

Université de Gafsa
Institut Supérieur des Sciences Appliquées et de Technologie de GAFSA
Département de Automatiques des systèmes industriels (ASI)



**Mise en place d'un système de contrôle commande au sein
de l'atelier TSP.**

Présenté et soutenu par :

Tradi Oumaima

En vue de l'obtention de

Master professionnel en Systèmes Embraqués

Sous la Direction de :

Dr. Zammel Sami

Soutenu le 00/00/2021

Devant le jury composé de :

Président :

Rapporteur :

Membres :

2022/2023

DÉDICACE

C'est avec profonde gratitude et sincères mots, Que nous dédions ce modeste travail de fin d'étude à mes chers parents ; qui ont sacrifié leur vie pour notre réussite et nous avons éclairé le chemin par leurs conseils judicieux. Nous pourrons leurs rendre un peu de ce qu'ils ont fait pour moi, Que dieu leur prête tout le bonheur et la longue vie. Nous dédions aussi ce travail à mes frères, mes familles, mes amis, A tous nos professeurs qui nous avons enseignés et à tous ceux qui nous sommes chers.

REMERCIEMENTS

Tout d'abord nous tenons à remercier Dieu le tout puissant de nous avoir donné la santé et la volonté et de terminer ce projet. Nous adressons nos sincères remerciements en premier lieu à toute la famille de l'ISSAT. Nous remercions également le groupe chimique tunisien (GCT-GAFSA) de nous avoir donné l'occasion de faire ce travail et de nous avoir accueilli durant la période du stage. En outre, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de monsieur MOHAMED BOULAES et monsieur BELGACEM BARHOUMI, au sein le groupe chimique tunisien (GCT-GAFSA). Nous le remercions pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce projet, grâce à lui nous avons appris des nouvelles connaissances dans le domaine de régulation. Nous tenons également à exprimer nos vifs respects et nos vifs remerciements à notre encadreur monsieur ZAMMEL SAMI et monsieur SABER HADJ ABDALLAH pour le temps qu'elle nous a consacré et son soutien à la fois scientifique et moral ainsi que sa gentillesse si précieuse qui nous a motivé à aller plus loin et à nous investir dans ce projet. Aussi, nous adressons nos vifs remerciements à tout le membre de jury pour avoir bien voulu, examiner et juger ce travail. Enfin, que toute personne qui a participé de près ou de loin à l'accomplissement de ce mémoire soit sincèrement remerciée et les enseignants qui ont participé à notre formation soient sincèrement remerciés.

RÉSUMÉ

Ce projet de mémoire vise à remplacer le régulateur classique de Groupe chimique tunisien a l'usine de M'dhilla par un autre régulateur numérique basé sur une carte électronique. Le régulateur classique présente beaucoup de problèmes au niveau de traitement des données, ces problèmes et due au vieillissement des matériels existant qui est implanté depuis les années 80. Aussi, il existe un manque de pièces de rechange au niveau de marché international.

C'est pour cela, on a proposé ce projet pour renouveler la régulation au sein de l'usine, ainsi, de mettre à jour les régulateurs de l'industrie avec les nouvelles technologies existant dans le marché avec l'intégration des techniques de supervision.

Ce projet sert à implémenter un régulateur numérique, c'est un régulateur industriel à fabriquer basé sur une carte Arduino pour assurer la commande et le contrôle de la régulation du niveau du réservoir et de la température ambiante. La transmission et l'échange de données (affichage de mesure, commande, transmission de consigne, etc.) se feront par l'interface IHM du poste de contrôle équipé du logiciel LabVIEW, qui permet d'afficher et de suivre en temps réel l'avancement du procédé.

ABSTRACT

This dissertation project aims to replace the traditional Tunisian Chemical Group regulator at the M'dhilla factory with another digital regulator based on an electronic card. The classic regulator has many problems in terms of data processing, these problems are due to the aging of existing equipment which has been installed since the 80s. Also, there is a lack of spare parts on the international market.

That is why this project was proposed to renew the regulation within the factory, thus, to update the regulators of the industry with the new technologies existing in the market with the integration of the techniques of supervision.

This project is used to implement a digital regulator, it is an industrial regulator to be manufactured based on an Arduino board to ensure the command and control of the regulation of the level of the tank and the ambient temperature. The transmission and exchange of data (measurement display, command, setpoint transmission, etc...) will be done by the HMI interface of the control station equipped with the LabVIEW software, which makes it possible to display and monitor in real time the progress of the process.

TABLE DES MATIÈRES

| | |
|--|-----|
| TABLE DES MATIÈRES | vii |
| LISTE DES FIGURES | x |
| INTRODUCTION GENERALE | 1 |
| 1 PRÉSENTATION DE L'USINE ET L'ÉTUDE DE RÉGULATEUR FoxBORO 762CNA | 3 |
| 1.1 INTODUCTION | 4 |
| 1.2 PRÉSENTATION DU PROJET | 4 |
| 1.3 PRÉSENTATION DE L'ORGANISME D'ACCUEIL | 4 |
| 1.3.1 Groupe Chimique Tunisien (GCT) | 4 |
| 1.3.2 L'usine de MDHILLA | 5 |
| 1.3.3 Les unités de production dans l'usine | 7 |
| 1.4 L'ÉTUDE DE RÉGULATEUR FOX BORO 762CNA | 9 |
| 1.4.1 Régulation | 9 |
| 1.4.2 Fonctionnement interne de régulateur | 10 |
| 1.4.3 Les boucles de régulation et de contrôle | 10 |
| 1.4.4 Régulateur modèle FOX BORO type 762CNA | 11 |
| 1.5 SOLUTION PROPOSÉE | 14 |
| 1.6 DÉFINITION DES BESOINS | 15 |
| 1.7 CONCLUSION | 16 |
| 2 DESCRIPTION LE DISPOSITIF PROGRAMMABLE ARDUINO | 17 |
| 2.1 INTRODUCTION | 18 |
| 2.2 ETUDE DE L'EXISTANT | 18 |
| 2.2.1 Analyse de l'existant | 18 |
| 2.2.2 Critique de l'existant | 18 |
| 2.3 DEFINITION DU MODULE ARDUINO | 19 |
| 2.4 CHOIX DE LA CARTE ARDUINO | 19 |
| 2.5 POURQUOI ARDUINO UNO | 20 |
| 2.6 CONSTITUTION DE LA CARTE ARDUINO UNO | 21 |

| | | |
|----------|--|-----------|
| 2.6.1 | Microcontrôleur | 22 |
| 2.6.2 | Entrées/Sorties numériques | 22 |
| 2.6.3 | Entrées analogiques | 23 |
| 2.7 | LANGAGE DE PROGRAMMATION | 23 |
| 2.7.1 | Structure générale d'un programme en langage Arduino | 23 |
| 2.7.2 | Les étapes de téléchargement du programme | 25 |
| 2.7.3 | Les fonctions IDE de l'environnement Arduino | 25 |
| 2.8 | FONCTIONNEMENT ET UTILISATION | 26 |
| 2.9 | CONCLUSION | 27 |
| 3 | COMMANDE, CONTROLE ET REGULATION DANS LABVIEW | 28 |
| 3.1 | INTODUCTION | 29 |
| 3.2 | LA PROGRAMMATION GRAPHIQUE | 29 |
| 3.3 | DOMAINE D'APPLICATION | 29 |
| 3.4 | ENVIRONNEMENT LABVIEW | 30 |
| 3.4.1 | FACE-AVANT / Diagramme | 30 |
| 3.4.2 | Palette d'outils | 32 |
| 3.4.3 | Palette de commandes | 32 |
| 3.4.4 | Palette de fonctions | 33 |
| 3.5 | DESCRIPTION DES SOUS MENUS DE COMMANDE ET DE FONCTIONS | 33 |
| 3.5.1 | Sous menus de commande | 33 |
| 3.5.2 | Sous menus de fonctions | 34 |
| 3.6 | STRUCTURE DANS LABVIEW | 35 |
| 3.6.1 | Structure de données | 35 |
| 3.6.2 | Structure de programme | 36 |
| 3.7 | COMMUNICATION ARDUINO ET LABVIEW (PILOT E/S INSTRUMENTS) | 37 |
| 3.7.1 | Interface et Bibliothèque VISA | 38 |
| 3.7.2 | Interface LabVIEW Arduino-LIFA | 39 |
| 3.7.3 | Interface Arduino-Maker hub (LINX) | 41 |
| 3.8 | CONCLUSION | 43 |
| 4 | ETUDE ET RÉALISATION PRATIQUE DU RÉGULATEUR | 44 |
| 4.1 | INTRODUCTION | 45 |
| 4.2 | PRESENTATION LE SYNOPTIQUE DE PROJET | 45 |
| 4.2.1 | Carte Arduino UNO | 45 |
| 4.2.2 | Les capteurs | 46 |
| 4.2.3 | Les Actionneurs | 49 |

| | | |
|-------|---|----|
| 4.2.4 | Le schéma général du module électronique | 51 |
| 4.3 | DEVELOPPEMENT DU PROGRAMME ET CONCEPTION HMI SOUS LABVIEW | 51 |
| 4.3.1 | Teste, réglage des paramètres | 53 |
| 4.3.2 | Programme et résultats de l'application de régulation de tempé- rature | 54 |
| 4.4 | CONCLUSION | 55 |
| | CONCLUSION GÉNÉRALE | 56 |
| | BIBLIOGRAPHIE | 57 |

LISTE DES FIGURES

| | | |
|------|--|----|
| 1.1 | l'architecture global du projet. | 4 |
| 1.2 | Structure et l'organigramme de l'usine Mdhilla. | 6 |
| 1.3 | Partenaires commerciaux. | 6 |
| 1.4 | Déroulement de la production. | 7 |
| 1.5 | Diagramme de description de l'unité de l'acide phosphorique. | 8 |
| 1.6 | structure interne du régulateur. | 9 |
| 1.7 | Affichage du contrôleur. | 12 |
| 1.8 | Schéma ou graphique caractérisant la solution proposée. | 15 |
| 2.1 | Description de la carte Arduino UNO | 21 |
| 2.2 | Structure d'un programme Arduino. | 24 |
| 2.3 | Les étapes de téléchargement du code. | 25 |
| 3.1 | Schéma synoptique de domaine d'application du LabVIEW. | 30 |
| 3.2 | Fenêtre de l'environnement de développement sur LabVIEW face-avant (à gauche) et Diagramme (à droite). | 31 |
| 3.3 | Palette d'outils. | 32 |
| 3.4 | Palette de commandes. | 32 |
| 3.5 | Palette de fonctions. | 33 |
| 3.6 | sous menu de commande. | 34 |
| 3.7 | Sous menu de fonction numérique. | 34 |
| 3.8 | Sous menu de fonction logique (booléenne). | 35 |
| 3.9 | Différents types de structures de données dans LabVIEW. | 36 |
| 3.10 | Bibliothèque boucles et structures données. | 37 |
| 3.11 | Interface LabVIEW-Arduino. | 38 |
| 3.12 | Ouverture et fermeture d'une VISA pilote d'instrument. | 38 |
| 3.13 | Exemple Pack VISA LabVIEW. | 39 |
| 3.14 | Interface LabVIEW Arduino. | 40 |
| 3.15 | Exemple VI LabVIEW Arduino. | 41 |
| 3.16 | Interface LabVIEW Arduino-Maker hub LINX. | 42 |
| 3.17 | Exemple sortie digital utilisent interface LabVIEW Arduino-LINX. | 42 |

| | | |
|------|---|----|
| 4.1 | Schéma synoptique. | 45 |
| 4.2 | Organigramme Programme Arduino. | 46 |
| 4.3 | Notion de capteur. | 47 |
| 4.4 | Capteur ultrason HC-SR04. | 47 |
| 4.5 | Câblage capteur ultrason HC-SR04. | 48 |
| 4.6 | Capteur DHT11. | 48 |
| 4.7 | Câblage Capteur DHT11. | 49 |
| 4.8 | Positionneur de vanne électropneumatique TROVIS 3793. | 49 |
| 4.9 | Câblage Positionneur de vanne électropneumatique TROVIS 3793 et capteur ultrason HC-RS04. | 50 |
| 4.10 | Schéma général du module électronique. | 51 |
| 4.11 | Programme de mesure et étalonnage ultrasonique HC-SR04. | 52 |
| 4.12 | Programme de pilotage et commande du positionneur. | 52 |
| 4.13 | Synoptique HMI de commande et régulation de niveau. | 53 |
| 4.14 | Synoptique HMI et résultats de régulation de la température. | 55 |
| 4.15 | Principe de la mesure par ultrason. | 58 |

INTRODUCTION GENERALE

Dans la plupart des processus de production industrielle, des critères de qualité des produits, de sécurité, de sûreté de fonctionnement et surtout de fiabilité des procédés, et la mise à disposition automatique de solutions pratiques pour répondre dans la mesure du possible à tous ces critères (conformité au cahier des charges) permettent d'évaluer les performances.).

La mise en œuvre de ces solutions consiste à exécuter des processus (pilotes). Le contrôle de processus comprend des activités telles que la planification, l'ordonnancement, la surveillance, la surveillance et le contrôle en temps réel. Le principe général du contrôle de processus automatique et manuel est connu sous le nom de contrôle en boucle ouverte ou contrôle en boucle fermée (contrôle de rétroaction).

La réglementation résume toutes les mesures matérielles et techniques utilisées pour maintenir les variables contrôlées à des valeurs égales à leurs points de consigne malgré les perturbations indésirables.

La régulation de la température, du niveau d'eau ou d'autres liquides est l'un des processus utilisés dans les applications industrielles, de laboratoire et quotidiennes, et les moyens techniques et efficaces de les contrôler correctement et de les réguler automatiquement doivent être mis en œuvre. Nous les traitons à l'aide de diverses techniques efficaces de traitement de l'information.

Ce mémoire traite une commande numérique, régulation manuel et automatique d'un réservoir d'eau et également un contrôle de la température en utilisant un calculateur ARDUINO et un PC doté par le logiciel de commande, contrôle et supervision le LABVIEW. Dans ce travail les objectifs ont été visés :

* Le premier chapitre aura regroupé la présentation de l'usine et l'étude de régulateur Foxboro 762cna.

* Le deuxième est de regrouper suffisamment d'informations sur une grande catégorie de cartes d'interfaçage Arduino : son langage de programmation, sa construction, son principe de fonctionnement.

* Nous abordons dans le troisième chapitre la programmation graphique à base du LABVIEW à plusieurs plateformes de l'Arduino et la communication via le port série où USB.

* Dans le dernier chapitre, nous intéressons essentiellement à la réalisation d'un régulateur à base Arduino et une interface de commande et contrôle par LABVIEW. Ainsi l'élaboration des interfaces graphiques, les graphes résultants et les interprétations des résultats.

PRÉSENTATION DE L'USINE ET
L'ÉTUDE DE RÉGULATEUR FoxBORO
762CNA

1

1.1 INTODUCTION

Dans ce chapitre, nous présenterons dans un premier temps les entreprises où se déroulent les stages. Il examine ensuite le système de commande et de contrôle numérique (DCS) sous ses aspects matériels et logiciels, et présente enfin la recherche et la critique des systèmes existants, suivies de suggestions de solutions adaptées qui répondent aux spécifications proposées par l'entreprise.

1.2 PRÉSENTATION DU PROJET

Ce projet fin d'étude a pour but de recherche vise à contourner le régulateur Foxboro 762cna. Ce contrôleur industriel est basé sur une carte d'acquisition Arduino pour contrôler et contrôler la régulation du réservoir et de la température. La transmission et l'échange de données (affichage des valeurs mesurées, transmission des commandes, des consignes , etc.) s'effectuent via l'interface IHM de la station de contrôle avec le logiciel LabVIEW permettant une visualisation en temps réel du déroulement du processus et surveillance.

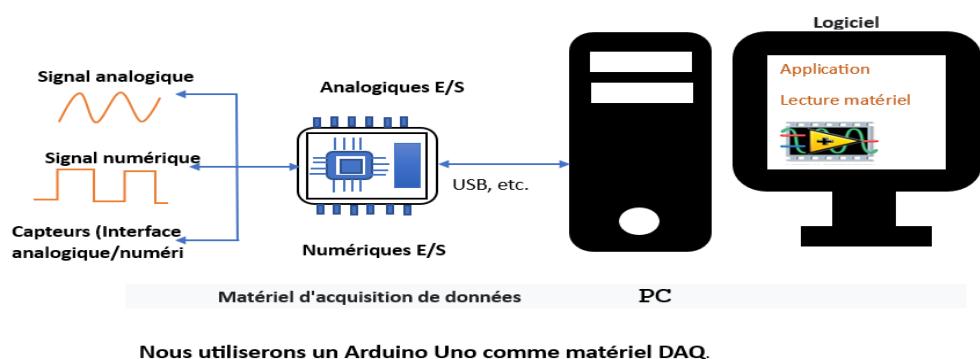


FIGURE 1.1 – l'architecture global du projet.

1.3 PRÉSENTATION DE L'ORGANISME D'ACCUEIL

Dans cette section, nous commençons par la présentations du GCT, l'usine de Mdhilla .

1.3.1 Groupe Chimique Tunisien (GCT)

Le phosphate est un minéral découvert depuis 1885 par le géologue français Philippe Thomas dans la région "METLAOUI" en Tunisie. Après de nombreuses années d'expérience dans l'exportation de mineraux de phosphate, la Tunisie s'est

tournée vers la conversion et la valorisation de ce minéral en implantant une industrie locale de production d'acide phosphorique et d'engrais minéraux au sein du groupe chimique tunisien. Le groupe chimique tunisien entend convertir le phosphate commercial en acide phosphorique (P_2O_5) et en engrais au sein de ses plateformes industrielles de Gabès, Sfax, Skira et Gafsa. Le GCT a encore renforcé sa position en tant que l'un des principaux producteurs d'engrais au monde. Au cours de la période de 1999 à 2011, les usines de GCT ont fonctionné à pleine capacité, ce qui a entraîné des niveaux de production très élevés et, par conséquent, une participation aux bénéfices très élevée.

1.3.2 L'usine de MDHILLA

Présentation

L'usine de MDHILLA est une unité industrielle fondée le 20 septembre 1981 pour produire du triple super phosphate (TSP) à base de phosphate et d'acide phosphorique. L'usine produit 460 000 tonnes de triple super phosphate granulaire par an. L'usine MDHILLA fonctionne 24 heures sur 24 et assure sa fonction principale de conversion du phosphate en triple super phosphate (TSP). La production de TSP nécessite un intermédiaire essentiel, à savoir le "Acide Sulfurique". L'usine fait partie de la division régionale GAFSA située dans le sud ouest du pays. L'usine est située à proximité d'une mine de phosphate et est reliée par un convoyeur de 4 km de long.

Structure et l'organigramme de l'usine

L'usine est composée de :

- Une unité de production d'acide sulfurique.
- Une unité de production d'acide phosphorique.
- Une unité de production du TSP.
- Une unité d'utilité qui contient :

*Deux chaudières.

*Une centrale thermique équipée d'un turbo-alternateur.

*4 compresseurs (de commande).

*Des installations de production d'eau.

Quatre ateliers de maintenances : mécanique, électrique, régulation, et engins :

- Un service de sécurité des biens et des personnes.
- Une salle de contrôle pour chaque service.
- Un atelier de stockage.

- Un bloc administratif.
- Un laboratoire chimique.

Comme les autres sociétés l'usine de MDHILLA s'organise selon une hiérarchie comme le montre la figure suivante :

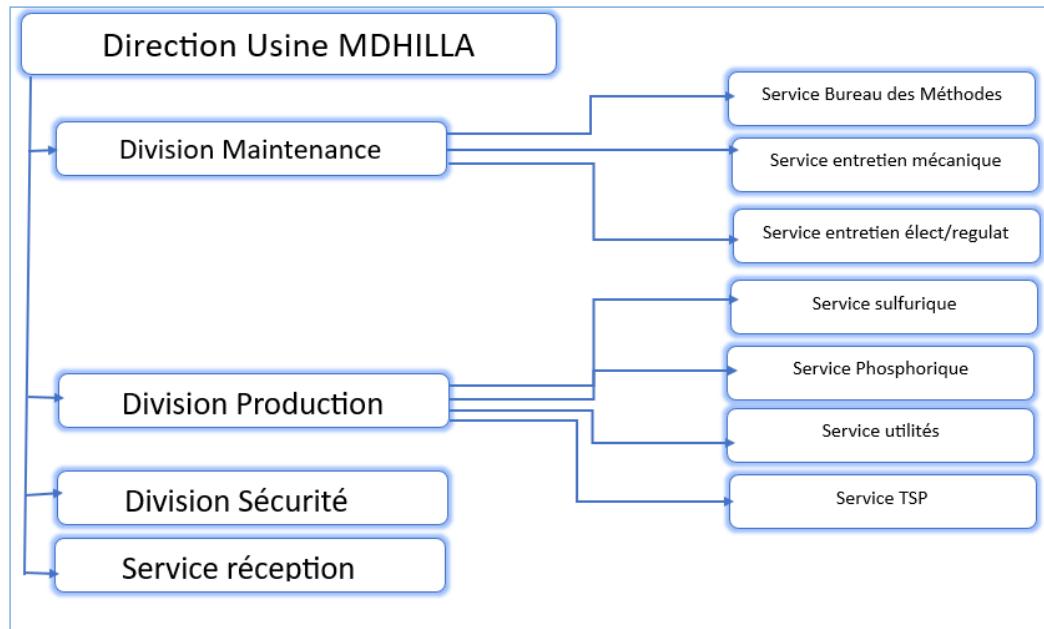


FIGURE 1.2 – Structure et l'organigramme de l'usine Mdhilla.

L'usine de MDHILLA a des relations commerciales avec des divers pays soit comme étant des clients partenaires tel que la Turquie, l'Egypte, l'Inde, la Chine et l'Europe de l'ouest (Voir la figure 1.3).



FIGURE 1.3 – Partenaires commerciaux.

La Déroulement de la production

Le rôle principal de l'usine est la production de TSP (triple superphosphate), qui se déroule en trois étapes (voir Figure 1.4). La première étape consiste à convertir le soufre brut en acide sulfurique. Dans la deuxième étape, le phosphate humide est attaqué avec de l'acide sulfurique pour le convertir en acide phosphorique. La troisième étape consiste à convertir l'acide phosphorique en TSP (Triple Super Phosphate) en utilisant du phosphate sec.

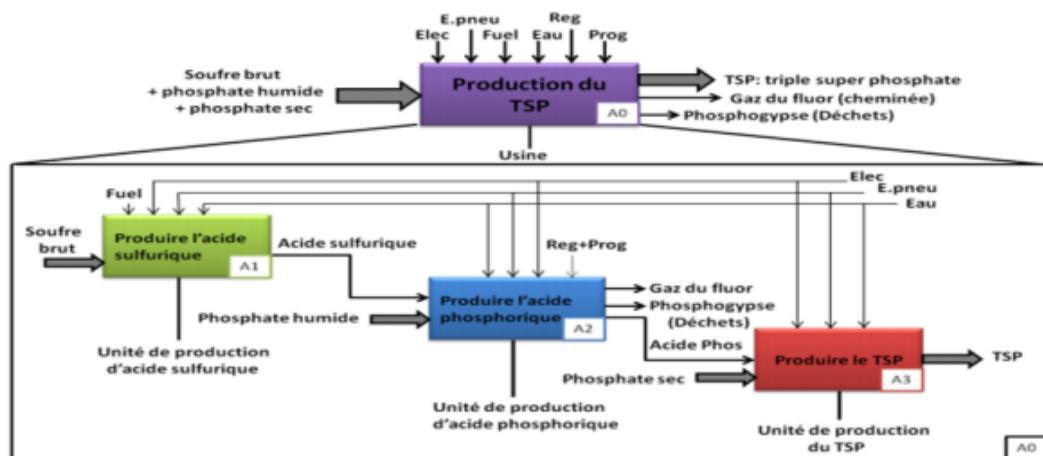


FIGURE 1.4 – Déroulement de la production.

1.3.3 Les unités de production dans l'usine

Les unités de production présentent par :

Présentation

L'usine Mdhilla comporte trois unités de production qui sont :

- * Unité de production d'acide sulfurique d'une capacité de 1500 tonnes de H_2SO_4 par jour.
- * Une unité de production d'acide phosphorique d'une capacité de 500 tonnes P_2O_5 par jour.
- * Deux lignes de production de triple super phosphate (TSP) d'une capacité de 700 tonnes par jour chacune. L'alimentation de ces unités en énergie électrique, eau et air comprimé s'effectue par l'unité de gestion des utilités.

• Unité sulfurique :

L'unité est destinée à produire 1500 tonnes d'acide sulfurique par jour concentré de 98,5 en H_2SO_4 en utilisant comme matière première le soufre solide. Le processus de fabrication de l'acide sulfurique.

- Unité de production de T.S.P :

Cette unité consiste à attaquer le phosphate sec avec l'acide phosphorique pour obtenir le TSP.

- Centrale utilités :

Les utilités assurent l'alimentation en eau, air comprimé, vapeur et électricité de l'usine. Elles comprennent :

Une centrale électrique avec un turboalternateur de 15.2 MW alimenté par de la vapeur HP surchauffée à 40 bars et 400C.

- * Une chaudière auxiliaire.
- * Une installation de préparation d'eau déminéralisée.
- * Un aérocondenseur de vapeur.
- * L'alimentation en eau de l'usine.
- * La d'air production d'air comprimé.
- * Une station d'air comprimé (compresseur).
- * Un groupe de condenseurs et réfrigérants.
- * Un tour de refroidissement d'eau recyclée.
- * Unité de production d'acide phosphorique :

Cette unité consiste en la conversion du phosphate humide en acide phosphorique par attaque à l'acide sulfurique, comme le montre la figure 1.5.

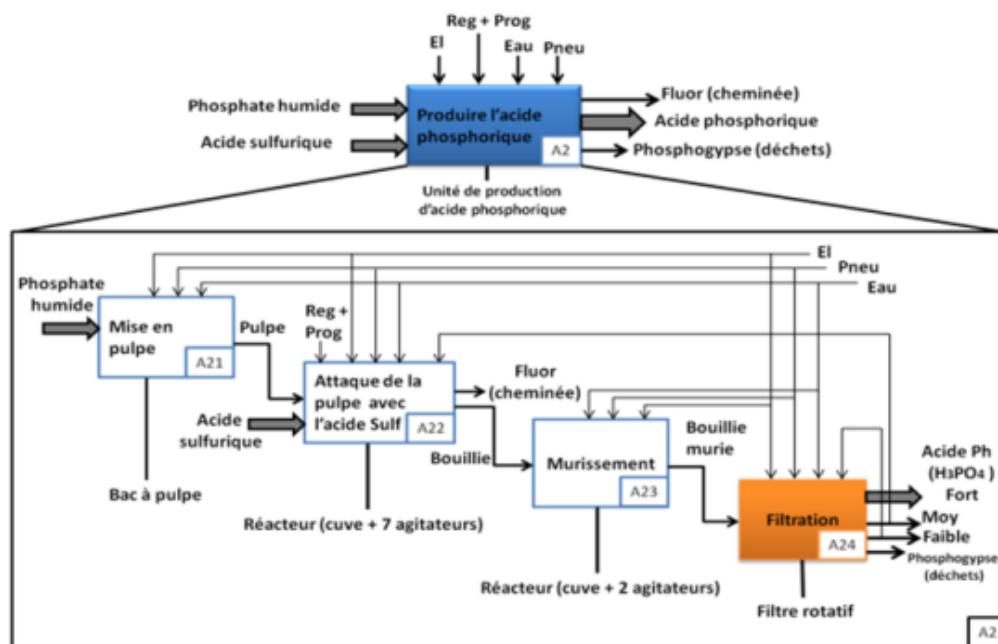


FIGURE 1.5 – Diagramme de description de l'unité de l'acide phosphorique.

Phosphate humide + acide sulfurique = acide phosphorique Le phosphate

est d'abord mélangé à de l'eau pour former une bouillie, puis introduit dans un réacteur équipé d'un agitateur central et de six périphériques et attaqué à l'acide sulfurique. Du gaz fluor s'échappe par la cheminée et une bouillie est obtenue. Cette bouillie est envoyée dans un deuxième réacteur équipé de deux agitateurs pour une bonne maturation. Enfin, ce furoncle est filtré à travers un filtre rotatif sous vide pour obtenir de l'acide phosphorique à 3 degrés (H_3PO_4).

L'acide concentré (acide fort) doit être stocké, l'acide moyen doit être renvoyé dans le réacteur primaire et l'acide faible doit rester sur le filtre pour laver la suspension.

1.4 L'ÉTUDE DE RÉGULATEUR FOXBORO 762CNA

Dans cette section, nous avons étudié le régulateur FoxBoro 762cna.

1.4.1 Régulation

Le mode de fonctionnement traditionnel est basé sur des contrôleurs pneumatiques et pose plusieurs problèmes qui conduisent à la nécessité d'utiliser de nouveaux systèmes de commande et de contrôle basés sur des applications analogiques-numériques. Le contrôleur est le cerveau de la boucle de régulation. Envoyez et recevez des signaux analogiques continus.

Les plus courants sont :

- Pneumatique «Pm» de 200-1000mBar.
- Electrique «Im» de 4-20mA.
- La mesure «M» venant d'un transmetteur.
- La cosigne «C» qui peut être interne ou externe.

La structure interne du régulateur peut être schématisée de la façon suivante :

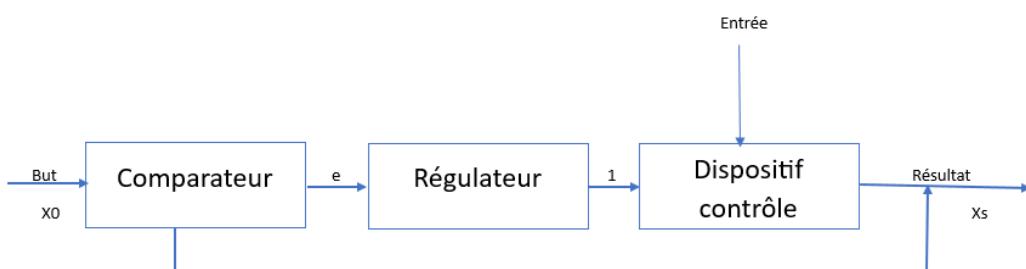


FIGURE 1.6 – structure interne du régulateur.

1.4.2 Fonctionnement interne de régulateur

Le signal de sortie (ou signal de commande) est fonction du signal d'entrée (différence entre la mesure et la consigne) $S=f(M-C)$. Les contrôleurs peuvent accepter des entrées analogiques ou numériques. Le principe de fonctionnement interne est entièrement numérique et tous les paramètres enregistrés sont variables même en cas de panne de courant. Un multiplexeur est connecté à l'entrée analogique. La tension d'entrée est appliquée à une capacité de multiplexeur connectée entre la masse et un circuit tampon. Une résistance d'entrée de 250 ohms convertit chaque entrée en une tension équivalente. Une entrée libre de potentiel interrompe la connexion reliant le pôle négatif à la terre.

1.4.3 Les boucles de régulation et de contrôle

o Régulation et contrôle de la température :

Il est composé de boucle de régulation mixte Dont : TC : régulateur de température de signal 4-20mA. Consigne température de vapeur 405C. la mesure de température est assurée par un thermocouple de type K (Nickel-chrome Nickel-allié) du régulateur et d'une vanne surchauffeur et le contrôle est assuré par un thermostat.

* Régulation et contrôle du niveau :

Le niveau d'eau dans le ballon chaudière a une grande importance donc nous avons soumis à des contraintes très sévères.

** Régulation du niveau :

Dont : LC : régulateur électronique numérique du niveau ballon chaudière.

FT : transmetteur avec un débitmètre électromagnétique.

LT : transmetteur différentiel intelligent.

FCV : vanne de contrôle de débit. La boucle de régulation de niveau comporte des régulateurs montés en cascade. La sortie du régulateur de niveau est une consigne pour le régulateur du débit.

** Contrôle de niveau :

Pour assurer un fonctionnement de la chaudière en toute sécurité, il est nécessaire d'implémenter un système de contrôle de sécurité. Par un contact à mercure qui peut déclencher ou arrêter une alarme.

LAHH : niveau très haut =>arrêt de la chaudière.

LAH : niveau haut =>enclenchement d'une alarme.

LAL : niveau bas =>enclenchement d'une alarme.

LALL : niveau très bas =>arrêt de la chaudière.

o Régulation de la température du combustible :

Dont :

TC : régulateur électronique température Consigne : température du combustible 105C.

L'utilisation du combustible dans ce cycle nécessite le chauffage à une température fixée comme consigne (105C).

* Différents régulateurs utilisés :

Nous avons recours à plusieurs régulateurs mais le plus utilisé et le plus performant est le modèle FOXBORO type 762CNA.

1.4.4 Régulateur modèle FOX BORO type 762CNA

o Description :

Le microcontrôleur 762CNA comprend un microprocesseur capable d'effectuer un contrôle proportionnel, intégral, dérivé (PID) pour deux boucles de contrôle indépendantes. Deux boucles peuvent également être configurées pour former une cascade à station unique ou un contrôleur à sélection automatique. De plus, le 762CNA offre de nombreuses fonctions de contrôle avancées telles que le réglage EXACT, l'addition, le calcul et les fonctions logiques complètes.

Le model FOXBORO 762CNA assure plusieurs types de régulations :

PID : l'algorithme peut être configuré en P, I, PI, PD ou en PID qui est le plus utilisé.

EXACT : avec ce type le temps de réponse du régulateur est réduit.

CASCADE : avec cette configuration la sortie du premier régulateur est la consigne du deuxième.

AUTO SELECTOR : deux régulateurs peuvent être combinés pour fournir une seule sortie qui peut être utilisée en mode double contrôle.

SPLIT RANGE : utilisé dans la régulation de température.

REMOTE OR LOCAL SETPOINT : la consigne peut être ajustée manuellement.

PANEL OR WORKSTATION : la surveillance du panel or work station peut être locale ou distante.

o La connexion à la source d'alimentation :

Le contrôle de 762CNA peut être commandé avec une tension de fonctionnement de 24,100,120,220,240 V cc, en vérifiant que notre puissance est la même comme cela est indiqué sur l'étiquette de données.

a) Affichage du contrôle :

Sauf que nous avons commandé une autre configuration, le contrôleur affichera quelque chose semblable à ce que montre la figure suivante :



FIGURE 1.7 – Affichage du contrôleur.

Avec :

- Affichage numérique supérieure : En mode NORMAL, l'affichage numérique supérieure monte TAG boucle ou valeur à l'échelle de la variable avec étiquette des unités d'ingénierie. Dans les modes de lecture et réglage, elle montre une catégorie de paramètre ou un message.
- Affichage numérique inférieure : En mode NORMAL, l'affichage numérique inférieur montre la valeur actuelle de la variable identifiée par un indicateur par Bargraph. Quand une alarme existe, elle affiche ID de variable. En lecture et SET, l'affichage montre un paramètre ou détail du message.
- Bar graph indicateur : Le Bargraph identifié peut être variable et sera affiché sur l'affichage inférieur numérique. Il y a aussi des postes «sans indicateur ».

- Gauche Bargraph : Il affiche la valeur actuelle de la variable n1 (en général le point de consigne).
 - Centre Bargraph : Il affiche la valeur actuelle de la variable n2 (généralement mesure).
 - Bargraph droit : Il affiche la valeur actuelle de la variable n3 (contrôle général de sortie).
 - Indicateur LED rouge de défaut : Lorsque On, indique une erreur matérielle en une basse tension en courant alternatif ou courant primaire, un signal d'alarme sera détecté sur l'afficheur.
 - W/P : Est un indicateur d'état de la station de travail (W) ou le panneau (P) de contrôle. W clignote si la communication échoue en mode W et recalé est mis à W. Ni W ou P sont allumés lorsque W/P est configuré OFF.
 - R/L : Est un indicateur d'état de la section locale (L) ou à distance (R) de consigne et pour Ratio (R) ou local (L) point de consigne. Ni R ou L sont allumés lorsque le point de consigne TYPE est configuré comme LOCAL.
 - A/M : Est un indicateur d'état de commande (A) pour l'Automatique et (M) pour Manuel.
 - Indicateur d'alarme : Il clignote lorsqu'il est actif ou stable lorsqu'il revient à la normale après avoir reconnu par l'indicateur d'alarme.
- Clavier
 - o Les Avantages et les Inconvénients :
 - **Avantages :
 - Interface graphique simple à utiliser.
 - Possibilité de branchement en cascade.
 - Détection de dépassements de niveau.
 - Indication de défaut (Alarme).
 - **Inconvénients :
 - Cout élevé.
 - Difficile à câbler.
 - Non compatible avec les équipements numériques.
 - Insertion de consigne et paramètres manuellement.
 - Entretien difficile.
 - Manque de pièces de rechange.
- Problématique :
- L'existant dans ce projet est le contrôle de pression et débit d'une ligne de production TSP dans l'usine du groupe chimique tunisien, le contrôle est assuré par des régulateurs FOXBORO762CNA. Ce type des régulateurs classiques fonctionne en boucle fermé qui est basé sur 3 grandeurs :

Une valeur mesurée, une consigne, une valeur nominale.

L'opérateur donne une consigne pour le régulateur, ce dernier fait des mesures à l'aide des capteurs, la valeur mesurée sera comparée avec la valeur nominale, s'il y a une différence de mesure le régulateur fait des régulation nécessaires (arrêt de moteur, ouverture d'une vanne, ...) pour obtenir la valeur nominale demandé par l'opérateur.

Le problème est que due aux nouvelles technologies, on a un manque en pièces de rechange en coté de régulateurs, ce problème influent à la continuité de production dans l'usine, ce qui entraîne une perte de temps et d'argent dans l'usine. C'est pour cela, on a proposé une solution qui est basé sur la migration du mode analogique vers le mode numérique. Cette solution est basée sur le changement des régulateurs existant par des cartes ARDUINO; la mise en place d'une application de supervision basé sur le LABVIEW, pour assurer la commande et la contrôle à distance de la ligne de production TSP.

1.5 SOLUTION PROPOSÉE

Pour pouvoir minimiser ces inconvénients, on se propose d'élaborer et mettre en place une application automatisé qui nous permet de :

- * Remplacer ce régulateur par une carte Arduino.
- * Régler le niveau, la température et la pression.
- * Supervision sur LabVIEW.

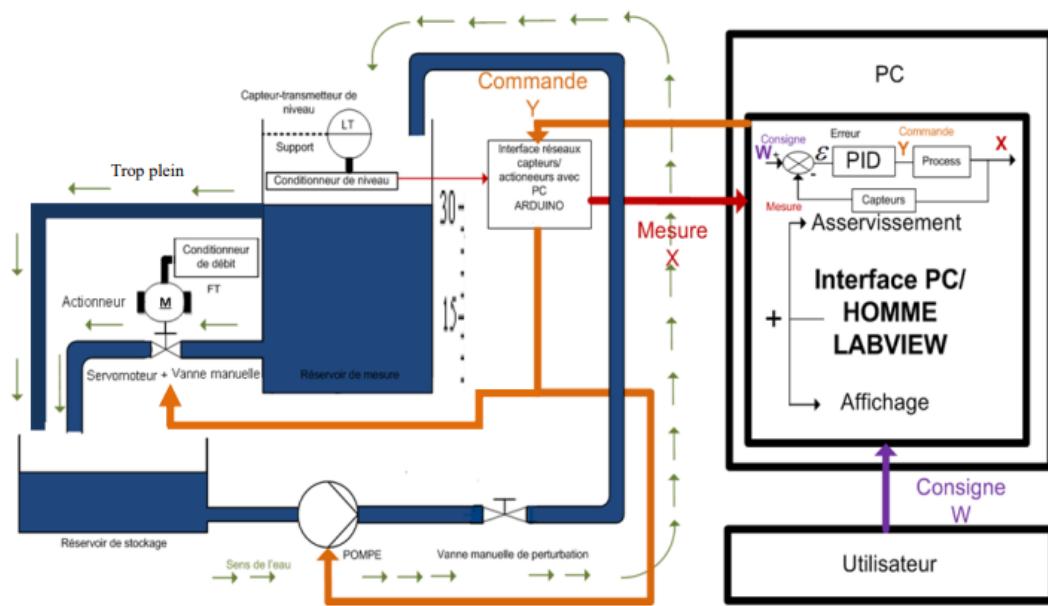


FIGURE 1.8 – Schéma ou graphique caractérisant la solution proposée.

La régulation est effectué par un programme développé dans le logiciel LABVIEW, qu'on le présenterons plus tard, mais le contrôle se fait avec la carte Arduino UNO. Nous expliquent brièvement la composition de notre maquette :

.Un réservoir qui permettent l'approvisionnement de l'évolution de niveau de TK.

.Le capteur à ultrasons (numéro de référence HC-SR04) est un émetteur-récepteur qui utilise des ondes ultrasonores pour mesurer la distance des objets et lire le niveau réel dans le réservoir de TK.

.En tant que module d'automatisation électronique basé sur la carte Arduino Uno, ce module permet principalement de traiter les signaux des capteurs, de contrôler les actionneurs et de communiquer avec les stations de surveillance et de contrôle HMI via Labview.

.La station de contrôle PC comprend une interface graphique Labview pour la programmation de la carte Arduino, la visualisation des mesures et l'enregistrement des données reçues du module via le port série USB.

1.6 DÉFINITION DES BESOINS

- Minimiser les pertes de temps en maintenance.
- Elaborer un système de supervision avancée.
- Migration du mode analogique vers mode numérique.

- Changement des régulateurs classiques par des cartes Arduino.
- Gain du temps d'entertain.
- Gestion d'historique.
- Faible cout.
- Installation facile (simple).
- Commuincation a distance supervisé.

1.7 CONCLUSION

Dans ce chapitre, nous avons mis en évidence les configurations, les descriptions de divers systèmes de contrôle industriels, et les contrôles les plus utilisés dans le monde industriel préfèrent stabiliser efficacement les systèmes d'automatisation.

DESCRIPTION LE DISPOSITIF PROGRAMMABLE ARDUINO

2

2.1 INTRODUCTION

Aujourd’hui, l’électronique est de plus en plus remplacée par l’électronique programmée. Aussi appelés systèmes embarqués ou informatique embarquée. Son but est de simplifier les schémas électroniques, réduisant ainsi l’utilisation de composants électroniques et réduisant ainsi le coût de fabrication des produits. Cela permet des systèmes plus complexes et efficaces dans moins d’espace. Depuis que l’électronique existe, sa croissance a été rapide et continue à ce jour. L’électronique est désormais accessible à tous ceux qui : Ce que nous apprenons dans ce travail est une combinaison d’électronique et de programmation. En fait, on parle d’électronique embarquée. C’est un sous-domaine de l’électronique qui vous permet de combiner la puissance de la programmation avec la puissance de l’électronique.

2.2 ETUDE DE L’EXISTANT

L’étude de l’existant consiste à mettre à plat, de façon aussi claire que possible, le critique et de donnée des solutions pour résoudre le problème de remplacement du régulateur par la carte Arduino et supervision en LabVIEW.

2.2.1 Analyse de l’existant

Dans cette partie nous avons défini l’analyse du mode de fonctionnement du régulateur actuel. Le mode de fonctionnement existant menant est l’usage du régulateur par l’intermédiaire d’une interface qui contient plusieurs boutons et par l’intervention d’un administrateur. Chaque bouton est spécifique à une tache bien déterminé.

2.2.2 Critique de l’existant

Dans cette partie nous avons développé les problèmes qui peuvent influencer sur le déroulement du travail du régulateur :

- Le mal implémentation du matériel.
- La température.
- La pression.
- Lourdeur du temps d’exécution des instructions et du travail.
- Le travail manuellement.

La sécurité des personnels et des informations Partant du problème résidant actuellement chez Groupe Chimique de Gafsa, l’idée de notre projet consiste à

remplacer ce régulateur par une carte Arduino avec une supervision sur LabVIEW.

2.3 DEFINITION DU MODULE ARDUINO

Les modules Arduino sont des circuits imprimés matériels libres (plates-formes de contrôle). Les plans de la carte elle-même sont publiés sous licence libre avec certains composants carte :

Microcontrôleurs et composants complémentaires non inclus License gratuite. Un microcontrôleur programmé peut analyser et générer des signaux pour accomplir diverses tâches. Arduino est utilisé dans de nombreux domaines Applications telles que l'électronique industrielle et automobile. La modélisation, Non seulement dans la domotique, mais aussi dans d'autres domaines comme l'art contemporain et les pilotes Contrôlez des robots, contrôlez des moteurs, créez des jeux de lumière, communiquez avec les humains Ordinateurs avec et sans interfaces IHM, contrôle d'appareils mobiles (modélisation).

Chaque module Arduino dispose d'un régulateur de tension +5V et d'un oscillateur quartet 16 MHz (ou résonateur céramique sur certains modèles). Pour programmer cette carte, Utilisez le logiciel Arduino IDE.

2.4 CHOIX DE LA CARTE ARDUINO

Il existe actuellement plus de 20 versions de modules Arduino. Je vais en citer quelques-uns dans l'ordre.

Pour clarifier l'évaluation de ce produit scientifique et académique :

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le LilyPad Arduino, une conception de minimalist pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.

- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- L'Arduino Duemillanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega32U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- L'Arduino Esplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

Parmi eux, j'ai choisi la carte Arduino Uno. Principale préoccupation de cette carte est destinée à faciliter la mise en œuvre de telles commandes. Plus à ce sujet plus tard. Arduino fournit un environnement de développement basé sur des outils open source. Comme interface de programmation. Injection de programmes convertis l'environnement est sous forme de code "HEX" dans la mémoire du microcontrôleur facile avec la connexion USB. De plus, la bibliothèque de fonctions "prête à l'emploi Main" est également destinée au traitement des E/S. Cette carte est basée sur microcontrôleur Arduino Uno et composants complémentaires.

2.5 POURQUOI ARDUINO UNO

Il existe plusieurs modèles fabriqués par : ATMEL. Le choix dépend de plusieurs critères de sélection dont le développeur doit tenir compte du :

- * Type du microcontrôleur.
- * Nombre d'entrées/sorties.
- * Liaison d'entrées/sorties.
- * Conversion analogique numérique et numérique analogique.
- * Mémoire RAM, ROM, EPROM interne ou externe, sa taille.
- * Vitesse d'horloge, temps d'exécution d'une multiplication, d'une division.
- * Bus de données 8bits / 16bits.

- * Les logiciels de programmation (assembleur, c, micro, ...).
- * Les émulateurs pour la mise au point des applications.
- * Les évolutions prévisibles du composant, son prix, les sources.

2.6 CONSTITUTION DE LA CARTE ARDUINO UNO

Arduino est un outil pour construire des appareils qui peuvent interagir avec l'environnement qui les entoure. Peut être utilisé pour connecter des capteurs qui détectent le son, la lumière ou les vibrations. Ces capteurs peuvent être utilisés pour allumer les lumières, changer les couleurs, démarrer les moteurs, etc. Arduino est le système magique au cœur de toute cette action. Arduino se présente généralement sous la forme d'une carte électronique bleue de la taille d'une paume. Cette carte comporte des inscriptions blanches qui permettent d'identifier différents éléments. Tous les composants et circuits de la carte sont visibles et accessibles. Arduino est un microcontrôleur, c'est donc un ordinateur très simple. Il ne peut pas faire beaucoup de choses à la fois, mais il fait très bien ce qu'on lui dit de faire. Il existe de nombreux types de microcontrôleurs, mais ce qui rend l'Arduino unique, c'est qu'il est conçu pour les débutants, tout en étant adaptable à des projets plus importants.

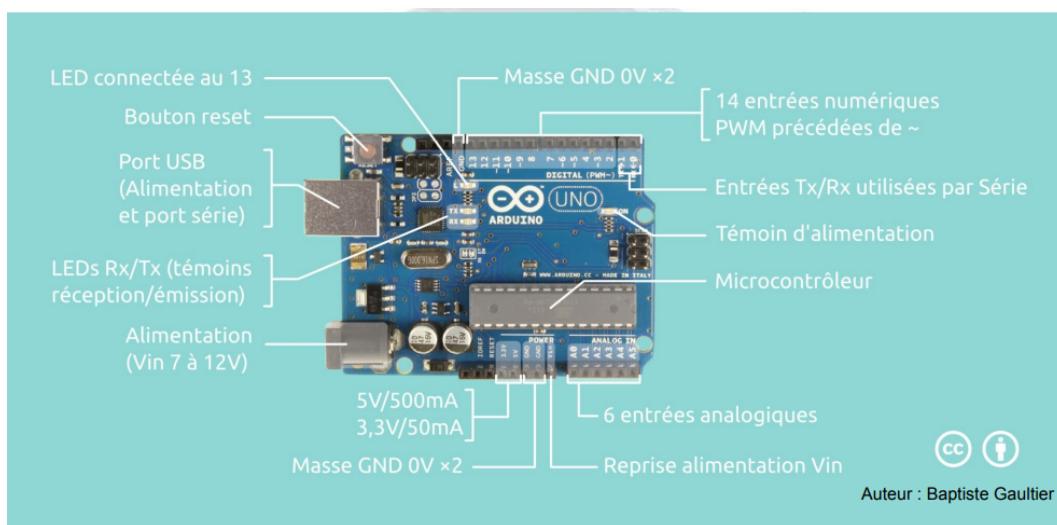


FIGURE 2.1 – Description de la carte Arduino UNO

La carte de microcontrôleur Arduino UNO est la première version stable de la carte Arduino. Il utilise un microcontrôleur avec une architecture ATmega328p fonctionnant à 16 MHz. Il dispose de 32 ko de mémoire flash pour la réception des programmes, de 2 ko de SRAM (mémoire à accès aléatoire) et de 1 ko

d'EEPROM (mémoire morte pour les données). Il y a 14 broches d'entrée/sortie numériques, dont 6 peuvent être utilisées comme sorties PWM (Pulse Width Modulation). Ces six entrées analogiques permettent également de mesurer des grandeurs analogiques.

2.6.1 Microcontrôleur

Il s'agit d'un circuit intégré qui regroupe plusieurs éléments dans un petit espace sur une puce et est un système à microprocesseur avec des périphériques intégrés tels que le stockage de données et des programmes pouvant être utilisés comme système embarqué. L'architecture de la carte Arduino a été publiée en open source et sa philosophie globale est basée sur le monde libre au sens le plus large. Les microcontrôleurs sont de plus en plus utilisés dans les applications embarquées telles que les téléphones et les automobiles. Un microcontrôleur se compose de quatre parties principales :

- Un microprocesseur dont la fonction est le traitement des informations, composé d'une unité arithmétique et logique (UAL), d'un bus de données, d'adresse et de commande, ayant pour tâche l'exécution du programme embarqué dans le microcontrôleur.
- Une mémoire de données (RAM ou EEPROM) dans laquelle seront stockées les données temporaires nécessaires aux calculs qui est une mémoire de travail volatile.
- Une mémoire de programme (flash), contenant les instructions du programme à exécuter. Il s'agit ici d'une mémoire non volatile.
- La dernière partie correspond aux ressources auxiliaires qui sont :
 - * Des ports d'entrées / sorties (parallèle ou série).
 - * Des Timers servant à générer ou mesurer des signaux.
 - * Des convertisseurs A/N pour le traitement des signaux analogiques.

2.6.2 Entrées/Sorties numériques

La carte possède 14 broches numériques (numérotées de 0 à 13) qui sont programmables avec les commandes `pinMode ()`, `digitalWrite ()` et `digitalRead ()`. Ces broches fonctionnent à 5V et chacune il fournit ou reçoit un courant maximum de 40 mA avec une résistance interne (pull-up) de 20-50 kOhm (déconnecté par défaut). Ce registre interne est activé par une instruction `digitalWrite (broche, HIGH)` sur la broche d'entrée.

En plus, certaines broches ont des fonctions spécialisées par exemple :

- Transmission et réception série : les broches 0 (RX) et 1 (TX) permet de recevoir et transmettre les données séries de niveau TTL et ces pins sont connectés aux pins correspondants de l'USB-TTL puce Serial ATmega8U2.
- Interruptions Externes : les broches 2 et 3 peuvent être configurées pour déclencher une interruption sur un niveau bas, sur un front montant, descendant et sur un changement de valeur.
- Modulation à largeur de bande (PWM : Pulse Width Modulation) : les broches 3, 5, 6, 9, 10, 11 peuvent fonctionner en mode PWM pour faire varier la puissance du signal envoyé sur 8 bits à l'aide de l'instruction analogWrite () .

2.6.3 Entrées analogiques

La carte UNO dispose de 6 entrées analogiques (A0 à A5), chacune pouvant fournir des mesures avec une résolution de 10 bits (0 à 1023) grâce à la fonction analogRead () du langage Arduino. Par défaut, ces broches mesurent des tensions entre 0V, correspondant au niveau 0, et 5V, correspondant au niveau 1023. () Instruction du langage Arduino.

2.7 LANGAGE DE PROGRAMMATION

Un langage de programmation est un langage qui permet de créer un ensemble d'instructions qui sont traduites en langage machine à l'aide d'un compilateur. L'avantage du langage Arduino est qu'il est basé sur le langage C/C++, prenant en charge toute la syntaxe standard du langage C et certains outils C++. De plus, des bibliothèques sont disponibles pour faciliter la communication avec les appareils attachés à la carte (afficheurs LCD, afficheurs 7 segments, capteurs, servomoteurs, etc.). Pour programmer en utilisant le langage Arduino, certaines règles doivent être respectées. Notez que l'exécution du programme Arduino est séquentielle. Autrement dit, les instructions sont exécutées les unes après les autres. Tout d'abord, le compilateur doit vérifier l'existence de deux structures obligatoires pour tout programme écrit en langage Arduino.

- Initialisation et réglage de la partie entrée/sortie (réglage de la fonction ()).
- Un corps qui s'exécute en boucle (la fonction Loop ()).

2.7.1 Structure générale d'un programme en langage Arduino

Pour utiliser l'IDE Arduino standard (arduino.exe), entrez votre code dans la fenêtre prévue, compilez le programme et téléchargez-le sur votre carte Arduino. La carte doit être connectée à votre ordinateur avec un câble USB. Le modèle de la

carte Arduino et le port série auquel elle est connectée doivent être déclarés dans les menus Tools/Board Type et Tools/Serial Port IDE. Une fois le programme compilé téléchargé sur le microcontrôleur, il commencera à s'exécuter. La fonction setup () ne s'exécute qu'une seule fois après la mise sous tension et après chaque redémarrage. D'autre part, la fonction Loop () s'exécute dans une boucle. La Figure 2.2 montre la structure générale d'un programme IDE.

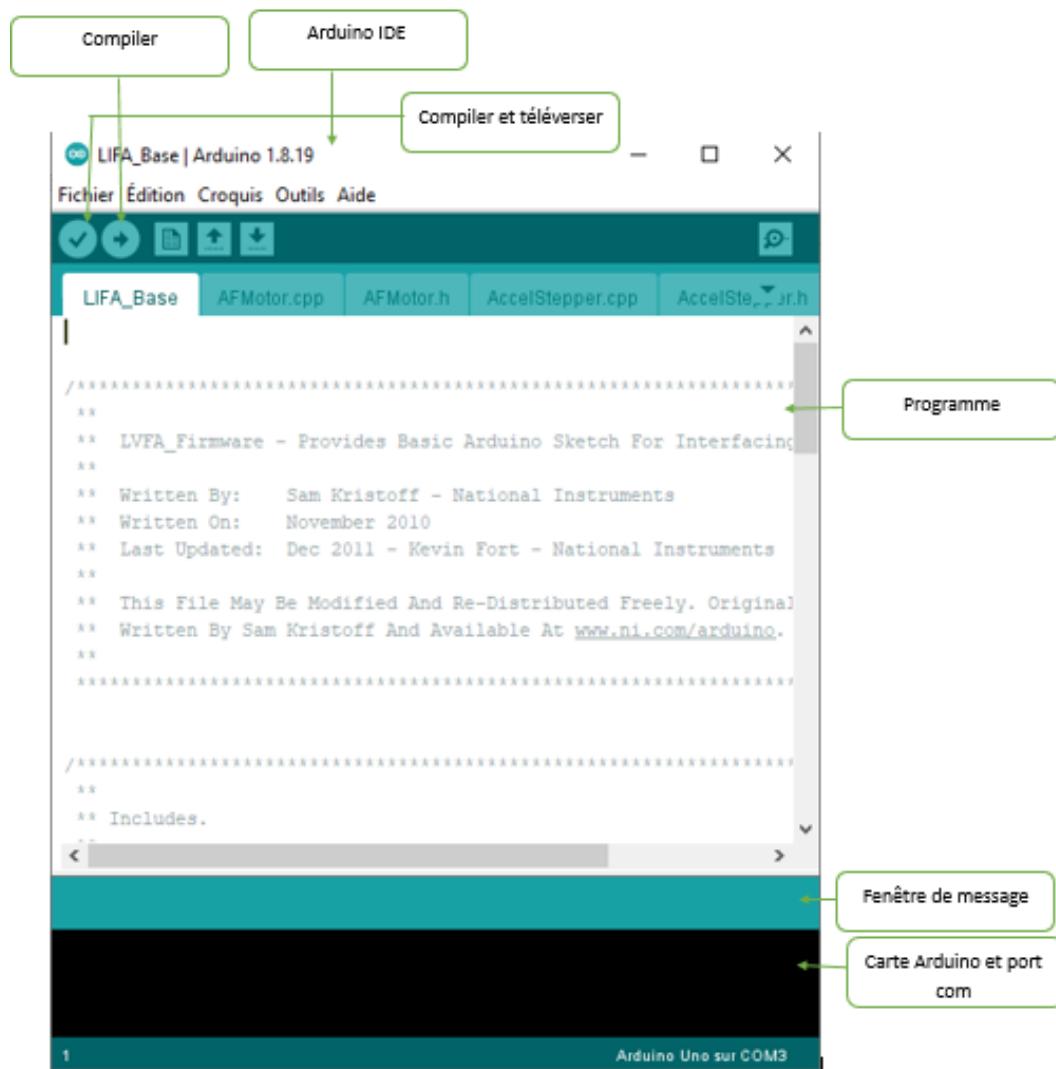


FIGURE 2.2 – Structure d'un programme Arduino.

Un programme Arduino comporte trois parties principales :

1. La partie déclaration des variables (optionnelle).
2. La partie initialisation et configuration des entrées/sorties : la fonction setup () .
3. La partie principale qui s'exécute en boucle : la fonction Loop () .

2.7.2 Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

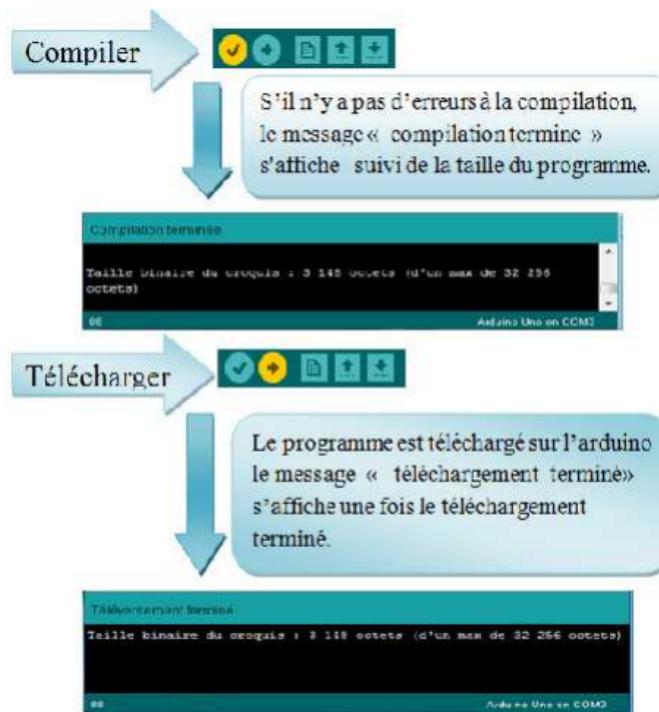


FIGURE 2.3 – Les étapes de téléchargement du code.

2.7.3 Les fonctions IDE de l'environnement Arduino

Nous citons quelques fonctions de base manipulons les bibliothèques de l'Arduino

Fonctions d'initialisation

- * Begin () : initialise communication avec Arduino "maître".
- * Begin (adresse) : initialise communication avec Arduino "esclave".

Fonctions mode maître

- * requestFrom (adresse, quantité) : demande de données à un esclave.
- * beginTransmission (adresse) : débute communication avec un esclave (ouvre stockages donnés à envoyer avec write).
- * endTransmission () : envoie des données vers esclave.
- * write () : écrit les données à envoyer vers esclave.
- * available () : teste ces données disponibles en provenance esclave (cf requestFrom).
- * read () : lit les données en provenance de l'esclave.

Fonctions mode esclave

- * write () : envoie les données vers le maître après requête.
- * available () : test ces données disponibles en provenance du maître (cf onReceive).
- * read () : lit données en provenance maître.
- * onReceive (fonction) : définit la fonction à appeler sur réception de données en provenance du maître.
- * onRequest (fonction) : définit la fonction à appeler sur requête du maître.

Fonctions obsolètes

- * send () : obsolète.
- * receive () : obsolète.

PWM

2 à 13 et 44 à 46. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite () .

2.8 FONCTIONNEMENT ET UTILISATION

L'utilisation de la carte Arduino est très simple : connectez la carte à votre ordinateur (généralement avec un câble USB), lancez "Arduino IDE" (le logiciel

qui permet de programmer la carte), configurez vos composants, connectez-les à la carte, codez et chargez votre programme.

2.9 CONCLUSION

Dans ce chapitre, nous avons détaillé les éléments électroniques de base Arduino UNO utilisés pour mettre en œuvre le module de régulation. Cela offre des possibilités créatives presque illimitées si vous disposez du bon équipement.

Il est clair que l'Arduino permet d'assurer les principaux protocoles d'interfaçage : communication bidirectionnel, acquisition (conversion) et traitement des différents types de signaux avec une bonne précision et rapidité avec une mémoire de programme de données importantes.

COMMANDÉ, CONTRÔLE ET REGULATION DANS LABVIEW

3

3.1 INTODUCTION

Dans ce chapitre, on va représenté le logiciel de simulation graphique utilisé et la liaison avec la carte Arduino.

3.2 LA PROGRAMMATION GRAPHIQUE

LabVIEW est une plate-forme de programmation pour le contrôle d'instruments et l'analyse de données. Contrairement à la nature séquentielle des langages textuels, LabVIEW est basé sur un environnement de programmation graphique utilisé pour le contrôle, l'instrumentation, la mesure et les tests automatisés à partir d'un PC. A cet égard, nous avons présenté l'interface graphique produite par ce logiciel et ses fonctionnalités. LabVIEW est entièrement équipé pour communiquer avec du matériel tel que : GPIB, VXI, PXI, RS-232, RS-485 et périphériques d'acquisition de données cablés. LabVIEW a une fonctionnalité intégrée pour connecter des applications à Internet via le serveur Web de LabVIEW, prenant en charge des protocoles standard tels que TCP/IP et ActiveX. Grâce à LabVIEW, nous prouvent créer des applications compilées 32 bits qui nous offrir des vitesses d'exécution rapides nécessaires pour des solutions personnalisées d'acquisition de données, de test, de mesure et de contrôle. LabVIEW comprend des bibliothèques étendues pour l'acquisition, l'analyse et le traitement de données basés sur des fonctions, l'affichage et le stockage de données. LabVIEW comprend également toutes les constructions de programmation des langages traditionnels. Définissez des points d'arrêt et animez l'exécution du programme pour exécution étape par étape pour un débogage et un développement plus facile. Parce que LabVIEW a la capacité de communiquer avec des périphériques, il peut donc être utilisé pour envoyer des données à ces périphériques (par exemple des robots) pour les contrôler, même si ce n'est pas leur objectif premier. Le programme LabVIEW est appelé instrument virtuel ou VI (Virtual Instrument).

3.3 DOMAINE D'APPLICATION

Les utilisations de LabVIEW sont de plus en plus diversifiées. A la base, LabVIEW était destiné au contrôle d'instruments de laboratoire. LabVIEW offre à des milliers d'utilisateurs satisfaits un moyen plus rapide de programmer l'Instrumentation, l'acquisition de données et les systèmes de commande. En utilisant LabVIEW pour réaliser le prototype, la conception, les tests et la mise en application des systèmes d'instrumentation, nous pouvant réduire aussi le temps de

développement du système et en augmenter la productivité de 4 à 10 fois. Cependant, les possibilités de communications offertes (RS232, USB, Ethernet, GPIB...), la diversité des bibliothèques disponibles, la possibilité de créer très rapidement des Interfaces graphiques efficaces en font un outil de plus en plus utilisé par les industriels.



FIGURE 3.1 – Schéma synoptique de domaine d'application du LabVIEW.

LabVIEW fournit des outils de débogage uniques que vous pouvez utiliser pour vérifier le mouvement interactif des données dans votre programme. Ils vous permettent également de visualiser les valeurs de données lorsqu'elles sont transmises d'une fonction à une autre via un fil. LabVIEW suit un modèle de flux de données pour l'exécution des VIs. Le nœud du diagramme s'exécute après avoir reçu toutes les entrées requises. Lorsqu'un nœud s'exécute, il produit des données de sortie, qui sont envoyées au nœud suivant via un flux de données.

3.4 ENVIRONMENT LABVIEW

Les programmes LabVIEW sont appelés instruments virtuels ou VIs car ils ressemblent et se comportent comme de vrais instruments, tels que des oscilloscopes et des multimètres. Chaque VI utilise des fonctions qui manipulent les entrées d'interface affichez l'utilisateur ou d'autres sources et affichez ces informations ou déplacez-les vers d'autres fichiers ou ordinateurs.

3.4.1 FACE-AVANT / Diagramme

LabVIEW étant plus particulièrement destiné à l'instrumentation, ce programme apparaît constitué comme un appareil de mesure : face-avant et diagramme.

FACE-AVANT

La face-avant est l'interface avec l'utilisateur reproduisant les différents contrôles (action de l'utilisateur vers l'instrument, donc entrée d'une variable) et les différents indicateurs (visualisation d'une variable).

Diagramme

Le diagramme (initialement blanc) où l'on retrouve les contrôles, les indicateurs ainsi que des opérateurs il correspond aux circuits internes de l'appareil et constitue le cœur du programme.

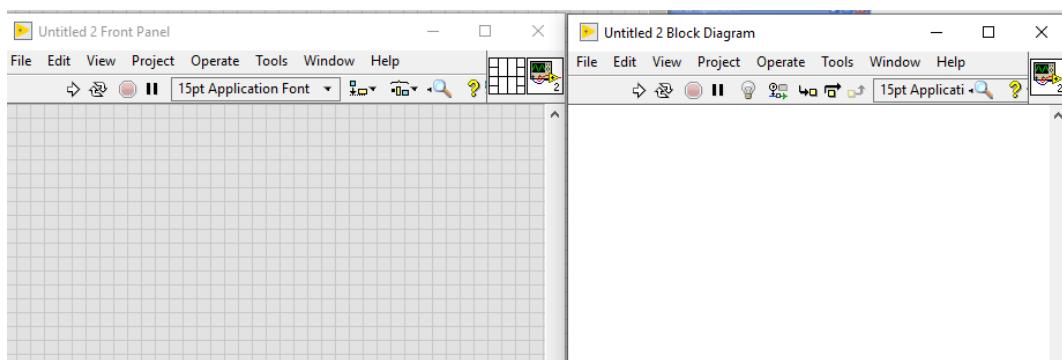


FIGURE 3.2 – Fenêtre de l'environnement de développement sur LabVIEW face-avant (à gauche) et Diagramme (à droite).

Chaque VI affiche une icône, dans le coin supérieur droit des fenêtres de la face-avant et du diagramme. Un connecteur est un ensemble de terminaux correspondant aux commandes et aux indicateurs du VI qui sont accessibles.

Menus hiérarchisés communs aux fenêtres du diagramme et du panneau avant :

File : Pour manipuler les fichiers (créer, ouvrir, sauver, imprimer).

Edit : Pour éditer la fenêtre active (copier, coller).

View : Pour voir la hiérarchie des VI en mémoire et rappeler les palettes Tools ou Controls P.

Operate : Pour choisir le mode de fonctionnement Tools.

Window : Pour changer la fenêtre active (« panneau avant » /diagramme).

Help : Pour afficher la fenêtre d'aide de LabVIEW et accéder à l'aide en ligne (Online Reference).

3.4.2 Palette d'outils

Elle est disponible sur la face-avant et sur le diagramme, elle contient les outils nécessaires pour faire fonctionner et modifier la face avant et les objets du diagramme.

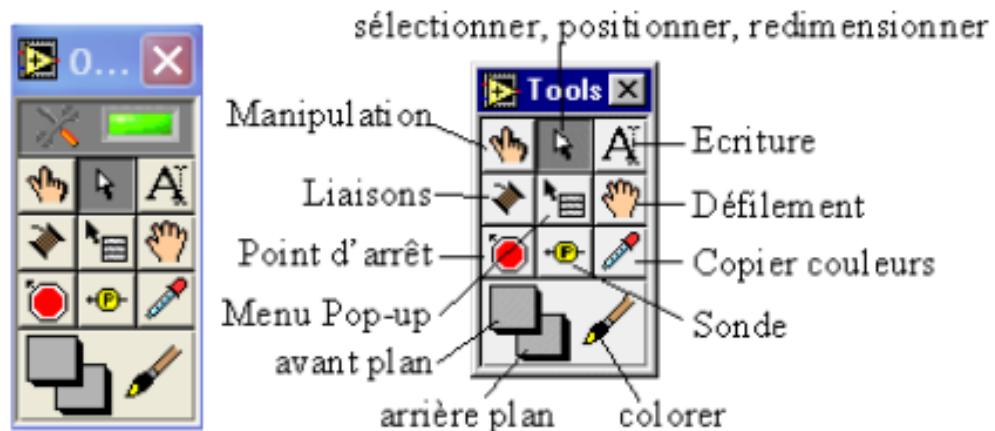


FIGURE 3.3 – Palette d'outils.

3.4.3 Palette de commandes

Elle est disponible uniquement sur la face-avant, elle contient les commandes, les éléments graphiques et indicateurs de la face-avant nécessaire pour créer l'interface utilisateur. Ceux-ci sont hiérarchisés par type de données ou par grandes familles d'objets.

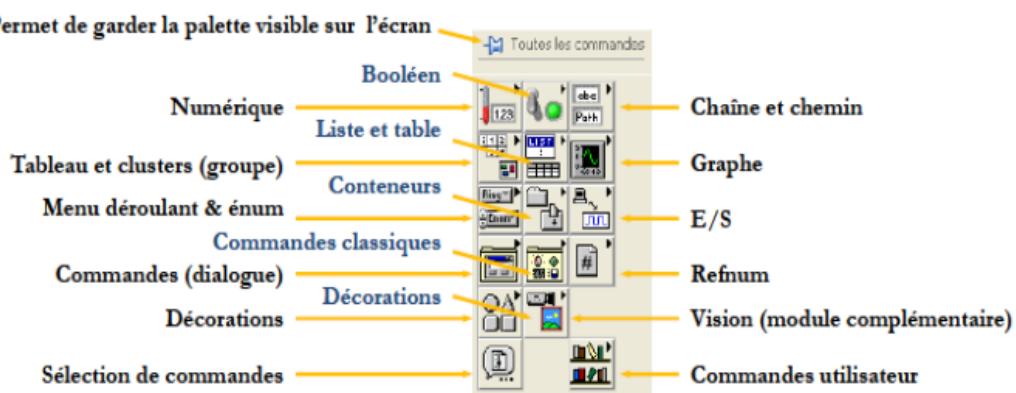


FIGURE 3.4 – Palette de commandes.

3.4.4 Palette de fonctions

Elle est accessible uniquement dans le diagramme par les mêmes méthodes que celle de commandes. Elle contient l'ensemble des fonctions nécessaire de LabVIEW regroupées par type de fonctionnalités (Programmation, acquisition de données, traitement mathématiques et arithmétique...).

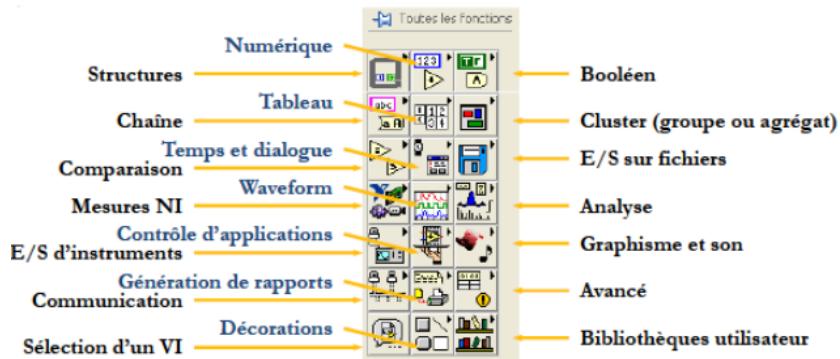


FIGURE 3.5 – Palette de fonctions.

3.5 DESCRIPTION DES SOUS MENUS DE COMMANDE ET DE FONCTIONS

Grâce au sous-menu Commandes et fonctions, les programmeurs peuvent manipuler les graphiques pour obtenir des interfaces de commande et de contrôle arbitraires. Ce sont donc les éléments nécessaires pour pouvoir créer une interface utilisateur.

3.5.1 Sous menus de commande

Il touche les commandes et les affichages qui composent l'interface utilisateur, tels que : Voltmètres, boutons, affichages numériques et analogiques, boutons rotatifs, etc. Tous ces éléments sont situés sur le panneau avant et sont accessibles à partir de la palette de commandes.

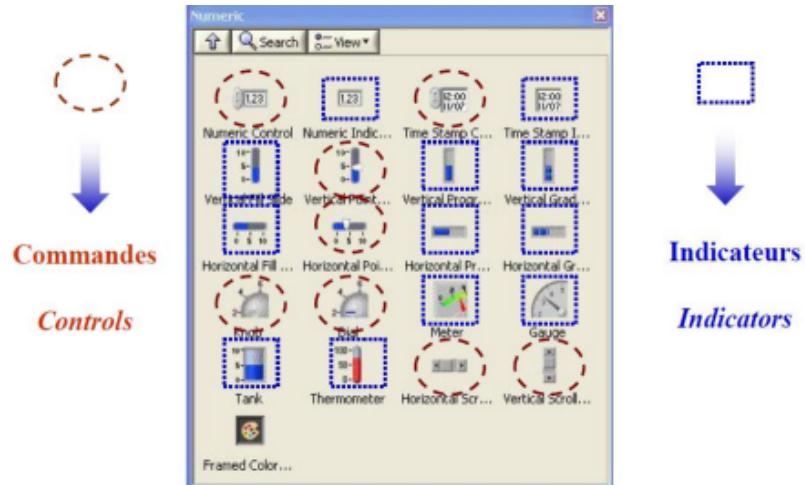


FIGURE 3.6 – sous menu de commande.

3.5.2 Sous menus de fonctions

LabVIEW dispose d'une bibliothèque de fonctions logiques et arithmétiques qui manipulent les bits et facilitent les opérations arithmétiques au niveau des équations simples et complexes.

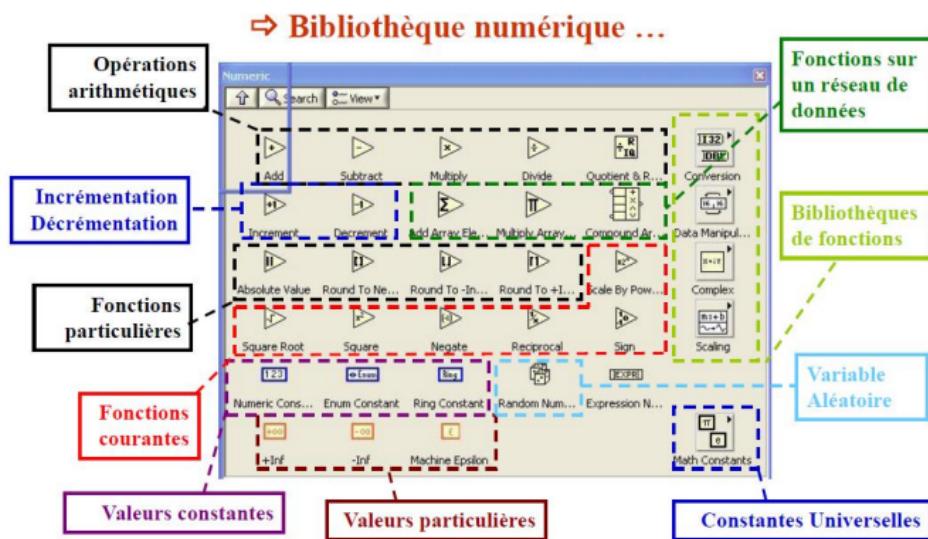


FIGURE 3.7 – Sous menu de fonction numérique.

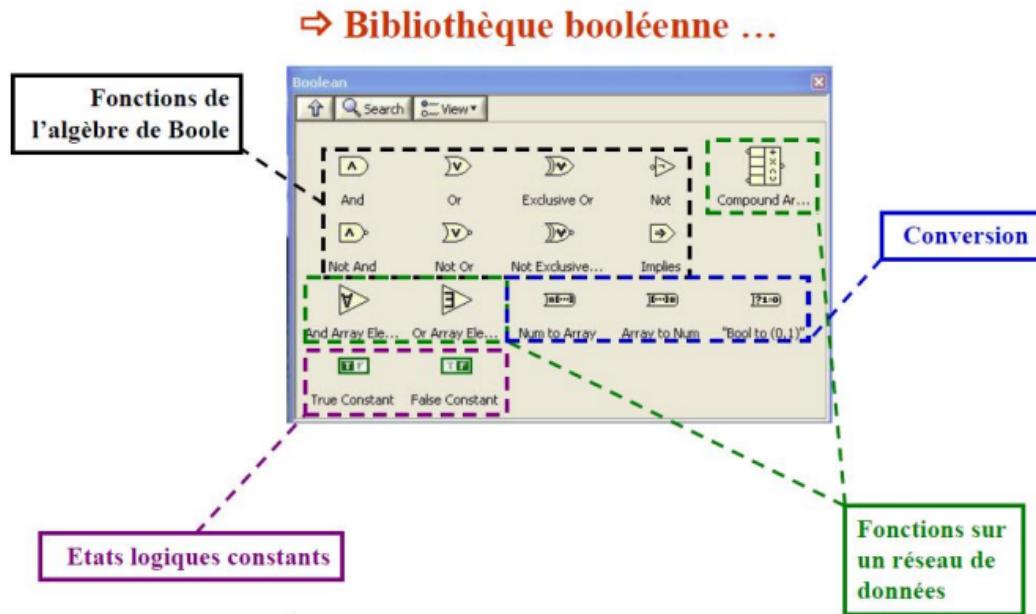


FIGURE 3.8 – Sous menu de fonction logique (booléenne).

3.6 STRUCTURE DANS LABVIEW

Dans cette section , on va étudier les structures dans labview.

3.6.1 Structure de données

LabVIEW utilise un langage fortement typé et des données ou structures de données arbitraires ne peuvent être utilisées que dans les fonctions qui autorisent ce type. En fait, LabVIEW prend en charge les types scalaires, les types entiers (signés ou non, code 8, 16 ou 32 bits), les types réels (16, 32 ou 64 bits), les types booléens, les types chaîne, les types de tableau (matrice) et les clusters. Notez la forme et la couleur différentes des éléments représentant ces données et les liens résultant de ces éléments.

Ces fils associés à des variables sont utilisés pour transporter des données entre les bornes vers les nœuds du V.I.

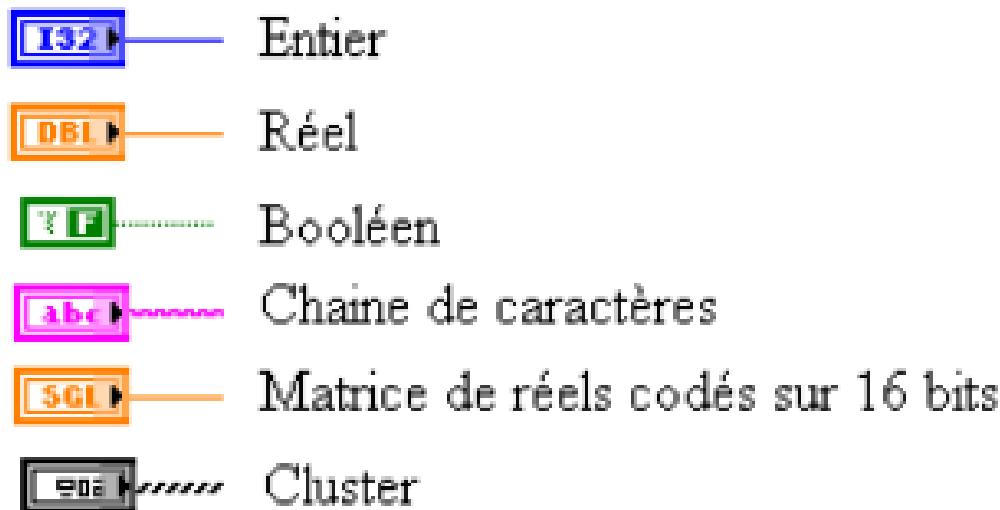


FIGURE 3.9 – Différents types de structures de données dans LabVIEW.

Le langage permet aussi de créer des structures de données plus élaborées.

3.6.2 Structure de programme

LabVIEW utilise un langage flot de données pur qui a été enrichi de quatre types de structures : la séquence, deux structures d’itération (la boucle « Pour où FOR » avec un nombre d’itérations fixe et la boucle « Tant Que où While » avec un nombre d’itérations soumis à condition) et la structure de choix.

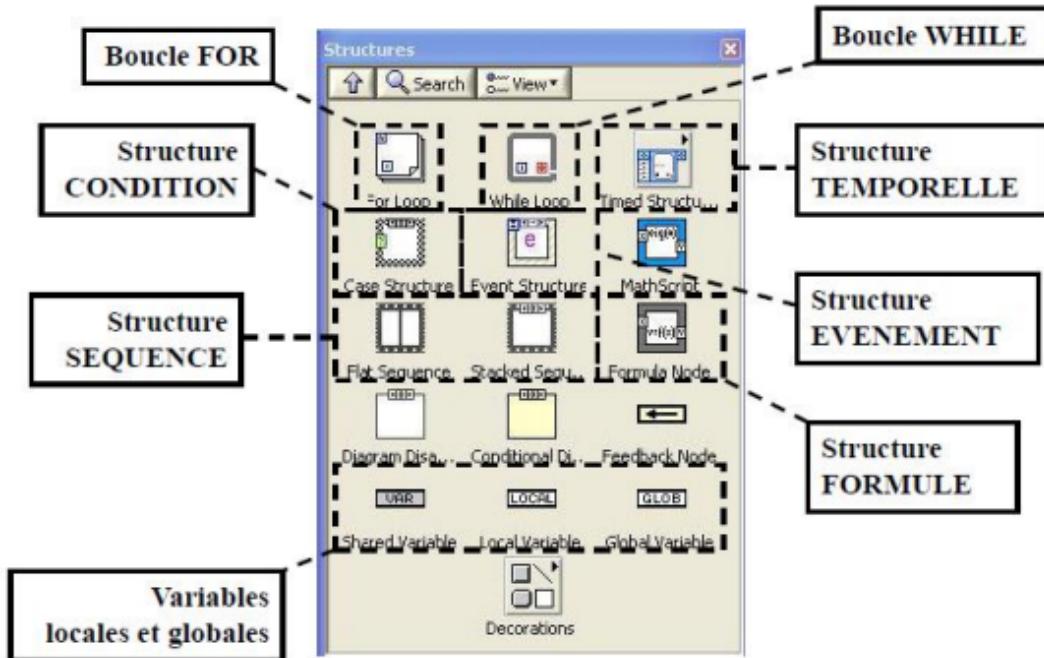


FIGURE 3.10 – Bibliothèque boucles et structures données.

3.7 COMMUNICATION ARDUINO ET LABVIEW (PILOT E/S INSTRUMENTS)

Cette partie montre comment établir une communication entre l'interface LabVIEW et la carte Arduino à l'aide de protocoles de communication tels que Serial (UBS) et à l'aide de diverses bibliothèques et plates-formes qui assurent le contrôle et le contrôle des périphériques d'E/S. de données en temps réel.

Les différents interfaces et bibliothèques utilisées sont :

- * Interface et Bibliothèque VISA (NI Serial Communication).
- * Interface Arduino-LIFA (LabVIEW Interface for Arduino).
- * Interface ET plateforme Arduino-Maker hub (LINX).

Le but est de présenter une technique de couplage d'une carte Arduino à LabVIEW. Pour ce faire, ajoutez un package spécial (PackageManager) à LabVIEW et contrôlez l'Arduino directement depuis LabVIEW.

Les étapes suivantes décrivent comment installer, configurer et intégrer la carte Arduino avec LabVIEW.



FIGURE 3.11 – Interface LabVIEW-Arduino.

Donc les éléments nécessaires pour l'installation et afin d'établir la communication sont :

- Installation du LabVIEW.
- Installation de l'Arduino IDE.
- Installation NI-VISA.
- Gestionnaire de paquets VI (Package Manager (VIPM)).
- Après l'utilisateur doit être installé les Interfaces LabVIEW pour Arduino à partir du Package Manager La bibliothèque Arduino-LabVIEW (LINX) ou la bibliothèque (LIFA BASE), et d'autres si nécessaire.

3.7.1 Interface et Bibliothèque VISA

Pour communiquer avec vos instruments, nous utilisons principalement les fonctions VISA (Visual Instrument Software Architecture) que LabVIEW utilise pour les drivers d'instruments.

Les fonctions VISA vous permettent de communiquer avec les pilotes d'E/S installés sur votre système, tels que RS232 série, USB, GPIB, Ethernet, VXI ou PXI Link. Ces pilotes sont écrits pour différents types de communication.



FIGURE 3.12 – Ouverture et fermeture d'une VISA pilote d'instrument.

* Les fonctions VISA les plus couramment utilisées pour communiquer avec les instruments de mesure sont les fonctions VISA :

- VISA clean : permet d'effacer les tampons d'entrée et de sortie de l'appareil.
- VISA open : permet d'ouvrir une session sur le périphérique spécifié par le nom de ressource VISA et renvoie un identifiant de session qui peut être utilisé pour appeler toute autre opération de ce périphérique.
- VISA Read : de lire le nombre d'octets spécifié à partir du périphérique ou de l'interface spécifié par le nom de ressource VISA et renvoie les données dans le tampon de lecture.
- VISA close : permet de fermer la session.

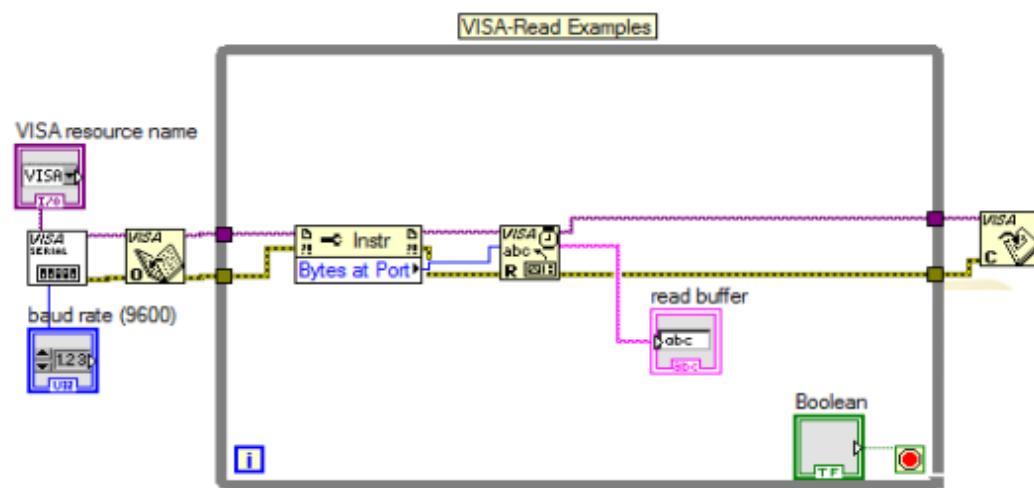


FIGURE 3.13 – Exemple Pack VISA LabVIEW.

3.7.2 Interface LabVIEW Arduino-LIFA

LIFA signifie LabVIEW Interface pour Arduino, LabVIEW Interface for Arduino (LIFA) est une boîte à outils qui permet aux développeurs de capturer des données à partir de microcontrôleurs Arduino.

Il est important d'effectuer les étapes suivantes une fois pour créer un programme et établir la communication avec l'interface LabVIEW de l'Arduino LIFA :

- Installation dans LabVIEW la bibliothèque ou le module Arduino LIFA via le VI Package Manager (VPM).
- Téléchargement via l'IDE Arduino le Firmware LIFA base (fichier LIFA base.ino) sur la carte Arduino cible, ce programme Firmware est disponible par défaut aux libraires de l'Arduino dans le fichier LIFABase.
- Lorsque le module Arduino est installé, la palette Arduino et ces bibliothèques est disponible dans la palette des fonctions du LabVIEW.

La figure ci-dessous présente le module Arduino.

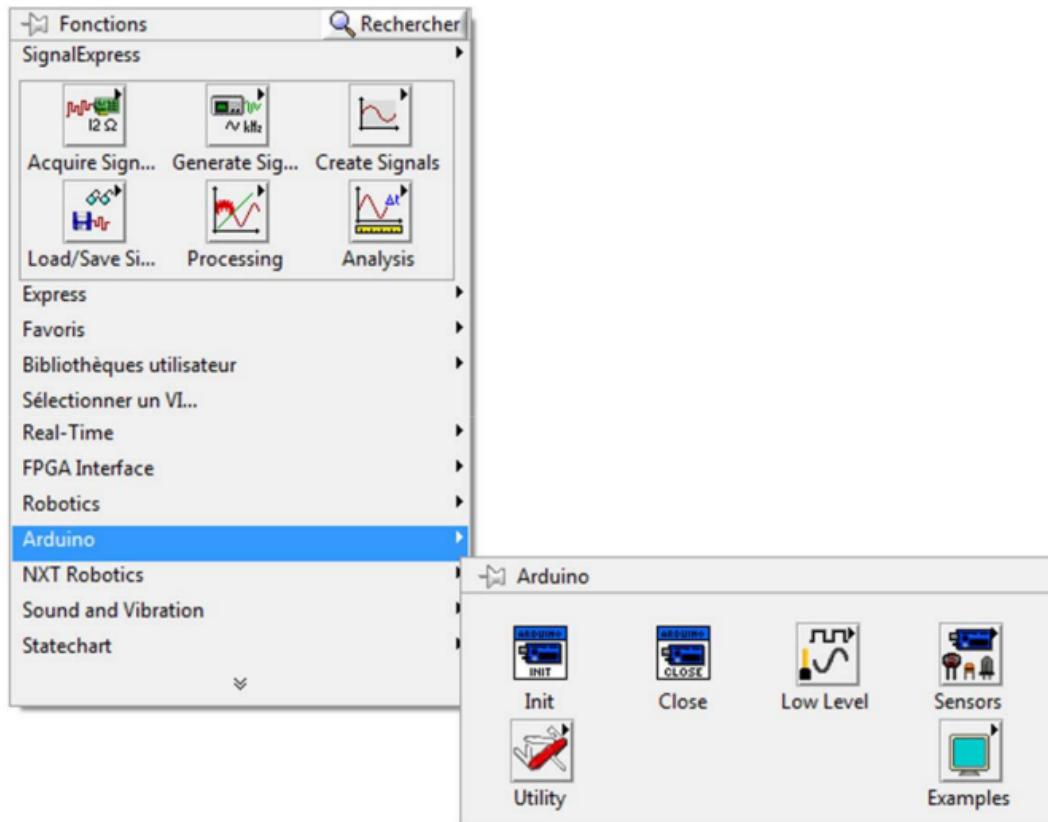


FIGURE 3.14 – Interface LabVIEW Arduino.

Pour démontrer certaines des fonctionnalités utilisées dans l’interface LabVIEW Arduino (LIFA), nous allons passer par la réalisation d’un ou deux exemples de programmes. Le premier programme montre comment créer un programme de configuration de broches et accéder à la sortie numérique d’écriture Arduino.

La seconde montre comment lire une entrée analogique à travers un potentiomètre.

* Exemple :

La carte Arduino a une LED connectée au pin 13. Vous pouvez créer votre premier programme sans rien connecter.

Eclairage Led L disponible sur carte Arduino. Utilise la palette Arduino.

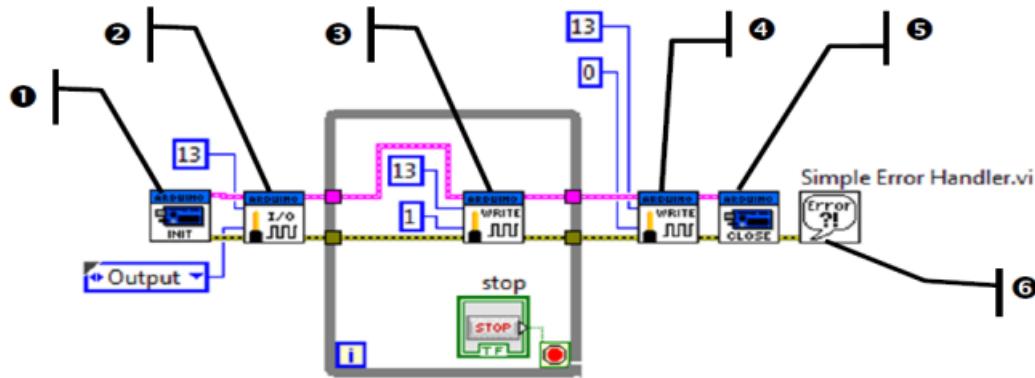


FIGURE 3.15 – Exemple VI LabVIEW Arduino.

- 1) Init – Initialise la carte Arduino.
- 2) Set Pin Mode - Déclare si le Pin est une entrée ou une sortie. Dans notre cas, le Pin 13 est une sortie.
- 3) Digital Write - Ecrit la valeur indiquée - ici 1 - sur le Pin digital concerné – ici 13 -.
- 4) Digital write. Ecrit 0 sur le Pin 13.
- 5) Arduino Close – ferme la carte Arduino.
- 6) Gestionnaire d’erreurs : Se trouve dans la palette Programmation – Dialogue/IU (Interface utilisateur) – Gestionnaire d’erreurs simples.
Affiche une erreur avec son code s’il y a lieu.

3.7.3 Interface Arduino-Maker hub (LINX)

Le package LabVIEW Interface for Arduino (LIFA) a été remplacé par LINX. LINX fournit des VIs LabVIEW faciles à utiliser pour interagir avec les plates-formes embarquées populaires telles que Arduino, Chip KIT, myRIO, etc.

LINX 3.0 a étendu le logiciel pour prendre en charge Raspberry Pi et BeagleBone Black, permettant à la carte d'exécuter du code LabVIEW. Avec LINX, vous pouvez commencer à envoyer des données à votre PC en quelques secondes avec les VIs de capteur intégrés, ou accéder aux E/S numériques, aux E/S analogiques, SPI, I2C, UART, PWM, et plus encore avec les VIs de périphériques.

LINX est sa couche d'abstraction matérielle principale, vous permettant d'utiliser une grande variété de périphériques matériels avec une seule interface LabVIEW. Selon votre appareil, vous pouvez y accéder de deux manières :E/S distantes ou locales.

Pour pouvoir utiliser l'interface LINX, vous devez d'abord installer le package

Makerhub-LINX via VIPM (VI Package Manager). LINX fournit et intègre dans sa boîte à outils plusieurs fonctionnalités présentées dans la figure ci-dessous.

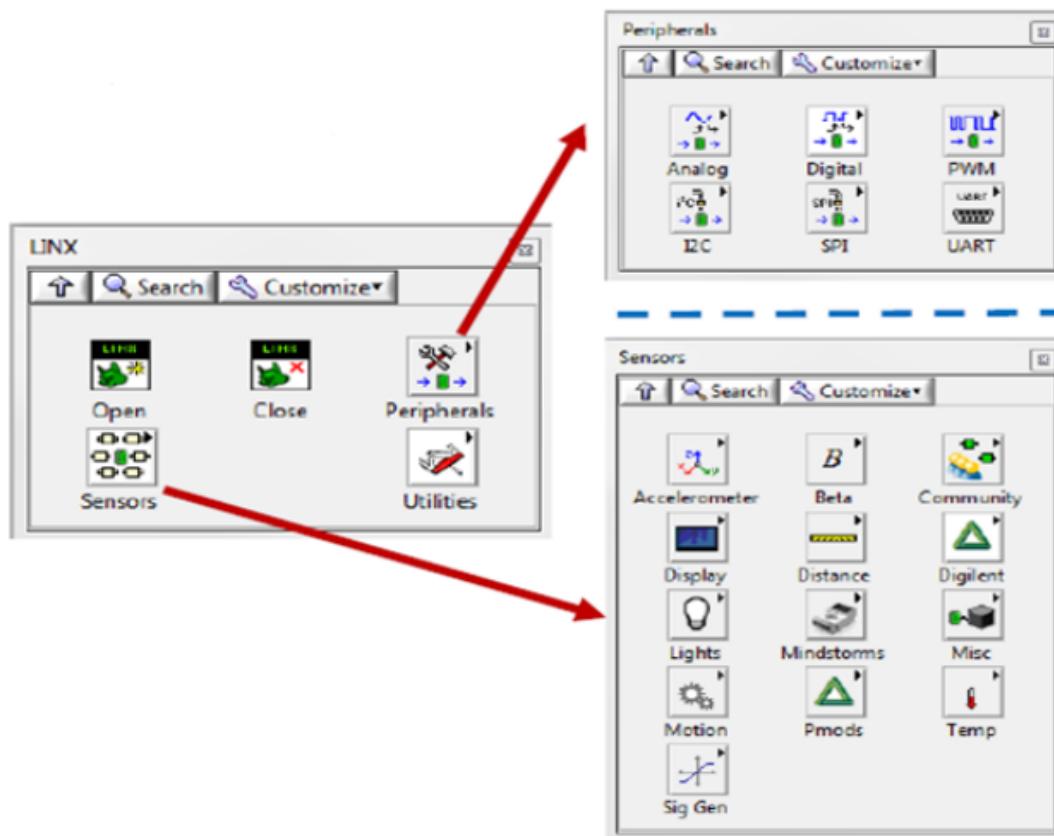


FIGURE 3.16 – Interface LabVIEW Arduino-Maker hub LINX.

* Exemple : Cet exemple montre comment utiliser la palette LabVIEW Arduino LINX pour écrire une valeur sur un canal de sortie numérique Arduino connecté à une LED et surveiller la sortie avec le VI ci-dessous.

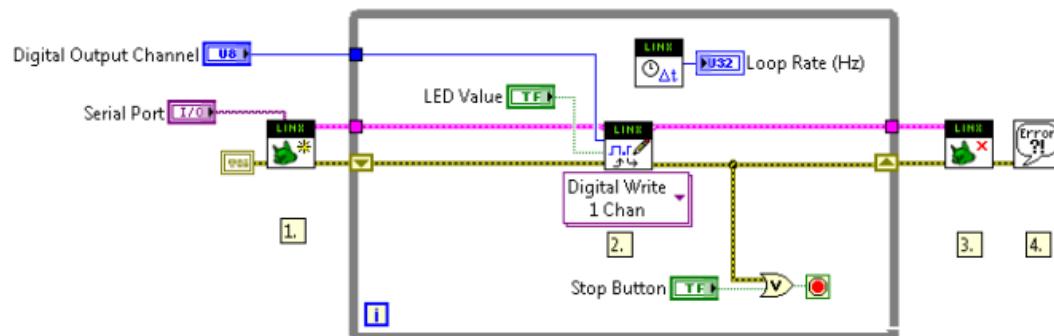


FIGURE 3.17 – Exemple sortie digital utilisent interface LabVIEW Arduino-LINX.

- 1) Open a connection to the LINK device.
- 2) Write the value to the specified DO channel.
- 3) Close the connection to the LINK device.
- 4) Handle Errors.

3.8 CONCLUSION

Ce chapitre a introduit la partie logicielle utilisée dans la programmation de l'interface de commande et de contrôle de notre travail. La partie logicielle est consacrée à la présentation du logiciel LabVIEW, ses fonctionnalités, sa structure et les bibliothèques les plus couramment utilisées.

ETUDE ET RÉALISATION PRATIQUE
DU RÉGULATEUR

4

4.1 INTRODUCTION

Dans ce chapitre, nous considérons les parties matérielles et logicielles du projet, la structure générale du modèle et les différentes parties électroniques et mécaniques utilisées dans la réalisation réelle du système de contrôle commande (régulation de niveau et de température). La réalisation d'un projet de contrôle de niveau et de température nécessite la réalisation d'études électromécaniques et logicielles pour assurer le développement correct et optimal et une série d'étapes pour pouvoir mener à bien l'opération du système de contrôle. La réalisation pratique est divisée en deux grandes parties, une première application de la régulation de niveau, une deuxième sert au contrôle et au régulation de la température.

4.2 PRÉSENTATION LE SYNOPTIQUE DE PROJET

La maquette de régulation de notre projet est constituée de des éléments électroniques sont :

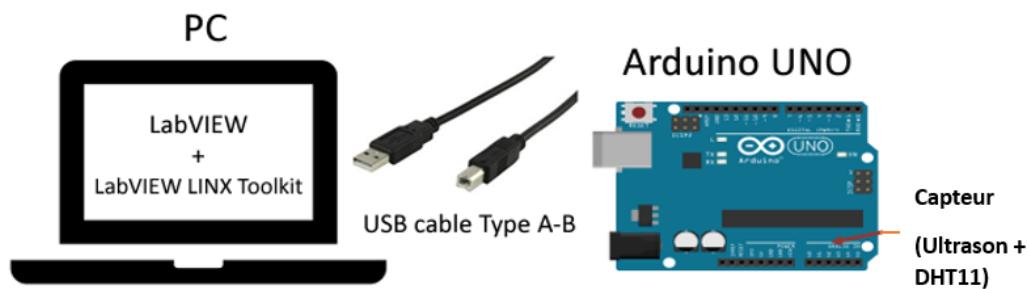


FIGURE 4.1 – Schéma synoptique.

4.2.1 Carte Arduino UNO

Un module Arduino est généralement construit autour d'un microcontrôleur Atmel et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Le microcontrôleur peut être programmé de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques, éclairage, chauffage. . .), le pilotage d'un robot, de l'informatique embarquée, etc.

- Organigramme :

Notre programme Arduino permet de lire les valeurs des capteurs de niveau (ultrason HC-RS04) et température et humidité (DHT11) puis les connecter avec

le programme LabVIEW pour afficher dans un LCD la valeur de chaque capteur et le message correspondant.



FIGURE 4.2 – Organigramme Programme Arduino.

4.2.2 Les capteurs

Le capteur est un composant très important dans la chaîne de mesure et voilà les principales étapes de leur mesure :

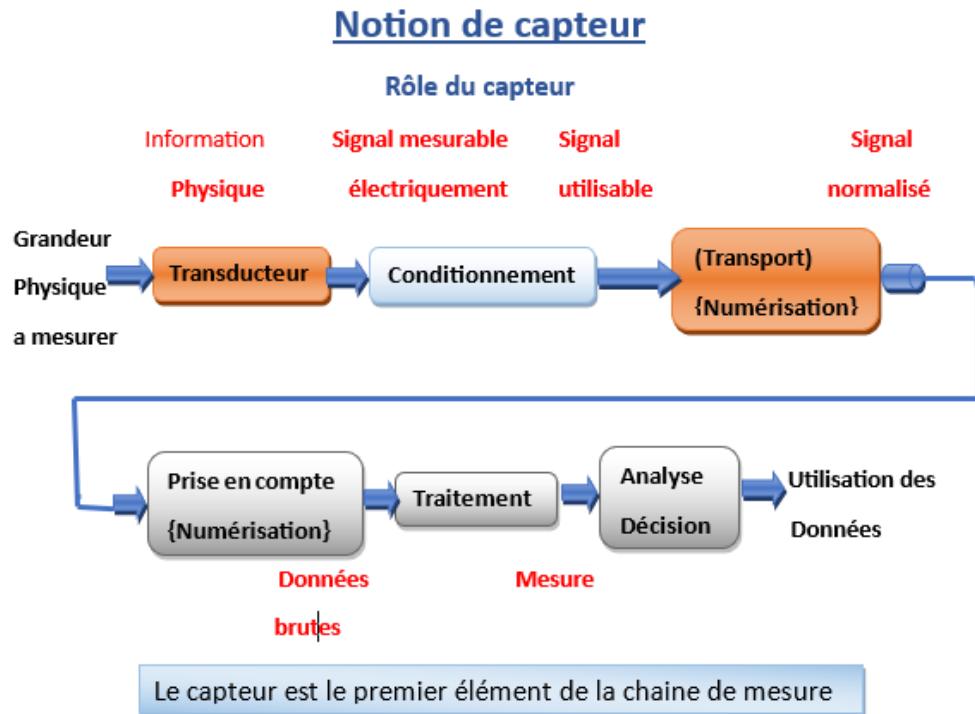


FIGURE 4.3 – Notion de capteur.

Ultrason HC-RSo4

Ce module dispose simplement de 4 pins de sortie : VCC, TRIG, ECHO, GND.



FIGURE 4.4 – Capteur ultrason HC-SR04.

Par conséquent, la connexion avec un micro-ordinateur est très facile. Voici le processus complet :

Lorsque la broche "TRIG" est pulsée vers le haut (5 V) pendant au moins 10 micro(s), le module commencera à lire. A la fin de la mesure, "Pulse In ()" avec la broche "ECHO" passant au niveau haut (5V) lorsqu'un objet est détecté devant. Et la distance sur laquelle l'obstacle existe est proportionnelle à la durée de cet impact.

*Caractéristiques HC-SR04 :

- Alimentation : 5 Vcc.
- Consommation : 15 mA.
- Fréquence : 40 kHz.
- Portée : de 2 cm à 4 m.
- Déclenchement : impulsion TTL positive de 10 micro(s).
- Signal écho : impulsion positive TTL proportionnelle à la distance.
- Calcul : distance (cm) = impulsion (micro(s)) / 58.
- Dimensions : 45 x 21 x 18 mm.

Pour connecter ce capteur avec Arduino, il est nécessaire respecter le montage représenté dans la figure suivante :

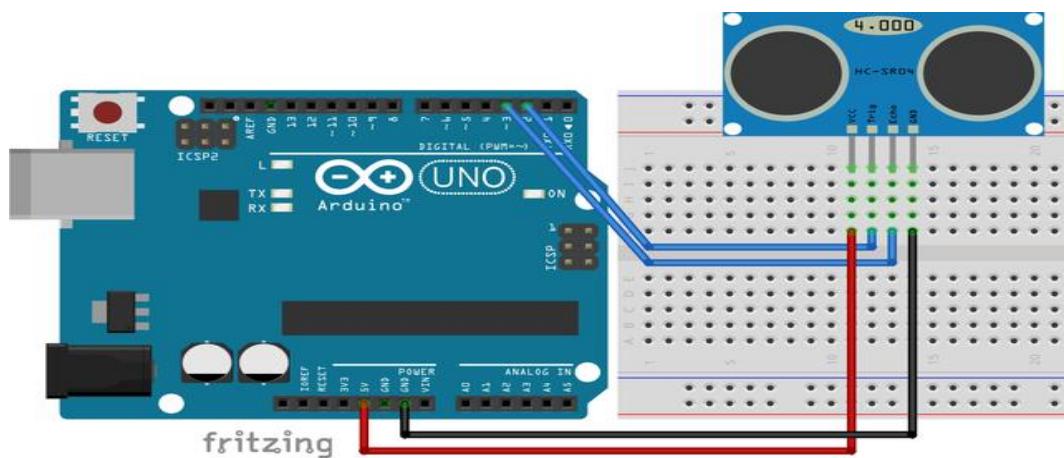


FIGURE 4.5 – Câblage capteur ultrason HC-SR04.

DHT11

Ce module dispose simplement de 4 pins de sortie : VCC(5V), DATA, GND.

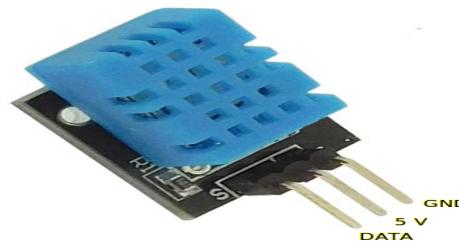


FIGURE 4.6 – Capteur DHT11.

Le capteur DHT11 peut mesurer la température de 0 à 50°C avec une précision

de +/- 2°C, et mesurer l'humidité relative de 20 à 80 avec une précision de +/- 5%. Le DHT11 est alimenté en 5V et a une consommation de courant nominale comprise entre 0,5mA et un maximum de 2,5mA. Il comporte 4 broches, comme le montre la figure 10 : dont une broche d'alimentation Vcc (5V ou 3,3V), une broche de données pour la communication de mesure, qui doit être connectée à l'alimentation via une résistance de 407 kOhms. N/C ne doit pas être câblé GND est la masse GND du capteur.

Pour connecter ce capteur avec Arduino, il est nécessaire respecter le montage représenté dans la figure suivante :

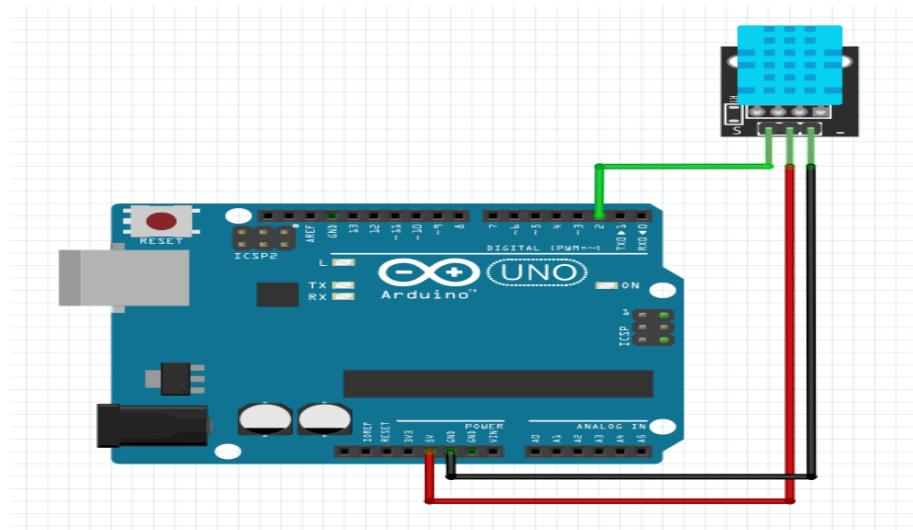


FIGURE 4.7 – Câblage Capteur DHT11.

4.2.3 Les Actionneurs

Dans notre réalisation nous avons utilisé le Positionneur de vanne électropneumatique TROVIS 3793 pour piloter et contrôler le mécanisme mécanique d'ouverture et fermeture de la vanne de régulation de niveau.



FIGURE 4.8 – Positionneur de vanne électropneumatique TROVIS 3793.

- CARACTÉRISTIQUES :

Type : électropneumatique.

Mouvement : rotatif, linéaire.

Configuration : double effet, simple effet.

Indice de protection : IP66.

Course de la tige : Max : 300 mm (12 in).

Min : 4 mm (0 in).

Pression : Max : 14 bar (203,1 psi).

Min : 0 bar (0 psi).

Température ambiante : Max : 85 °C (185 F).

Min : -50 °C (-58 F).

Angle de rotation : Max : 170 °C. Min : 24 °C.

Poids : Max : 2 kg (4,41 lb). Min : 1 kg (2,2 lb).

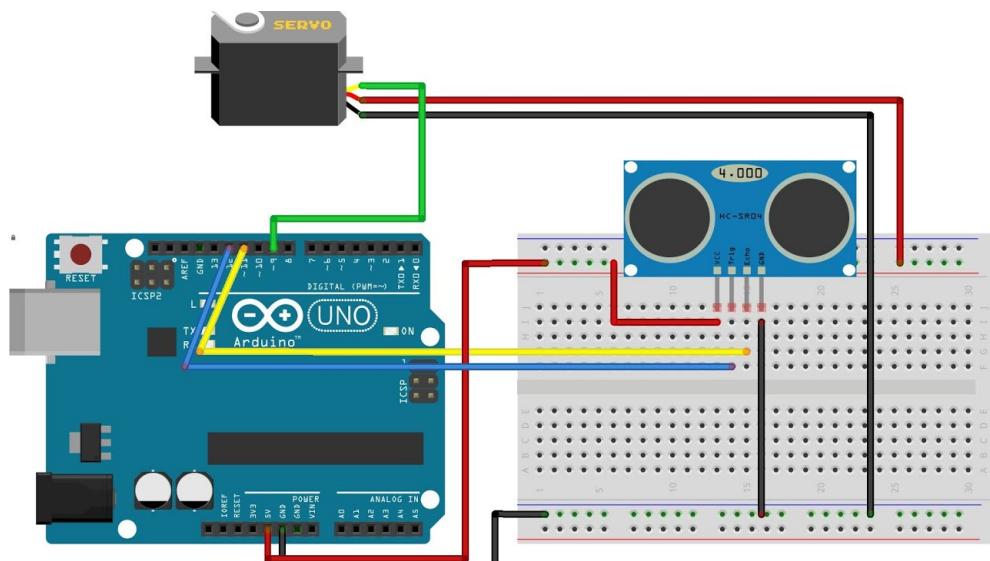


FIGURE 4.9 – Câblage Positionneur de vanne électropneumatique TROVIS 3793 et capteur ultrason HC-RS04.

4.2.4 Le schéma général du module électronique

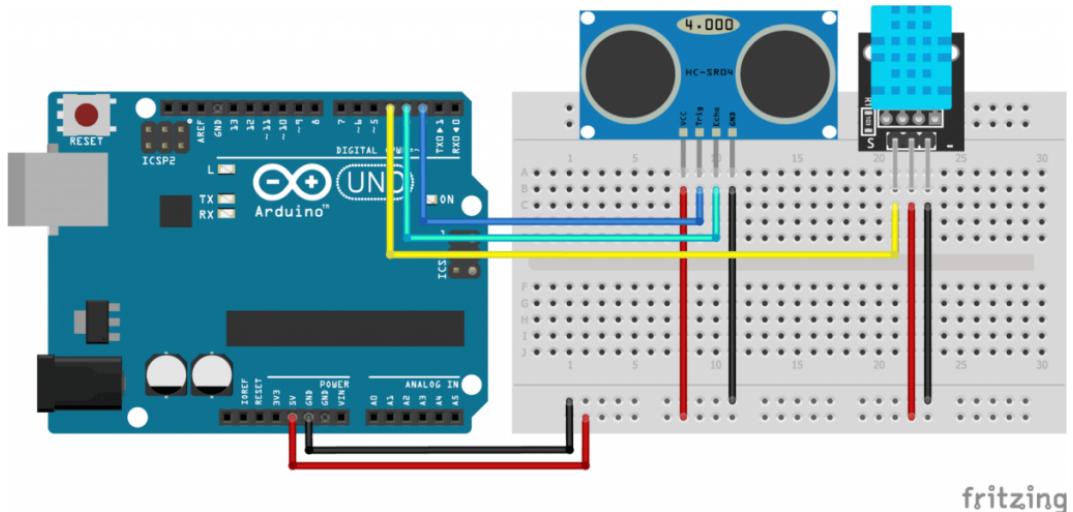


FIGURE 4.10 – Schéma général du module électronique.

4.3 DEVELOPPEMENT DU PROGRAMME ET CONCEPTION HMI SOUS LABVIEW

Pour cette partie software nous allons diviser le travail en deux grandes parties, une pour la programmation et conception HMI de la régulation de niveau, la deuxième partie sert au contrôle de la régulation de la température. Nous avons procédé aux tests de chacun des composants électroniques du système (capteur à ultrason, positionneur et capteur dht11) séparément pour vérifier leur bon fonctionnement et leur étalonnage via la bibliothèque LabVIEW LINX. Le Toolkit LabVIEW LINX ajoute la prise en charge des plates-formes embarquées Arduino.

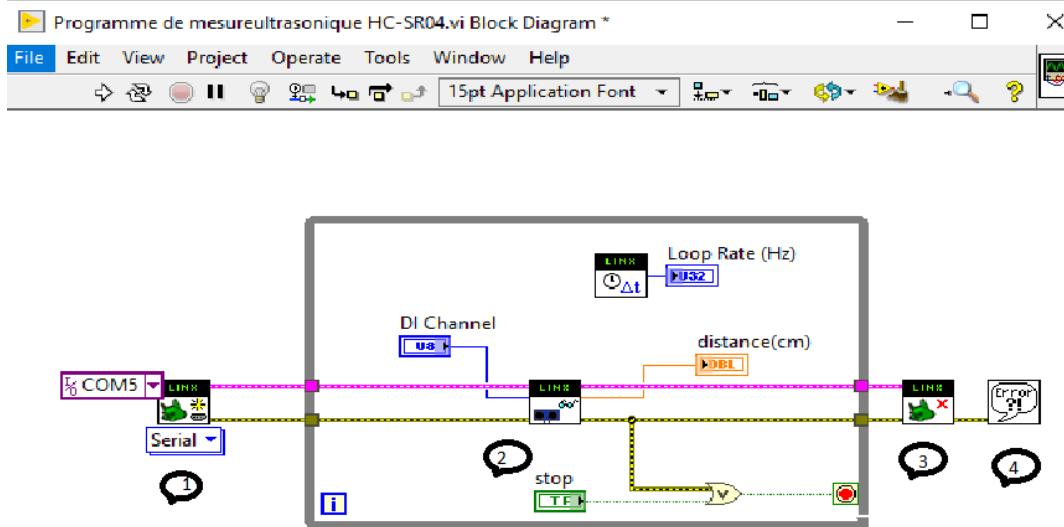


FIGURE 4.11 – Programme de mesure et étalonnage ultrasonique HC-SR04.

- 1) Ouvrez une connexion à l'appareil LINX.
- 2) Lire la valeur du capteur à ultrasons connecté au canal d'entrée numérique spécifié.
- 3) Fermé la connexion à l'appareil LINX.
- 4) Gérer l'erreur.

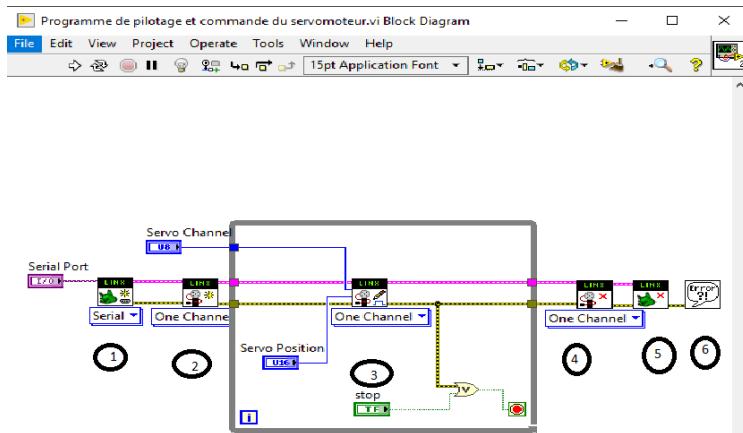


FIGURE 4.12 – Programme de pilotage et commande du positionneur.

- 1) Ouvrez une connexion à l'appareil LINX.
- 2) Ouvrir le canal servomoteur spécifique. Notes bien que le canal ne peut pas être utilisé comme DIO tant que le servo n'pas été fermé.
- 3) Écrivez une largeur d'impulsion sur le servo pour définir le positon.

- 4) Fermer le canal Servomoteur.
- 5) Fermé la connexion à l'appareil LINX.
- 6) Gérer l'erreur.

Après avoir confirmé que tous les éléments du système fonctionnaient correctement, j'ai regroupé tout le code en un seul VI, retesté les connexions du système et effectué un contrôle manuel des vannes.

Une fois que j'ai eu le contrôle manuel, j'ai programmé un VI complet qui pouvait enregistrer le comportement naturel de ce type de système.

Le résultat final de notre HMI de supervision suivant :

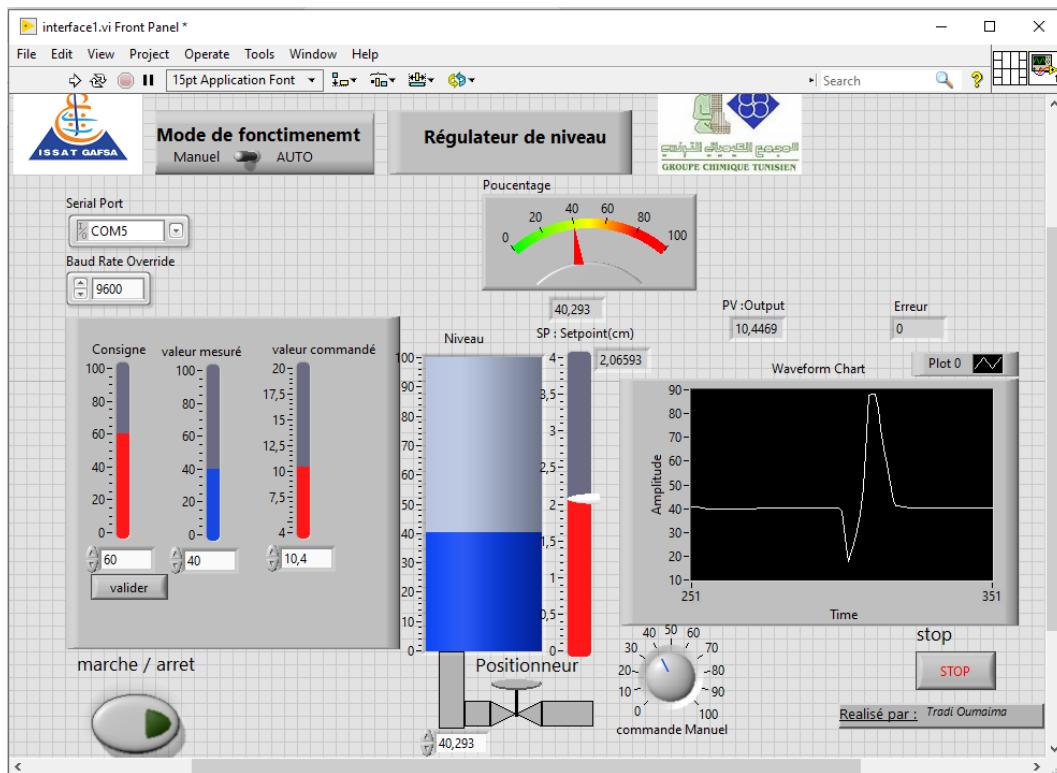


FIGURE 4.13 – Synoptique HMI de commande et régulation de niveau.

4.3.1 Teste, réglage des paramètres

Après avoir préparé le programme et le modèle, procéder au raccordement de tous les appareils (PC, modules électroniques et leurs périphériques), mettre en place la partie communication logicielle et régler les paramètres de fonctionnement selon les consignes du mode de fonctionnement. :

- L'utilisateur doit sélectionner le port COM (COM5) du PC et sa vitesse (9600) figurent sur l'HMI.

- Introduit la consigne SP désiré en cm.

- Sélection au premier lieu le mode de fonctionnement en manuel.
- Saisir les paramètres de départ.

Après avoir démarré l'IHM en mode manuel, observez comment les servos se déplacent et comment les capteurs HC-SR04 communiquent et mesurent la distance.

4.3.2 Programme et résultats de l'application de régulation de température

Cette partie de l'application utilise le capteur DHT11 pour mesurer la température ambiante, l'humidité, et acquiert, traite et affiche les valeurs et les données reçues de ce capteur en temps réel via la bibliothèque d'entrées analogiques Labview LINX.

Pour exécuter la régulation manuel (bibliothèque LINX dans Labview), collectez des mesures, des points de consigne et des paramètres de réglage pour exécuter le contrôle et conserver les valeurs souhaitées. Ce dernier envoie un signal de commande et délivre une valeur proportionnelle à l'actionneur. La source de chaleur attend le point de consigne.

Après avoir l'HMI de supervision de la température, nous avons procédure à faire des tests en mode automatique et voici les résultats suivants :

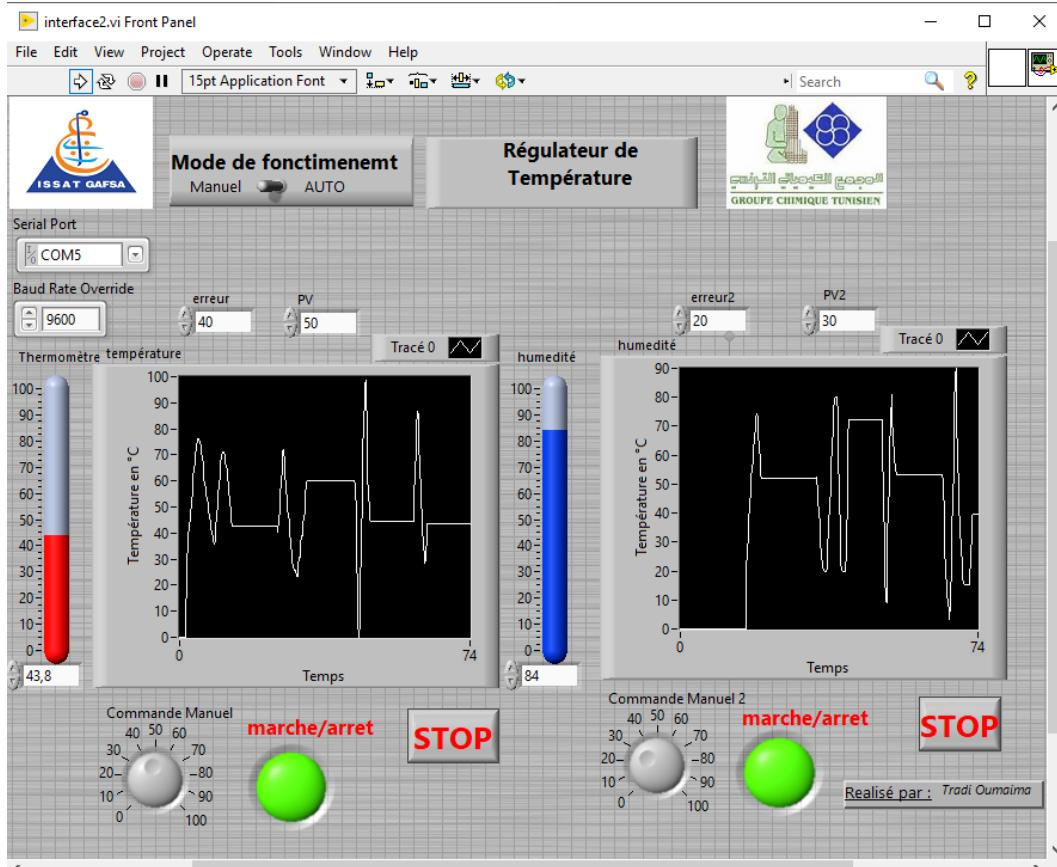


FIGURE 4.14 – Synoptique HMI et résultats de régulation de la température.

4.4 CONCLUSION

Dans ce chapitre, nous avons étudié les parties soft des modules électroniques utilisés pour le contrôle de la température et du niveau de liquide. Ce module électronique est constitué d'une carte Arduino UNO qui réalise l'acquisition, la conversion analogique-numérique et la gestion des périphériques (capteurs, actionneurs, etc.). Pour contrôler le système, nous avons utilisé le logiciel LabVIEW pour créer une interface graphique IHM. Cela permet de suivre et de contrôler en temps réel l'avancement des processus industriels. Cependant, Nous avons testé au préalable. Ce test nous a permis de vérifier le bon fonctionnement et la fiabilité de la carte électronique avant de la connecter au port PC. Le régulateur sert à la régulation de niveau et complète également la régulation de température.

CONCLUSION GÉNÉRALE

Le but de ce mémoire est de se concentrer sur la mise en œuvre des stratégies de contrôle les plus couramment utilisées dans l'industrie. Dans ce travail, nous présentons nos recherches et la mise en œuvre expérimentale de cette commande numérique. Cette commande numérique permet de contrôler la température ambiante et niveau d'un TK. La commande est une commande de station PC avec le logiciel LabVIEW et une carte basée sur Arduino (microcontrôleur). Les chapitres 2 et 3 consistent en des présentations détaillées du matériel et des logiciels de programmation utilisés dans cette pratique. La dernière partie consiste d'abord à rechercher et réaliser un module de contrôle électronique basé sur la carte Arduino UNO. Ce module sera une jonction entre la station PC-HMI et le processus via un port COM série. Le second est l'implémentation et le développement de logiciels de type HMI (Human Machine Interface) par LabVIEW installés sur PC. Cela permet un contrôle visuel et une surveillance en temps réel de tous les paramètres du processus. Tous les matériels et logiciels sont mis en service et testés étape par étape dans les modes de fonctionnement manuel et automatique pour garantir un fonctionnement et une fiabilité corrects avant d'être mis à la disposition de l'utilisateur. Enfin, nous avons programmer l'Arduino en tant que carte d'acquisition de données et une bonne communication avec le logiciel LabVIEW. De plus, ces deux technologies ont l'avantage de permettre le développement et l'optimisation de tout procédé industriel ou mécatronique. Plusieurs propositions seront étudiées pour un développement futur. Je peux suggérer de développer davantage ce point :

- Créez un fichier de stockage de données Excel et un tampon d'alarme pour aider les techniciens à diagnostiquer les problèmes.
- En connectant le module au réseau à l'aide du protocole de communication Ethernet TCP/IP, le problème de distance entre le PC de contrôle et le module est éliminé. Cela vous permettra de déployer ce module sur votre système de serveur OPC à l'aide de l'API.

BIBLIOGRAPHIE

- [1] : Régulation industrielle –Notion de Base Article PDF.
- [2] : Prouvost, P. (2005). Automatique : contrôle et régulation, Dunod.
- [3] <http://blog.acversailles.fr/technopeguy/public/Programmation/Arduino.pdf> mai 2017.
- [4] <https://www.flossmanualsfr.net/-booki/arduino/arduino.pdf> Mai 2017.
- [5] [www.cours-gratuit.com–Course LabView-id5006.pdf.](http://www.cours-gratuit.com-Course LabView-id5006.pdf)
- [6] [www.cours-gratuit.com–Course LabView-id4491.pdf.](http://www.cours-gratuit.com–Course LabView-id4491.pdf)
- [7] [www.cours-gratuit.com–Course LabView-id5009.pdf.](http://www.cours-gratuit.com–Course LabView-id5009.pdf)
- [8] [www.cours-gratuit.com–Course LabView-id5007.pdf.](http://www.cours-gratuit.com–Course LabView-id5007.pdf)
- [9] Guide d’utilisateur du logiciel Labview.
- [10] [https://www.labviewmakerhub.com/doku.php?id=get-labview.](https://www.labviewmakerhub.com/doku.php?id=get-labview)

ANNEXE

Principe de la mesure par ultrason

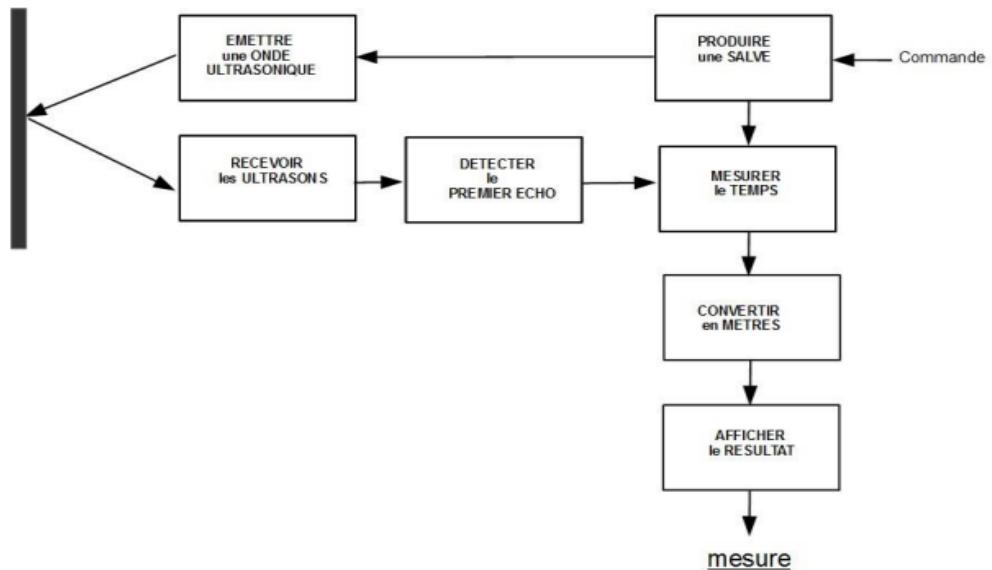
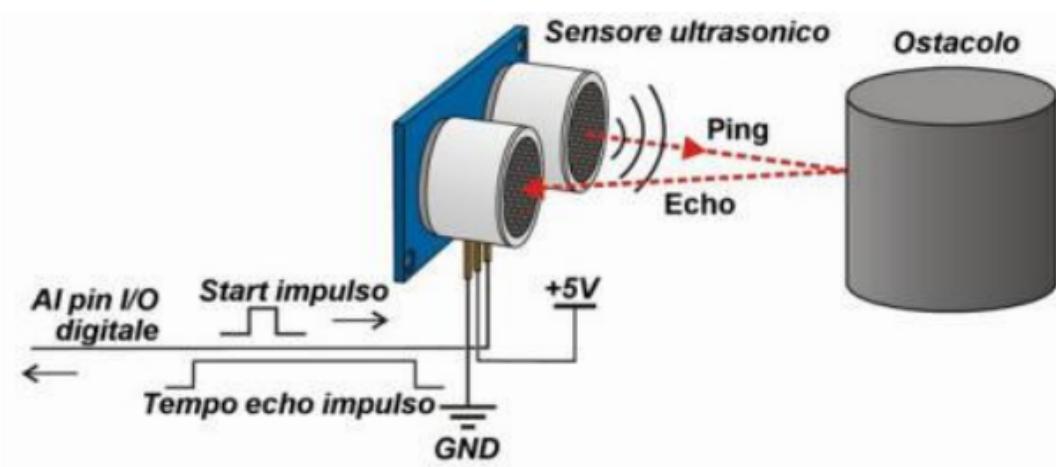


FIGURE 4.15 – Principe de la mesure par ultrason.



الملخص

يهدف مشروع الأطروحة هذا إلى تنفيذ وحدة تحكم رقمية ، ووحدة تحكم صناعية ضبي تعتمد على لوحة اردوينو للتحكم والتحكم في مستوى الخزان وتنظيم درجة الحرارة المحيطة. سيتم نقل البيانات وتبادلها (عرض القياس ، والأمر ، ونقل نقطة الضبط ، وما إلى ذلك) من خلال واجهة حاسوبية لوحدة التحكم المجهزة ببرنامج بيثون ، مما يجعل من الممكن عرض ومراقبة التقدم في الوقت الفعلي . العملية.

الكلمة المفتاحية : أردوينو ، إشراف ، بطيئاً ، تحكم ، قيادة ، حاسوبية ، تنظيم ، المجمع الكيميائي التونسي.

Résumé

Ce projet de mémoire vise à implémenter un régulateur numérique, un régulateur industriel à fabriquer basé sur une carte Arduino pour la commande de le contrôle de la régulation du niveau du réservoir et de la température ambiante. La transmission et l'échange de données (affichage de mesure, commande, transmission de consigne, etc.) se feront par l'interface IHM du poste de contrôle équipé du logiciel Labview, qui permet d'afficher et de suivre en temps réel l'avancement du procédé.

Mot clé : Arduino , supervision , LabVIEW , contrôle , commande , HMI , régulation , Groupe Chimique Tunisien .

Abstract

This dissertation project aims to implement a digital controller, a DIY industrial controller based on an Arduino board for command and control of tank level and ambient temperature regulation. .The transmission and exchange of data (measurement display, command, setpoint transmission, etc.) will be done by the HMI interface of the control station equipped with the Labview software, which makes it possible to display and monitor in real time the progress of the process.

Key word: Arduino, supervision, LabVIEW, control, command, HMI, regulation, Tunisian Chemical Group.