	Modbus ASCII :	22
2.8.2	Modbus RTU :	22
2.9	Trame d'échange requête/réponse pour Modbus RTU	22
2.10	Conclusion :	25

CHAPITRE 03: Réalisation Matérielle et Logicielle

3.1	Introduction	27
3.2	Le circuit complet	27
3.3	Première test: La communication en utilisant un convertisseur USB RS485:	28
3.3.1	Description de la partie matérielle :	29
3.3.1.1	Présentation de La carte Arduino UNO :	29
3.3.1.2	Caractéristiques de la carte Arduino UNO :	29
3.3.1.3	Avantages de la carte Arduino UNO :	30
3.3.1.4	Module convertisseur MAX485 RS 485 TTL vers RS485 5V :	30
3.3.1.5	USB à RS485 Convertisseur Adaptateur :	31
3.3.2	Description de la partie logicielle :	31
3.3.2.1	Arduino IDE :	31
3.3.2.2	Structure d'un programme Arduino :	32
3.3.2.3	L'interface du logiciel arduino :	32
3.3.2.4	La partie programmation du premier test:	33
3.4	Deuxième teste	39
3.4.1	Description de la partie logicielle :	40
3.4.1.1	Logiciel vijeo Designer :	40
3.5	Troisièmes tests :	43
3.5.1	Description de la carte ARCELI TTL à RS485 Adaptateur 485 Serial Port UART LevelConvertir Module 3.3V 5V :	43
3.5.2	La partie programmation du troisième test:	43
3.6	Quatrième test :	44
	45
3.6.1	Capteur de courant ACS712.....	45
3.6.2	La partie programmation du quatrième test:	46
3.7	Cinquième test : Ce test consiste d'ajouter un capteur de tension.	46
	46

3.7.1 Capteur de tension :.....	47
3.7.2 La partie programmation du Cinquième test:	48
3.8 Sixième test :	48
3.8.1 Panneau solaire Condor :.....	48
3.9 Conclusion :.....	51
Conclusion Générale :	53
Références bibliographiques.....	54

Liste Des Figures

CHAPITRE 1

Figure 1:Interface Homme Machine	4
Figure 2:Les équipements connectables sur les terminaux Advanced Panels TactilesMAGELIS	6
Figure 3:Description Advenced Panels MAGELIS XBTGT2330 en face avant.....	6
Figure 4:Description Advenced Panels Magelis XBTGT2330 en face arrière.....	7

CHAPITRE 2

Figure 5:Principe de fonctionnement du Modbus.....	17
Figure 6:La liaison série type RS 232.....	19
Figure 7:La liaison série type RS 485.....	19
Figure 8:Trame de Modbus RTU	23
Figure 9:Trame de réponse d'un esclave Modbus.....	24

CHAPITRE 3

Figure 10:shéma global.....	27
Figure 11:la communication en utilisant un convertisseur USB RS485	28
Figure 12:La carte Arduino UNO	29
Figure 13:Module convertisseur MODBUS vers RS485 5V	31
Figure 14:USB à RS485 Convertisseur.....	31
Figure 15:Interface de logiciel Arduino	32
Figure 16:Les boutons de logiciel Arduino	33
Figure 17:Interface du logiciel Simplymodbus master	35
Figure 18:schéma de la Deuxième teste.....	39
Figure 19:Réalisation pratique du Deuxième test.....	40
Figure 20:Interface du logiciel vijeo designer	42
Figure 21:Le premier programme dans logiciel vijeo designer	43
Figure 22:Carte ARCELI TTL à RS485 Adaptateur	43
Figure 23:Réalisation pratique du troisième test	44
Figure 24:schéma de quatrième test.....	45
Figure 25:Capteur de courant ACS712.....	45
Figure 26:schéma de Cinquième test.	46
Figure 27:Capteur de tension.....	47
Figure 28:Réalisation pratique du Cinquième test.....	47
Figure 29:Panneau solaire utilisé.....	48
Figure 30:Réalisation pratique du sixième test.....	49
Figure 31:Traçage du courant et tension	49
Figure 32:caractéristique de panneaux solaire.....	50
Figure 33:puissance de panneaux solaire	50

Liste Des Tableaux

CHAPITRE 1

Tableau 1:Interface série COM1	8
Tableau 2:Interface série COM2	8
Tableau 3:Interfaces série Ethernet.....	9
Tableau 4:Interface série USB et la carte mémoire	9
Tableau 5:Interface série avec un connecteur SUB-D 9 points via un câble RS-422/RS-485	11
Tableau 6: Interface utilisée pour connecter un câble série RS-485	12
Tableau 7: l'interface série COM2	13

CHAPITRE 2

Tableau 8 :Zones mémoire des esclaves Modbus.....	21
Tableau 9:Code fonctions de la trame Modbus	24

CHAPITRE 3

Tableau 10:Caractéristiques de la carte Arduino UNO.....	29
--	----

Liste des abréviations

HMI Interface homme machine

PLC Contrôleur logique programmable

V voltage

USB *Universal Serial Bus*

API Application Programming Interface

GND le principe de la terre en électricité

PDU Protocol Data unit

RTU Remote Terminal Unit

ASCII *American Standard Code for Information Interchange*

PV panneaux photovoltaïque

ملخص:

يتكون هذا المشروع من برمجة بطاقة اردوينو أونو لضمان عرض البيانات والملفات للإشراف على الألواح الكهروضوئية على واجهة بين الإنسان والآلة 'Schneider MAGELISXBTGT2330' والذي تمت برمجته بواسطة برنامج مصمم Vijeo الكلمات الرئيسية :

Schneider MAGELIS XBTGT2330, arduino UNO, vijeo designer. IV tracer

RÉSUMÉ:

Le présent projet consiste à la programmation d'une carte arduino UNO pour assurer l'affichage des données et la supervision des panneaux photovoltaïque sur une interface homme machine "Schneider MAGELISXBTGT2330" qui est programmer par le logiciel Vijeo designer.

Mots clés : Schneider MAGELIS XBTGT2330, arduino UNO, Vijeo designer .traceur IV

ABSTARCT:

This project consists of programming an arduino uno card to ensure the display of data and the supervision of photovoltaic panels on a man-machine interface " Schneider MAGELIS XBTGT2330 " which is programmed by the vijeo designer software.

Keywords: Schneider MAGELIS XBTGT2330, arduino uno, vijeo designer. IV tracer

Introduction générale

Introduction générale

Les systèmes de contrôle industriels continuent d'évoluer et dans le monde d'aujourd'hui, les tâches que les opérateurs doivent accomplir peuvent changer fréquemment. Pour gérer cette complexité, les contrôles doivent être flexibles et ergonomiques. C'est l'avantage de l'HMI. Avec une HMI vous pouvez facilement communiquer avec les machines et obtenir des données opérationnelles sur l'ensemble de vos équipements et de vos installations.

Dans le domaine de l'automatisation, les écrans tactiles sont des HMIs très populaires afin de centraliser le contrôle d'un procédé sur un seul écran. Ainsi, il est possible d'afficher plusieurs informations et de mettre à la disposition de l'opérateur des commandes qui affecteront le procédé.

Les HMIs permettent aussi de remplacer des stations de boutons. Ils sont surtout utilisés en complément avec un API (automate programmable industriel) pour avoir un affichage des états des entrées/sorties et des alarmes du système.

Une HMI peut prendre plusieurs formes. Il peut s'agir d'un écran autonome, d'un tableau de bord attaché à un autre équipement ou d'une tablette. Quoi qu'il en soit, son but premier est de permettre aux utilisateurs de visualiser des données quant aux opérations et de contrôler les machines. Les opérateurs pourraient, par exemple, utiliser une HMI pour voir quelles bandes transporteuses sont enclenchées ou pour ajuster la température d'un réservoir d'eau industriel.

D'habitude, les HMI sont câblés avec les APIs, pour des raisons pédagogiques mais aussi pour des solutions économiques, nous avons essayé de remplacer l'API par une carte arduino.

Ce mémoire est divisé en deux parties une partie théorique et une autre pratique.

Le premier chapitre, présente l'utilisation des HMIs, une description matérielle de notre HMI utilisée dans ce projet

Introduction générale

Le deuxième chapitre, présente des Principes de fonctionnement du Modbus, Les types de données Modbus , Quelques Caractéristique Du Modbus

Le troisième chapitre sera consacré à plusieurs test pour assurer la communication entre la carte arduino et HMI dans le but d'afficher le courant et la tension des panneaux photovoltaïque.

CHAPITRE1:

Interface homme machine

CHAPITRE 01: Interface homme machine

1.1 Introduction:

Le développement de l'industrie a mis en avant une demande plus forte pour le fonctionnement des équipements industriels, des schémas de principe aux pupitres de contrôle et de commande, en passant par les écrans tactiles, avec un seul objectif, à savoir maximiser la supervision et le fonctionnement des équipements industriels. Équipement dans nos recherches, nous nous concentrerons sur les interfaces homme-machine spécifiquement pour la supervision des systèmes industriels[1].

1.2 Définition :

L'interface homme-machine (communication homme-machine) est définie comme un ensemble de dispositifs matériels et logiciels permettant aux utilisateurs d'interagir avec des systèmes interactifs, où les installations industrielles sont représentées par des graphiques répartis sur plusieurs pages. Chaque page en appelle une autre, couvrant l'ensemble de l'installation, vous pouvez donc démarrer la pompe, voir le niveau du réservoir, ouvrir la vanne, etc. Cliquez simplement avec la souris sur le PC ou touchez la surface de l'écran tactile. [1]



Figure 1:Interface Homme Machine

1.3 Utilisation d'HMI :

L'interface homme-machine améliore la communication entre les humains et les machines, augmentant ainsi l'efficacité. Les appareils HMI dans les environnements industriels peuvent surveiller et/ou contrôler les équipements de processus sur un seul écran, affichant ainsi plusieurs éléments d'information et fournissant aux opérateurs des commandes qui affectent le processus.

Les HMI peuvent également être utilisées pour remplacer les stations de boutons. En plus des automates, ils sont principalement utilisés pour afficher l'état des entrées/sorties et des alarmes système. Contrôlez des moteurs asynchrones via des variateurs de vitesse connectés à des moniteurs PLC et HMI.

L'interface invoque 3 principales fonctions d'interaction humaine :

- Tactile (commandes via boutons, écran tactile, clavier, pavé numérique)
- Vision (surveillance de l'écran, surveillance du faisceau)
- Audible (alertes sonores, blips) (by schneider Electric, 1996 2017)

1.4 Conception d'interface homme-machine :

Les placements importants de la plupart des projets d'informatisation et d'automatisation visant à concevoir des systèmes industriels homme-machine de plus en plus complexes rendent nécessaire la prise en compte de l'humain dans leur approche holistique. Sur ce sujet, des outils, des techniques, des méthodes et des modèles issus de toutes les technologies et sciences humaines émergent actuellement qui peuvent contribuer au développement des systèmes homme-machine. La conception de l'interface homme-machine repose principalement sur :

- Analyse et modélisation de systèmes techniques
 - Analyser et modéliser les tâches homme-machine et les acteurs impliqués dans les systèmes homme-machine
- Spécification de l'image
 - Environnement graphique pour générer des images
 - Évaluation du système homme-machine.[1]

1.5 Description de la partie matérielle :

Mode de fonctionnement des bornes

Le Figure suivant présente les équipements pouvant être connectés au pupitre tactile avancé MAGELIS selon les deux modes de fonctionnement :

1.5.1 Mode édition :

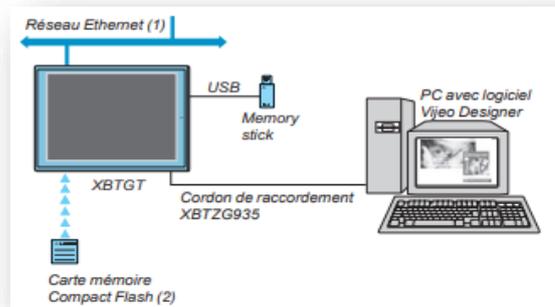


Figure 2: Les équipements connectables sur les terminaux Advanced Panels TactilesMAGELIS

1.5.2 Description Panels MAGELIS :

La face avant est représentée par la figure suivante :



Figure 3: Description Advanced Panels MAGELIS XBTGT2330 en face avant

- 1 Un écran tactile d'affichage de synoptiques (5,7" monochrome ou couleur).
- 2 Un voyant multi couleur (vert, orange et rouge) indiquant le mode de fonctionnement du terminal.[2]

1.5.3 Fonctionnement du terminal :

La face arrière est représentée par la figure suivante :

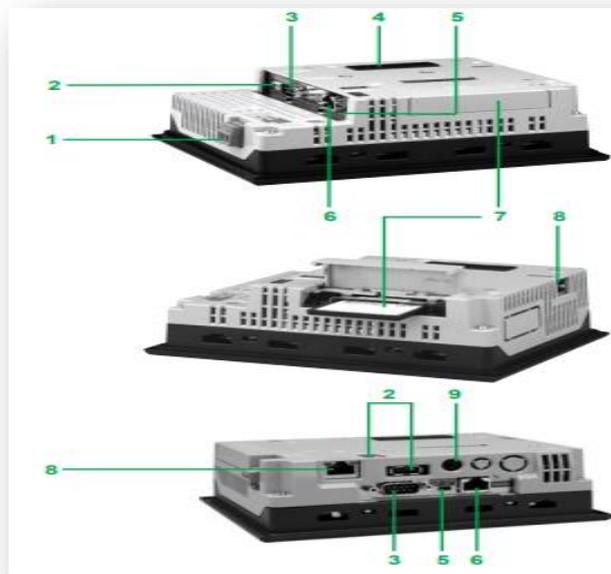


Figure 4:Description Advanced Panels Magelis XBTGT2330 en face arrière

- 1 Bornier débrochable à vis pour alimentation c 24 V.
- 2 Un connecteur USB principal de type A pour la connexion et le transfert de périphériques Applications et communication Prise terminal Modicon M340.
- 3 connecteurs SUB-D mâle 9 contacts pour liaison série RS 232C ou RS 422/485 vers API (COM1).
- 4 Interface de l'unité d'extension de la carte de communication du bus de terrain (réseau d'appareils, Profibus DP).
- 5 Un commutateur de polarisation de liaison série COM2 pour Modbus.
- 6 Connecteur RJ45 pour liaison série RS 485 (COM2).
- 7 Fente pour carte Compact Flash avec couvercle. Sur XBTGT2930 uniquement :
- 8 Un connecteur RJ45 pour liaison Ethernet TCP/IP, 10BASE-T/100BASE-TX. Sur XBTGT2430 uniquement :
- 8 Un connecteur RJ45 pour liaison Ethernet TCP/IP, 10BASE-T/100BASE-TX.

9 Connecteur mini jack pour sortie audio. [2]

Le tableau ci-dessous représente l'interface série COM1 des équipements des séries XBT GT 2000/4000/5000/6000/7000

Tableau1:Interface série COM1

Interface	La description
Interface série COM/COM1 D-Sub9	
Transmission asynchrone	R0S-232C / RS-422-485
Longueur des données	7 ou 8 bits
Bit d'arrêt	1 ou 2 bits
Aucun, impair ou pair	Vitesse de transmission des données
Parité	2 400 à 115 200 points de base 62

1.5.4 Les interfaces de l'HMI**1.5.4.1 Interface série COM2 :**

Le tableau suivant représente l'interface série COM2 des unités des séries XBT GT2000/4000/5000/6000/7000

Tableau2:Interface série COM2

Interface	La description
Interface série COM2 RJ45	
Transmission asynchrone	Données RS-485
Longueur	7 ou 8 bits
Bit d'arrêt	1 ou 2 bits
Parité	Aucun, impair ou pair
Vitesse de transmission des données	2 400 bps à 187,5 Kbps Ethernet

1.5.4.2 Interface Ethernet :

Le tableau suivant représente les interfaces série Ethernet disponibles pour les séries XBT GT

Tableau3: Interfaces série Ethernet

Interface	La description
Ethernet RJ45	IEEE 802.3, 10Base-T/100Base-TX (sauf pour XBT GT1105/2110/2120/2220 et XBT GK2120 : aucun. USB

1.5.4.3 Interface USB et carte mémoire

Le tableau suivant représente l'interface série USB et la carte mémoire disponibles pour les séries XBT GT XBT GT2000/4000/5000/6000/7000, série XBT GK Interface série COM1, XBT GH COM

Tableau4: Interface série USB et la carte mémoire

Interface	La description
USB TYPE-A	I/F hôte USB 1.1
Emplacement pour carte CF (TYPE-II) (sauf XBT GT1105/1135/1335/2110)	Flash compact

Le tableau suivant décrit l'interface série avec un connecteur SUB-D 9 broches via un câble RS-232C.

Broche	Connexion	Broche	Nom du signal	du	Direction	Sens
		1	CD		Saisir	Détection de porteuse
		2	RD(RXD)		Saisir	Recevoir des données
		3	SD (TXD)		Production	Envoyer des données
		4	ER(DTR)		Production	Borne de données Prêt
		5	SG		-	Masse du signal
		6	6 DR (DSR)		Saisir	Ensemble de données prêt
		7	RS (RTS)		Production	Demande d'envoi
		8	8 CS (CTS)		Saisir	Envoyer Possible
		9	CI(RI)/CCV		Saisir	Affichage de l'état appelé/+5V5% Sortie 0,25A
		Coquille	FG		-	Mise à la terre du cadre (commun avec SG)

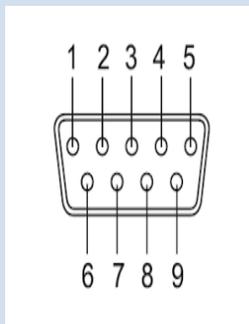
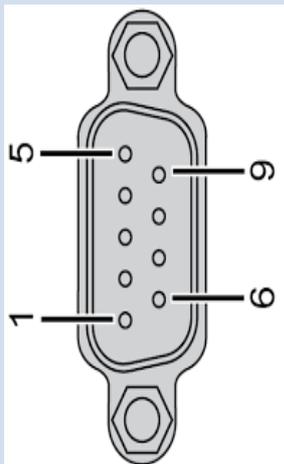


Tableau5:Interface série avec un connecteur SUB-D 9 points via un câble RS-422/RS-485

Broche	Connexion	Broche	Nom du signal	Direction	Sens
		1	RDA	Saisir	Recevoir des données A (+)
		2	RDB	Saisir	Recevoir des données B (-)
		3	DPS	Production	Envoyer des données A (+)
		4	ERA	Production	Terminal de données prêt A (+)
		5	SG	-	Masse du signal
		6	BSC	Saisir	Envoyer Possible B (-)
		7	SDB	Production	Envoyer des données B (-)
		8	CSA	Saisir	Envoi possible (A)
		9	ERB	Production	Terminal de données prêt B (-)
		Coquille	FG	-	Mise à la terre du cadre (commun avec SG)

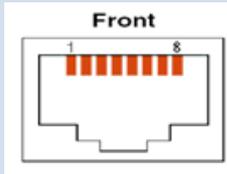


1.5.4.4 Interface série COM2 avec câble RS-485 :

Cette interface est utilisée pour connecter un câble série RS-485 à une unité de la série XBT GT2000/4000/5000/6000/7000 ou XBT GK. Un connecter RJ45 à 8 broches est utilisé [3]

Tableau6: Interface utilisée pour connecter un câble série RS-485

Connexion de broche	Pin	Nom du signal	du	Direction	Sens
	1	Non connecté	-	-	-
	2	Non connecté	-	-	-
	3	Non connecté	-	-	-
	4	D1	Sortie	entrée	Transférer des données (RS-485)
	5	D0	Sortie	entrée	Transférer des données (RS-485)
	6	RTS	Production		Demande d'envoi
	7	Non connecté	-	-	-
	8	SG	-	-	Masse du signal



1.5.4.5 Interface série COM2:

Le tableau suivant représente l'interface série COM2 des unités des séries XBT GT2000/4000/5000/6000/7000 et XBT GK Interface[3]

Tableau7: l'interface série COM2

Interface	La description
Interface série COM2 RJ45	
Transmission asynchrone	RS-485
Longueur des données	7 ou 8 bits
Bit d'arrêt	1 ou 2Bit
Parité	Aucun, pair ou impair
Vitesse de transmission des données	2,400bps à 187.5Kbps

1. 6avantages de l'interface homme-machine :

Les HMI se présentent sous de nombreuses formes, des écrans intégrés aux machines, aux écrans d'ordinateur, en passant par les tablettes, vous permettant de communiquer avec n'importe quel système de production. Améliorer l'efficacité et les performances de la machine, plus économique et plus sûre. Satisfaire les besoins physiques et psychologiques des utilisateurs et améliorer leur satisfaction.

Fonctionne dans des environnements difficiles. A des graphismes riches et un bon écran rétroéclairé.

Dispose de ports USB, Ethernet et carte SD. Avec capacité d'enregistrement de données[4]

1.7 Conclusion :

Le choix de la bonne HMI est tout aussi important que le choix de l'automate lui-même. Une HMI bien conçue doit faire plus que contrôler le processus. Il doit être sûr, fiable et rentable, et il doit fournir à l'opérateur une vue d'ensemble du processus. L'HMI est le principal point de contact entre l'utilisateur et le processus.

Il doit exécuter toutes les fonctions avec un minimum d'effort, augmentant la productivité pour satisfaire les utilisateurs.

CHAPITRE2: Protocol De communication
Modbus

CHAPITRE02: Protocol De communication Modbus

2.1 Introduction

Modbus est un protocole de communication industriel introduit par Modicon en 1979 et est généralement utilisé dans les automates programmables ou les équipements industriels. Aujourd'hui, il est devenu une norme de "protocole ouvert" dans le domaine de l'automatisation et de la communication industrielle, et c'est le moyen de communication des équipements industriels le plus couramment utilisé. L'un des avantages du protocole Modbus est sa flexibilité et sa facilité de mise en œuvre. La plupart des appareils et équipements embarqués tels que les microcontrôleurs, les automates, les capteurs intelligents, etc. sont équipés d'une interface Modbus et sont capables de communiquer en Modbus. Conçu à l'origine pour les lignes de communication série câblées, Modbus a maintenant été étendu aux normes de communication sans fil et de mise en réseau TCP/IP. Le protocole Modbus permet la communication entre plusieurs appareils connectés sur le même réseau, par exemple, un système mesurant la température et l'humidité du four peut transmettre ses résultats à un ordinateur de processus via Modbus.[5]

2.2 Historique de modbus

1979: Création de Modbus par MODICON (Modular Digital Controller).

1994 : Modicon fusionne avec Schneider (Télemécanique / April / Square D).

2003 : Transfert de compétences de Schneider vers Modbus-IDA.

2004 : Pré-norme internationale IEC62030.

2004 : Leader mondial Modbus/TCP (840 000 nœuds).

2005 : Modbus devient une norme chinoise. [6]

2.3 Principe de fonctionnement du modbus

Dans le protocole MODBUS, le maître initie la communication en demandant qu'une action soit effectuée par l'esclave ou tous les esclaves. Seul le maître a le droit d'initier la communication avec tous les esclaves.

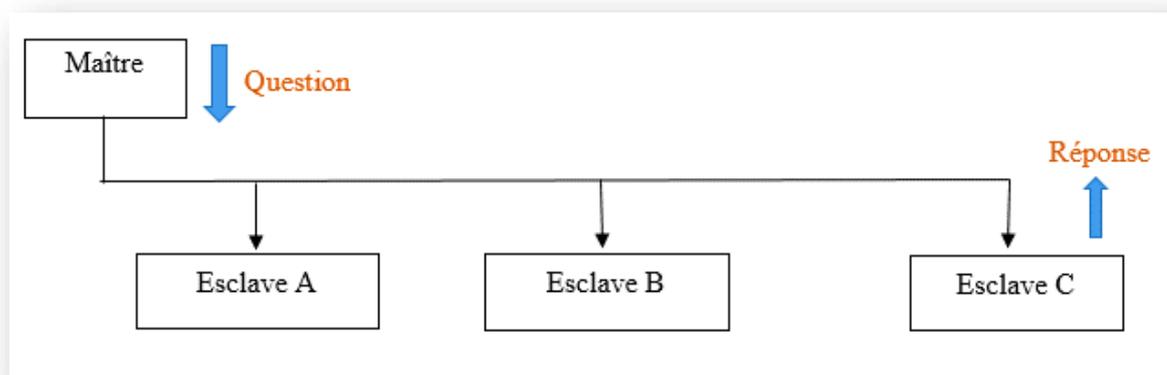


Figure 5:Principe de fonctionnement du Modbus

Les esclaves qui sont toujours à l'écoute interceptent la totalité des échanges de messages sur le bus auquel ils sont connectés. Chaque esclave est identifié par sa propre adresse unique (un esclave ne peut pas avoir la même adresse qu'un autre esclave connecté au BUS). Il ne répondra que si le message lui est adressé. Le maître envoie une requête et attend une réponse. Les deux esclaves ne peuvent pas communiquer entre eux. [7]

2.3.1 La requête

La requête contient un code fonction qui indique à l'esclave adressé quel type d'action Exiger. Les données contiennent des informations supplémentaires requises par l'esclave pour exécuter cette fonction. Le champ d'octet de contrôle permet à l'esclave de s'assurer de l'intégrité du contenu en question.

2.3.2 La réponse

Lorsque l'esclave envoie une réponse après une transaction normale, La réponse fait écho à la réponse contenue dans la question. Les données sont les données collectées par l'esclave, telles que la valeur d'un registre. En cas d'erreur, modifiez le code de la fonction en indiquant que la réponse est un Réponse d'erreur. Les données contiennent alors un code d'exception qui permet de connaître le type d'erreur. Le champ de contrôle permet au maître de

confirmer que le message est valide. Après que l'esclave a reçu une question, il vérifie la cohérence de la trame. Oui Un paramètre illégal (code fonction, adresse, valeur...) a été détecté ou la station n'a pas pu exécuter la requête.[6]

2. 4 Architecture du réseau Modbus

Le réseau Modbus est un réseau composé d'un seul maître et d'esclaves, qui communiquent par une liaison série asynchrone. On peut utiliser n'importe quel support de transmission RS 232 ou RS 485, mais la liaison RS 485 est la plus répandue car elle autorise le mode « multipoints ».[6]

2.4.1 La liaison RS 232

C'est la norme de communication série la plus connue. Un port série RS232 est disponible sur la plupart des PC standard. Il s'agit d'un type point à point et se compose de lignes Rx, Tx et GND.

RS232 ne permet qu'à un maître et un esclave de communiquer sur chaque ligne. Il fonctionne en mode full duplex et sa vitesse de communication peut atteindre jusqu'à 115 kbits/s.

En RS232, la distance entre deux appareils n'est généralement pas supérieure à 15 m. Si vous n'avez pas besoin d'ajouter plusieurs esclaves sur la même ligne, vous devrez utiliser une liaison RS422 ou RS485 plus adaptée.

L'inconvénient du RS232 est qu'il n'est pas adapté aux environnements avec beaucoup de bruit ou d'interférences (risque d'interférence de transmission).[5]

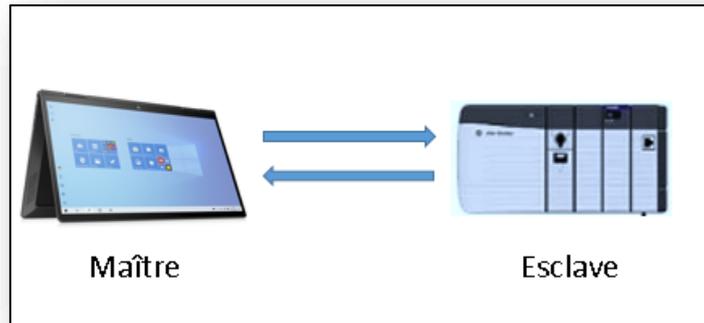


Figure 6:La liaison série type RS 232

2.4.2 La liaison RS 485

Les médias de type RS485 sont souvent en half duplex c'est-à-dire la transmission s'effectue via 2 fils.

Ils permettent de faire communiquer jusqu'à 32 périphériques sur la même ligne de données et sur une distance pouvant aller jusqu'à 1200 m sans répéteurs.

A noter que l'on peut obtenir du full duplex en utilisant 4 fils de transmission au lieu de 2. Cela permet d'avoir un débit de transmission plus rapide. Chaque périphérique esclave peut aussi communiquer avec les 32 autres périphériques. Les protocoles de communication RS422 et RS485 sont multi-drop c'est à dire plusieurs périphériques peuvent communiquer sur la même ligne de données.[5]

Le RS485 a comme avantages d'être immunisé contre les bruits ou parasites.

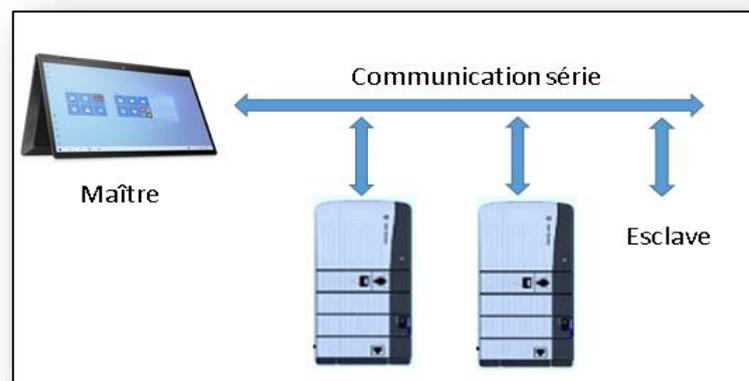


Figure 7:La liaison série type RS 485

Les caractéristiques générales d'une liaison RS 485 sont:

- Topologies possibles : point à point ou multipoints de type bus.
- Topologie déconseillée : en étoile.
- Distance minimale entre 2 points : 27 cm.
- Longueur maximale d'une dérivation : 1m.
- Vitesse de transmission : 4800/9600/19200 Bauds.
- Débit nécessaire à l'utilisation : optimisation des liaisons en fonction du besoin, maximum 10 Mbits/s.[6]

2.5 Les types de données Modbus

Il y a deux types de données dans Modbus : les bobines et les registres.

2.5.1 Les bobines :

sont simplement des bits. Les bits peuvent être activés (1) ou désactivés (0).

Certaines bobines représentent des entrées, ce qui signifie qu'elles contiennent l'état d'une entrées physiques discrètes (tout ou rien) ou elles représentent des sorties, ce qui signifie qu'elles contiennent l'état d'un signal de sortie physique discret.

2.5.2 Les registres : uniquement des données de registre non signées de 16 bits. La valeur de registre peut être de 0 à 65535 (hexadécimal de 0 à FFFF). Les valeurs négatives ne sont pas représentées, les valeurs supérieures à 65 535 ne sont pas représentées et les données réelles telles que 200.125 Ne sont pas représentées.

Les registres sont divisés en registres d'entrée et en registres de stockage. Comme les bobines d'entrée, les registres d'entrée indiquent l'état d'une entrée externe sous la forme d'une valeur comprise entre 0 et 65535.

L'objectif initiale d'un registre d'entrée était de refléter la valeur d'une entrée analogique. C'est une représentation numérique d'un signal analogique comme une tension ou un courant. La plupart des équipements Modbus actuels ne sont pas des équipements d'E/S et les registres d'entrée fonctionnent simplement 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. Aujourd'hui, les registres de stockage sont utilisés comme stockage de données pour les appareils.[7]

2.6 Zones mémoires du Modbus

Le protocole Modbus fonctionne avec un maître et jusqu'à 247 esclaves qui fournissent des informations au maître.

Les mémoires des esclaves MODBUS sont organisées en 4 zones comme indiquer dans la table suivante :

Tableau8 :Zones mémoire des esclaves Modbus

Plage d'adresse	Adresse	Accès	Type de la variable
00001 à 09999	0000 à 270 E	Lire - Ecrire	Bobine
10001 à 19999	0000 à 270E	Lire seulement	Entrée discrète
30001 à 39999	0000 à 270E	Lire seulement	Registre d'entrée
40001 à 49999	0000 à 270E	Lire - Ecrire	Registre général

2.6.1 Zone des bobines (coils) :cette zone de mémoire est organisée en bits (de 00001 à 09999) qui contiennent les valeurs des sorties discrètes Tout ou Rien.

2.6.2 Zone de mémoire d'entrée (entrées discrètes) :cette zone est organisée en bits, où se trouvent les valeurs des entrées discrètes, la plage des adresses dans cette zone est entre 10001 et 19999.

2.6.3 Zone des registres d'entrées :sont des registres de 16 bits qui contiennent les valeurs des entrées analogiques, elle occupe les adresses : 30001 à 39999.

2.6.4 Zone des registres à usage général (holding registers) :en unités de 16 bits, occupant les adresses du 40001 à 49999 où se trouve les valeurs de sorties.[7]

2.7 Quelques Caractéristique Du Modbus

Certaines caractéristiques du protocole Modbus sont fixes, telles que le format de trame, les séquences de trames, la gestion des erreurs de communication, les conditions d'exception et les fonctions exécutées. Les autres caractéristiques sont facultatives. Il s'agit du support de transmission, des caractéristiques de transmission et du mode de transmission (RTU ou ASCII). Les caractéristiques de l'utilisateur sont définies sur chaque appareil et ne peuvent pas être modifiées lorsque le système est en cours d'exécution.[7]

2.8 Modes de transmissions et format de trames

Il existe deux modes de transmission dans le protocole Modbus : RTU et ASCII, qui déterminent comment les messages Modbus sont encodés.

2.8.1 Modbus ASCII :

Le début de chaque message est marqué par un caractère ASCII ":" et la fin de chaque message se termine par un caractère ASCII de retour chariot/saut de ligne (CR/LF). Cela permet de modifier l'espacement entre les octets dans un message, ce qui le rend adapté à la transmission via certains modems. Les données des messages Modbus ASCII utilisent des caractères ASCII. Pour chaque octet de 8 bits, 1 bit de départ, 7 bits de données, 1 bit de parité et 1 bit d'arrêt sont envoyés, pour un total de 10 bits. Les messages Modbus ASCII se terminent par une somme de contrôle d'erreur appelée contrôle de redondance longitudinale (LRC).[7]

2.8.2 Modbus RTU :

Dans le modbus RTU, les octets sont envoyés consécutivement, sans espaces entre eux, et un espacement de 3-1/2 caractères entre les messages afin que le logiciel d'interface Modbus sache quand commencer un nouveau message. Pour chaque octet de 8 bits, 1 bit de départ, 8 bits de données, 1 bit de parité et 1 bit d'arrêt sont envoyés, pour un total de 11 bits par octet. Chaque message Modbus RTU se termine par une somme de contrôle d'erreur appelée contrôle de redondance cyclique (CRC).[7]

2.9 Trame d'échange requête/réponse pour Modbus RTU

Une trame MODBUS RTU se compose d'une série de caractères hexadécimaux et contient les informations suivantes :

- Numéro d'esclave (1 octet) (le numéro 00 est réservé aux messages de diffusion)
- Code fonction (1 octet)
- Données (n octets)
- CRC (2 octets)

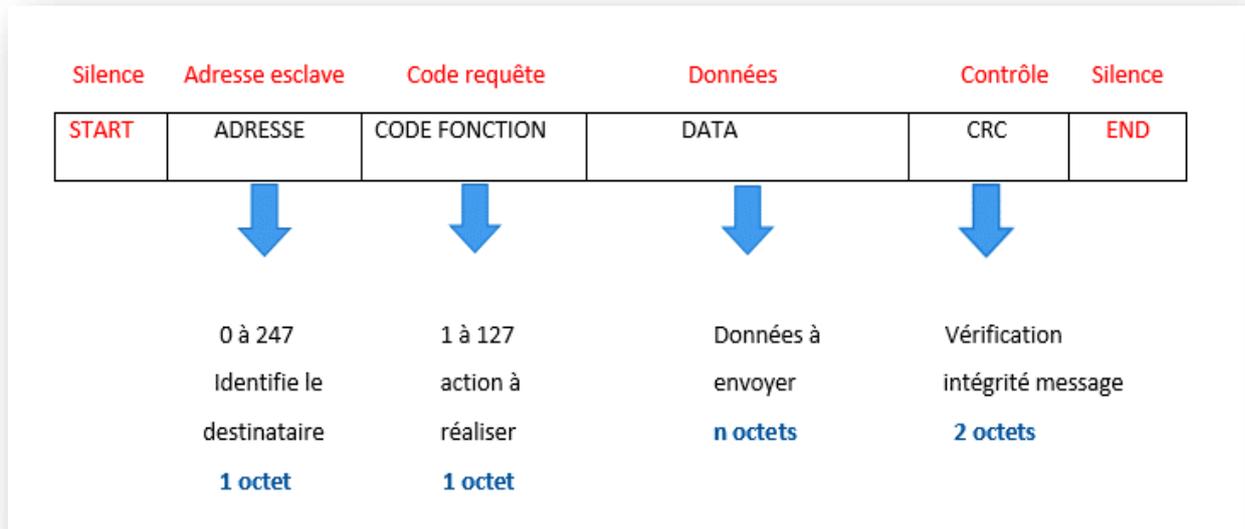


Figure 8:Trame de Modbus RTU

Modbus permet la communication maître/esclave entre des appareils connectés via un bus ou un réseau. Dans le modèle OSI, Modbus est au niveau 7. Modbus est un protocole de requête/réponse qui fournit des services spécifiés par des codes de fonction. Les codes de fonction Modbus sont des éléments d'une requête/réponse Modbus PDU (Protocol Data Unit).

La demande contient :

- Adresse esclave : il s'agit de l'adresse du nœud Modbus RTU dans la plage 1 à 255. 0 est réservé aux messages de diffusion et d'écriture uniquement.
- Un code fonction indiquant à l'esclave adressé le type d'opération demandée.
- Champ de données : Encodé en n mots, de 00 à FF hex, contenant les informations permettant à l'esclave de décoder le message.
- Un champ de contrôle d'erreur CRC (CyclicRedundancy Check) : contient une valeur codée sur 16 bits.

Le tableau suivant montre les différents codes fonction qui existent :

Tableau9:Code fonctions de la trame Modbus

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 bit de sortie
06	Ecriture de 1 mot de sortie
07	Lecture de statu d'exception
08	Accès aux compteurs de diagnostic
0F	Ecriture de n bit de sortie
13	Reset de l'esclave après erreur non recouverte

La trame de réponse de l'esclave comprend : Elle prend la forme suivante :

- Champ d'adresse contenant l'adresse de l'esclave.
- Le même code de fonction que le message envoyé par l'hébergeur.
- Champ de données contenant des informations sur la demande de l'hôte.
- Le champ Error Control contient une valeur codée sur 16 bits. La valeur est le résultat du CRC calculé à partir du message.[7]

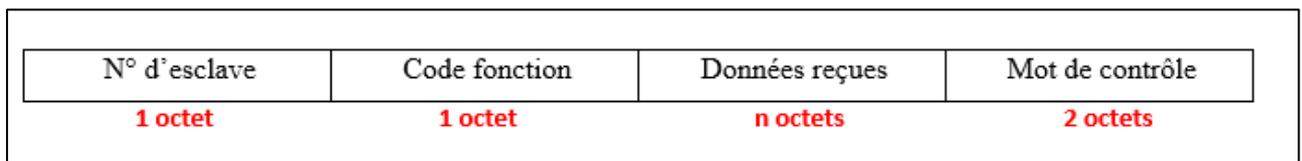


Figure 9:Trame de réponse d'un esclave Modbus

2.10 Conclusion :

Le protocole MODBUS permet une communication aisée dans tous les types d'architectures de réseau. Tous les types d'appareils (automates, HMI, panneaux de contrôle, variateurs, contrôle de mouvement, périphériques d'E/S...) peuvent utiliser le protocole MODBUS pour initier un fonctionnement à distance. La même communication est possible sur des lignes série et des réseaux Ethernet TCP/IP. Les passerelles permettent la communication entre de nombreux types de bus ou de réseaux en utilisant le protocole MODBUS.

***CHAPITRE 3 : Réalisation Matérielle et
Logicielle***

CHAPITRE 03:Réalisation Matérielle et Logicielle

3.1 Introduction

Après avoir présenté dans le chapitre précédent le protocole de communication Modbus qu'on a utilisée pour assurer la communication entre la carte arduino et l'écran tactile MAGELIS Schneider, on décrira dans ce chapitre les étapes suivies dans sa programmation.À cet égard nous avons divisée ce chapitre en plusieurs tests, en premier temps nous avons établir la communication entre la carte arduino et un câble USB à partir d'un module RS485, et à la deuxième pas en a remplacer le câble USB par l'HMI MAGELIS Schneider.

3.2 Le circuit complet

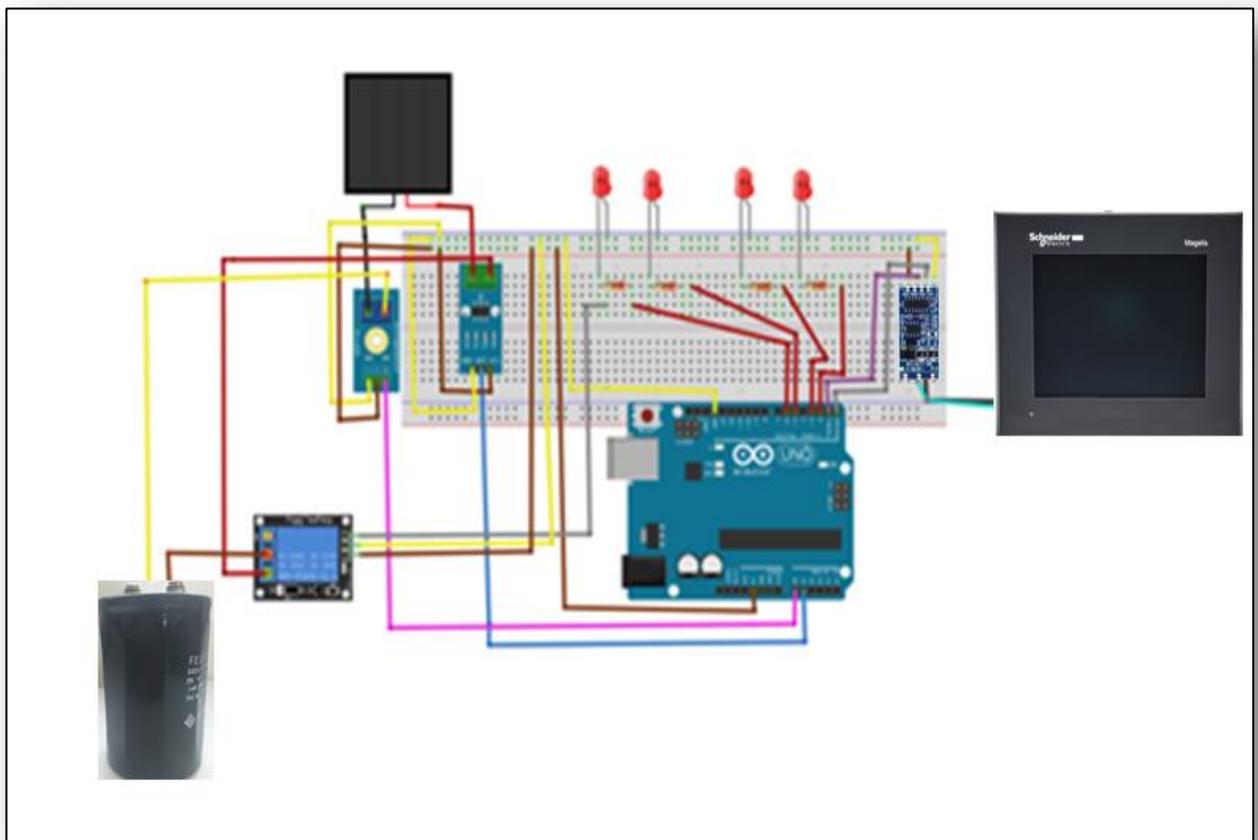


Figure 10:shéma global

On a essayé de réaliser le schéma global représenté dans la figure (9), et pour cela en passe par plusieurs tests :

3.3 Première test: La communication en utilisant un convertisseur USB RS485:

Dans ce premier test en a essayé de réalisé la transmission des données par l'utilisation d'un PC et la carte arduino pour allumer les quatre LED.

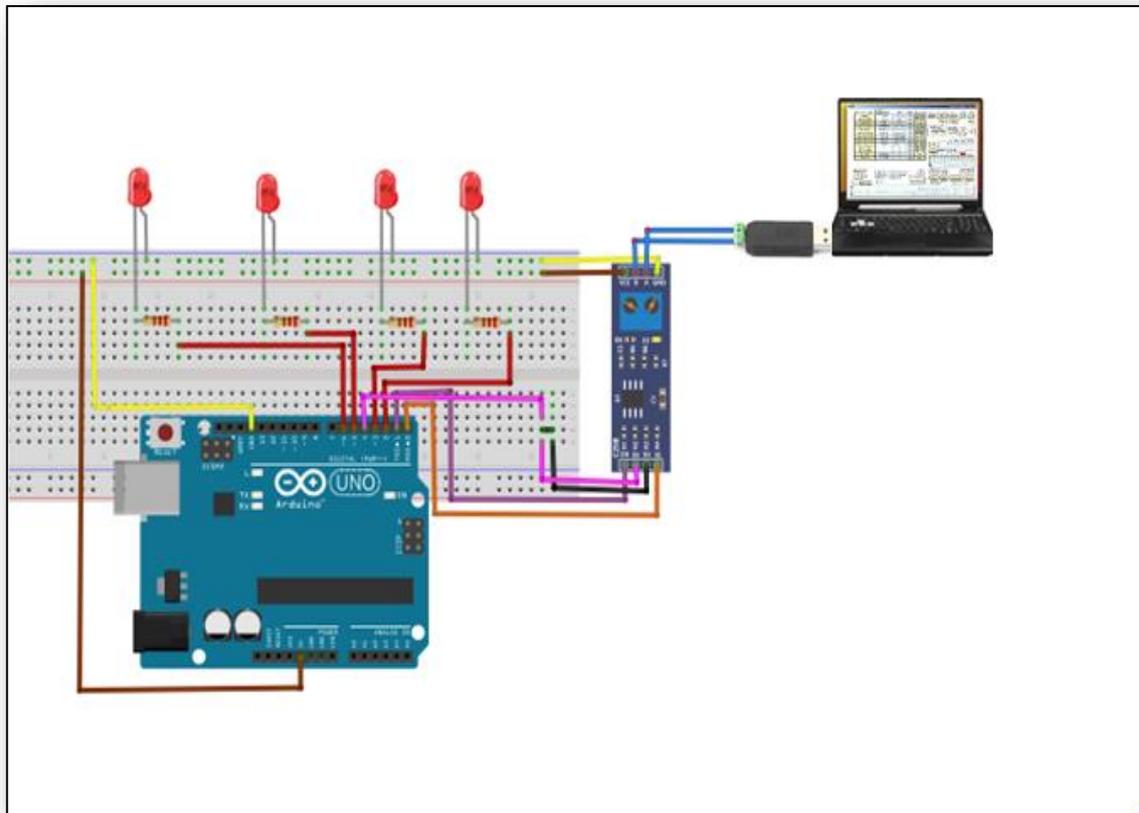


Figure 11: la communication en utilisant un convertisseur USB RS485

3.3.1 Description de la partie matérielle :

3.3.1.1 Présentation de La carte Arduino UNO :

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif. Constituée d'une carte électronique et d'un environnement de programmation. Sans tout connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.[8]

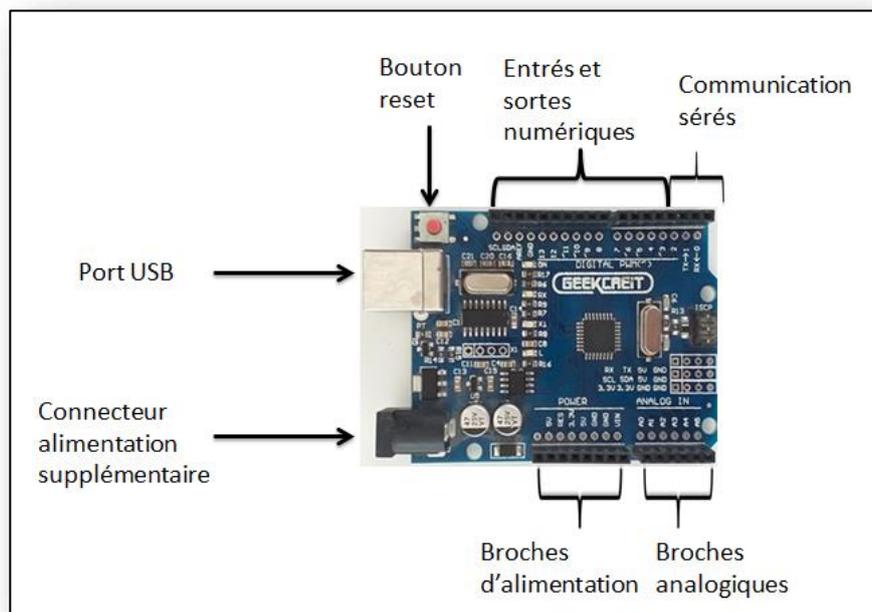


Figure 12:La carte Arduino UNO

3.3.1.2 Caractéristiques de la carte Arduino UNO :

Tableau10:Caractéristiques de la carte Arduino UNO

Microcontrôleur	A Tmega328
Tension d'alimentation interne	5 V
Tension d'alimentation (recommandée)	7 à 12 V. limite 8-6 à 20 V
Entrées / sorties numériques	14 dont 6 sorties PWM
Entrées analogiques	6
Courant max par broches E/S	40mA
Courant max sur Sortie 3.3V	50mA
Mémoire Flash	32KB dont 0.5 KB utilisée par le boot loader
Mémoire SRAM	2KB
Mémoire EEPROM	1KB
Fréquence horloge	16 MHz

3.3.1.3 Avantages de la carte Arduino UNO :

- Pas cher
- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows,....
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso etc...)
- Existence de « shield » (boucliers en français) : ce sont des cartes supplémentaires qui se

Connectent sur le module Arduino pour augmenter les possibilités comme par exemple :

Afficheur graphique couleur, interface ethernet, GPS, etc...

Par sa simplicité d'utilisation, Arduino est utilisé dans beaucoup d'applications comme

L'électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des

Domaines différents comme l'art contemporain ou le spectacle. [9]

3.3.1.4 Module convertisseur MAX485 RS 485 TTL vers RS485 5V :

Une puce MAX485 de la carte à faible consommation d'énergie pour Communication RS-485, limite l'émetteur-récepteur de la vitesse de réponse.

- À bord des bornes à pas de 5,08 (mm) 2P, pour faciliter le câblage de communication RS-485
- Puce toutes les broches ont été conçues par le microcontrôleur pour contrôler le fonctionnement
- Tension de fonctionnement: 5V
- Dimensions de la carte: 46 (mm) x12 (mm)[10]

La longueur de câble maximale dans les communications RS-485 est 4000 pieds ou environ 1200 mètres Une directive générale, cependant, est que le produit de la longueur de ligne (en mètres) et du débit de données (en bits par seconde) ne doit pas être supérieur à 108. Par exemple, un câble de 20 mètres permet un débit de données maximum de 5 Mbits/s.[11]

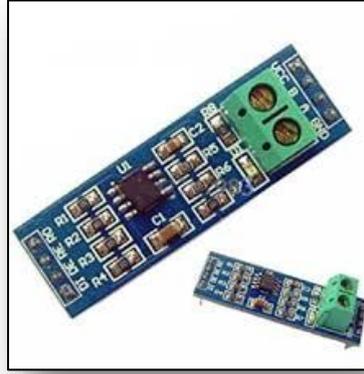


Figure 13:Module convertisseur MODBUS vers RS485 5V

3.3.1.5 USB à RS485 Convertisseur Adaptateur :

Un adaptateur permet de relier un système RS485 sur un port USB

- USB connecteur: à votre PC
- RS485 connecteur: à votre RS485 dispositif
- pas besoin d'alimentation externe, alimenté par USB port.



Figure 14:USB à RS485 Convertisseur

3.3.2 Description de la partie logicielle :

3.3.2.1 Arduino IDE :

Le logiciel Arduino IDE, basé sur le langage C++, possède une bibliothèque de Développement riche. Une fois, le programme tapé ou modifié, il sera transféré et mémorisé dans la carte à travers de la liaison USB.

Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information.[12][13]

3.3.2.2 Structure d'un programme Arduino :

Le programme Arduino est composé de 3 parties principales :

La partie optionnelle : déclaration des constantes et des variables.

La partie initialisation et configuration des entrées/sorties dans la fonction Setup ()

La partie principale qui s'exécute en boucle (la fonction Loop), elle est réservée aux instructions à effectuer par le programme.[12]

3.3.2.3 L'interface du logiciel arduino :

L'interface du logiciel Arduino se présente de la façon suivante :

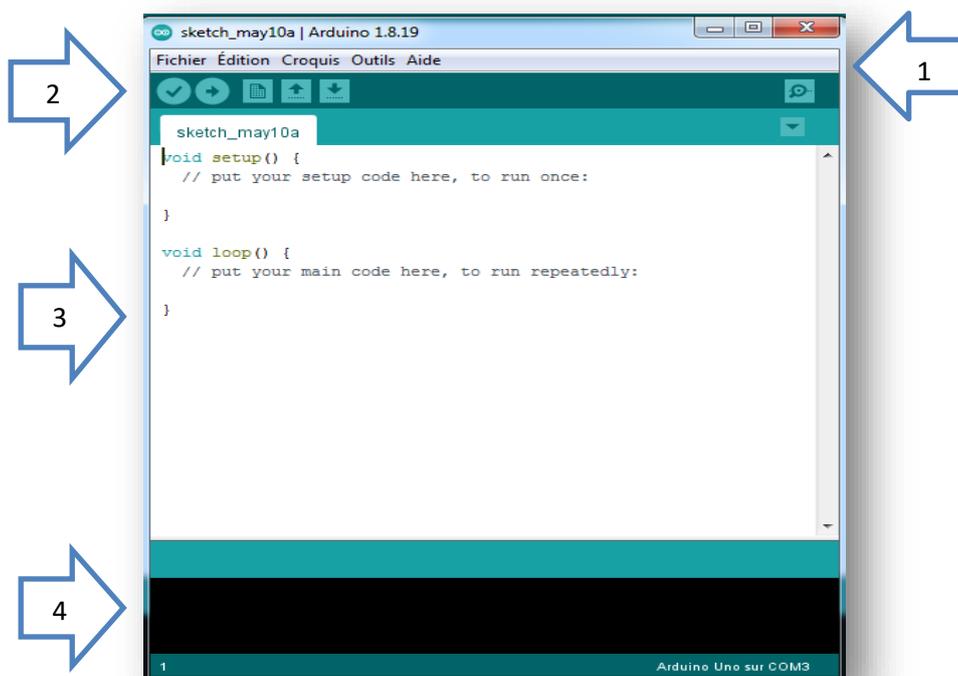


Figure 15:Interface de logiciel Arduino

1. Options de configuration du logiciel
2. Boutons pour la programmation des cartes
3. Programme à créer

4. Débogueur (affichage des erreurs de programmation)

Les boutons :

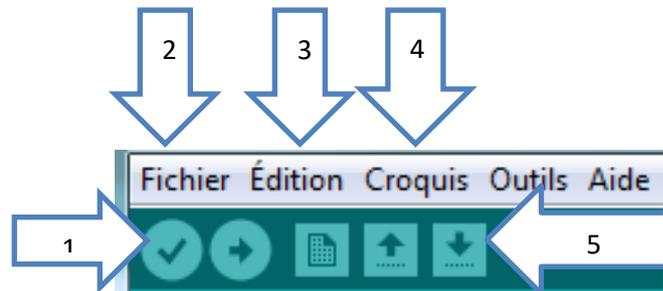


Figure 16:Les boutons de logiciel Arduino

1. Permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le Programme
2. Compiler et envoyer le programme vers la carte
3. Créer un nouveau fichier
4. Charger un programme existant
5. Sauvegarder le programme en cours

3.3.2.4 La partie programmation du premier test:

- En premier temps en doit installer la bibliothèque modbus.

```
#include<ModbusRtu.h>
```

- On définit les led 1, 2, 3,4 broché sur les pins 2, 3, 5,6 associées aux sorties digitales D'arduino.

```
#define led1 2
#define led2 5
#define led3 3
#define led4 6
```

- On suite on définit le bus d'objets pour la classe modbus et un tableau initialisé avec trois valeurs 0 pour le stockage des données.

```
Modbus bus;  
uint16_t modbus_array[] = {0,0,0,0,0,0,0,0};
```

- Dans la fonction void setup on définit les led comme des sorties.

```
void setup()  
{  
  
    pinMode(led1,OUTPUT);  
    pinMode(led2,OUTPUT);  
    pinMode(led3,OUTPUT);  
    pinMode(led4,OUTPUT);  
}
```

- On suite on définit le ID d'esclave modbus comme 1 et 1 connecté via RS-485 et 4 connecté à la broche DE & RE du module RS-485
- On définit aussi la vitesse de communication série sur 19200 bauds

```
Serial.begin(19200);  
  
bus = Modbus(1,1,4);  
bus.begin(19200);
```

- Dans la fonction loop on fait Cette instruction qui est Utilisée pour recevoir ou écrire une valeur depuis le maître.

```
bus.poll(modbus_array,sizeof(modbus_array)/sizeof(modbus_array[0]));
```

- On fait des boucles if pour allumer les quatre led.

```

if (modbus_array[0] == 0)
{
    digitalWrite(led1,LOW);
}
else
{
    digitalWrite(led1,HIGH);
}
    
```

3.3.2.5 Simplymodbus master :

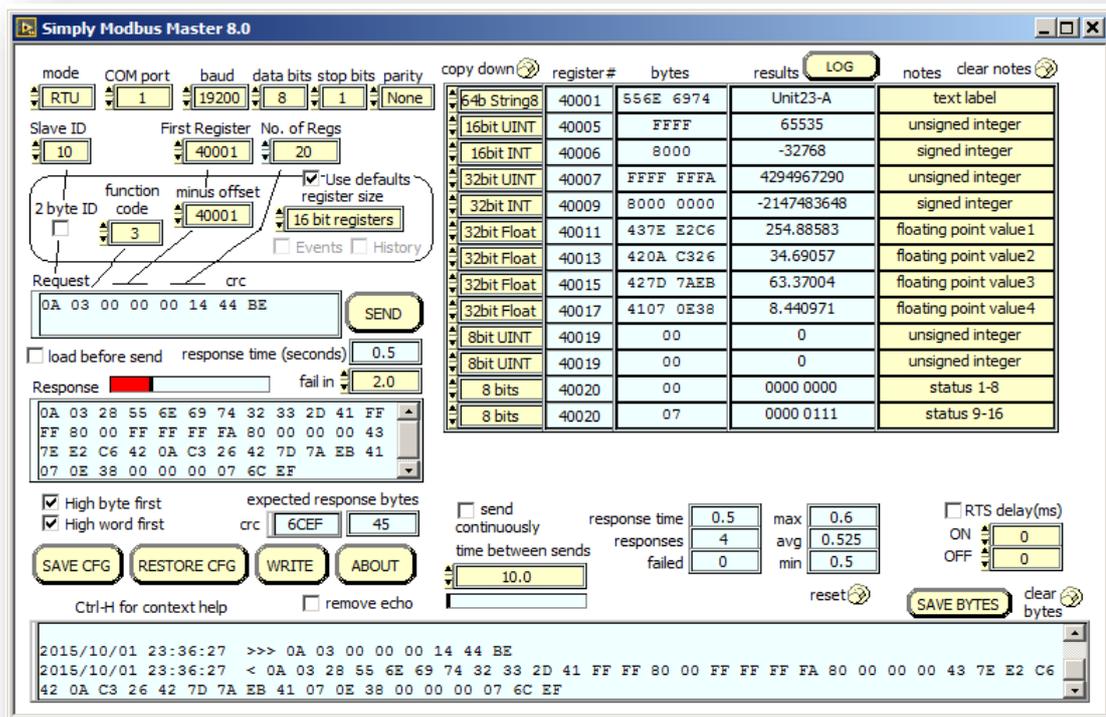


Figure 17: Interface du logiciel Simplymodbus master

Jaune : Commandes (entrées) pouvant être modifiées par l'utilisateur.

Bleu : Indicateurs (sorties) qui ne peuvent pas être modifiés directement

mode
 Select Modbus RTU

COM port baud data bits stop bits parity


Port COM: port série du PC vers un appareil esclave modbus. Plage : COM1 à COM99

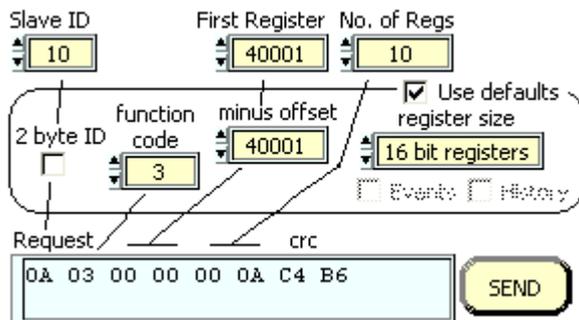
baud : Le débit en bauds (bits par seconde) de la connexion série.

bits de données : le nombre de bits de données dans chaque octet, le mode RTU en nécessite 8, le mode ASCII est généralement de 7 mais peut être de 8.

bits d'arrêt : le nombre de bits d'arrêt pour chaque octet envoyé. Les valeurs autorisées sont 1, 1,5 ou 2.

parité : la valeur du bit de parité pour chaque octet envoyé. Les valeurs autorisées sont aucune, impaire, paire, marque ou espace.

3.4.2.1 Construire une chaîne de requête :



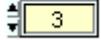
Slave ID
 Le 1er octet de la chaîne de commande

L'adresse d'unité de l'appareil esclave à partir duquel obtenir des données.

plage normale : 1 à 247

2 byte ID La définition d'un ID esclave supérieur à 255 cochera automatiquement cette case et autorisera les ID esclave jusqu'à 65535.

Reste allumé pour toutes les adresses jusqu'à ce que l'ID esclave soit abaissé en dessous de 256 et que la case soit décochée manuellement.

function
code
 Le 2ème octet dans la chaîne de requête

Peut être réglé automatiquement

Utilisé pour sélectionner la table à lire

La fenêtre de lecture prend en charge les codes de fonction 01, 02, 03 et 04

First Register Les 3e et 4e octets de la chaîne de requête


Le numéro de bobine ou de registre au début du bloc à lire

minus offset Cette valeur est soustraite du premier registre pour donner l'adresse de données
utilisée dans la requête


No. of Regs Les 5e et 6e octets de la chaîne de requête

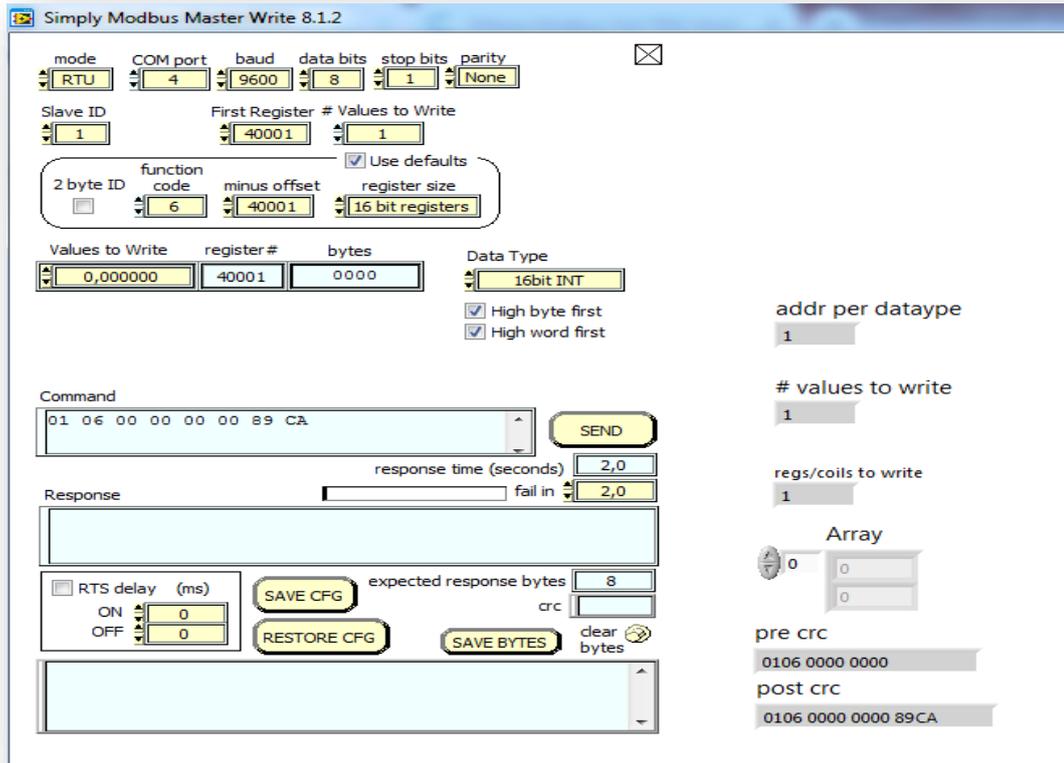

La quantité de registres ou de bobines dans le bloc à lire.

register size Modbus La taille des registres du bloc à lire.

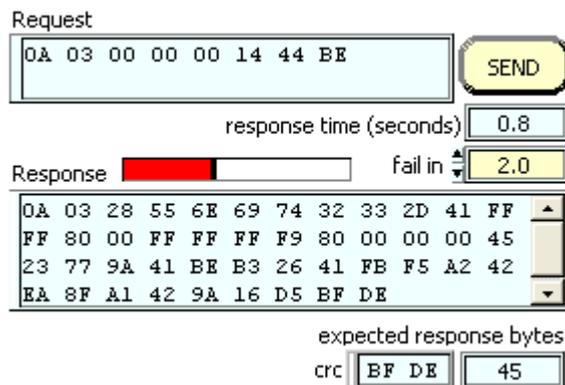

Cette valeur doit être définie sur des registres de 16 bits pour lire les registres modbus standard et réglez les bobines sur 1 bit modbus standard

Les registres 32 bits doivent être utilisés uniquement pour lire les bobines pour le modbus Enron.

Pour envoyer une donnée on appuie sur le bouton WRITE. La fenêtre suivante apparaît où les données à envoyer sont saisies.



Appuyez sur le bouton SEND. Le programme surveille alors le port série et affiche tous les octets reçus dans l'indicateur de réponse.



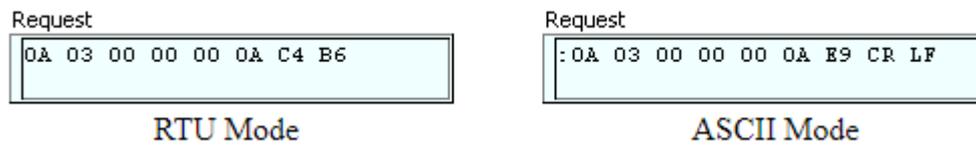
Lors de la réception, le temps de réponse compte avec une barre de progression.

La réception s'arrête lorsque les octets de réponse attendus sont reçus ou que le temps atteint la valeur d'échec saisie.

L'indicateur « octets de réponse attendus » est calculé à partir des paramètres d'entrée.

Le temporisateur de réponse s'arrête lorsque ce nombre d'octets est reçu.

Le crc attendu est calculé à partir des octets de la réponse.



crc Les 2 derniers octets de la requête RTU sont le contrôle de redondance cyclique. Ce sont des octets de détection d'erreur

lrc Les messages en mode ASCII sont précédés de deux points et le crc est remplacé par un contrôle de redondance longitudinale, un retour chariot et des caractères de saut de ligne. [2]

3.4 Deuxième teste: la communication entre la carte arduino et L'HMI :

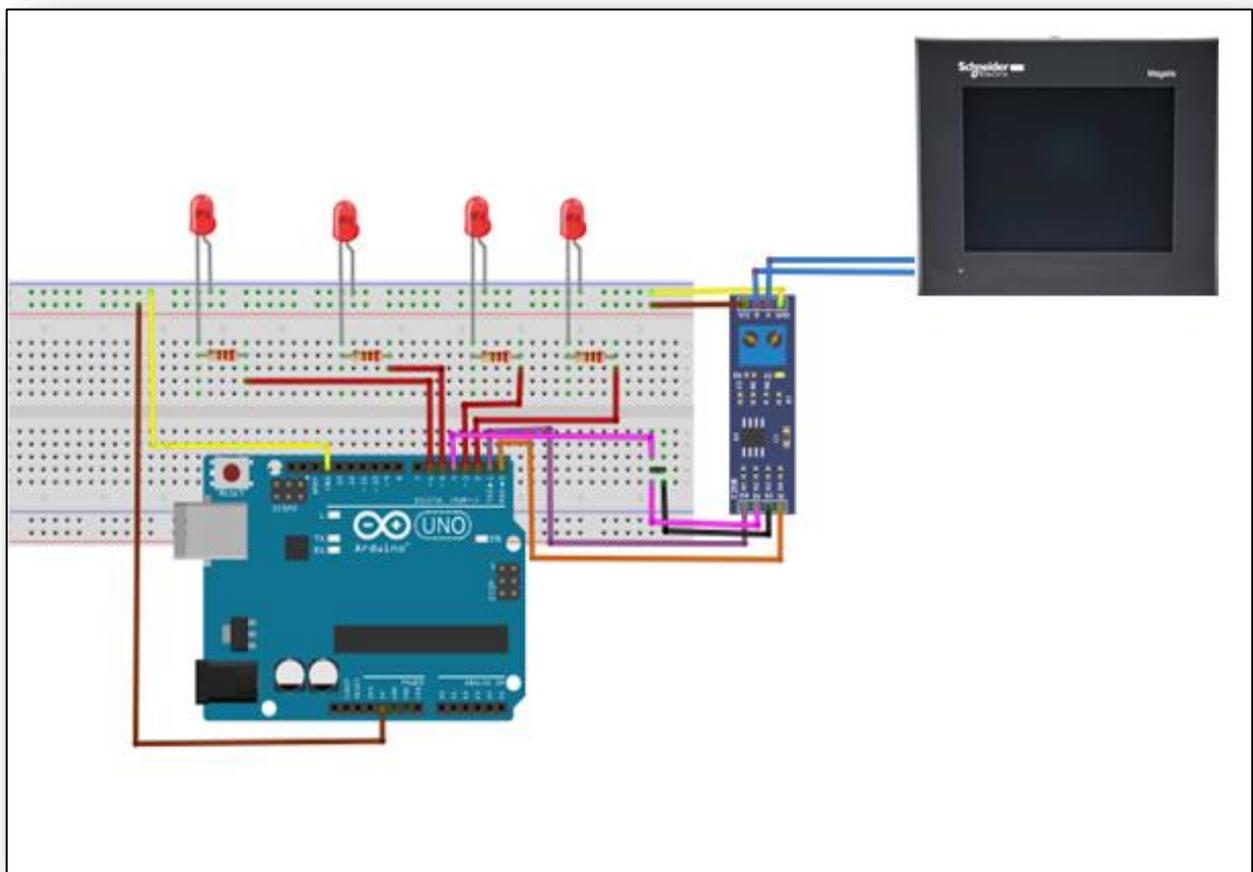


Figure 18:schéma de la Deuxième teste

Dans la deuxième expérience, nous avons déconnecté le fil USB et l'ordinateur et connecté le périphérique HMI qui est programmé avec le programme vijeo designer.

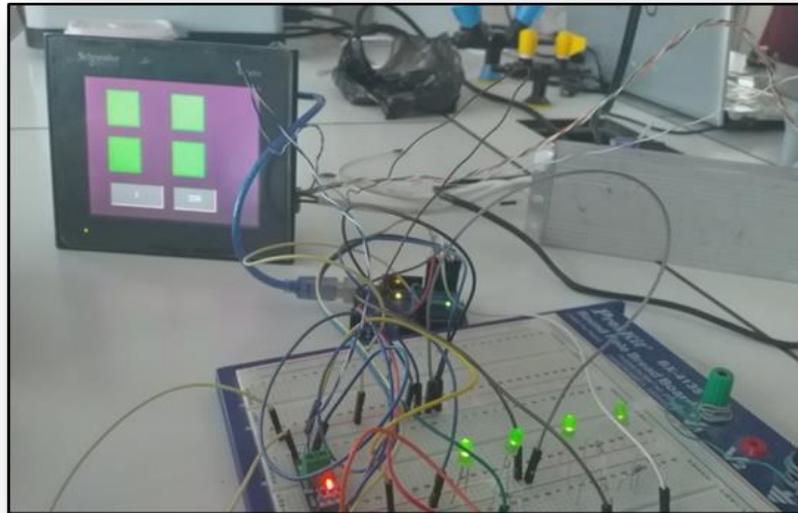


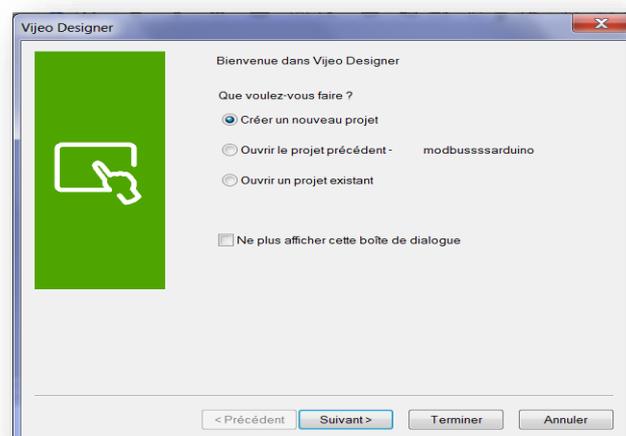
Figure 19:Réalisation pratique du Deuxième test

3.4.1 Description de la partie logicielle :

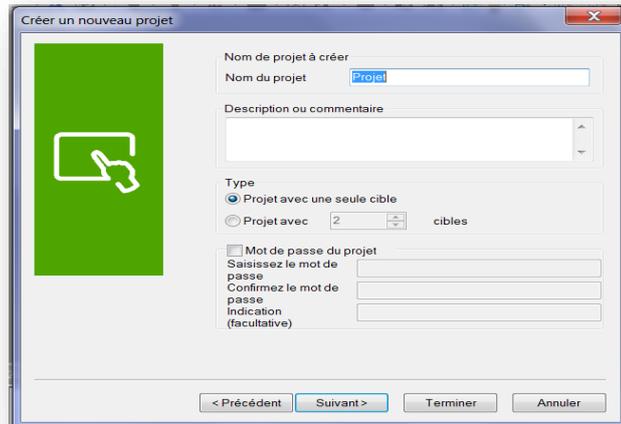
3.4.1.1 Logiciel vijeo Designer :

Pour créer un nouveau projet dans vijeo Designer en suit les étapes suivantes :

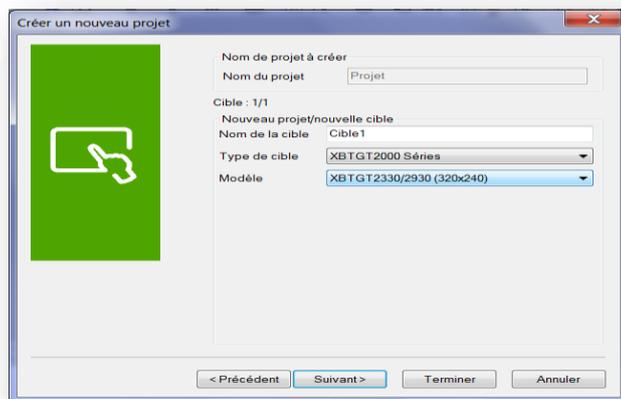
- D'abord en cliquant sur créer un nouveau projet



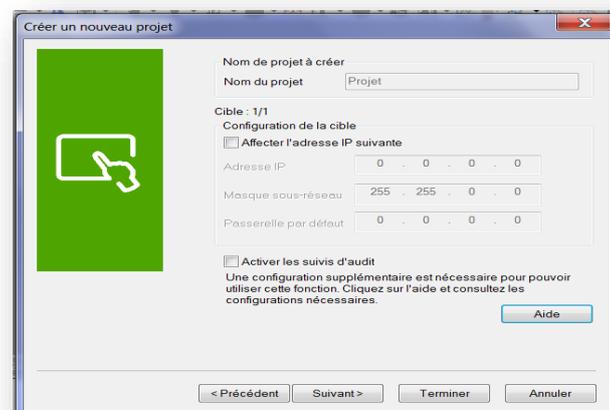
- En suite en choisit le non du notre projet



- après en choisit le nom du série d'HMI utilisée dans notre projet



- Dans la figure suivante en affecter l'adresse IP si en travail par modbus TCP/IP ou Ethernet sinon en laissé tel qu'elle est et en clique sur suivant



- Le premier programme dans logiciel vijeo designer pour allumer les quatre led.

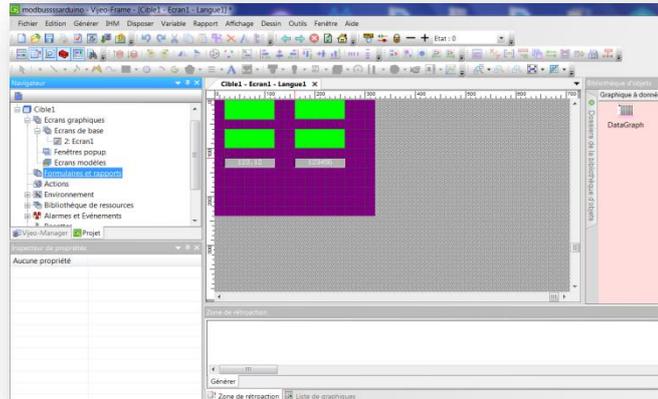


Figure 21:Le premier programme dans logiciel vijeo designer

3.5 Troisièmes tests :

Dans ce test en remplace juste le convertisseur MAX485 RS 485 TTL 5V par un ARCELI TTL à RS485 Adaptateur 485 Serial Port UART Level Converter Module 3.3V 5V parce que ça rendre la communication plus robuste.



Figure 22:Carte ARCELI TTL à RS485 Adaptateur

3.5.1 Description de la carte ARCELI TTL à RS485 Adaptateur 485 Serial Port UART Level Convertir Module 3.3V 5V :

Un module qui permet de transférer l'interface TTL du microcontrôleur vers le module RS485. Il est généralement utilisé pour l'automatisation industrielle. Ce module ajoute une conception antibrouillage.[14]

3.5.2 La partie programmation du troisième test:

Dans ce teste on ajoute des instructions qui permet de contrôler l'intensité lumineuse de la troisième led.

```
analogWrite(led3, modbus_array[2]);
```

```
int pwm = modbus_array[2];
```

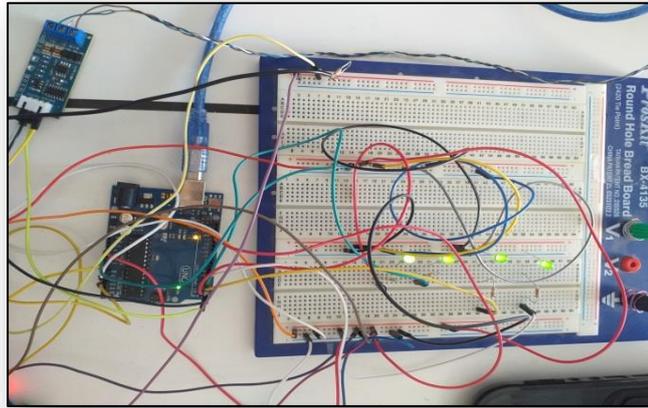


Figure 23:Réalisation pratique du troisième test

3.6 Quatrième test :

Dans ce test on a essayé d'ajouter un capteur de courant.

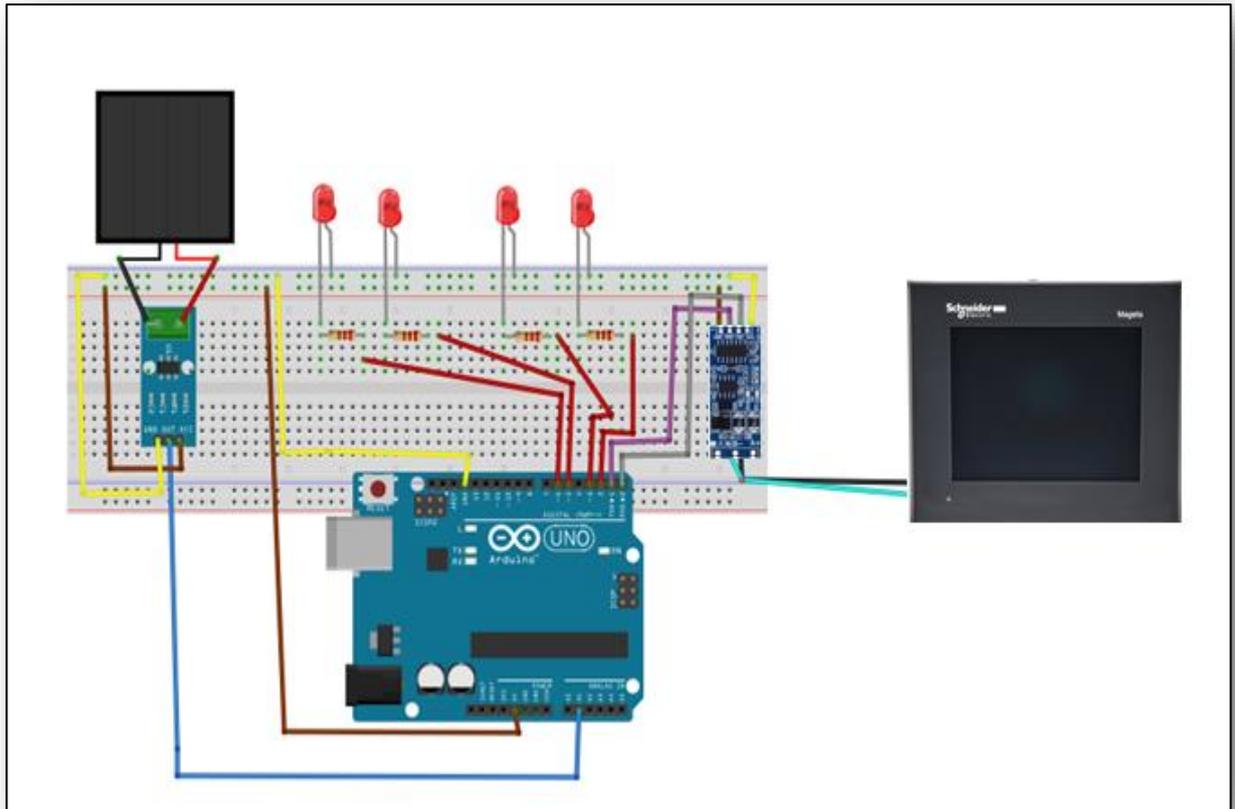


Figure 24:schéma de quatrième test

3.6.1 Capteur de courant ACS712

Le dispositif ACS712 Allegro offre un moyen économique et précis de détection de courants AC et DC. Il est basé sur le principe de l'effet Hall, qui a été découvert par le Dr Edwin Hall en 1879 selon ce principe, quand un conducteur de courant est placé dans un champ magnétique déposé, une tension est générée sur ses bords perpendiculaires à la direction à la fois du courant et du champ magnétique. **La Figure(25)**représente le capteur de courant ACS712 utilisé dans notre travail.[15]

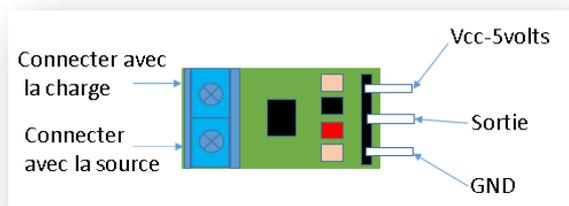
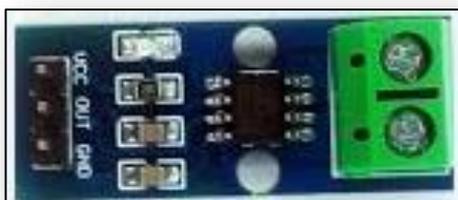


Figure 25:Capteur de courant ACS712

3.6.2 La partie programmation du quatrième test:

- On définit les variables utilisés pour le calcul du courant d'après le datasheet.

```
float Ain =0.0;  
float Aout =0.0;  
int a=1000;  
int b=185;
```

- Dans la fonction loop on fait les instructions du calcul.

```
void loop() {  
  
    float val= analogRead (A1);  
    float Ain = ((val*5)/1023)-2.5;  
    float c = (float)a / b;  
    Aout= Ain*(c);  
}
```

3.7 Cinquième test : Ce test consiste d'ajouter un capteur de tension.

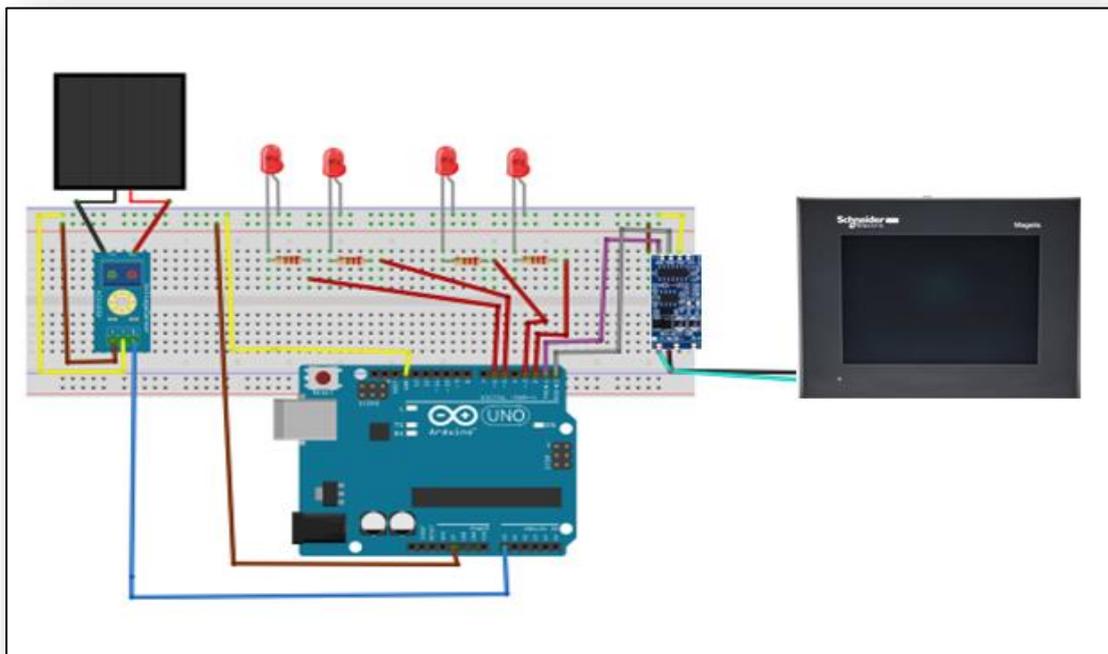


Figure 26:schéma de Cinquième test.

3.7.1 Capteur de tension :

La figure suivante représente un capteur de tension :

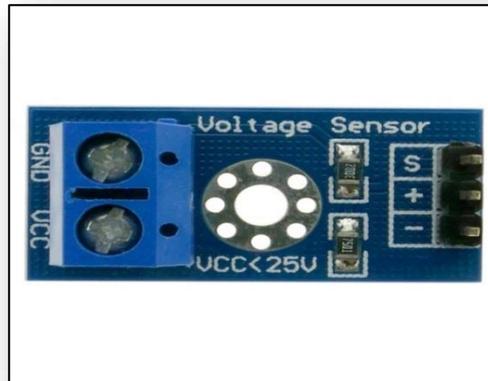


Figure 27:Capteur de tension

Ce module est basé sur le principe un diviseur de tension, peut réduire la tension de la borne d'entrée de tension module ne peut pas être supérieure à $5\text{ V} \times 5 = 25\text{ V}$ (3.3 V).

- La figure suivante représente la réalisation pratique du Cinquième test :

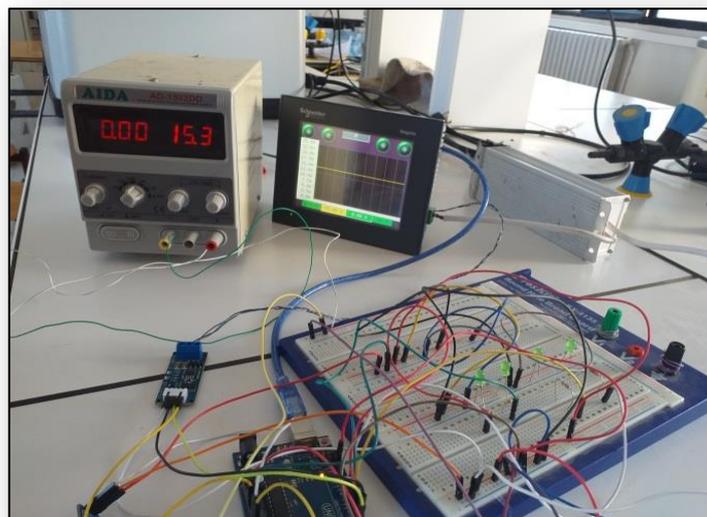


Figure 28:Réalisation pratique du Cinquième test

3.7.2 La partie programmation du Cinquième test:

- On définit les variables du capteur de tension utilisée d'après son datasheet.

```
float vout=0.0;
float vin=0.0;
float R1=37000.0;
float R2=7450.0;
```

- Ces instructions utilisées pour convertir la donnée analogique en une donnée numérique.

```
modbus_array[4]= analogRead(A0);
vout=(analogRead(A0)*5.0)/1023;
vin=vout/((R2)/(R1));
modbus_array[4]=vin;
```

3.8 Sixième test : Dans ce dernier test on a ajouté le panneau solaire et la capacité pour tracer les caractéristiques de ce panneau.

3.8.1 Panneau solaire Condor :

Le panneau utilisé est de type CEM 100M-36 représenté sur la figure suivante :



Figure 29:Panneau solaire utilisé

Puissance maximal (P_{max}) 100 W

Tension à point maximal (V_{pm}) 18.5 V

Courant à puissance maximale (I_{pm}) 5.41A

Tension de circuit ouvert (V_{co}) 22.3 V

Courant de court-circuit (I_{sc}) 5.84 A

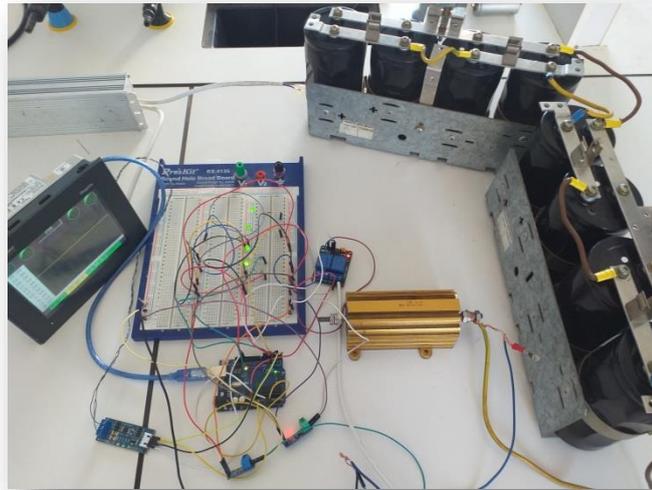


Figure 30:Réalisation pratique du sixième test



Figure 31:Traçage du courant et tension

- Les figures suivantes représente les caractéristique de panneaux (I (V),P) , on les tracer par l'EXCEL .

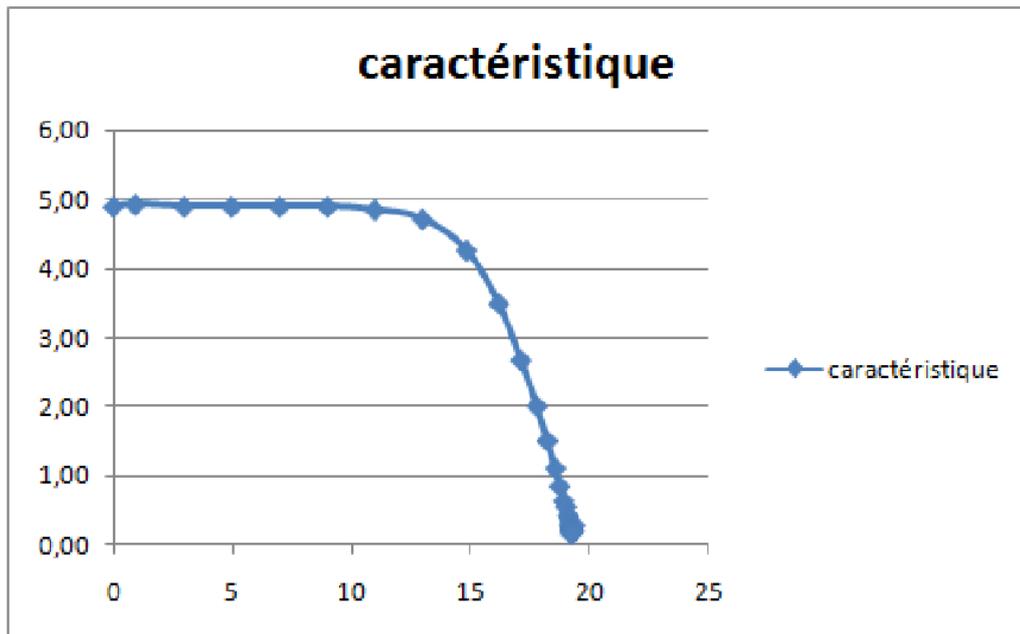


Figure 32:caractéristique de panneaux solaire

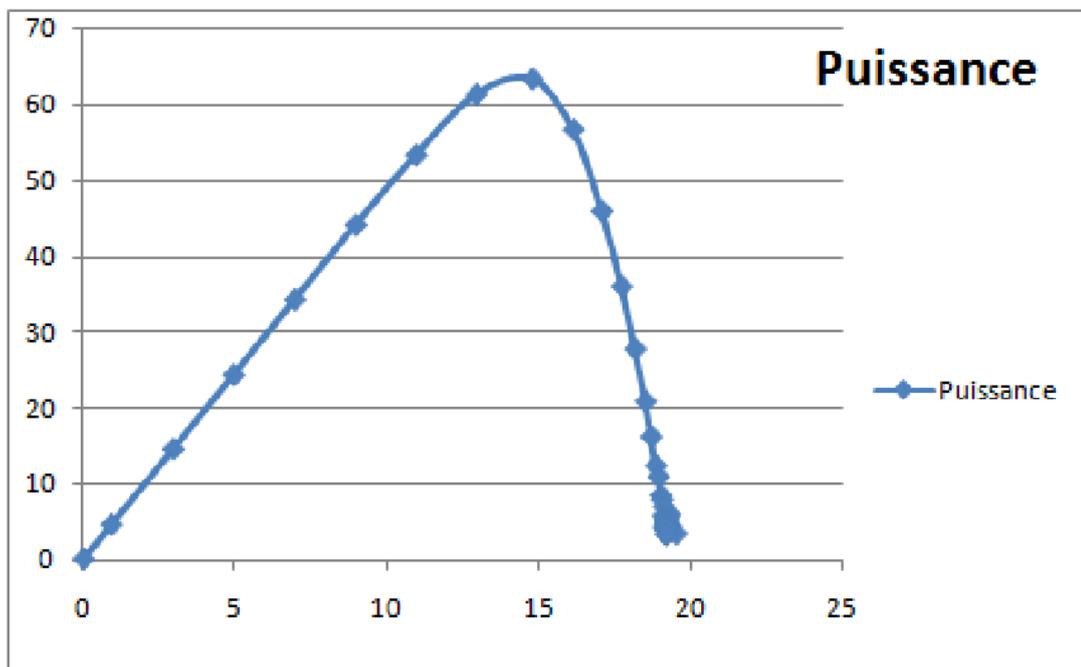


Figure 33:puissance de panneaux solaire

3.9 Conclusion :

Dans ce chapitre, on a réalisé plusieurs tests, le premier test a pour but de communiquer l'arduino et le PC et d'utiliser un logiciel libre pour visualiser les trames dans une communication MODBUS RTU, le test le plus important et la communication entre l'arduino et l'HMI pour tracer la caractéristique I-V, dont on a utilisé deux capteurs, un pour la mesure de courant I et un autre pour la tension V, une banc de capacités est utiliser pour Tracer I(V).

Conclusion Générale

Conclusion Générale :

Dans ce mémoire, Notre travail consiste à réaliser une communication entre la carte arduino et HMI schneider MAGELIS pour tracer les caractéristiques (le courant, la tension, la puissance) des panneaux photovoltaïques.

À cet égard, nous avons divisé notre projet en plusieurs parties:

En premier temps, nous avons fait une communication entre la carte arduino et le PC par utilisant d'un convertisseur USB / rs485, la carte Arduino Uno est utiliser comme esclave. Dans cette étape on a utilisé déférentes logiciel pour tester notre programme est déceler les erreurs susceptibles d'empêcher le bon fonctionnement de l'application embarquée.

Dans un deuxième temps, on fait la communication entre l'arduino et l'HMI, dans ce cas, arduino utilisée comme maitre. L'étape finale était la réalisation d'une l'interface Homme Machine qui permet de tracer des caractéristiques des PV en utilisant le logiciel vijeo designer.

Ce travail nous a permis de maitre en pratique les connaissances théoriques acquises toute le long de notre formation et d'enrichir nos connaissances sur les systèmes automatisés industriels, ainsi que d'améliorer nos compétences en programmation.

Références bibliographiques

- [1] P. Minist, L. Recherche, S. Universit, S. Facult, and D. Technologie, “Année 2013,” 1955.
 - [2] C. Mars, “Interfaces Homme- Machine 2,” 2017.
 - [3] M. X. B. T. Gt and X. B. T. Gk, “Magelis XBT GT, XBT GK, XBT GH,” 2011.
 - [4] “Définition simple d’IHM.” <https://www.proface.com/fr/node/49508> (accessed Jun. 24, 2022).
 - [5] Automatic-sense.com, “Guide du modbus pour les nuls,” pp. 1–11, 2016.
 - [6] G. Modicon and M. D. Controller, “2.1 Introduction,” pp. 19–26, 2005.
 - [7] E. Oued, “MEMOIRE DE FIN D ’ ETUDE Remerciements,” 2018.
 - [8] C. Des Syst, “Mémoire De Fin D ’ étude De Conception et réalisation d ’ un système Domotique par GSM,” 2017.
 - [9] M. Nourelhouda and R. Djamil, “Remerciements”.
 - [10] “Module convertisseur MAX485 RS 485 de TTL à RS485 5V.” <https://www.arduiner.com/fr/prodotto/module-convertisseur-max485-rs-485-de-ttl-rs485-5v/> (accessed Jun. 24, 2022).
 - [11] “communication.” <https://www.eltima.com/fr/article/rs485-communication-guide/> (accessed Jun. 24, 2022).
 - [12] R. Algerienne, D. Et, and U. A. M. Oulhadje-bouira, “Mémoire de fin d’étude,” 2019.
 - [13] A. Krama, “Faculté des Sciences Appliquées Domaine : Sciences et technologies Spécialité : Electrotechnique Industrielle Etude et réalisation d ’ une carte de contrôle par Arduino via le système Androïde Soutenu publiquement,” no. June 2015, 2018.
 - [14] “TTL à RS485 Adaptateur.” www.didactico.tn/produit/module-scm-ttl-vers-rs485-vers-uart-serie/
 - [15] M. Professionnel, “Conception et Réalisation d “ un Traceur de Caractéristiques Electriques des Panneaux PV à Base d “ Arduino,” 2019.
-