

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



Université MOULOUD MAMMERRI de Tizi-Ouzou

Faculté de Génie Électrique et informatique

Département d'électronique

Mémoire de fin d'études

En vue de l'obtention du diplôme d'ingénieur d'état en Électronique

Option: Contrôle

Thème

Conception et réalisation
d'un système d'acquisition et de commande
d'un four de chimie avec une interface Labview

Propose et dirigé par:
Mr. LAGHROUCHE M.

Réalisé par:
TOUMI Toufik
GUERBAS Nazim

PROMOTION 2011 - 2012

Remerciements

Nous remercions tout d'abord dieu qui nous a donné la force et le courage pour terminer nos études et élaborer ce modeste travail.

Nos remerciements à notre encadreur Mr. LAGHROUCHE qui a bien voulu nous encadrer, et de nous avoir encouragé tout au long de notre travail.

Nos remerciements sont destinés également à tous nos enseignants pour l'effort fourni pendant la durée de nos études.

Nous remercions également le président du jury ainsi que les membres du jury qui nous ont honorés pour évaluer et juger notre modeste travail.

Table des matières

Cahier de Charge

Introduction générale

1. Description du système

Introduction	13
Schéma synoptique	13
I. Thermocouple :	14
I.1. Introduction	14
I.2. Capteur de température «Thermocouple».....	14
I.3. Principes de mesure	14
I.4. Types Thermocouple.....	17
I.5. Circuit de compensation de Thermocouple type S	17
II Bloc de commande :	18
II.1. Introduction	18
II.2. Microcontroller PIC16F876A.....	18
II.2.1 Les particularités électriques:	18
II.2.2 Brochage du PIC16F876A	18
II.2.4. Schémas fonctionnelle	19
II.2.3 Organisation du 16F876:.....	21
II.3 Convertisseur analogique numérique AD7524	22
II.3.1 Introduction	
II.3.2 Description générale	22
a- Rappels:.....	22
b- Différents types de convertisseurs numériques analogiques (CNA)	23
c- Schéma fonctionnel et Fonction de transfert d'un CNA	23
d- Caractéristiques principales des CNA	23
e- Principe de fonctionnement des CNA.	24
II.4 Branchement de convertisseur analogique numérique AD7524.....	25
III. Bloc d'acquisition	26
III.1 Introduction	26
III.2 Description de la carte Arduino	26
III.2.1 Caractéristique	26
III.2.2 Connexion de la carte Arduino au bloc de commande	27
• Principe du bus I2C	27
• La prise de contrôle du bus	27
• La transmission d'un octet.....	28
• La transmission d'une adresse	28
• Ecriture d'une donnée	28

IV. Gradateur de puissance	29
IV.1 Principe	29
IV.2. Les différents modes de fonctionnement	29
• Fonctionnement par train d'ondes ou mode Tout-ou-Rien	29
• Fonctionnement par train d'ondes syncopé ou mode TAKT	30
• Fonctionnement par train d'ondes syncopé rapide ou mode QTM	30
• Fonctionnement par angle de phase ou mode VAR	30
• Fonctionnement en mixte ou mode SSSD	30
IV.3. Régulateurs de puissance THYRITOP 30	31
1. Généralités	31
2. Caractéristiques générales	31
3. Circuit de commande	31
4. Circuit de puissance	31
V. Bloc d'alimentation	32
2. Interface graphique et codage	
Introduction:	36
I. Introduction à la programmation graphique	36
I.1. programmation flot de contrôle	36
I.2. Programmation flot de données	36
II. L'environnement LabVIEW	37
• Palette d'outils	37
• Palette de commandes	38
• Palette de fonctions	38
II.1. Structure de données dans LabVIEW	39
II.1.1. Le type tableau (structure de données homogènes)	39
II.1.2. Le type « cluster » (structure de données hétérogènes)	40
II.2 Structures de programmation	40
II.2.1. Structure de séquence	40
II.2.2. Les structures itératives	41
II.2.3. La structure de choix	43
II.3. Traitement numérique	43
II.3.1 Les fonctions prédéfinies	43
II.3.2. Boîtes à outils mathématique	44
II.4. Bibliothèques de commande	44
II.4.1. Bibliothèques « Analog sampling »	45
II.4.2. Bibliothèques « I2C »	45

III. Interface & programme graphique	46
III.1. Description de schéma-bloc général	46
III.2. Programme graphique	47
III.3. Interface graphique.....	48
IV. Programme du micro-contrôler PIC16F86A	49
IV.1. Présentation de MikroC Pro for pic®.....	49
IV.1.1 Interface principale me MikroC	49
IV.2. Organigramme	51
IV.3. Programme en C	52
IV.3.1 Programmation et tests	53
3. Réalisation pratique et mise en œuvre.....	
I. Circuit électrique et imprimé.....	57
I.1. Création du circuit électrique	57
I.2. Développement ds cartes	58
• Impression de PCB sur filme transparent	58
• L'insolation.	58
• La révélation.....	58
• La gravure.....	59
• L'implantation des composants:	59
• Circuit d'alimentation	59
• Circuit d'alimentation	60
II. Mise en œuvre du systeme au laboratoire.....	61
Conclusion générale	
Bibliographie	

Cahier de Charge

Dans le cadre de notre projet de fin d'études, il nous a été demandé de concevoir et réaliser un système d'acquisition et de contrôle de température d'un four de chimie qui se trouve au département de recherche doctoral (Laboratoire de Physique et Chimie des Matériaux) qui se trouve au sein de notre université .

Ce système doit incrémenter la température du four de une degré chaque minute ou chaque dix minutes selon la configuration de l'utilisateur pour une durée bien déterminé par ce dernier, ce four sert à l'analyse de la réaction des matériaux en fonction de la température . Tous ce système sera commandé par un ordinateur qui sera relié au four via une liaison USB entre le PC et la carte de commande du four à concevoir.

Pour réaliser le système nous allons utilisé les éléments suivant qui sont avec le four :

- Un capteur de température de type thermocouple type S
- Une résistance qui fournis une température de 1400° C
- Un variateur analogique monophasé

Introduction générale

De nos jours le monde change et évolue progressivement, de nouvelles créations apparaissent tous les jours, les technologies se développent sans cesse et les systèmes d'acquisition sont très utilisés dans plusieurs domaines.

Cette rapide évolution qui a touché tous les domaines n'a pas épargné le secteur de l'électronique qui à son tour connaît des changements profonds qui les mènent vers une meilleure perfection en permettant de faciliter beaucoup de tâches qui étaient difficiles auparavant.

Parmi les contraintes rencontrées dans ce domaine, on trouve les difficultés relatives à la mesure et l'acquisition des paramètres suivants : la vitesse, la température...etc.

En effet la température tout comme la pression, la vitesse ou le débit, intervient en tant que variable, dans les phénomènes liés aux fluides liquides ou gazeux, de même que les corps solides exerçant ou subissant des variations de température.

De ce fait, la température est un paramètre important dans des disciplines aussi variées que la thermodynamique, l'acoustique, la mécanique des fluides, la mécanique des solsEtc.

Toutes ces activités font appel à des chaînes instrumentales dont les capteurs de température sont les premiers maillons, ceux-ci délivrent les données relatives aux gaz, à la vapeur, à l'eau ou d'autres éléments déterminant le bon fonctionnement des machines mécaniques ou systèmes conditionnant le déroulement d'un processus.

Notre présent travail consiste à concevoir et réaliser un système de commande d'un four de chimie afin de suivre l'évolution et le comportement des matériaux pendant l'expérimentation.

Chapitre 1

Présentation du système

Introduction:

Ce chapitre porte sur la conception que nous avons proposé en tenant compte des contraintes imposées par le cahier de charges.

Schéma synoptique

Comme le montre la figure 1 le schéma synoptique du système est constitué de 6 partie ,

- Un thermocouple Type S
- Une partie de commande
- Une partie d'acquisition
- Un Gradateur
- Une alimentation
- Une Interface Labview

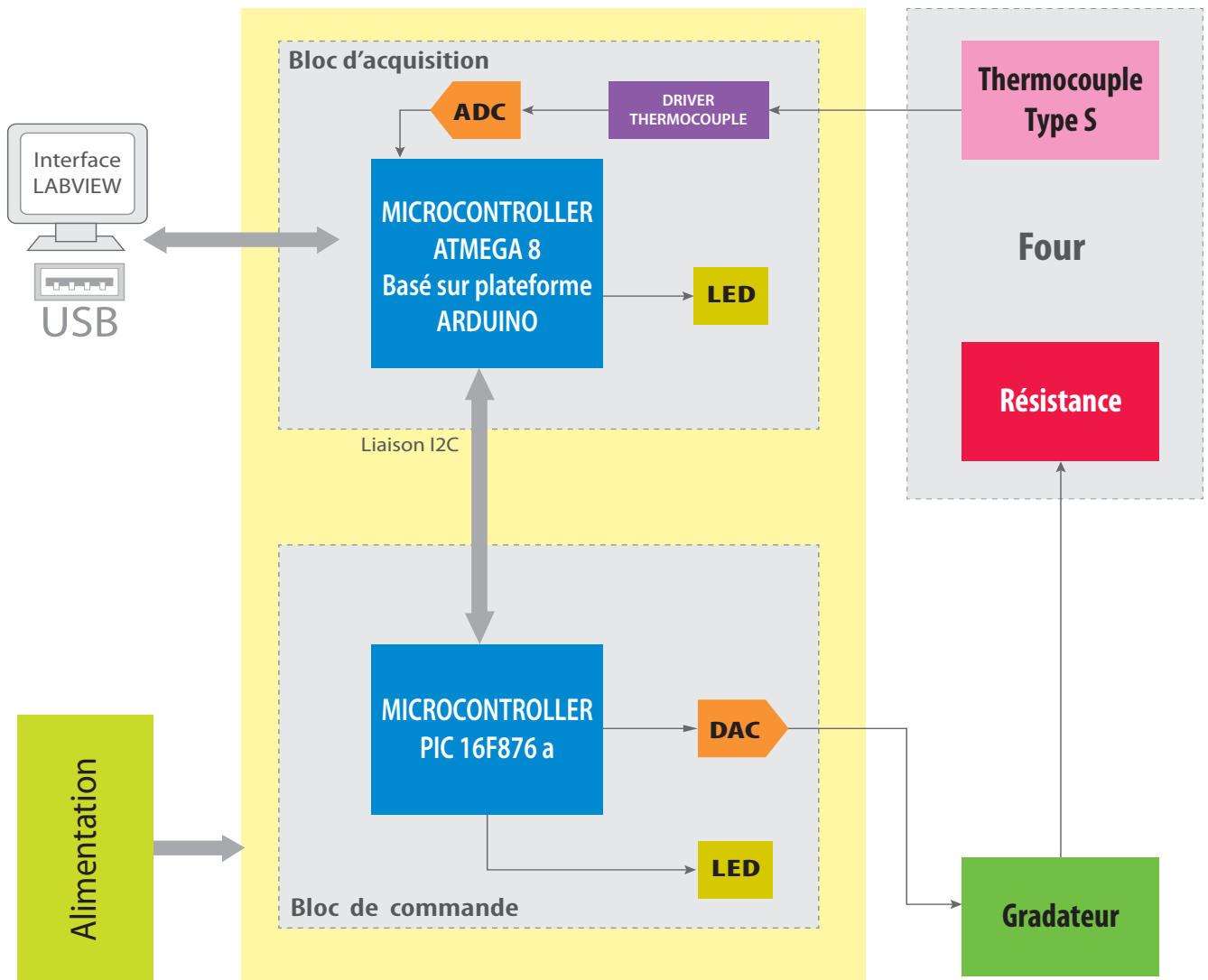


Figure 1: schéma synoptique

I. Thermocouple :

I.1. Introduction

Le Thermocouple type S et le capteur de température utilisé dans notre conception, ce capteur est l'élément essentiel pour l'acquisition.

I.2. Capteur de température « Thermocouple »

Les thermocouples ou couples thermoélectriques (CTE) sont, en physique, des couples de matériaux dont l'effet Seebeck est utilisé pour la mesure de température. Ils sont bon marché et permettent la mesure dans une grande gamme de températures. Leur principal défaut est leur imprécision : il est relativement difficile d'obtenir des mesures avec une erreur inférieure à $0,1\text{ }^{\circ}\text{C}$ - $0,2\text{ }^{\circ}\text{C}$.

I.3. Principes de mesure

Maintenant que nous savons qu'un thermocouple génère une tension dont la valeur est fonction de la température et du coefficient de Seebeck (α) de la jonction des deux métaux dissemblables, il ne reste plus qu'à mesurer celle-ci à l'aide d'un voltmètre puis d'exprimer, par calcul, la tension mesurée en température.

• Connexion sur un voltmètre

Connectons un thermocouple Platine-Rhodium (Type S) sur les bornes d'un voltmètre et, après calcul la sensibilité = $5\mu\text{V}/^{\circ}\text{C}$, nous trouvons une valeur de température qui n'a rien à voir avec l'ambiance dans laquelle se trouve le thermocouple.

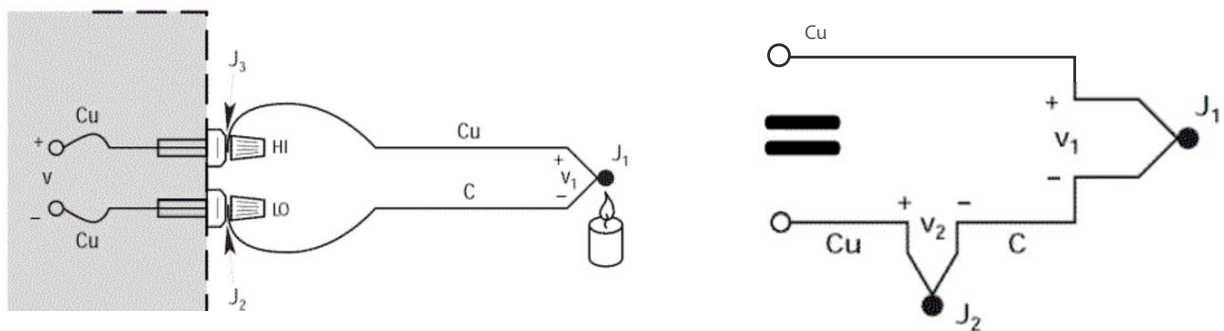


Figure 2 : Connexion d'un Thermocouple sur un voltmètre

En connectant le thermocouple Platine-Rhodium sur les bornes en cuivre du voltmètre, nous avons créé deux nouvelles jonctions métalliques : J, jonction cuivre sur cuivre qui ne crée pas de tension thermoélectrique et J qui, étant constituée de deux métaux différents (Cuivre/Constantan) génère une tension thermoélectrique (V) qui vient en opposition avec la tension V que nous voulions mesurer.

En fin de compte, en se référant au schéma équivalent, la tension résultante mesurée par le voltmètre est égale à $V_1 - V_2$, c'est-à-dire qu'elle est proportionnelle à la différence de température entre J_1 et J_2 .

• **Référence de la jonction externe**

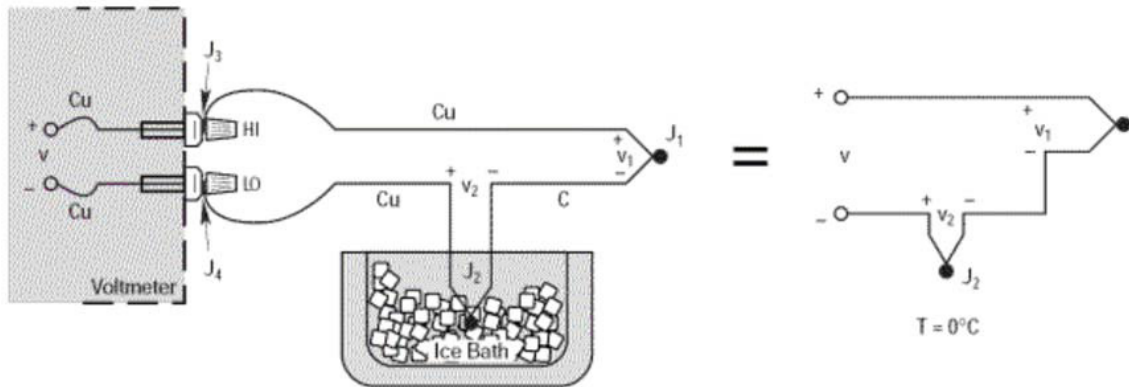


Figure 3 : Connexion d'un Thermocouple sur un voltmètre avec jonction froide

Une manière simple de déterminer exactement et facilement la température de la jonction J est de la plonger dans un bain de glace fondante, ce qui force sa température à 0°C (273,15 K). On pourra alors considérer J comme étant la jonction de référence. Le schéma à donc maintenant une référence 0°C sur J.

La lecture du voltmètre devient: $V = (V_1 - V_2)$ équivalent à $\alpha (t_J - t_J)$.

Écrivons la formule avec des degrés Celsius : $T_J (°C) + 273,15 = t_J (K)$.

Et la tension V devient : $V = V_1 - V_2 = \alpha [(t_J + 273,15) - (t_J + 273,15)] = \alpha (T_J - T_J) = \alpha (T_J - 0) = \alpha T_J$

Nous avons utilisé ce raisonnement pour souligner que la tension V de la jonction J, dans le bain de glace, n'est pas zéro volt. C'est une fonction de la température absolue. Cette méthode est très précise car la température 0°C (Point de fusion de la glace), peut être facilement et précisément contrôlée.

Le point de fusion de la glace est utilisé par les organismes de normalisation comme point de référence fondamental pour leurs tables de tension de thermocouple. Ainsi, à la lecture de ces tables, nous pourrions convertir directement la tension V en température T_J .

• **Utilisation d'un circuit de compensation**

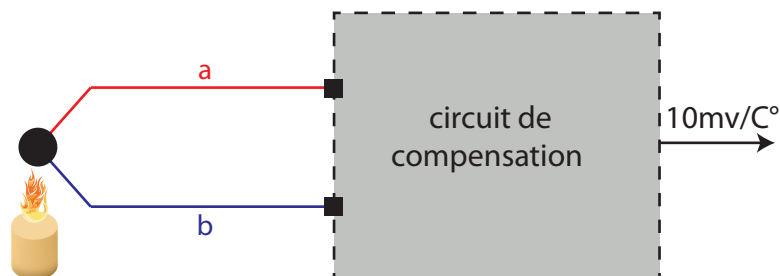


Figure 4 :synoptique de circuit de compensation de Thermocouple

Cette méthode consiste simplement à brancher le thermocouple à l'entrée du circuit de compensation adéquat au type de thermocouple avec jonction de référence à 0°C et en sortie on trouve une tension qui correspond à 10mV/C°.

• Convertir la tension en température

Maintenant que nous avons utilisé soit la compensation logicielle, soit la compensation matérielle, afin d'obtenir une jonction de référence à 0°C, nous devons convertir la tension V mesurée en température.

Malheureusement, les relations entre tension et température des thermocouples ne sont pas linéaires.

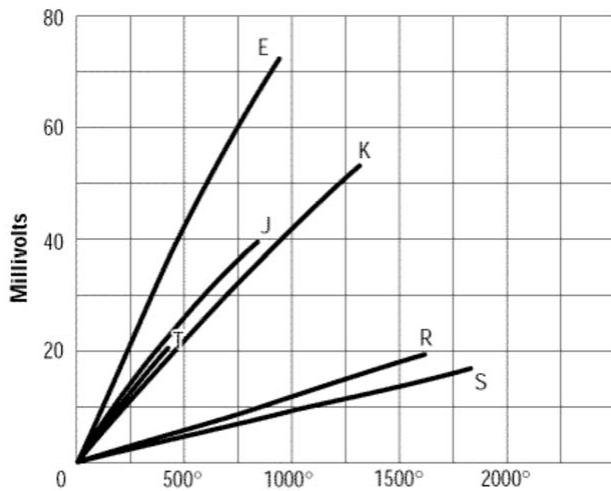


Figure 5: Tension des thermocouples en fonction de la température

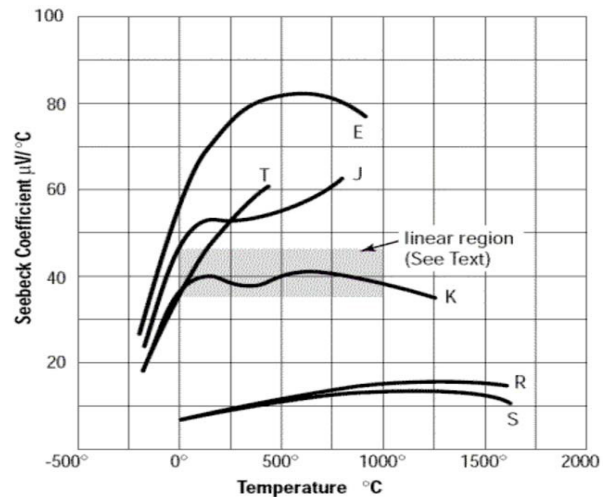


Figure 6: Coefficient de Seebeck (α) en fonction de la température

• Calcul à partir des tables

Après avoir lu la valeur de la tension V , par exemple 8,35687 mV, avec un thermocouple de type K (Chromel/Alumel), regardons dans la table ITS-90 :

°C	0	1	2	3	4	5	6	7	8	9	10
190	7.739	7.779	7.819	7.859	7.899	7.939	7.979	8.019	8.059	8.099	8.138
200	8.138	8.178	8.218	8.258	8.298	8.338	8.378	8.418	8.458	8.499	8.539
210	8.539	8.579	8.619	8.659	8.699	8.739	8.779	8.819	8.860	8.900	8.940

Nous pouvons voir que cette valeur est située entre T 205 °C (8,338 mV) et T 206 °C (8,378 mV)

Effectuons un calcul par interpolation entre les valeurs 205 et 206 °C:

$$8,35687 - 8,338 = 0,01887 \text{ mV (Reliquat de tension au dessus de 205 °C)}$$

$$8,378 - 8,338 = 0,040 \text{ mV pour une différence de 1 °C}$$

$$0,01887 / 0,040 = 0,471 \text{ °C en plus}$$

La température est donc de $205 + 0,471 = 205,471 \text{ °C}$

L'équation est :

$$= 205 + [(8,35687 - 8,338) / (8,378 - 8,338)] = 205,471 \text{ °C}$$

$$T^{\circ}\text{C} = T^{\circ}\text{C}_{inf} + \frac{V - V_{inf}}{V_{sup} - V_{inf}} \dots (1)$$

I.4 Type de Thermocouple

Il existe différents types de thermocouples, correspondant chacun à une gamme de température ou à une plus ou moins grande précision. Leurs propriétés peuvent également dépendre du type d'isolation (gaine) utilisée pour les fils métalliques.

Dans notre projet de fin d'étude on a utilisé un thermocouple de type S qui a les caractéristiques suivante:

- Composition : Platine-Rhodium (10 %) / Platine
- Adapté aux températures élevées (0 à 1 650 °C).
- Couleurs selon CEI 584-3 (+ / -) : Orange / Blanc
- Couleurs selon NF C 42-323 1985 (+ / -) : Jaune / Vert
- Résolution de $5\mu\text{V}/^\circ\text{C}$

I.5. Circuit de compensation de Thermocouple type S

On a utilisé le circuit LTK001 qui est un amplificateur de thermocouple fourni avec un compensateur de jonction froide. En séparant la fonction d'amplification et de compensation, le problème d'augmentation de la température de compensation est pratiquement éliminé.

La sortie est un signal linéaire $10\text{mV}/^\circ\text{C}$ qui est injecté à l'entrée d'un diviseur de tension de 3 résistances de $1.00\text{K}\Omega$ en série pour faire une résistance de $3.00\text{K}\Omega$ et une résistance de $1.00\text{K}\Omega$ pour avoir $1/4$ de tension d'entrée = $2.5\text{mV}/^\circ\text{C}$ qui sera ensuite injecté à l'entrée analogique de l'Arduino.

$$U_s = U_e \cdot (R_5 / (R_6 + R_5)) \rightarrow U_s = 10\text{mV} / (1 + 3) \Rightarrow U_s = 1/4 U_e = 2.5\text{mV}/^\circ\text{C} \dots (2)$$

On a 11Bit de résolution de l'entrée analogique à 5V donc 2048 échantillon de $\sim 2.5\text{mV}/\text{échantillon}$
 $5\text{V} / 2^{11\text{bits}} = 2048 \dots (3)$

Avec cette résolution on peut atteindre une température de 2048 $^\circ\text{C}$, donc on satisfait largement notre cahier de charge .

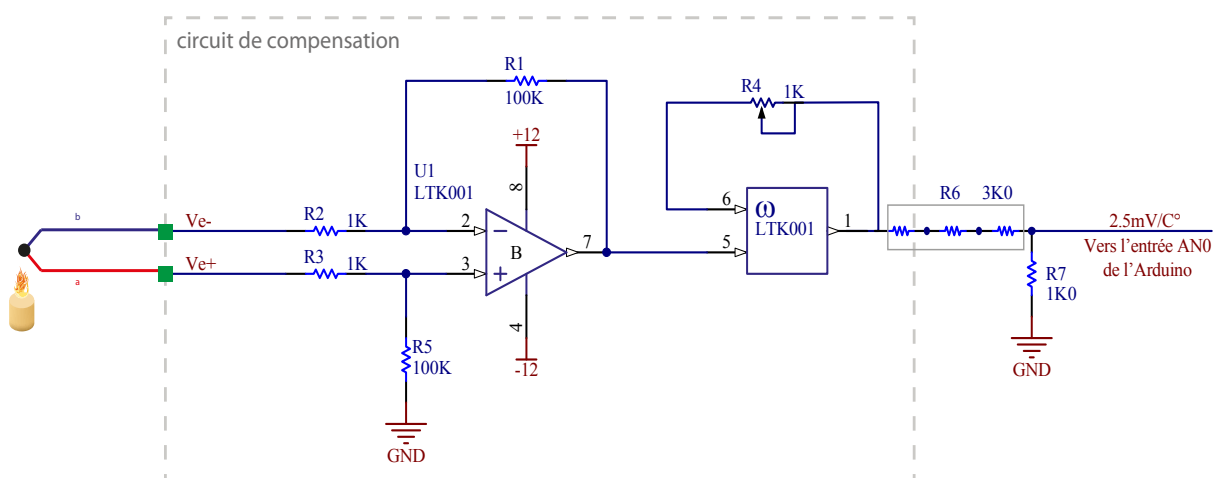


Figure 7 Circuit de compensation de Thermocouple type S

II Bloc de commande :

II.1. Introduction

Cette partie de commande est composé d'un microcontrôleur PIC16F876A d'un circuit d'horloge, d'une alimentation, d'un convertisseur analogique numérique DAC et des leds d'indication.

II.2. Microcontroller PIC16F876A

Le PIC16F876A est un micro-contrôleur de MICROCHIP®, fait partie intégrante de la famille des Mid-Range (16) dont la mémoire programme est de type flash (F) de type 876 et capable d'accepter une fréquence d'horloge maximale de 20Mhz

II.2.1 Les particularités électriques:

On constate que sur le schéma de la figure 7 deux connexions « VSS » qui sont reliées à la masse. En fait, en interne, ces pins sont inter-connectés. La présence de ces 2 pins s'explique pour une raison de dissipation thermique.

Les courants véhiculés dans le Pic sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles. Le constructeur a donc décidé de répartir les courants en plaçant deux pins pour l'alimentation VSS, bien évidemment, pour les mêmes raisons (dissipation thermique), ces pins sont situées de part et d'autre du PIC, et en positions relativement centrales, et des pins entrée/sortie (PORTA, PORTB, PORTC)

II.2.2 Brochage du PIC16F876A

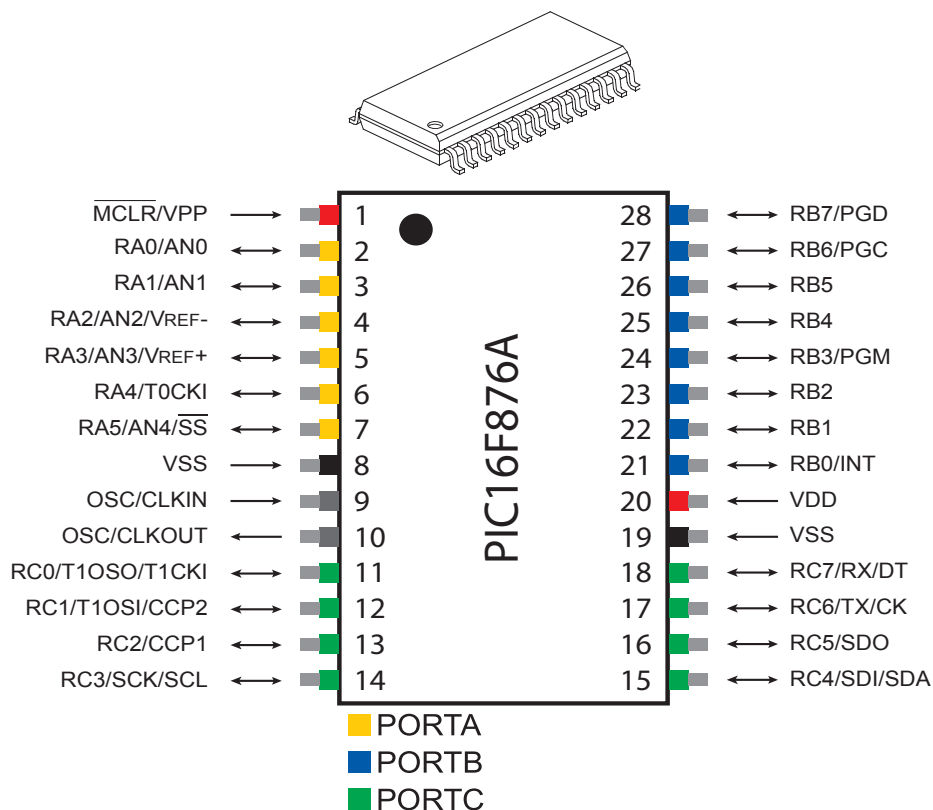


Figure7: Brochage du PIC16F876A

II.2.3 Organisation du 16F876:

La mémoire du 16F876 est divisée en 3 parties. Pour mieux le connaître regardons de près son Schéma fonctionnel (*Figure 8*).

- **La mémoire programme:**

La mémoire programme est constituée de 8K mots de 14 bits. C'est dans cette zone que nous allons écrire notre programme.

- **La mémoire eeprom:**

La mémoire eeprom (Electrical Erasable Programmable Read Only Memory), est constituée de 256 octets que nous pouvons lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents.

- **La mémoire RAM et organisation:**

La mémoire RAM est celle que nous allons sans cesse utiliser. Toutes les données qui y sont stockées sont perdues lors d'une coupure de courant.

La mémoire RAM disponible du PIC16F876A est de 368 octets. Elle est répartie de la manière suivante:

- 1) 80 octets en banque 0, adresses 0x20 à 0x6F
- 2) 80 octets en banque 1, adresses 0xA0 à 0XEF
- 3) 96 octets en banque 2, adresses 0x110 à 0x16F
- 4) 96 octets en banque 3, adresses 0x190 à 0x1EF
- 5) 16 octets communs aux 4 banques, soit 0x70 à 0x7F = 0xF0 à 0xFF
0x17F=0x1F0 à 0x1FF.

- **Watchdog:**

Sous ce nom étrange nous allons découvrir une fonction capable de surveiller le bon fonctionnement du programme que le micro contrôleur exécute. Le rôle du Watchdog (ou chien de garde) est de «reseter» le micro contrôleur si l'on ne remet pas à zéro périodiquement (à intervalle définissable) un registre interne grâce à l'instruction clrwdt(clear watchdog), si le programme tourne par exemple dans une boucle sans fin (bug) , il ne peut remettre à 0 le chien de garde et ainsi le micro contrôleur se réinitialise afin de relancer le programme.

- **Le TIMER:**

Un timer est un registre interne au micro contrôleur, celui-ci s'incrémente au grès d'une horloge, ce registre peut servir par exemple pour réaliser des temporisations, ou bien encore pour faire du comptage (par l'intermédiaire d'une broche spécifique : RA4/TOKI).

Le PIC 16F876 A possède trois timers sur 8 bits (il compte jusqu'à 256) configurable par logiciel.

• Les ports entrée/sortie:

On dispose de 22 broches d'entrées/sorties, chacune configurables soit en entrée soit en sortie (PORTA, PORTB, PORTC). Un registre interne au PIC, nommé TRIS, permet de définir le sens de chaque broche d'un port d'entrées/sorties. En règle générale, un bit positionné à « 0 » dans le registre TRIS donnera une configuration en sortie pour la broche concernée ; si ce bit est positionné à « 1 », ce sera une broche d'entrée.

• Particularité du port A:

Le 16F876 dispose de 5 canaux d'entrée analogique. Nous pouvons donc échantillonner successivement jusqu'à 5 signaux différents avec ce composant. Les pins utilisés sont les pins AN0 à AN4 (qui sont en fait les dénominations analogiques des pins RA0 à RA3 + RA5).

On peut noter également que les pins ANx sont des pins d'entrée. Il n'est donc pas question d'espérer leur faire sortir une tension analogique. Ceci nécessiterait un convertisseur numérique/analogique dont n'est pas pourvu notre PIC.

• Particularités du port B:

Hors de sa fonction principale autant que ports d'entrées /sorties, on note la pin RB0 qui, en configuration d'entrée, est de type « trigger de Schmitt » quand elle est utilisée en mode interruption « INT », La lecture simple de RB0 se fait, elle, de façon tout à fait classique, en entrée de type TTL.

Encore il y a (RB3-RB6-RB7) qui peuvent servir dans la programmation (en mode LVP) en cas d'absence de programmeur commercial.

• Particularités du port C:

C'est un port tout ce qu'il y a de plus classique, Or qu'il a deux pins qu'on utilisera plus tard dans la communication série avec le PC à travers (TX et RX) (pin17 et 18).

• L'oscillateur:

L'horloge système peut être réalisée soit avec un quartz (a), soit avec une horloge extérieure (b), soit avec un circuit RC(c). Dans ce dernier cas, la stabilité du montage est limitée.

La fréquence maximale d'utilisation va dépendre du micro-contrôleur utilisé. Le suffixe indiqué sur le boîtier donne la nature de l'horloge à utiliser et sa fréquence maximale.

• MCLR:

La broche MCLR permet de réaliser un Reset du circuit quand elle est placée à 0V.

II.2.4. Schémas fonctionnelle

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F876	8K	368 Bytes	256 Bytes

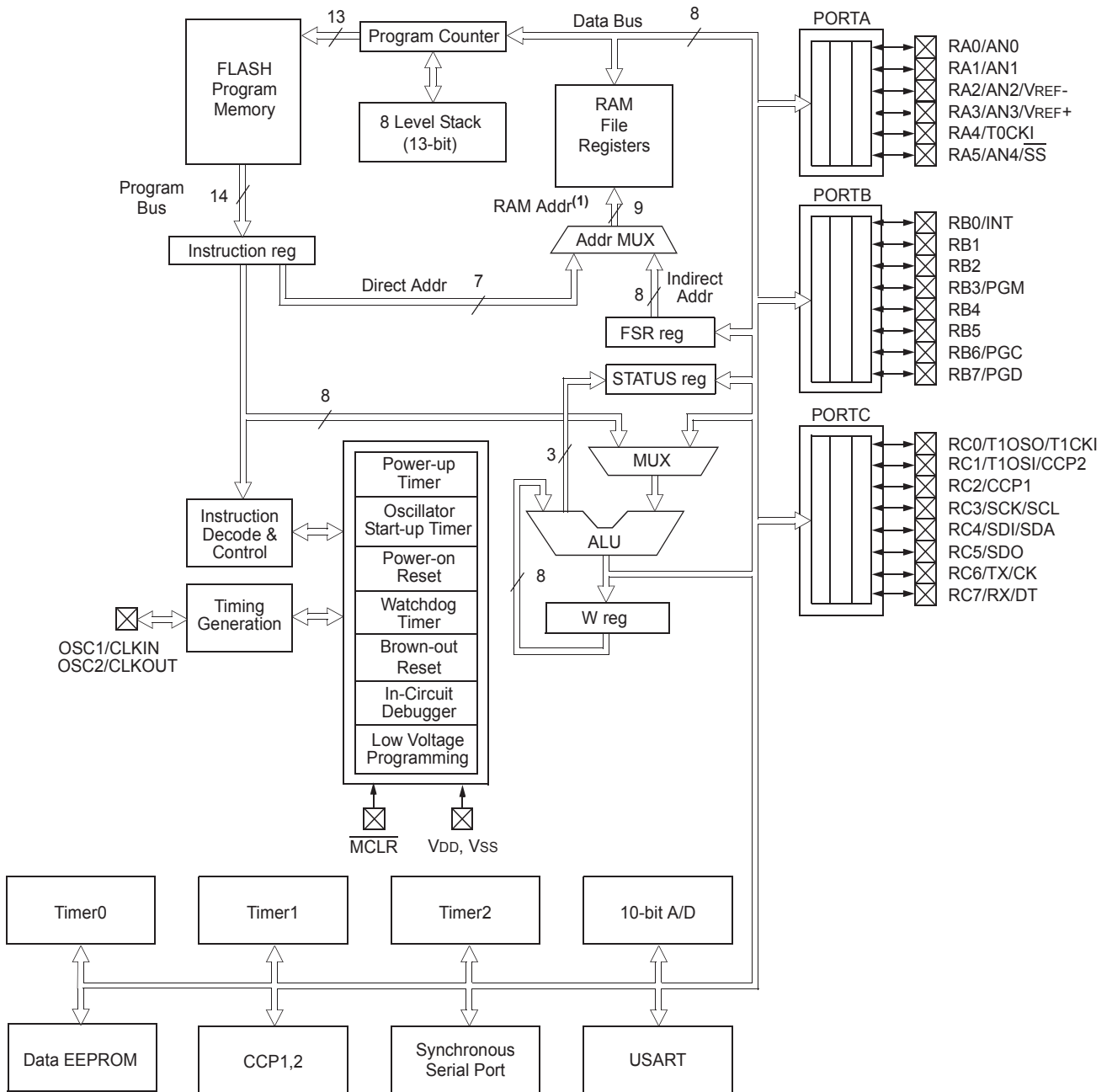


Figure8. Schéma fonctionnelle

II.3 Convertisseur analogique numérique AD7524

II.3.1 introduction

De nombreux systèmes électroniques utilisent la technique numérique, à base de microprocesseurs ou de microcontrôleurs pour les avantages qu'elle présente par rapport à la technique analogique : facilité de traitement de l'information selon des fonctions complexes (filtrage, compression...), mémorisation possible des informations, faible sensibilité au bruit, Lorsque les informations issues de capteurs sont des grandeurs analogiques ou que les actionneurs doivent être commandés par des signaux analogiques, il est nécessaire de procéder à des conversions de données : Conversion analogique numérique (CAN) et numérique analogique (CNA).

La grandeur physique à convertir peut-être de nature quelconque : vitesse, température, force. Le capteur permet de convertir cette grandeur en un signal analogique (par ex : dynamo tachymétrique pour le captage de la vitesse ou jauge d'extensiométrie pour le captage d'une force).

Le convertisseur analogique numérique (CAN) va convertir le signal analogique en une suite de mots numériques qui pourront être compris et traités par le calculateur (microprocesseur).

De même, le calculateur pourra générer en entrée du CNA des mots numériques qui seront convertis en un signal analogique par le CNA (convertisseur Numérique Analogique), et dans notre cas on vas utiliser se dernier pour convertir les donné numérique sortie du pic16F876A en une grandeur analogique pour contrôler le gradateur monophasé fournie par le laboratoire .

II.3.2 Description générale

a- Rappels:

Dans un système de numération de base B, une quantité Q quelconque se représente à l'aide d'une suite de symboles (B correspondra au nombre de valeurs distinctes que peuvent prendre chaque symbole).

$$a_{n-1}, a_{n-2}, \dots, a_i, \dots, a_2, a_1, a_0 \quad \dots (3)$$

Chaque symbole a_i est caractérisé par sa position, ou rang, précisé par l'indice i , celui ci définit automatiquement le poids B^i du symbole a_i . Ainsi la valeur Q est par définition:

$$Q = a_{n-1}.B^{n-1} + a_{n-2}.B^{n-2} + \dots + a_i.B^i + \dots + a_2.B^2 + a_1.B + a_0 \quad \dots (4)$$

La conversion numérique analogique, consiste à transformer une information disponible sous forme binaire en une information analogique. Soit N l'information numérique, et A la quantité analogique correspondante.

$$N = (a_{n-1}, a_{n-2}, \dots, a_i, \dots, a_2, a_1, a_0) \quad \dots (5)$$

$$A = q.(a_{n-1}.2^{n-1} + a_{n-2}.2^{n-2} + \dots + a_i.2^i + \dots + a_2.2^2 + a_1.2 + a_0) \quad \dots (6)$$

- q est la quantité analogique élémentaire (quantum de base), sa valeur est fonction dans le cas des convertisseurs, du nombre de bits et de l'excursion (pleine échelle appelé FSR= full scale range).
Ex: pour un convertisseur 8 bits, si $FSR=10V$, $q=10/(2^8 -1)=10/255=39mV$ (il y a 255 intervalles).
- a_{n-1} est le bit de poids le plus fort ou MSB (Most Significant Bit).
- a_0 est le bit de poids le plus faible ou LSB (Least Significant Bit).

b- Différents types de convertisseurs numériques analogiques (CNA)

Contrairement aux convertisseurs analogiques numériques, il existe peu de types différents de convertisseurs numériques analogiques. On distinguera les convertisseurs à réseaux R/2R et les convertisseurs à rapport cyclique variables très utilisés dans les micro-contrôleurs.

c- Schéma fonctionnel et Fonction de transfert d'un CNA

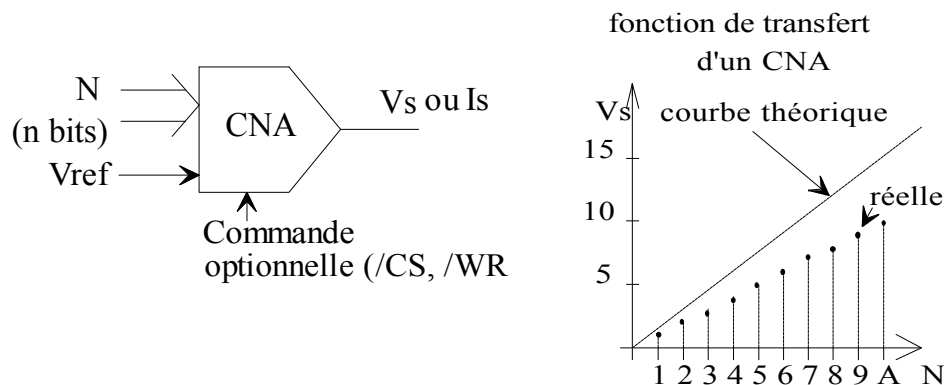


Figure9: Schéma fonctionnel et Fonction de transfert d'un CNA

d- Caractéristiques principales des CNA.

Pour les caractéristiques de CNA on distingue :

- Valeur du LSB ou de l'INCREMENT: Représente la variation de la tension de sortie lorsque l'on incrémente (augmente d'une unité) le nombre binaire d'entrée.(utilisé aussi pour les CAN)
- FSR: (full scale range) Dynamique de sortie = écart entre le minimum et le maximum de la tension de sortie.(utilisé aussi pour les CAN)
- RÉOLUTION: exprimée en nombre de bits du convertisseur ou en nombre d'incrément (LSB) du convertisseur.
- TEMPS D'ÉTABLISSEMENT: temps mis par la tension de sortie pour passer de zéro à la pleine échelle lorsque N passe de 0 à Nmax.
- LINÉARITÉ, erreurs de gain et d'offset, s'expriment de la même façon que pour les CAN.
- POLARITÉ: Un convertisseur Numérique analogique peut être considéré comme un multiplicateur. En effet il effectue une multiplication entre la valeur de la référence et le nombre à convertir à une

constante près $\left[V_S = N \times \frac{V_{ref}}{2^n} \right]$. Selon que N est exprimé en code signé ou que v_{ref} peut être positive ou négative on dira que l'on a à faire à un convertisseur deux cadrans ou quatre cadrans si les deux propriétés lui sont acquises.

• Précision et monotonie

Lorsque l'on passe d'un nombre N à N+1 la tension de sortie passe de $V_{S_N} \pm \epsilon$ à $V_{S_{N+1}}$ le cas le plus défavorable étant $V_{S_N} + \epsilon$ à $V_{S_{N+1}} - \epsilon$ si $\epsilon < 0.5$ LSB alors la tension de sortie diminue au lieu d'augmenter le convertisseur n'est pas monotone.

e- Principe de fonctionnement des CNA.

• Convertisseur à réseau R/2R.

Ce sont les convertisseurs les plus répandus, ils possèdent une bonne précision car les résistances mises en jeu sont toutes égales à R ou 2R.

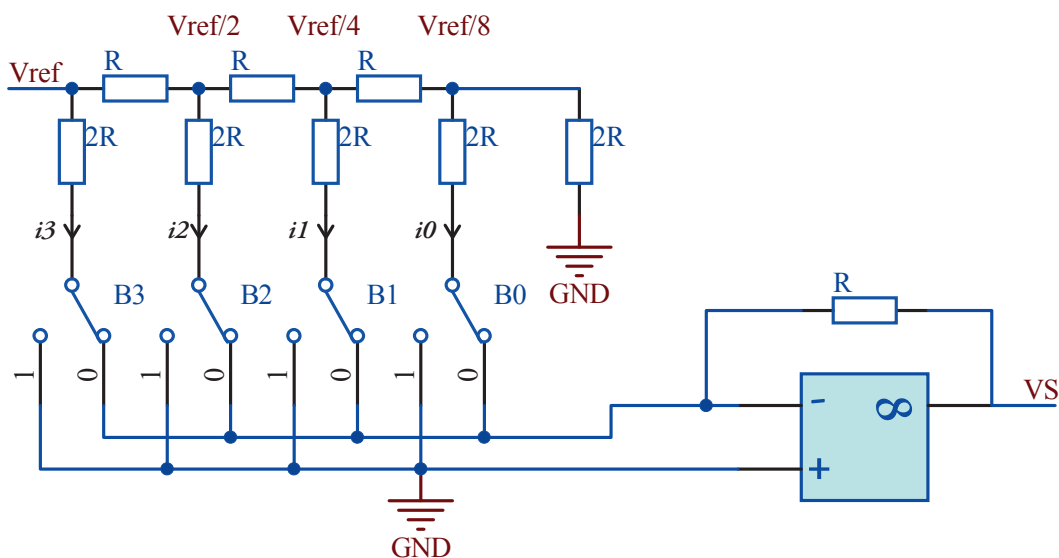


Figure10 Schéma de principe Convertisseur à réseau R/2R.

On exprime la tension V_S en fonction des éléments du montage et montrer que l'on peut la mettre sous la forme:

$$V_S = -\frac{V_{ref}}{2^n} [2^3 b_3 + 2^2 b_2 + 2b_1 + b_0] \quad \dots (7)$$

Ces convertisseurs dans un grand nombre de cas délivrent deux courants complémentaires l'un de l'autre, par contre la résistance de contre-réaction R est prévue dans le boîtier ce qui améliore la précision et la dérive thermique.

Le fait de ne pas intégrer d'amplificateur rend le montage plus souple et permet des applications autres que la conversion comme par exemple les amplificateurs à gain programmables numérique.

II.4 Branchement de convertisseur analogique numérique AD7524

Les pins A0...A7 des AD7524 sont connecté au port B du PIC16F876 simultanément en mode multiplexeur et grâce à la pin CS# on peut sélectionner soit le premier boitier ou le deuxième CS#=0 boitier 1 sélectionné CS#=1 boitier 2 sélectionné. On a fixé Vref du premier AD7524 à 5V et Vref du 2eme c'est la sortie du premier pour avoir une variation de 0V à 10V sur 16 bit.

Les deux amplificateurs sont ajoutés pour amplifier le signal issue des CNA et pour adapter l'impédance de sortie du convertisseur a fin d'attaquer le gradateur pour faire varier la température du four.

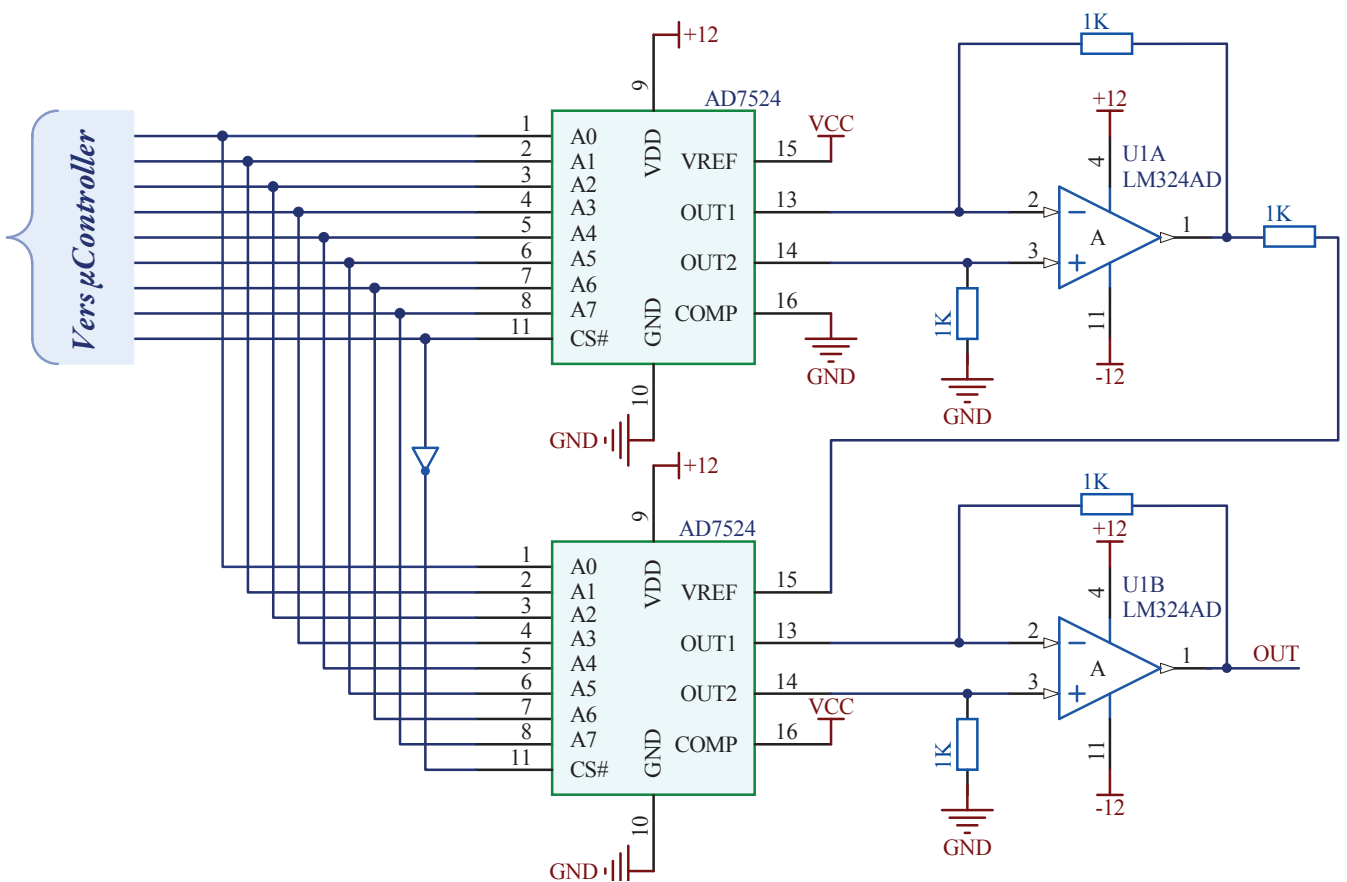


Figure11 Schéma de Branchement de CAN AD7524

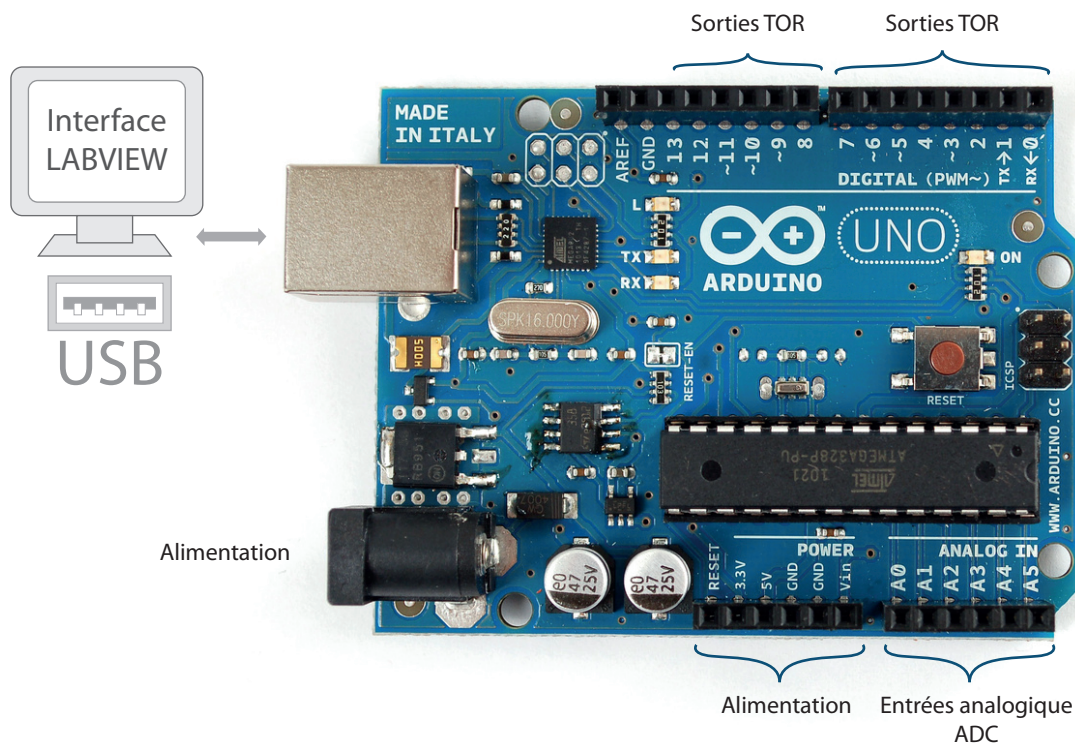
III. Bloc d'acquisition

III.1 Introduction

Cette partie d'acquisition est basé sur une plateforme open-source de prototypage Arduino uno et d'un thermocouple de type S et son circuit de compensation.

III.2 Description de la carte Arduino

L'Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328 de ATMEL. Il dispose de 14 entrées /sorties TOR broches (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, d'un oscillateur à quartz 16 MHz, d'une connexion USB, une prise de courant, d'une embase ICSP, et un bouton de ré-initialisation. Il contient tout le nécessaire pour soutenir le microcontrôleur, il suffit de le connecter à un ordinateur avec un câble USB.



III.2.1 Caractéristiques

• Micro-contrôleur	ATmega328
• Tension de fonctionnement	5V
• Tension d'entrée (recommandé)	7-12V
• Tension d'entrée (limites)	6-20V
• Pins E/S Numérique	14 (6 PWM)
• Pins entrée analogique	6
• Courant par Pin	40 mA
• Mémoire flash	32 KB
• SRAM	2 KB
• EEPROM	1 KB

III.2.2 Connexion de la carte Arduino au bloc de commande

Pour connecter l'Arduino au bloc de commande, on a utilisé le protocole I2C qui n'utilise que deux lignes de signal SDA & SCL (et les masses correspondantes bien sûr) permet d'échanger des informations sous forme série avec un débit pouvant atteindre 400 KBits par seconde.

Les données sont transmises en série de manière synchrone; cela signifie que les informations sont envoyées à la suite sur le même fil (une donnée par coup d'horloge), contrairement à une communication parallèle où plusieurs données sont envoyées en même temps, mais sur différents fils.

• Principe du bus I2C

A fin d'éviter les conflits électriques les Entrées/Sorties SDA et SCL sont de type «Collecteur Ouvert». Cela permet ainsi la présence de plusieurs maîtres sur le bus.

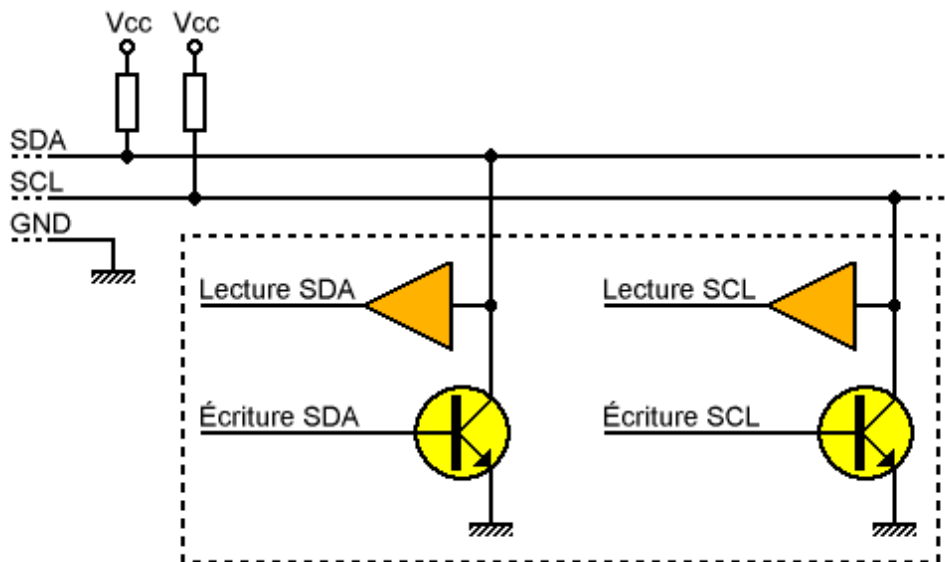


Figure11 Structure d'E/S d'un module I2C :

• La prise de contrôle du bus

Pour prendre le contrôle du bus, il faut que celui-ci soit au repos (SDA et SCL à '1'). Pour transmettre des données sur le bus, il faut donc surveiller deux conditions particulières :

- * La condition de départ. (SDA passe à '0' alors que SCL reste à '1')
- * La condition d'arrêt. (SDA passe à '1' alors que SCL reste à '1')

Lorsqu'un circuit, après avoir vérifié que le bus est libre, prend le contrôle de celui-ci, il en devient le maître. C'est lui qui génère le signal d'horloge.

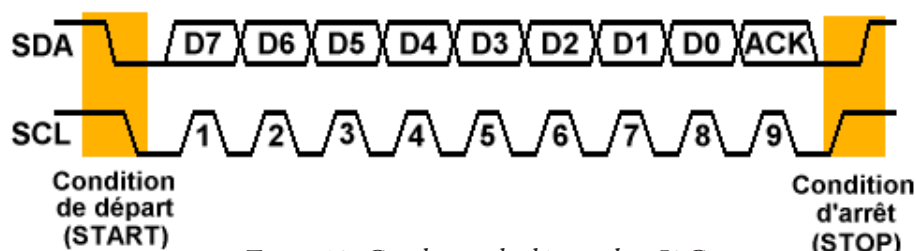


Figure12 Condition de départ bus I2C

• **La transmission d'un octet**

Après avoir imposé la condition de départ, le maître applique sur SDA le bit de poids fort D7. Ensuite on valide la donnée en appliquant pendant un instant un niveau '1' sur la ligne SCL. Lorsque SCL revient à '0', il recommence l'opération jusqu'à ce que l'octet complet soit transmis, ensuite en envoient alors un bit ACK à '1' tout en scrutant l'état réel de SDA. L'esclave doit alors imposer un niveau '0' pour signaler au maître que la transmission s'est effectuée correctement. Les sorties de chacun étant à collecteurs ouverts, le maître voie le '0' et peut alors passer à la suite.

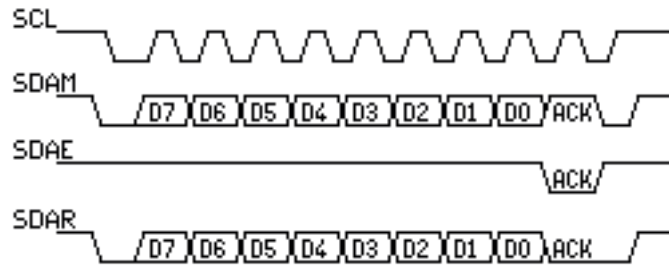


Figure13: Chronogramme de transmission d'un octet

• **La transmission d'une adresse**

Le nombre de composants qu'il est possible de connecter sur un bus I2C étant largement supérieur à deux, il est nécessaire de définir pour chacun une adresse unique.

L'adresse d'un circuit, codée sur sept bits, est défini d'une part par son type et d'autre part par l'état appliqué à un certain nombre de ces broches. Cette adresse est transmise sous la forme d'un octet au format particulier.

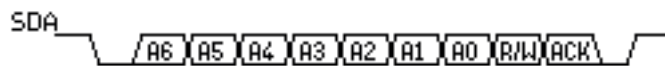


Figure14: Chronogramme de transmission d'une adresse

On remarque ici que les bits D7 à D1 représentent les adresse A6 à A0, et que le bit D0 et remplacé par le bit de R/W qui permet au maître de signaler s'il veut lire ou écrire une donnée. Le bit d'acquiescement ACK fonctionne comme pour une donnée, ceci permet au maître de vérifier si l'esclave est disponible.

• **Ecriture d'une donnée**

L'écriture d'une donnée par le maître ne pose pas de problème particulier :

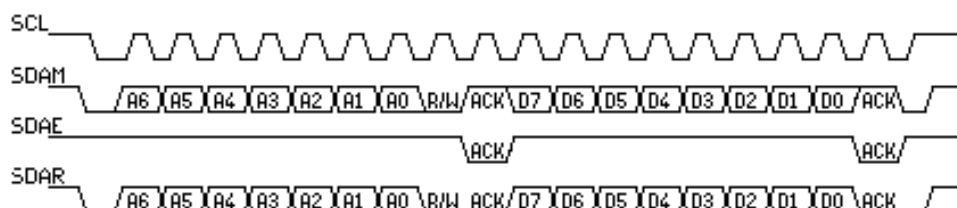


Figure15 :Chronogramme d'écriture d'une donnée

IV. Gradateur de puissance

Un gradateur est un dispositif de l'électronique de puissance destiné à modifier un signal électrique dans le but de faire varier sa tension et son intensité efficace de sortie et de modifier ainsi la puissance dans la charge. Ce dispositif est utilisé sur des tensions alternatives.

IV.1 Principe

Le gradateur utilise un thyristor pour faire varier la tension efficace en sortie du montage. Pour les équipements de forte puissance les gradateurs peuvent être réalisés par des groupes de thyristors montés en anti-parallèle, ou bien éventuellement, par des associations thyristors-diodes dans le cas de raccordement à des réseaux polyphasés.

Un thyristor est assimilable à une diode de forte puissance qui laisse passer le courant lorsqu'elle reçoit un signal de commande sur une 3e électrode, la "gâchette". Pour commuter un courant alternatif, on monte donc deux thyristors antiparallèles. La charge mise en série avec ces thyristors est commandée, en agissant sur le signal de gâchette. Cette commande est effectuée par des circuits électroniques, qui pilotent la gâchette en fonction d'un signal de consigne généralement issu d'un régulateur.

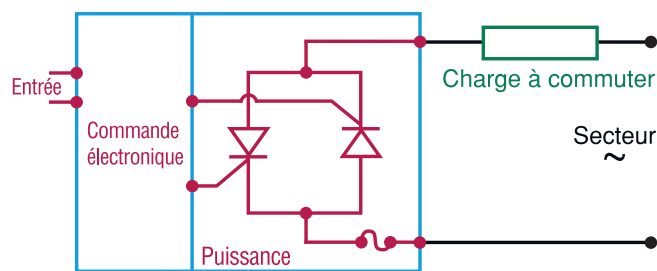
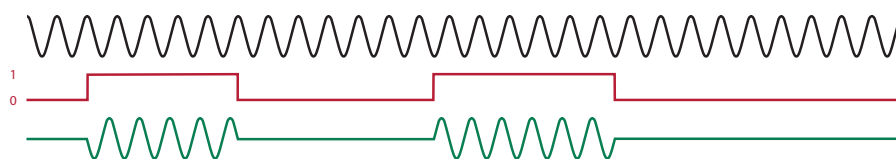


Figure16: Schéma de Principe d'un gradateur

IV.2. Les différents modes de fonctionnement

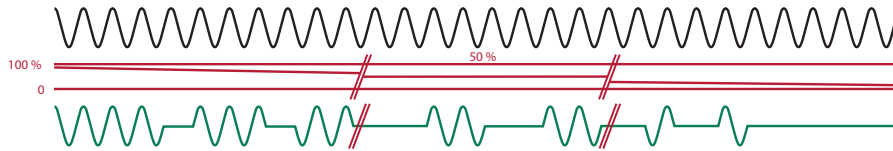
• Fonctionnement par train d'ondes ou mode Tout-ou-Rien

Les thyristors conduisent quand une tension est appliquée sur l'entrée commande. Sans tension de commande, les thyristors sont bloqués. Enclenchement et déclenchement se font au zéro de tension. Ce mode est adapté aux installations les plus simples, possédant une certaine inertie, et lorsque qu'une légère oscillation est admissible de part et d'autre de la valeur désirée. Les thyristors conduisent jusqu'à atteindre cette valeur. La conduction reprendra lorsque l'écart sera suffisant.



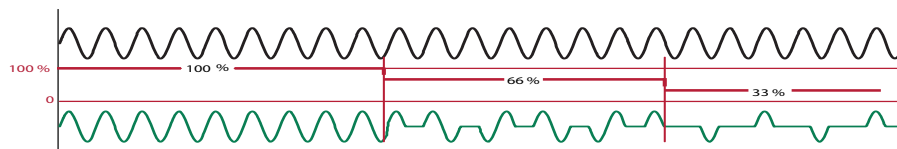
• Fonctionnement par train d'ondes syncopé ou mode TAKT

Consiste à commuter des paquets d'alternances du secteur, selon un temps de cycle de base défini par l'utilisateur. Ce mode permet de moduler finement la puissance en fonction de la demande.



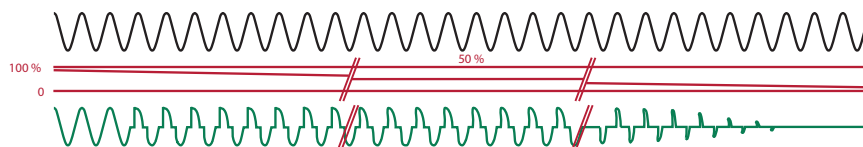
• Fonctionnement par train d'ondes syncopé rapide ou mode QTM

Ce mode se rapproche très fortement du mode angle de phase avec quelques avantages supplémentaires : commutation au zéro de tension et pas de composante continue. Ce nouveau mode breveté consiste à commuter des demi-alternances de 10 ms et ceci sur toute la plage de consigne de 0 à 100 %. En conséquence, la température de l'élément de chauffe est donc plus constante. Pour les applications en infrarouge, le scintillement des lampes infrarouges devient imperceptible à l'œil.



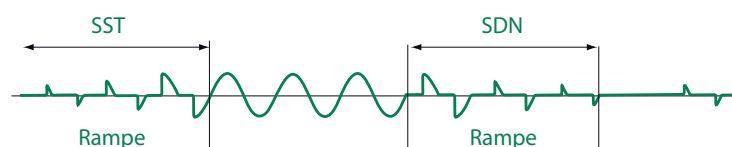
• Fonctionnement par angle de phase ou mode VAR

Découpe chaque "alternance secteur", transmise à la charge, d'un angle proportionnel à la demande de puissance. Ce mode est bien adapté à la commande de charges composées de transformateurs ou de résistances chauffantes présentant un $R_{chaud} / R_{froid} \pi 1$. Il permet en effet un démarrage en douceur, par ouverture progressive de l'angle de conduction, limitant ainsi l'appel de puissance lors de la mise en service de l'installation.



• Fonctionnement en mixte ou mode SSSD

Ce mode est une combinaison du fonctionnement par train d'ondes et par angle de phase. Il constitue une amélioration du mode train d'ondes, car il fonctionne en angle de phase, à l'enclenchement (SST) ainsi qu'au déclenchement (SDN). Il limite les perturbations sur le réseau pour la commande de charges importantes.



IV.3. Régulateurs de puissance THYRITOP 30

Le Thyritop 30 sont des régulateurs statiques de puissance avec commande électronique pour toutes les charges résistives ou inductives, en monophasé ou triphasé.

1. Généralités

Le THYRITOP 30 est un régulateur de puissance électronique à thyristors capable de communiquer avec la carte de commande . Il sera par la suite également appelé régulateur de puissance ou régulateur, peut être employé partout où des courants ou des tensions sont à commander ou à réguler dans le domaine du Process. Le THYRITOP 30 se distingue par plusieurs modes, de fonctionnement et de régulation, par une bonne compatibilité avec les systèmes de régulation de température et les automatismes, par une grande précision de régulation et une manipulation facilitée par l'utilisation d'un processeur 16 bits.

2. Caractéristiques générales

Régulation :

- U et U2 à $\pm 2,5 \%$
- I et I2 à $\pm 1,5 \%$ (modèles HRL1)
- P à $\pm 2,5 \%$ (modèles HRLP1)

Surveillance

- Rupture partielle de charge
- Sous ou sur-tension
- Sous ou sur-intensité
- Signalisation de défauts 3 LED : fusible, ΔI , ΔU
- Alarme : relais (modèles HRL1)
- Isolation entrée/sortie selon VDE 0160

3. Circuit de commande

- Analogique : 0/4...20 mA ($R_i = 250 \Omega$)
- 0...5 V ($R_i = 44 \text{ k}\Omega$)
- 0...10 V ($R_i = 88 \text{ k}\Omega$)
- Potentiométrique
- Numérique via modules de bus

4. Circuit de puissance

- Intensité nominale : 30A
- Tension nominale : 230 V
- Fréquence du réseau : 47 Hz à 63 Hz
- Rigidité diélectrique : 2 kV



Figure17 :Le THYRITOP 30

V. Bloc d'alimentation

On a utilisé une alimentation symétrique à base d'un transformateur à point milieu qui va transformé le 220V AC en +/- 12V AC, l'ensemble D1, D2, D3 et D4 forment une redresseur de tension, les deux capacités C4 et C7 de 1000 μ F assure un bon filtrage et en fin les régulateurs qui délivre une tension et courant stable LM7812 pour le 12V, LM7912 pour -12V, LM7805 pour 5V et en fin LM317 qui a une référence réglable pour avoir une tension variable de 0 à +12V.

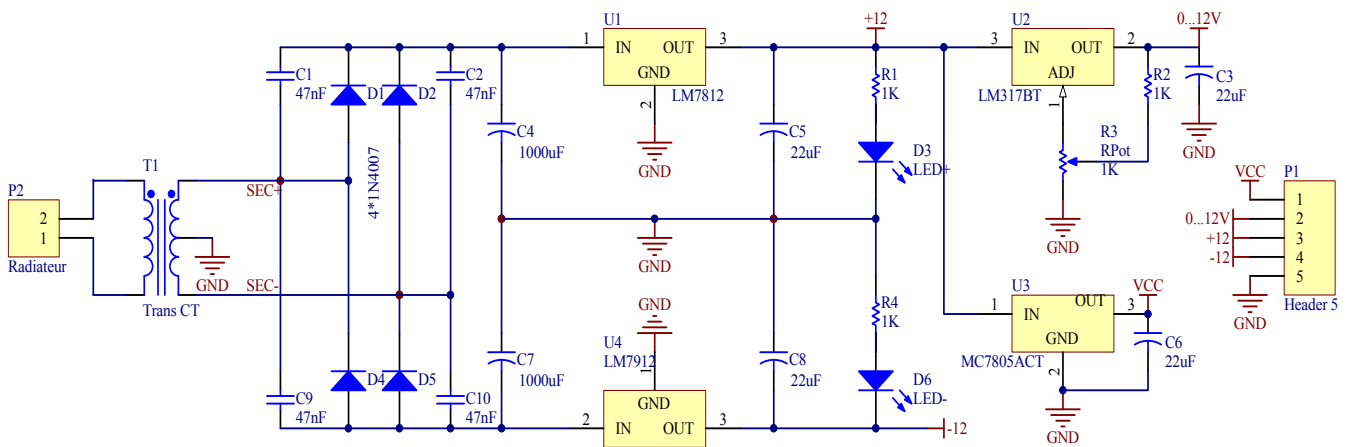


Figure18: Circuit d'alimentation

Chapitre 2

Interface graphique et codage

Introduction:

Ce chapitre porte sur la conception que nous avons proposé en tenant compte des contraintes imposées par le cahier de charges. A fin de facilité l'utilisation et la programmation du système de commande et l'interface graphique, Labview s'avère adapté dans le domaine de contrôle et l'expérimentation

I. Introduction à la programmation graphique

LABVIEW (Laboratory Virtual Instrument Engineering Workbench) est un langage de programmation dédié au 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 utilisant la notion flot de données pour ordonnancer les opérations. Enfaite il existe deux formes de programmation graphiques:

Une programmation dite flot de contrôle et une autre dite flot de donnés.

I.1. programmation flot de contrôle

Ils ont longtemps été utilisés pour décrire les algorithmes, ces représentations décrivent les programmes comme étant de noeud de calcul connecté par des arcs spécifiant quel calcul doit être effectué ensuite
Ex : Grafcet et Réseau de petri.

I.2. Programmation flot de donné

C'est une fonction analogue à la propagation du signal à travers un circuit électrique. Le digramme flot de données est un graphe acyclique qui peut être composé de 3 éléments suivants :

- Des terminaux : qui sont les liens avec l'extérieur qui représente la production où la consommation de données.
- Des Nœud qui sont les traitements à effectué et qui sont représenté par une figure géométrique pouvant contenir une image illustrant leur fonctionnalité
- Les arcs orienté : qui relie les nœuds et les terminaux et permettent d'indiqué le passage de données d'un nœud vers un autre.

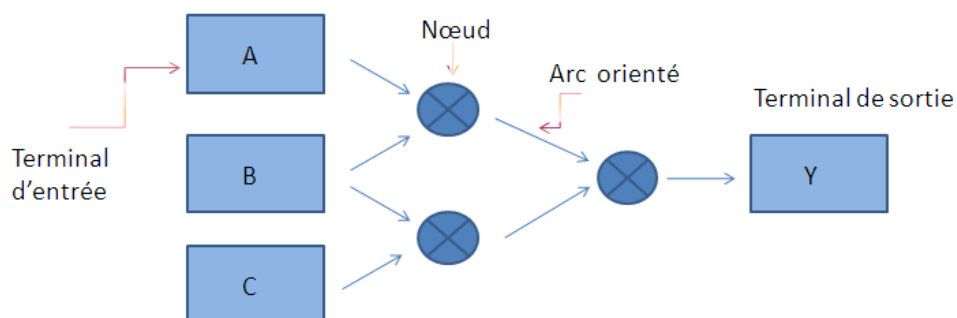
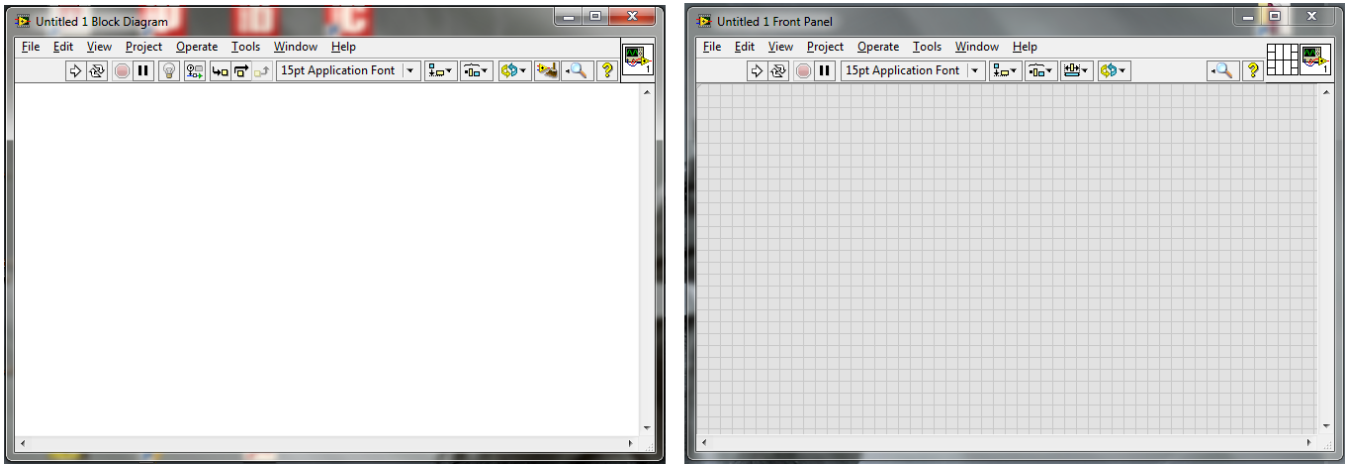


Figure20: Exemple de diagramme de flot de donnés

II. L'environnement LabVIEW

LabVIEW est centré autour du principe d'instrument virtuel (Virtual Instrument ou encore VI). IL se décompose en deux parties :

- la première partie (partie cachée ou interne) : elle contient l'algorithme du programme décrit sous la forme d'un diagramme flot de données en langage graphique.
- la seconde partie (partie visible) est constituée de l'interface utilisateur.



*Figure21: Fenêtre de l'environnement de développement sur LabVIEW
Face avant (à droite) et Diagramme (à gauche)*

Pour écrire un programme sur LabVIEW, on a besoin des « Palettes » qui nous offre la possibilité de modifier la face avant et le digramme de LabVIEW, on trouve trois palettes :

• 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.



Figure22: palette d'outils

• Palette de commandes

Elle est disponible uniquement sur la face-avant, elle contient les commandes et les indicateurs de la face-avant nécessaire pour créer l'interface utilisateur.

• Palette de fonctions

Elle est disponible uniquement sur le diagramme. Elle contient les objets nécessaire pour la programmation graphique comme les opérations d'arithmétique, d'E/S d'instrument, d'E/S de fichier et d'acquisition de données.

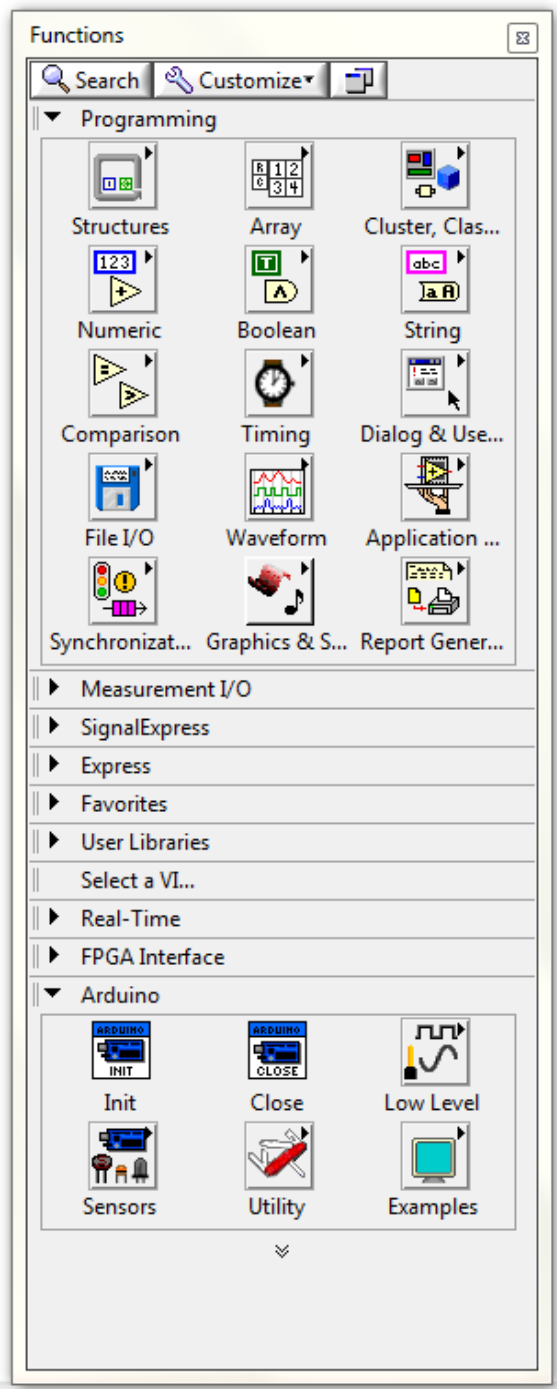


Figure23: Palette de fonctions

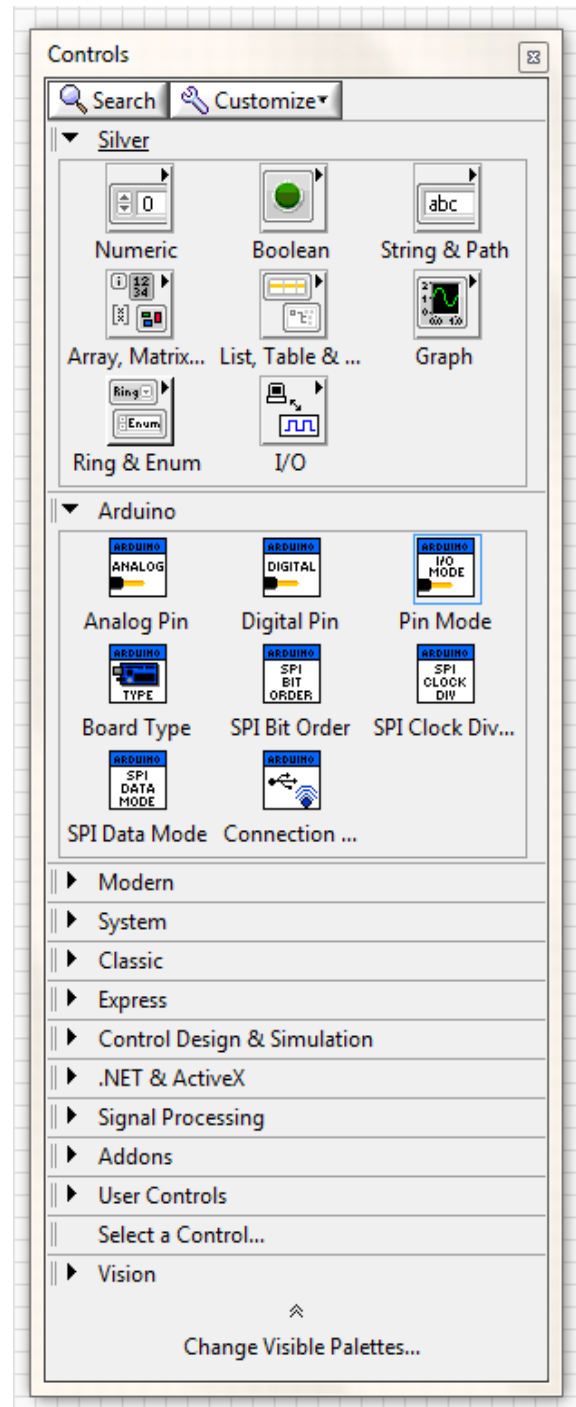


Figure24: Palette de commandes

II.1. Structure de données dans LabVIEW

LabVIEW utilise un langage fortement typé et toutes données ou structure de données ne peuvent être manipulées qu'avec des fonctions admettant ce type, en fait dans LabVIEW on trouve les types de base scalaire, les types entiers (signés ou non, codés sur 8, 16 ou 32 bits), le type réel (codé sur 16, 32 ou 64 bits), le type booléen et le type chaîne de caractères (figure IV.5). Il est important de noter que les éléments représentant ces données, ainsi que les liaisons issues de ces éléments, sont de forme et de couleur différente.

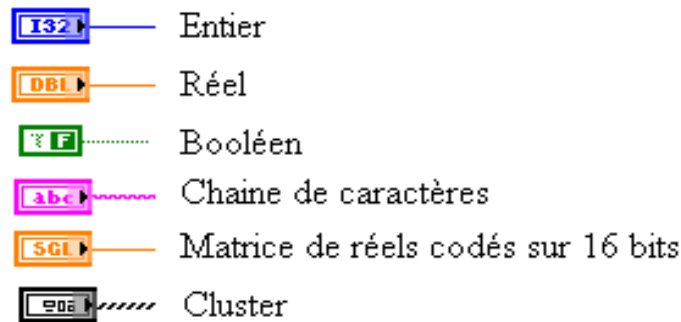


Figure 25 différents types de structures de données dans LabVIEW

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

II.1.1. Le type tableau (structure de données homogènes)

Comme tous les langages de programmations la présence de notion de tableau est obligatoire, puisqu'elle définit un outil de base pour le stockage et les traitements des informations.

La bibliothèque tableau dans LabVIEW est riche en fonctions prédéfinies, tel que, l'initialisation, l'indexation, l'inversement d'un tableau, la concaténation de deux tableaux, et plusieurs autres fonctions, bien sûr, la notion des tableaux peut être élargie pour contenir les matrices. Et comme toutes les autres bibliothèques, on peut l'enrichir avec des fonctions ou des procédures créées à partir des langages de bas niveau comme le langage C.

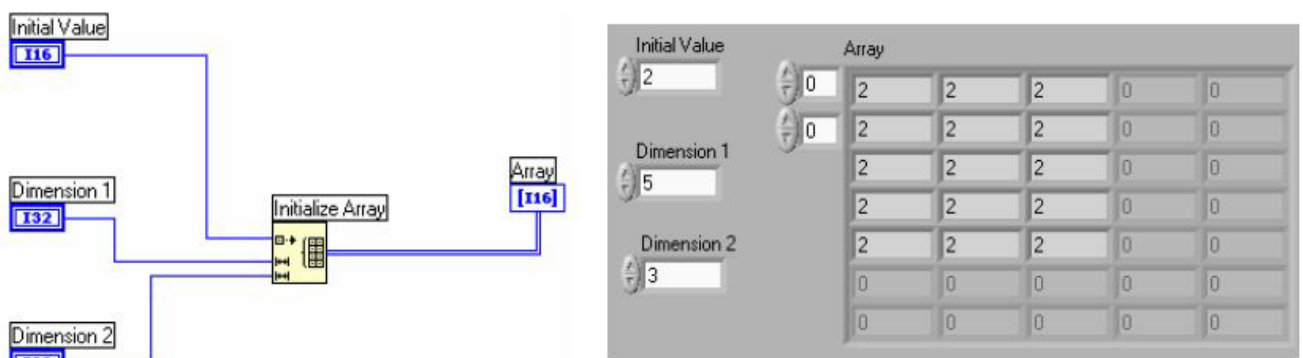


Figure 26: Exemple d'initialisation d'un tableau à 2 dimensions (matrice)

II.1.2. Le type « cluster » (structure de données hétérogènes)

Ce second constructeur de type est l'équivalent du « record » en Ada ou du « struct » en C. Les différents composants ou champs de ce groupe de données (cluster) peuvent être assemblés ou récupérés par des fonctions spécifiques.

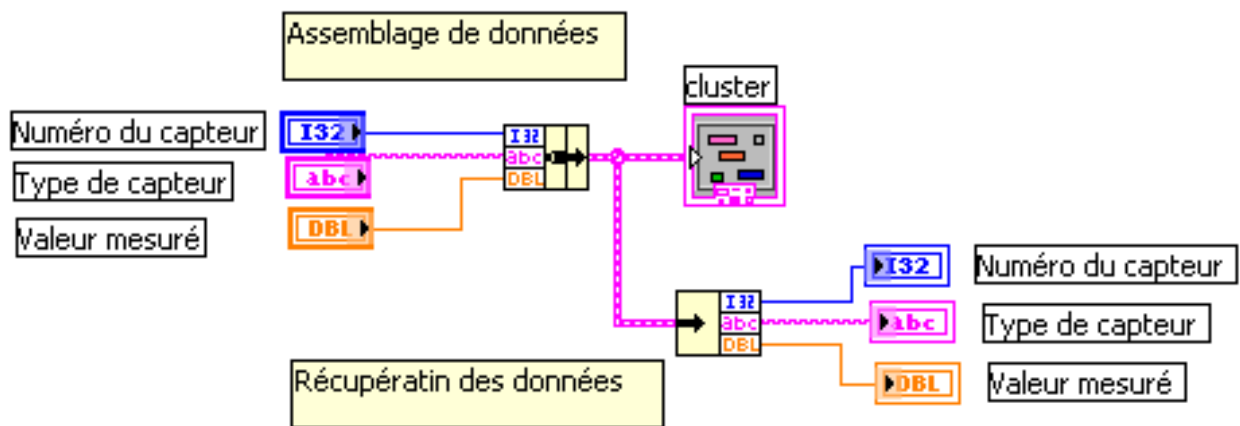


Figure 27: Exemple de manipulation des données avec le type cluster

II.2 Structures de programmation

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 » avec un nombre d'itérations fixé et la boucle « Tant Que » avec un nombre d'itérations soumis à condition) et la structure de choix.

II.2.1. Structure de séquence

La structure de « séquence » permet de spécifier l'ordre d'exécution de flots de données. Cette structure se présente sous la forme d'un cadre et a le statut d'un nœud

• *Exemple :*

Sur la figure ci-dessous, l'utilisation de la séquence s'impose : il s'agit d'initialiser un port série puis d'envoyer une chaîne de caractères. Le premier flot de données risque de provoquer des erreurs car rien n'empêche le nœud d'écriture sur le port série de s'exécuter avant le nœud d'initialisation de ce port : l'utilisation de la séquence permet d'ordonner correctement les opérations.

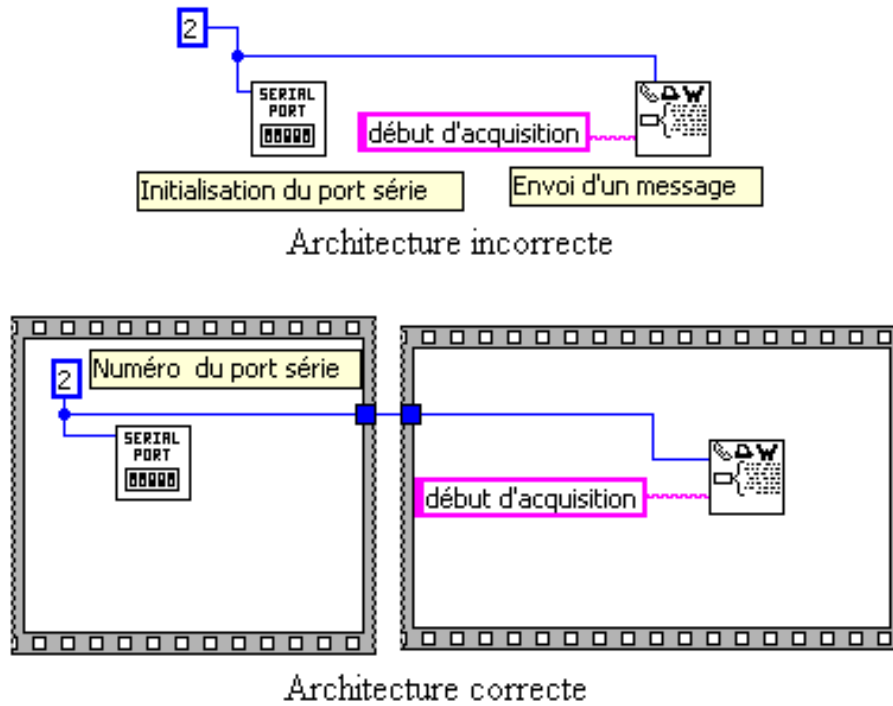


Figure28: Exemple d'utilisation de la structure de séquence

II.2.2. Les structures itératives

Les deux structures itératives, la boucle « Pour » et la boucle « Tant que », ont aussi le statut d'un nœud ordinaire. La boucle « Pour » permet d'exprimer la répétition (ou itération) pour un nombre de fois prédéterminé défini par une connexion d'entrée obligatoire: le nombre d'itérations à effectuer N . À l'intérieur de la boucle « Pour » se trouve un terminal d'entrée local générant l'entier indiquant l'indice d'itération de la boucle (i varie de 0 à $N-1$).

Exemple :

L'utilisation de cette structure, présentée sur la figure ci-dessous, concerne l'acquisition de N mesures avec un temps fixe entre chacune, défini par un nœud de temporisation. Les N données de sortie sont assemblées pour former un vecteur avec une auto-indexation.

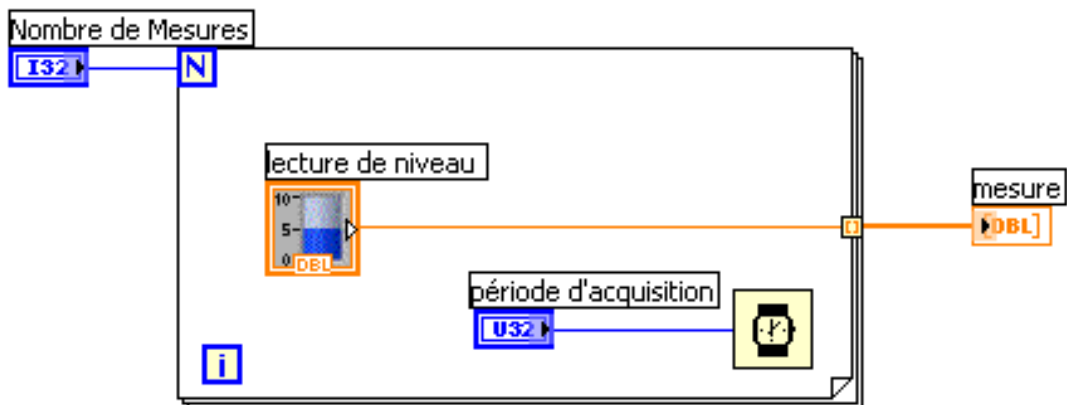


Figure29: Exemple d'utilisation de la structure itérative "pour"

La boucle « Tant Que » permet d'exprimer la répétition pour un nombre de fois non connu à l'avance. À l'intérieur de la boucle « Tant Que » se trouve un terminal d'entrée local générant l'entier indiquant l'indice d'itération de la boucle. Un terminal de sortie de type booléen permet d'arrêter la boucle lorsque la valeur « False » lui est envoyée.

• *Exemple :*

L'utilisation précédente décrite sur la figure ci-dessous, est reprise avec cette structure « Tant Que » en arrêtant l'acquisition pour une valeur trop grande de la mesure.

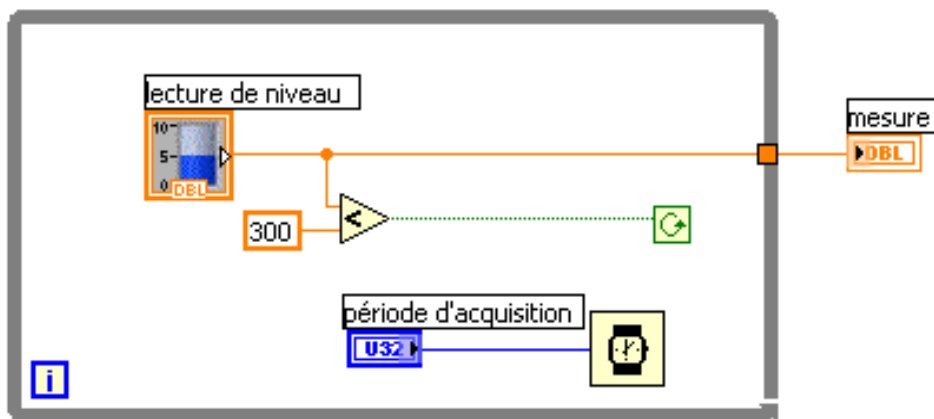


Figure30: Exemple d'utilisation de la structure itérative "tant que"

Dans les deux cas de structure itérative lorsque des tableaux de données arrivent ou partent des tunnels, ceux-ci ont la possibilité d'être automatiquement indexés : récupération un par un des éléments du tableau ou concaténation pour créer un tableau. Dans le cas des deux structures, il est possible de mémoriser des résultats produits lors d'itérations antérieures et de les récupérer ensuite. Ces registres à décalage sont des variables locales à une boucle.

• *Exemple :*

La figure ci-dessous, présente une utilisation de ces registres à décalage dans le cas de la boucle « Pour ». Il est important de remarquer que ces registres à décalage constituent un des effets de bord du langage. Il est de plus nécessaire d'initialiser ces mémoires locales pour réaliser une utilisation contrôlée et cohérente.

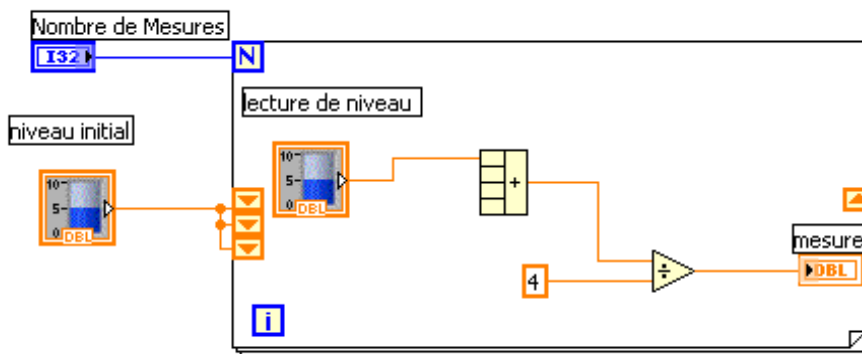


Figure31: Exemple d'utilisation des registres à décalage dans une boucle "pour"

II.2.3. La structure de choix

La dernière structure nécessaire est celle qui va permettre d'exprimer l'alternative (le choix). La sélection du cas exécuté est faite par la valeur de la variable connectée à l'entrée représentée par un point d'interrogation. L'identifiant du cas représenté est indiqué en haut de la structure.

• *Exemple :*

La figure ci-dessous présente une utilisation de la structure de choix. Si nous avons une action « vrais » alors il y a une acquisition de donnée sur le port série, et si l'action est « faux », alors il y aura pas d'acquisition.

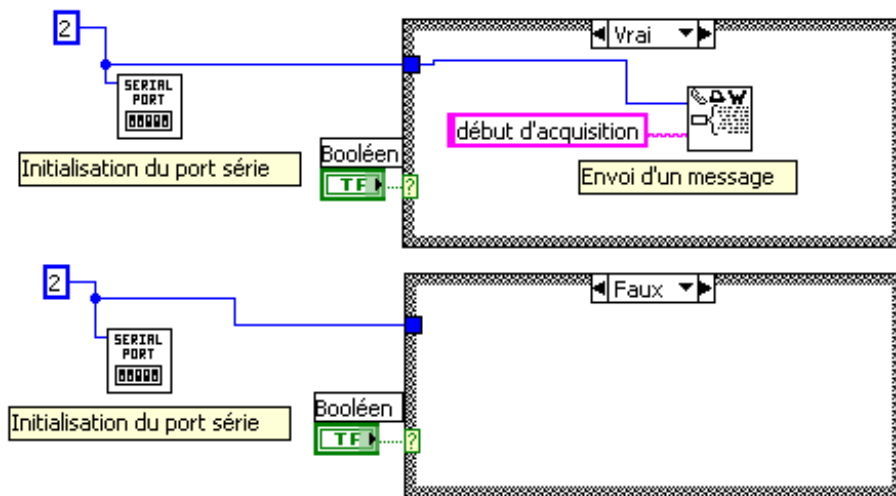


Figure32: Exemple d'utilisation de structure de choix.

II.3. Traitement numérique

II.3.1 Les fonctions prédéfinies

LabVIEW possède les instructions de base d'un langage de programmation permettant de traiter les différents types de données. Ainsi, nous avons des fonctions liées aux variables numériques (entiers, réels et complexes), aux variables booléennes, aux chaînes de caractères, et aux tableaux.

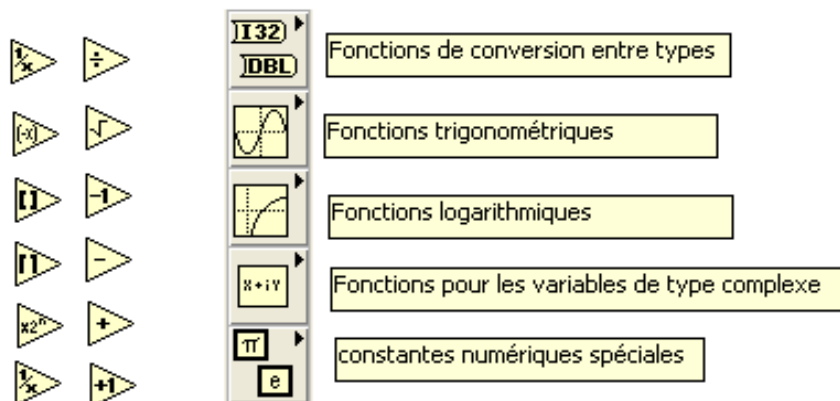


Figure33: Quelques instructions de traitement de données numérique dans LabVIEW

Nous trouvons aussi les opérateurs de test et de comparaisons liées à ces différents types de données. À partir des structures de contrôle, des fonctions et des opérateurs, de base, il est alors possible de traduire un algorithme quelconque et d'enrichir la bibliothèque des fonctions en utilisant le mécanisme d'encapsulation. Un diagramme complet est alors réduit à un noeud qui peut être ensuite réutilisé.

II.3.2. Boites à outils mathématique

LabVIEW contient aussi des boites de calcul mathématiques qui servent à introduire des commandes complexes tel que (linspace , ode45 ,sin, cos , etc..).

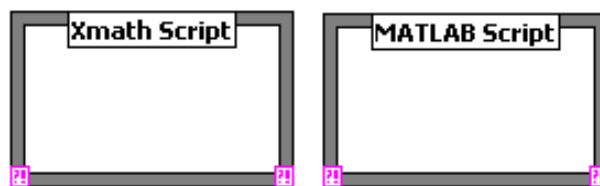


Figure34: Exemple de boîte de calcul mathématique dans LabVIEW

II.4. Bibliothèques de commande

LabVIEW possède des bibliothèques de fonctions spécialisées dans le domaine de la mesure, du test et du contrôle-commande. Ces bibliothèques de fonctions peuvent être réalisées soit à partir des fonctions et opérateurs de base vus précédemment, soit directement par des programmes écrits en langage de haut niveau, compilé et ensuite intégré sous la forme d'un nœud graphique.

Cette dernière possibilité est aussi offerte aux utilisateurs de cet environnement LabVIEW par l'intermédiaire de la construction de nœuds particuliers appelés « nœud d'interface vers code » (CIN : Code Interface Node). L'utilisation de ces fonctions qui peuvent être assez complexes est facilitée par une aide en ligne qui donne pour chaque fonction une description sommaire, souvent suffisante de ses spécifications.

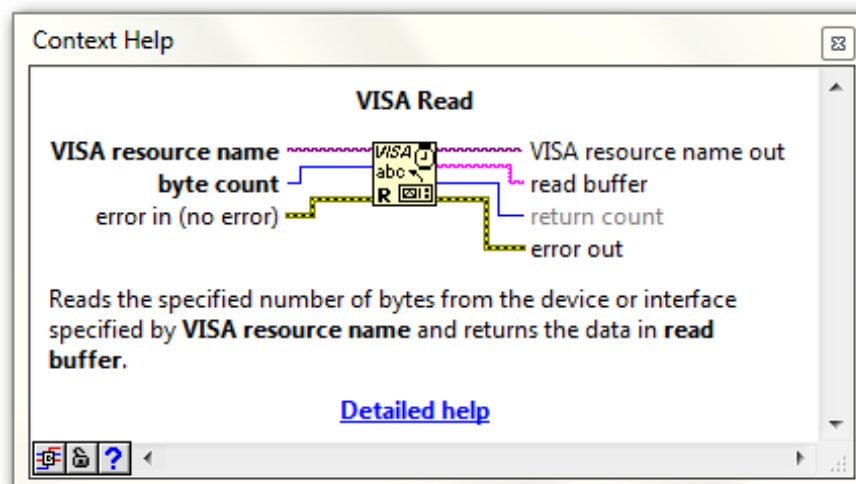


Figure35: Exemple de fenêtre d'aide offrant la description d'une fonction fournie dans la bibliothèque de LabVIEW

Dans notre projet on a besoin des bibliothèques spécifiques qui n'existe pas dans les versions standard de LabVIEW, mais dans des versions nommée (LabVIEW Arduino), enfaite nous avons utilisé des bibliothèques nommées « Analog sampling» .Il existe d'autre bibliothèque de communication telle que « I2C » et quelques autres commandes avancées.

On va présenter d'abord les deux bibliothèques de commande que nous avons utilisée dans notre projet :

II.4.1. Bibliothèques « Analog sampling»

Elle ajoute aux fonctions standards de LabVIEW un ensemble d'algorithmes d'acquisition des données analogique en temps réel , avec toutes les options que l'on trouve généralement dans les systèmes d'acquisition complets.

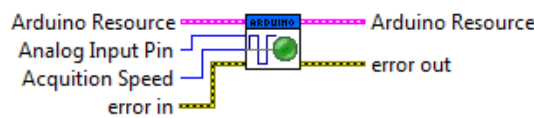


Figure36: Continuous Aquisition .vi

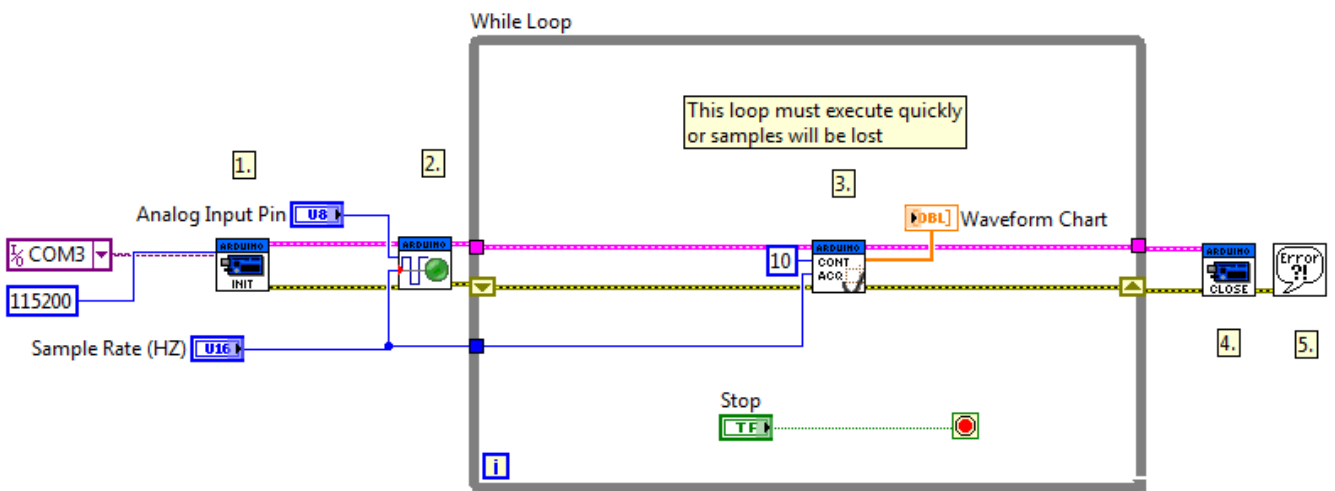


Figure37: Exemple de bibliothèque « Analog sampling»
Acquisition continu en temps réel

II.4.2. Bibliothèques « I2C»

Elle ajoute aux fonctions standards de LabVIEW un ensemble d'algorithmes de transmission de données série sur le bus I2C qui est compatible avec la carte Arduino.

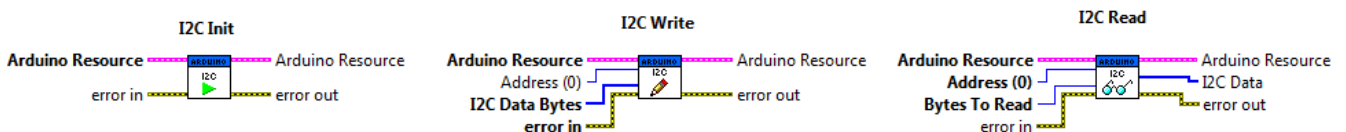


Figure38: Bibliothèques « I2C» i2c.vi

III. Interface & programme graphique

III.1. Description de schéma-bloc général

Au début on initialise la carte Arduino avec le port utilisé puis le bus i2c, après l'initialisation on entre dans la première boucle «Tant Que», cette boucle va être exécutée jusqu'à ce qu'on règle les paramètres d'acquisition (fréquence d'échantillonnage, temps d'acquisition, et mode de contrôle de gradateur $1^{\circ}\text{C}/\text{minute}$ ou $10^{\circ}\text{C}/\text{minute}$), puis transmette ces paramètres vers la carte de contrôle via le bus i2c, une fois le paramétrage est terminé on appuie sur le bouton «Démarrer l'acquisition» pour sortir de cette boucle et attaquer la deuxième boucle, cette dernière assure le rôle de la lecture de tension sur l'entrée analogique 0 de la carte Arduino et la transformer en température $2.5\text{mV}=1^{\circ}\text{C}$ (la tension de sortie du compensateur de thermocouple).

On met les échantillons de température sur un graphe en fonction de temps ($^{\circ}\text{C}/\text{s}$), ce graphe peut être réglé par l'utilisateur (la couleur, l'épaisseur, le style) et être exporté vers d'autres logiciels comme l'Excel de Microsoft ou Origin Lab.

Cette boucle reste active jusqu'à ce que le temps de fonctionnement soit écoulé ou arrêté par l'utilisateur on appuie sur le bouton arrête.

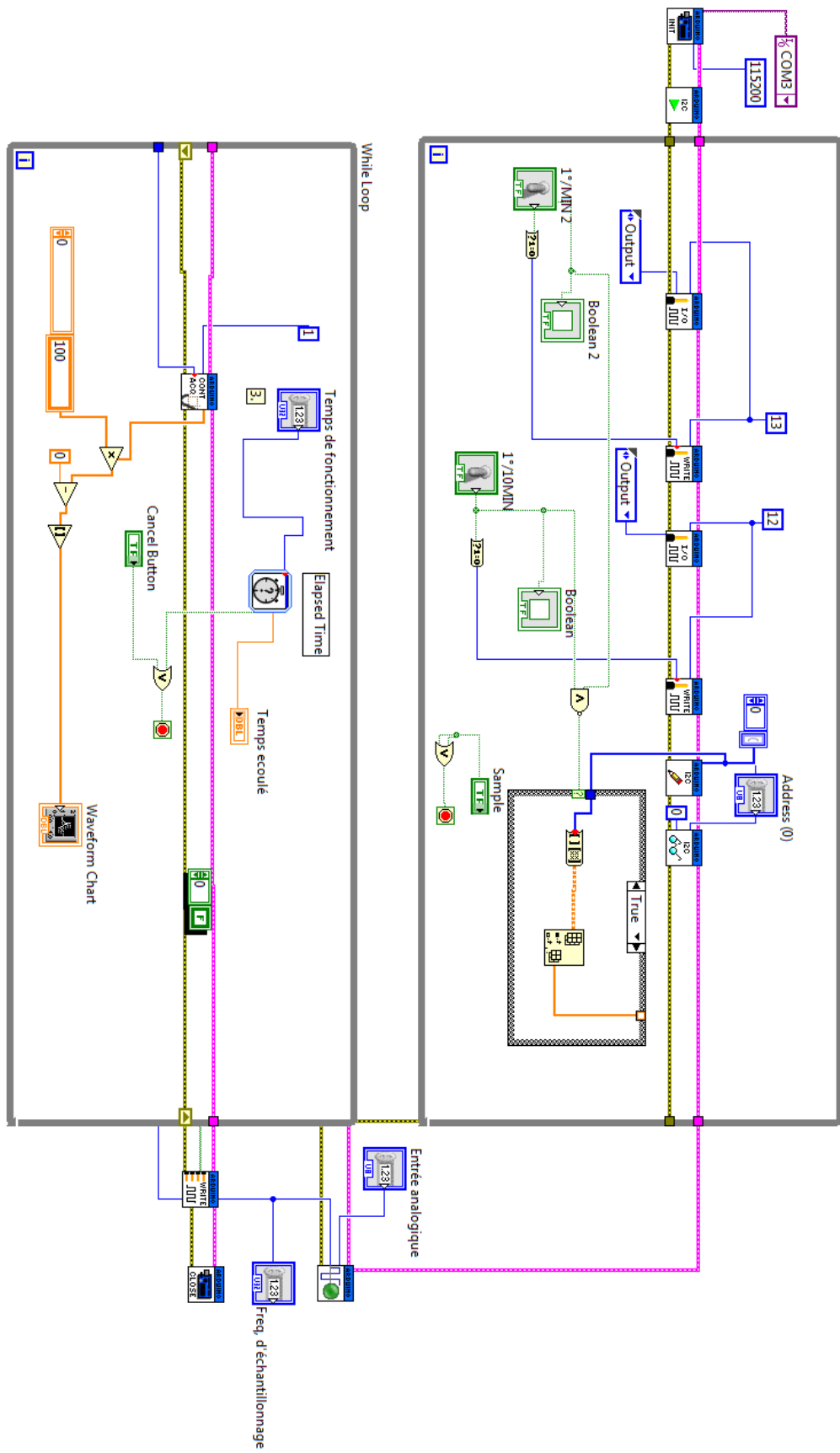


Figure39 Schéma-bloc général

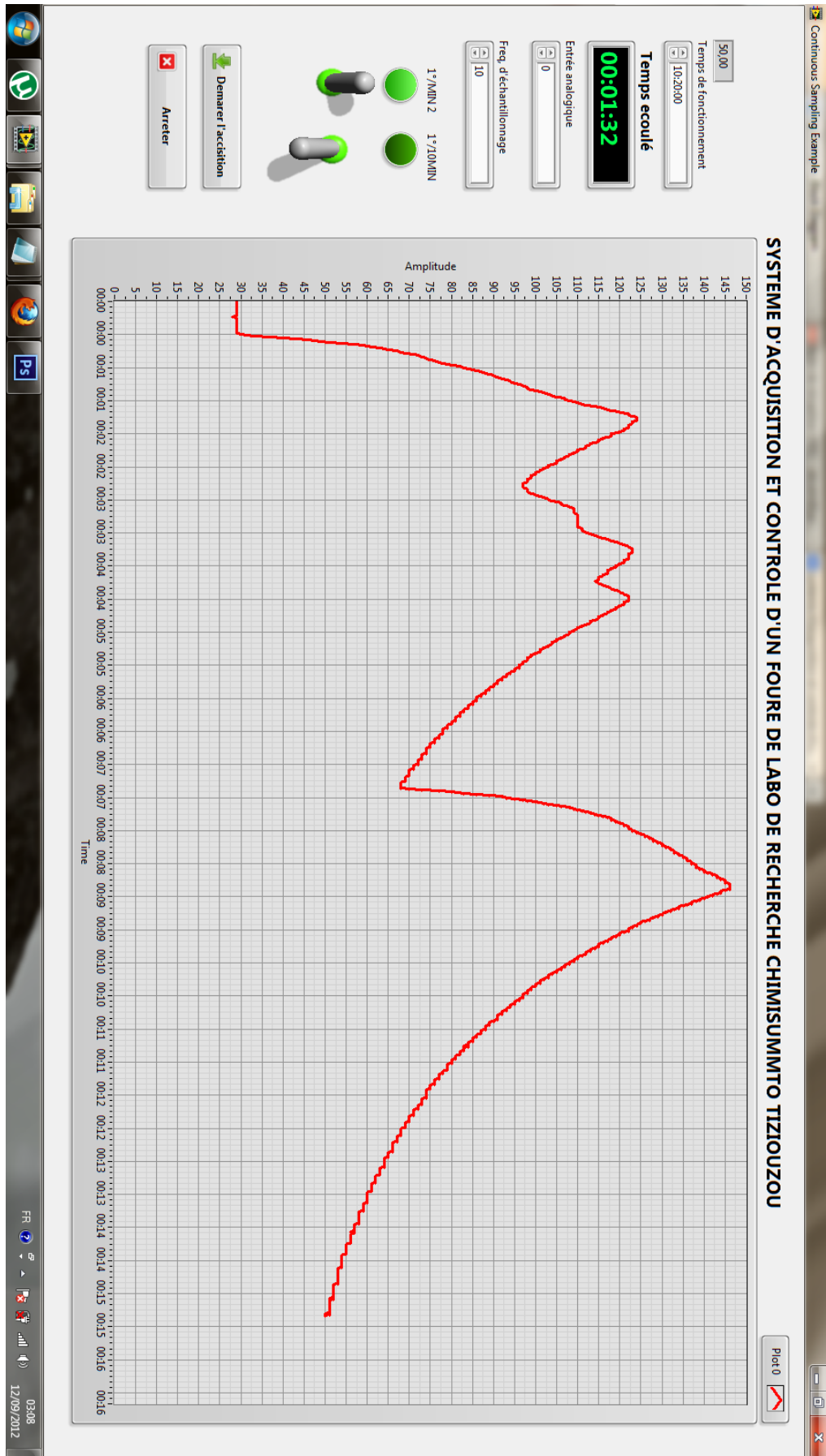


Figure40 :Interface graphique

IV. Programme du micro-contrôleur PIC16F86A

Pour la programmation du micro-contrôleur on a utilisé MikroC Pro for pic[®] de Mikroelektronika[™] qui utilise le langage C pour micro-contrôler Pic. Au début on a créé un organigramme pour faciliter la programmation puis on traduit cet organigramme en un programme en langage C. Après la compilation du programme on a testé son bon fonctionnement sur la carte de développement EasyPIC 6[®].

IV.1. Présentation de MikroC Pro for pic[®]

MikroC Pro for pic[®] est un compilateur C plein de fonctionnalités pour les PIC de Microchip[®]. C'est la meilleure solution pour développer le code pour les PIC. Il dispose d'IDE intuitif, puissant compilateur avec des optimisations avancées, beaucoup de matériel et de logiciels, des bibliothèques et des outils supplémentaires qui nous aident dans notre travail. Le compilateur est livré avec le fichier d'aide complet et beaucoup d'exemples de prêt-à-utiliser destinés à nous aider à démarrer en un rien de temps.

IV.1.1 Interface principale de MikroC

La figure 41 nous montre l'interface principale de MikroC avec description des parties .

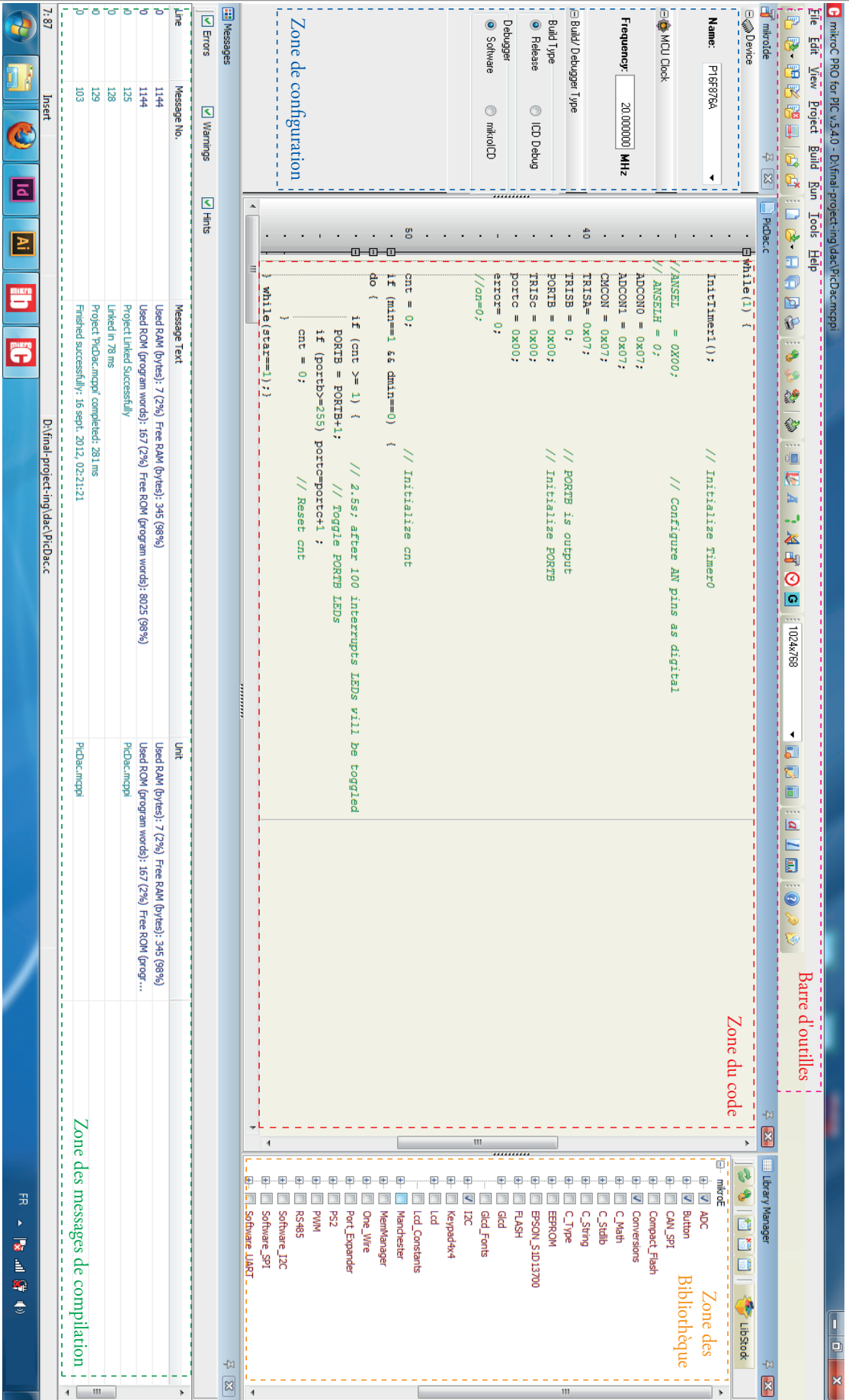


Figure 41: Interface principale me MikroC

IV.2. Organigramme

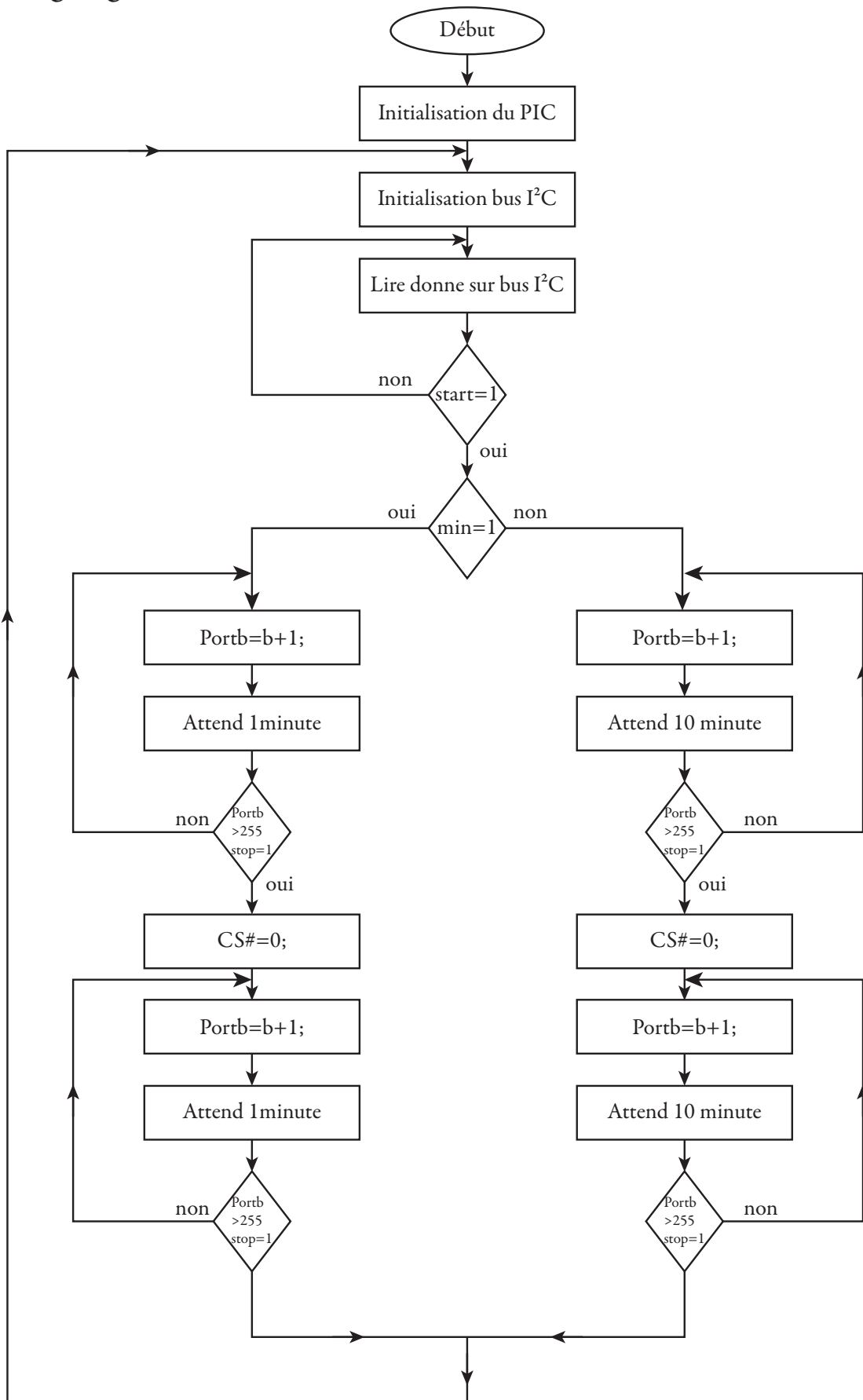


Figure42: Organigramme pour PIC16F876

IV.3. Programme en C

```

unsigned cnt,min,dmin,error,star,CS ;
void InitTimer1(){
    T1CON          = 0x21; // Configuration de Timer 1 pour
    TMR1IF_bit     = 0;   // une interruption chaque 100 ms
    TMR1H          = 0x3C;
    TMR1L          = 0xB0;
    TMR1IE_bit     = 1;
    INTCON         = 0xC0;
}
void Interrupt(){
    if (TMR1IF_bit){
        cnt++; // Increment counter
        TMR1IF_bit = 0;
        TMR1H      = 0x3C;
        TMR1L      = 0xB0;
    }
}
void main(){
    I2C1_Init(100000); // initialize I2C communication
    I2C1_Start();     // issue I2C start signal
    Delay_100ms();    // issue I2C stop signal
    Soft_I2C_Write(0xb7); // Address Arduino
    Soft_I2C_Write(2);
    Soft_I2C_Start();

    min = Soft_I2C_Read(1); // Lire les configuration
    dmin = Soft_I2C_Read(1); // sur le buse I2C de l'Arduino
    error = Soft_I2C_Read(1);
    star = Soft_I2C_Read(1);

    while(1) {
        InitTimer1(); // initialiser Timer0
        TRISA= 0x07;
        TRISB = 0; // PORTB en sortie
        PORTB = 0x00; // initialiser PORTB
        TRISC = 0x00; // PORTC en sortie
        portc = 0x00;
        cnt = 0; // initialiser cnt
        if (min==1 && dmin==0) {
            do {
                if (cnt >= 600) { // 600*100 ms = 1 minute
                    PORTB = PORTB+1; //incrementer PortB pour le Dac
                    if (portb>=255) CS = 0; }
                cnt = 0; // Reset cnt
            }
        }
    }
}

```

```

} while(star==1);}
  if (dmin==1 && min ==0 ) {
do {
  if (cnt >= 6000 ) { // 6000*100 ms = 10 minute
    PORTB = PORTB+1; //incrementer PortB pour le Dac
    if (portb>=255) CS = 0;
    cnt = 0; } // Reset cnt
} while(star==1 || ((dmin==1 && min==1) || (dmin==0 && min==0))); }
  if ((dmin==1 && min==1) || (dmin==0 && min==0)) {
do {
  if (cnt >= 10) {
    error = ~error;
    cnt = 0; }
} while((dmin==1 && min==1) || (dmin==0 && min==0)); }
}
}

```

IV.3.1 Programmation et tests

Après avoir injecter le code compilé dans le Pic via la carte de développement et brancher la carte Arduino a l'EasyPic via le bus i2c (3 fils RC6, RC7 GND) on a tester le fonctionnement, tout est bon !

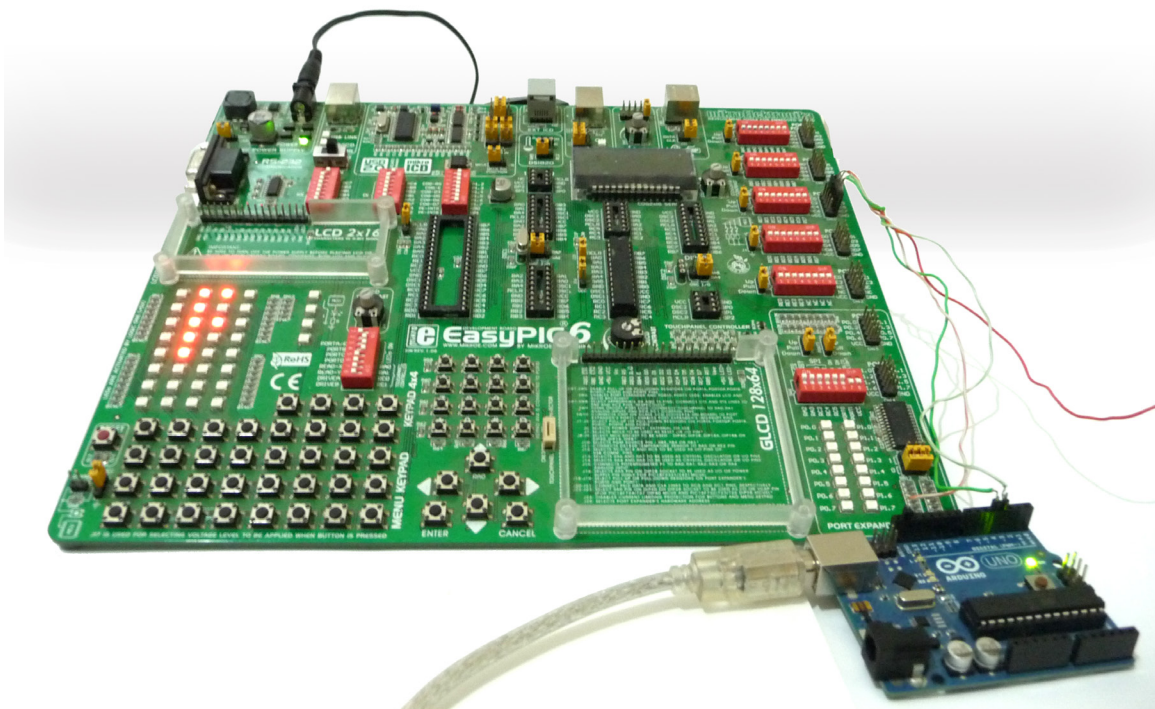


Figure43 Teste sur EasyPic6 avec Arduino

A présent tout notre système est détailler et tester soit la partie soft ou la partie hard, il reste juste a le réalisation pratique et mise en œuvre qui sera détailler dans le chapitre suivant.

Chapitre 3

Réalisation pratique et mise en œuvre

Après l'étude théorique qui avait pour objectif d'illustrer le fonctionnement de chaque partie du montage, et de spécifier les différents composants à utiliser, nous allons maintenant aborder la partie réalisation matérielle du système.

I. Circuit électrique et imprimé

Pour le dessin du circuit électrique et le développement du circuit imprimé on a utilisé le logiciel Altium Designer 10 Pour cela on a suivis les étapes suivantes:

I.1. Création du circuit électrique

On met en place tout les composants et connecteur utilise dans le circuit électrique,les nommer et attribuer a chaque composant son empreinte approprier a fin de pouvoir réaliser le circuit imprimé PCB.

Après avoir saisir les schémas électriques bous procédent maintenant a le transformer en circuit imprimé PCB en respectant les empreintes des composants et l'emplacement des connecteurs pour l'emboîtement avec la carte Arduino, pour cela on a générer une image 3D de la carte pour bien visualiser tous les emplacement des composants.

La figure suivante représente une Modélisation 3D de la carte de commande avec les différents composants implémenté et monté en superposition sur la carte Arduino a fin de réduire l'espace.

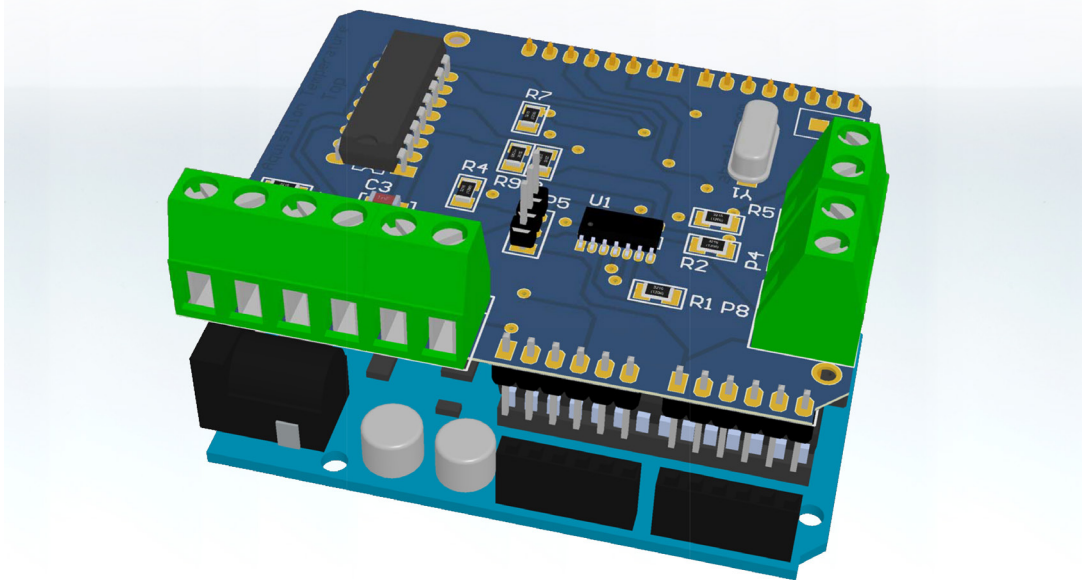
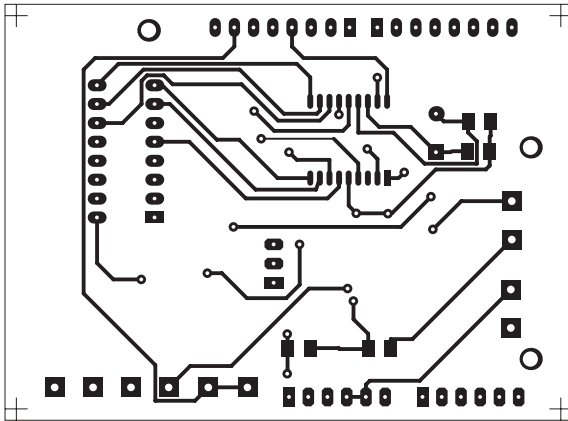


Figure 44: image 3D de la carte de commande monté sur l'Arduino

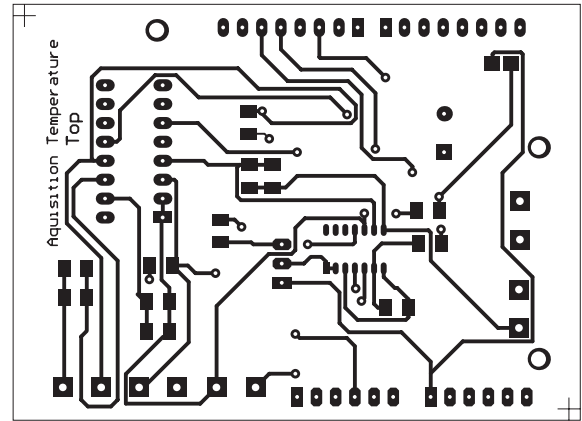
Après vitrification on procède alors au développement des cartes en passant par les étapes suivantes.

I.2. Développement ds cartes

- Impression de PCB sur film transparent



PCB Couche haut (Top)



PCB Couche bas (Bottom)

• L'insolation.

Retirer la pellicule protectrice de la plaquette pré sensibilisée. On superpose les typons sur la plaquette puis on passe à l'insolation. La partie exposée aux ultraviolets devient solvable dans le révélateur pour circuit imprimé.

- La révélation.

Après l'insolation on plonge les maquettes dans un bain de potasse avec une légère agitation pour éliminer toute la résine touchée par les rayons ultraviolets, cette opération dure entre 30s à 2mn. On termine par un rinçage à l'eau.

- La gravure.

Le cuivre non protégé est éliminé par une solution de perchlorure de fer chauffée. Dans ces conditions, la gravure complète se fera entre 15 et 25mn environ. La gravure étant terminée, le circuit imprimé est rincé à grande eau. Une fois le circuit sec, on vérifie que tout le cuivre est éliminé, les retouches nécessaires sont faites avec un grattoir. On vérifie l'absence de micro-coupure et de courts-circuits entre pistes, et au besoin on Utilise un l'ohmmètre pour cela.

• L'implantation des composants:

Après toutes les étapes précédentes le circuit imprimé est prêt pour le perçage. Toutes les connexions seront percées à l'aide d'une perceuse munie d'une forêt de 0.8mm de diamètre. Certains trous nécessitent un agrandissement pour s'adapter aux composants les plus volumineux. Pour l'implantation des composants, nous commençons par la mise en place des straps de liaisons. Ensuite, on passera à l'implantation des composants en commençant par les plus petits pour terminer par les plus grands. Nous souderons d'abord les composants passifs, résistances, condensateurs, ajustables puis les supports des circuits intégrés. On veille à ce que les composants polarisés soient bien orientés et qu'ils soient de même hauteur. Enfin on termine par les semi-conducteurs et les composants les plus volumineux puis les connecteurs.

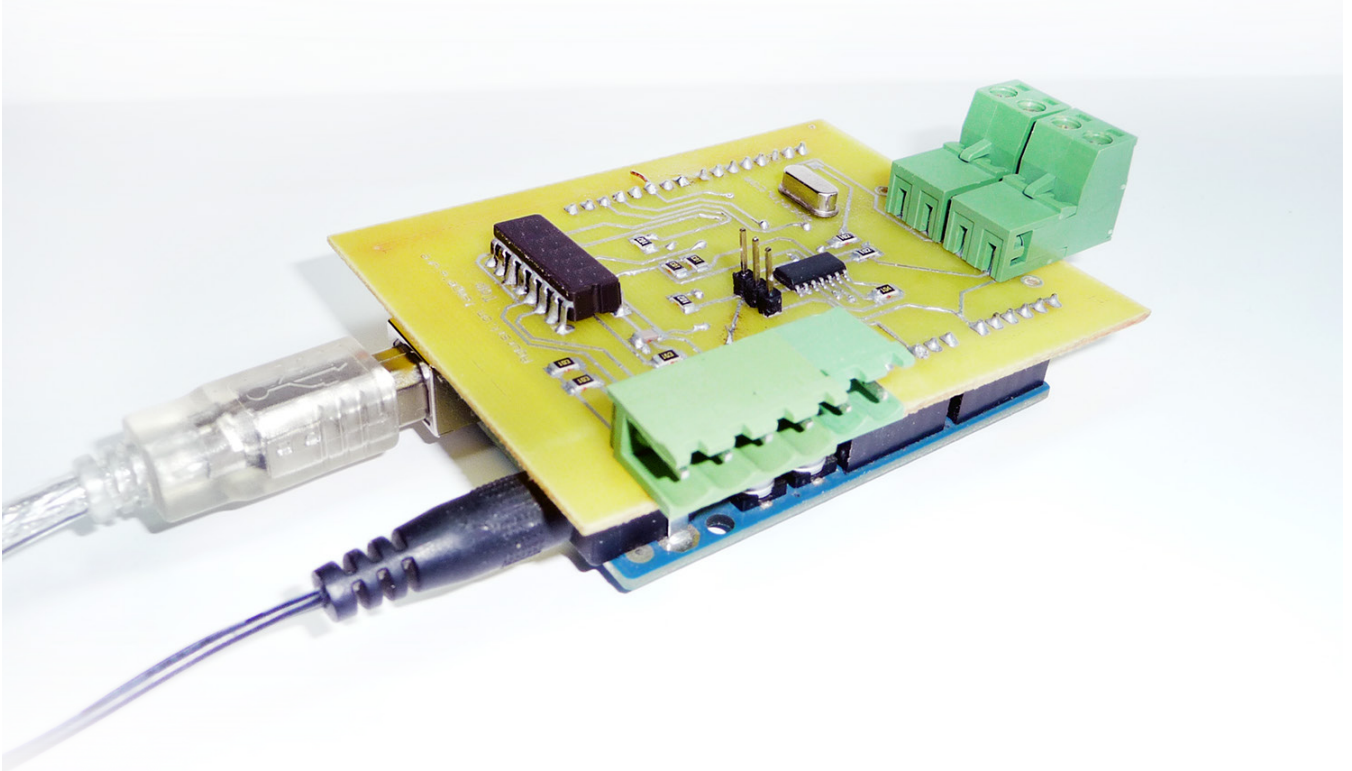
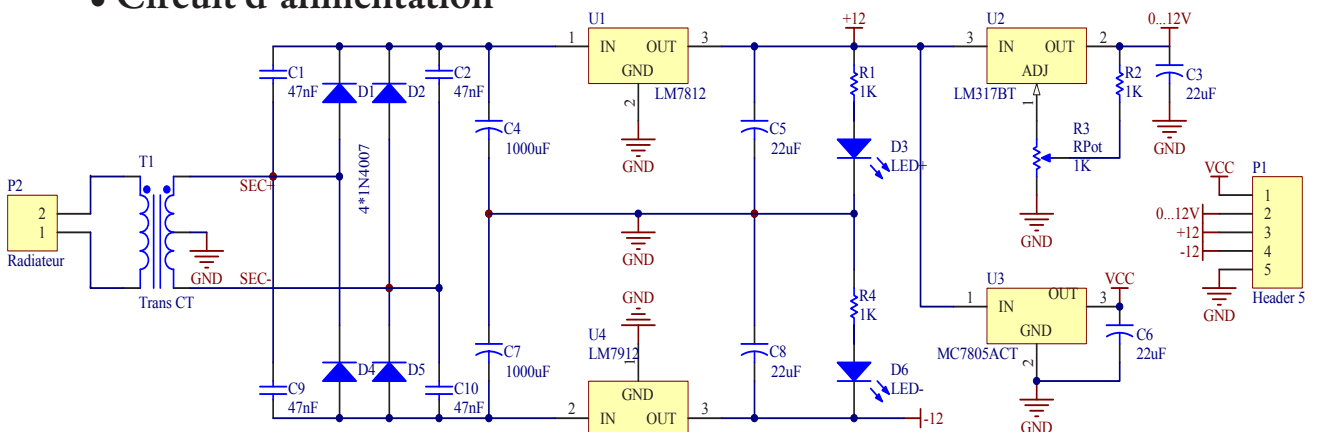
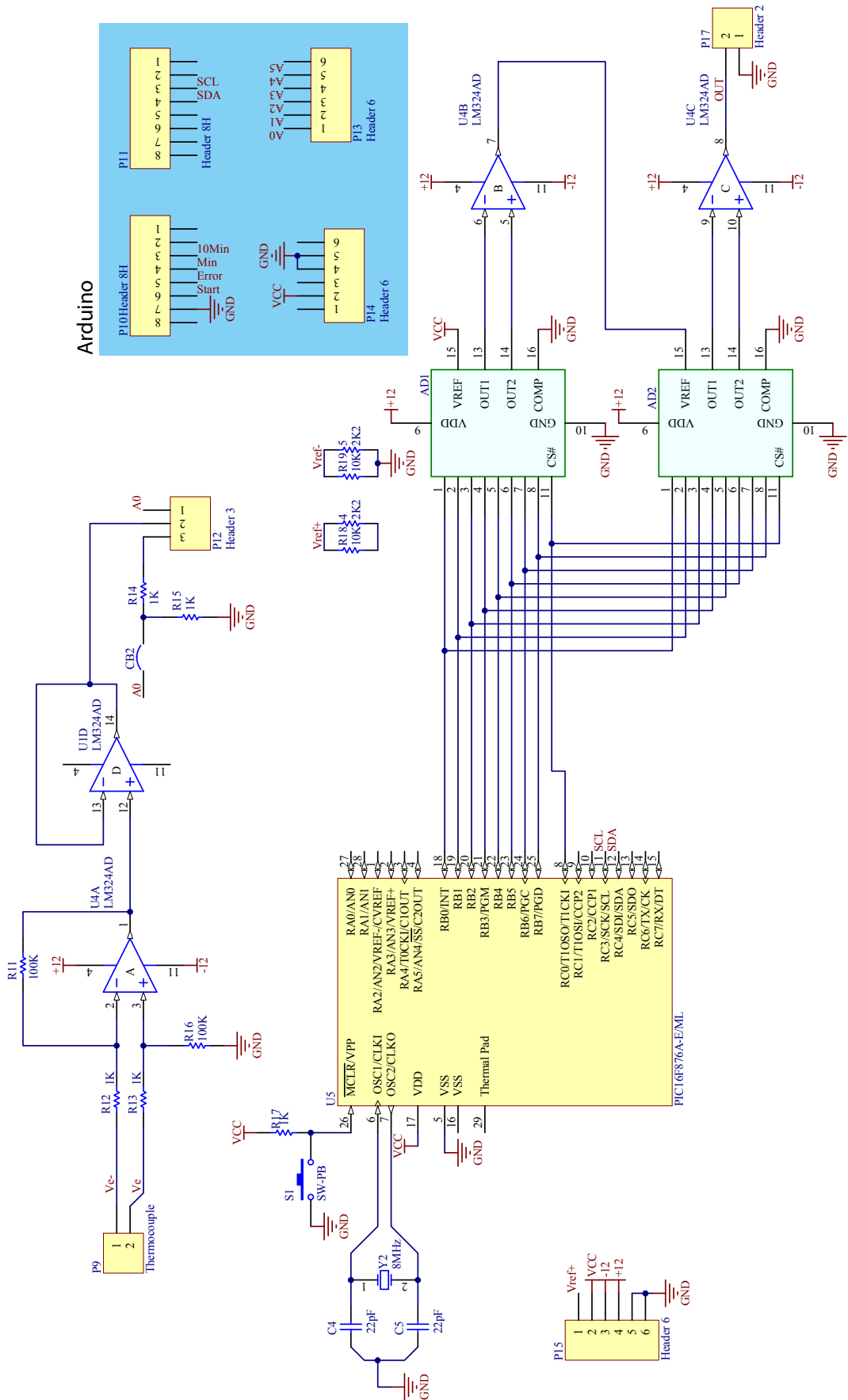


Figure 45: Carte de commande montée sur la carte Arduino

• Circuit d'alimentation

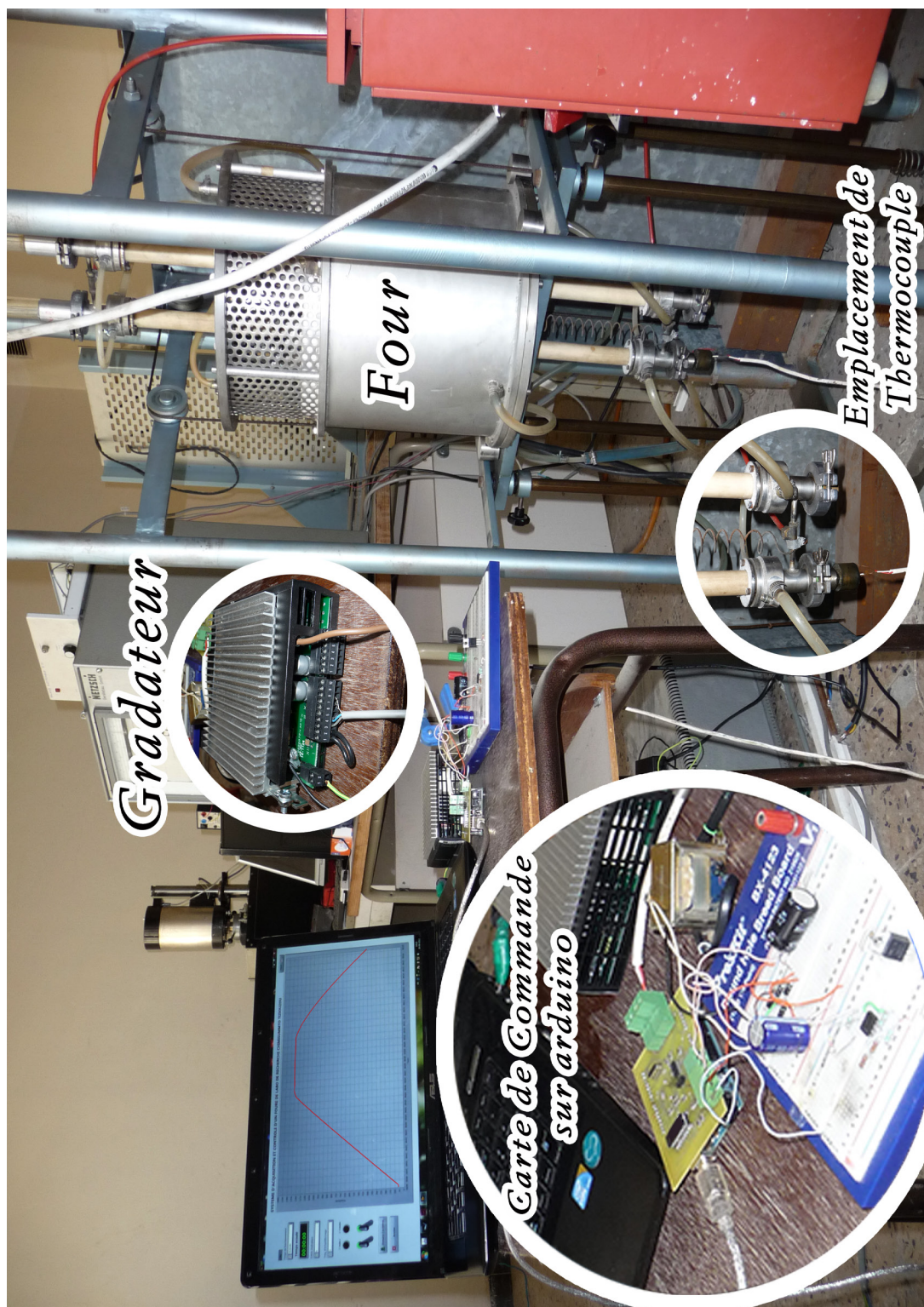


• Circuit globale

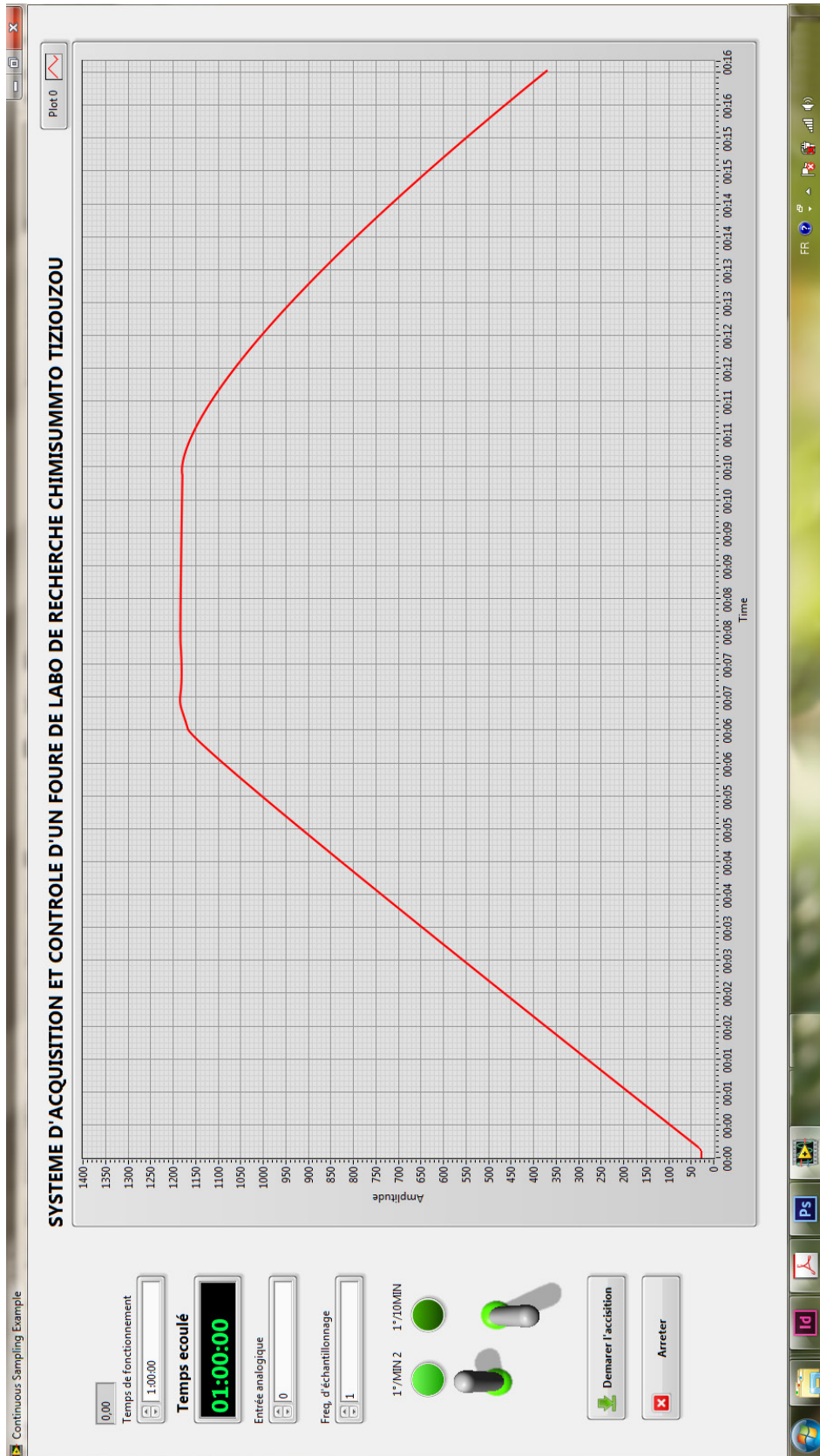


II. Mise en œuvre du système au laboratoire

L'image ci dessous représente le système mis en place au laboratoire de chimie il est composé d'un (thermocouple, gradateur, résistance de chauffage + notre système). Les résultats de l'évolution de la température dans le four en fonction du schéma de commande



L'image ci dessous représente les résultats de mesure de la température dans four via le thermocouple .
Les résultats son montré sous forme d'un graphe à l'aide de Labview



Conclusion générale

Au cours de ce projet de fin d'études, la conception et la réalisation de notre système d'acquisition et de commande du four de chimie nous a permis d'étudier les systèmes d'acquisitions et la programmation graphique «Labview.

Ce travail nous a permis donc d'enrichir nos connaissances dans plusieurs domaines et notamment dans le domaine de la commande numérique et de l'électronique analogique et numérique.

Pour conclure, nous espérons que ce modeste travail pourra servir de référence aux travaux futures des prochaines promotions et les inciter à s'intéresser d'avantage au coté pratique de l'électronique.

Bibliographie

- Labview Programmation et applications . Francis Cottet, Michel Pinard Collection: Technique et Ingénierie, Dunod/L'Usine Nouvelle 2009 - 2ème édition EAN13 : 9782100495801.
- Programmation en C des Pic. Christian Tavernier Dunod/L'Usine Nouvelle 2009. EAN13 : 97821007418931.
- Les alimentations électroniques. Pierre Mayé Collection: Technique et Ingénierie, Dunod 2012 - 2ème édition EAN13 : 9782100581559.
- Systèmes temps réel de contrôle-commande Conception et implémentation. Francis Cottet, Emmanuel Grolleau Collection: Technique et Ingénierie, Dunod/L'Usine Nouvelle 2005. EAN13 : 9782100078936.
- Aide-mémoire d'électronique analogique et numérique. Jean-Marc Poitevin Collection: Sciences Sup, Dunod 2008 2ème édition. EAN13 : 9782100515295
- Handbook of modern sensors. Physics, designs, and applications. JACOB FRADEN . Third edition
- Sensors handbook. Sabrie Soloman. Third edition. ISBN: 978-0-07-160571-7
- Aide-mémoire d'électronique générale . Bogdan Grabowski Collection: Aide-mémoire de l'ingénieur, Dunod/L'Usine Nouvelle 2008 - 5ème édition. EAN13 : 9782100519118
- <http://www.aviatechno.free.fr/thermo/>
- <http://www.datasheetcatalog.net>

Annexe

Table ITS-90 pour Thermocouple de Type S
Tension thermoélectrique en mV

°C	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
-50	-0.236										
-40	-0.194	-0.199	-0.203	-0.207	-0.211	-0.215	-0.219	-0.224	-0.228	-0.232	-0.236
-30	-0.150	-0.155	-0.159	-0.164	-0.168	-0.173	-0.177	-0.181	-0.186	-0.190	-0.194
-20	-0.103	-0.108	-0.113	-0.117	-0.122	-0.127	-0.132	-0.136	-0.141	-0.146	-0.150
-10	-0.053	-0.058	-0.063	-0.068	-0.073	-0.078	-0.083	-0.088	-0.093	-0.098	-0.103
0	0.000	-0.005	-0.011	-0.016	-0.021	-0.027	-0.032	-0.037	-0.042	-0.048	-0.053
°C	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
°C	0	1	2	3	4	5	6	7	8	9	10
0	0.000	0.005	0.011	0.016	0.022	0.027	0.033	0.038	0.044	0.050	0.055
10	0.055	0.061	0.067	0.072	0.078	0.084	0.090	0.095	0.101	0.107	0.113
20	0.113	0.119	0.125	0.131	0.137	0.143	0.149	0.155	0.161	0.167	0.173
30	0.173	0.179	0.185	0.191	0.197	0.204	0.210	0.216	0.222	0.229	0.235
40	0.235	0.241	0.248	0.254	0.260	0.267	0.273	0.280	0.286	0.292	0.299
50	0.299	0.305	0.312	0.319	0.325	0.332	0.338	0.345	0.352	0.358	0.365
60	0.365	0.372	0.378	0.385	0.392	0.399	0.405	0.412	0.419	0.426	0.433
70	0.433	0.440	0.446	0.453	0.460	0.467	0.474	0.481	0.488	0.495	0.502
80	0.502	0.509	0.516	0.523	0.530	0.538	0.545	0.552	0.559	0.566	0.573
90	0.573	0.580	0.588	0.595	0.602	0.609	0.617	0.624	0.631	0.639	0.646
°C	0	1	2	3	4	5	6	7	8	9	10
100	0.646	0.653	0.661	0.668	0.675	0.683	0.690	0.698	0.705	0.713	0.720
110	0.720	0.727	0.735	0.743	0.750	0.758	0.765	0.773	0.780	0.788	0.795
120	0.795	0.803	0.811	0.818	0.826	0.834	0.841	0.849	0.857	0.865	0.872
130	0.872	0.880	0.888	0.896	0.903	0.911	0.919	0.927	0.935	0.942	0.950
140	0.950	0.958	0.966	0.974	0.982	0.990	0.998	1.006	1.013	1.021	1.029
150	1.029	1.037	1.045	1.053	1.061	1.069	1.077	1.085	1.094	1.102	1.110
160	1.110	1.118	1.126	1.134	1.142	1.150	1.158	1.167	1.175	1.183	1.191
170	1.191	1.199	1.207	1.216	1.224	1.232	1.240	1.249	1.257	1.265	1.273
180	1.273	1.282	1.290	1.298	1.307	1.315	1.323	1.332	1.340	1.348	1.357
190	1.357	1.365	1.373	1.382	1.390	1.399	1.407	1.415	1.424	1.432	1.441
°C	0	1	2	3	4	5	6	7	8	9	10
200	1.441	1.449	1.458	1.466	1.475	1.483	1.492	1.500	1.509	1.517	1.526
210	1.526	1.534	1.543	1.551	1.560	1.569	1.577	1.586	1.594	1.603	1.612
220	1.612	1.620	1.629	1.638	1.646	1.655	1.663	1.672	1.681	1.690	1.698
230	1.698	1.707	1.716	1.724	1.733	1.742	1.751	1.759	1.768	1.777	1.786
240	1.786	1.794	1.803	1.812	1.821	1.829	1.838	1.847	1.856	1.865	1.874
250	1.874	1.882	1.891	1.900	1.909	1.918	1.927	1.936	1.944	1.953	1.962
260	1.962	1.971	1.980	1.989	1.998	2.007	2.016	2.025	2.034	2.043	2.052
270	2.052	2.061	2.070	2.078	2.087	2.096	2.105	2.114	2.123	2.132	2.141
280	2.141	2.151	2.160	2.169	2.178	2.187	2.196	2.205	2.214	2.223	2.232
290	2.232	2.241	2.250	2.259	2.268	2.277	2.287	2.296	2.305	2.314	2.323
°C	0	1	2	3	4	5	6	7	8	9	10
300	2.323	2.332	2.341	2.350	2.360	2.369	2.378	2.387	2.396	2.405	2.415
310	2.415	2.424	2.433	2.442	2.451	2.461	2.470	2.479	2.488	2.497	2.507
320	2.507	2.516	2.525	2.534	2.544	2.553	2.562	2.571	2.581	2.590	2.599
330	2.599	2.609	2.618	2.627	2.636	2.646	2.655	2.664	2.674	2.683	2.692
340	2.692	2.702	2.711	2.720	2.730	2.739	2.748	2.758	2.767	2.776	2.786
350	2.786	2.795	2.805	2.814	2.823	2.833	2.842	2.851	2.861	2.870	2.880
360	2.880	2.889	2.899	2.908	2.917	2.927	2.936	2.946	2.955	2.965	2.974
370	2.974	2.983	2.993	3.002	3.012	3.021	3.031	3.040	3.050	3.059	3.069
380	3.069	3.078	3.088	3.097	3.107	3.116	3.126	3.135	3.145	3.154	3.164
390	3.164	3.173	3.183	3.192	3.202	3.212	3.221	3.231	3.240	3.250	3.259
°C	0	1	2	3	4	5	6	7	8	9	10
400	3.259	3.269	3.279	3.288	3.298	3.307	3.317	3.326	3.336	3.346	3.355
410	3.355	3.365	3.374	3.384	3.394	3.403	3.413	3.423	3.432	3.442	3.451
420	3.451	3.461	3.471	3.480	3.490	3.500	3.509	3.519	3.529	3.538	3.548
430	3.548	3.558	3.567	3.577	3.587	3.596	3.606	3.616	3.626	3.635	3.645

440	3.645	3.655	3.664	3.674	3.684	3.694	3.703	3.713	3.723	3.732	3.742
450	3.742	3.752	3.762	3.771	3.781	3.791	3.801	3.810	3.820	3.830	3.840
460	3.840	3.850	3.859	3.869	3.879	3.889	3.898	3.908	3.918	3.928	3.938
470	3.938	3.947	3.957	3.967	3.977	3.987	3.997	4.006	4.016	4.026	4.036
480	4.036	4.046	4.056	4.065	4.075	4.085	4.095	4.105	4.115	4.125	4.134
490	4.134	4.144	4.154	4.164	4.174	4.184	4.194	4.204	4.213	4.223	4.233
°C	0	1	2	3	4	5	6	7	8	9	10
500	4.233	4.243	4.253	4.263	4.273	4.283	4.293	4.303	4.313	4.323	4.332
510	4.332	4.342	4.352	4.362	4.372	4.382	4.392	4.402	4.412	4.422	4.432
520	4.432	4.442	4.452	4.462	4.472	4.482	4.492	4.502	4.512	4.522	4.532
530	4.532	4.542	4.552	4.562	4.572	4.582	4.592	4.602	4.612	4.622	4.632
540	4.632	4.642	4.652	4.662	4.672	4.682	4.692	4.702	4.712	4.722	4.732
550	4.732	4.742	4.752	4.762	4.772	4.782	4.793	4.803	4.813	4.823	4.833
560	4.833	4.843	4.853	4.863	4.873	4.883	4.893	4.904	4.914	4.924	4.934
570	4.934	4.944	4.954	4.964	4.974	4.984	4.995	5.005	5.015	5.025	5.035
580	5.035	5.045	5.055	5.066	5.076	5.086	5.096	5.106	5.116	5.127	5.137
590	5.137	5.147	5.157	5.167	5.178	5.188	5.198	5.208	5.218	5.228	5.239
°C	0	1	2	3	4	5	6	7	8	9	10
600	5.239	5.249	5.259	5.269	5.280	5.290	5.300	5.310	5.320	5.331	5.341
610	5.341	5.351	5.361	5.372	5.382	5.392	5.402	5.413	5.423	5.433	5.443
620	5.443	5.454	5.464	5.474	5.485	5.495	5.505	5.515	5.526	5.536	5.546
630	5.546	5.557	5.567	5.577	5.588	5.598	5.608	5.618	5.629	5.639	5.649
640	5.649	5.660	5.670	5.680	5.691	5.701	5.712	5.722	5.732	5.743	5.753
650	5.753	5.763	5.774	5.784	5.794	5.805	5.815	5.826	5.836	5.846	5.857
660	5.857	5.867	5.878	5.888	5.898	5.909	5.919	5.930	5.940	5.950	5.961
670	5.961	5.971	5.982	5.992	6.003	6.013	6.024	6.034	6.044	6.055	6.065
680	6.065	6.076	6.086	6.097	6.107	6.118	6.128	6.139	6.149	6.160	6.170
690	6.170	6.181	6.191	6.202	6.212	6.223	6.233	6.244	6.254	6.265	6.275
°C	0	1	2	3	4	5	6	7	8	9	10
700	6.275	6.286	6.296	6.307	6.317	6.328	6.338	6.349	6.360	6.370	6.381
710	6.381	6.391	6.402	6.412	6.423	6.434	6.444	6.455	6.465	6.476	6.486
720	6.486	6.497	6.508	6.518	6.529	6.539	6.550	6.561	6.571	6.582	6.593
730	6.593	6.603	6.614	6.624	6.635	6.646	6.656	6.667	6.678	6.688	6.699
740	6.699	6.710	6.720	6.731	6.742	6.752	6.763	6.774	6.784	6.795	6.806
750	6.806	6.817	6.827	6.838	6.849	6.859	6.870	6.881	6.892	6.902	6.913
760	6.913	6.924	6.934	6.945	6.956	6.967	6.977	6.988	6.999	7.010	7.020
770	7.020	7.031	7.042	7.053	7.064	7.074	7.085	7.096	7.107	7.117	7.128
780	7.128	7.139	7.150	7.161	7.172	7.182	7.193	7.204	7.215	7.226	7.236
790	7.236	7.247	7.258	7.269	7.280	7.291	7.302	7.312	7.323	7.334	7.345
°C	0	1	2	3	4	5	6	7	8	9	10
800	7.345	7.356	7.367	7.378	7.388	7.399	7.410	7.421	7.432	7.443	7.454
810	7.454	7.465	7.476	7.487	7.497	7.508	7.519	7.530	7.541	7.552	7.563
820	7.563	7.574	7.585	7.596	7.607	7.618	7.629	7.640	7.651	7.662	7.673
830	7.673	7.684	7.695	7.706	7.717	7.728	7.739	7.750	7.761	7.772	7.783
840	7.783	7.794	7.805	7.816	7.827	7.838	7.849	7.860	7.871	7.882	7.893
850	7.893	7.904	7.915	7.926	7.937	7.948	7.959	7.970	7.981	7.992	8.003
860	8.003	8.014	8.026	8.037	8.048	8.059	8.070	8.081	8.092	8.103	8.114
870	8.114	8.125	8.137	8.148	8.159	8.170	8.181	8.192	8.203	8.214	8.226
880	8.226	8.237	8.248	8.259	8.270	8.281	8.293	8.304	8.315	8.326	8.337
890	8.337	8.348	8.360	8.371	8.382	8.393	8.404	8.416	8.427	8.438	8.449
°C	0	1	2	3	4	5	6	7	8	9	10
900	8.449	8.460	8.472	8.483	8.494	8.505	8.517	8.528	8.539	8.550	8.562
910	8.562	8.573	8.584	8.595	8.607	8.618	8.629	8.640	8.652	8.663	8.674
920	8.674	8.685	8.697	8.708	8.719	8.731	8.742	8.753	8.765	8.776	8.787
930	8.787	8.798	8.810	8.821	8.832	8.844	8.855	8.866	8.878	8.889	8.900
940	8.900	8.912	8.923	8.935	8.946	8.957	8.969	8.980	8.991	9.003	9.014
950	9.014	9.025	9.037	9.048	9.060	9.071	9.082	9.094	9.105	9.117	9.128
960	9.128	9.139	9.151	9.162	9.174	9.185	9.197	9.208	9.219	9.231	9.242
970	9.242	9.254	9.265	9.277	9.288	9.300	9.311	9.323	9.334	9.345	9.357
980	9.357	9.368	9.380	9.391	9.403	9.414	9.426	9.437	9.449	9.460	9.472
990	9.472	9.483	9.495	9.506	9.518	9.529	9.541	9.552	9.564	9.576	9.587
°C	0	1	2	3	4	5	6	7	8	9	10
1000	9.587	9.599	9.610	9.622	9.633	9.645	9.656	9.668	9.680	9.691	9.703
1010	9.703	9.714	9.726	9.737	9.749	9.761	9.772	9.784	9.795	9.807	9.819
1020	9.819	9.830	9.842	9.853	9.865	9.877	9.888	9.900	9.911	9.923	9.935
1030	9.935	9.946	9.958	9.970	9.981	9.993	10.005	10.016	10.028	10.040	10.051
1040	10.051	10.063	10.075	10.086	10.098	10.110	10.121	10.133	10.145	10.156	10.168
1050	10.168	10.180	10.191	10.203	10.215	10.227	10.238	10.250	10.262	10.273	10.285
1060	10.285	10.297	10.309	10.320	10.332	10.344	10.356	10.367	10.379	10.391	10.403
1070	10.403	10.414	10.426	10.438	10.450	10.461	10.473	10.485	10.497	10.509	10.520

1080	10.520	10.532	10.544	10.556	10.567	10.579	10.591	10.603	10.615	10.626	10.638
1090	10.638	10.650	10.662	10.674	10.686	10.697	10.709	10.721	10.733	10.745	10.757
°C	0	1	2	3	4	5	6	7	8	9	10
1100	10.757	10.768	10.780	10.792	10.804	10.816	10.828	10.839	10.851	10.863	10.875
1110	10.875	10.887	10.899	10.911	10.922	10.934	10.946	10.958	10.970	10.982	10.994
1120	10.994	11.006	11.017	11.029	11.041	11.053	11.065	11.077	11.089	11.101	11.113
1130	11.113	11.125	11.136	11.148	11.160	11.172	11.184	11.196	11.208	11.220	11.232
1140	11.232	11.244	11.256	11.268	11.280	11.291	11.303	11.315	11.327	11.339	11.351
1150	11.351	11.363	11.375	11.387	11.399	11.411	11.423	11.435	11.447	11.459	11.471
1160	11.471	11.483	11.495	11.507	11.519	11.531	11.542	11.554	11.566	11.578	11.590
1170	11.590	11.602	11.614	11.626	11.638	11.650	11.662	11.674	11.686	11.698	11.710
1180	11.710	11.722	11.734	11.746	11.758	11.770	11.782	11.794	11.806	11.818	11.830
1190	11.830	11.842	11.854	11.866	11.878	11.890	11.902	11.914	11.926	11.939	11.951
°C	0	1	2	3	4	5	6	7	8	9	10
1200	11.951	11.963	11.975	11.987	11.999	12.011	12.023	12.035	12.047	12.059	12.071
1210	12.071	12.083	12.095	12.107	12.119	12.131	12.143	12.155	12.167	12.179	12.191
1220	12.191	12.203	12.216	12.228	12.240	12.252	12.264	12.276	12.288	12.300	12.312
1230	12.312	12.324	12.336	12.348	12.360	12.372	12.384	12.397	12.409	12.421	12.433
1240	12.433	12.445	12.457	12.469	12.481	12.493	12.505	12.517	12.529	12.542	12.554
1250	12.554	12.566	12.578	12.590	12.602	12.614	12.626	12.638	12.650	12.662	12.675
1260	12.675	12.687	12.699	12.711	12.723	12.735	12.747	12.759	12.771	12.783	12.796
1270	12.796	12.808	12.820	12.832	12.844	12.856	12.868	12.880	12.892	12.905	12.917
1280	12.917	12.929	12.941	12.953	12.965	12.977	12.989	13.001	13.014	13.026	13.038
1290	13.038	13.050	13.062	13.074	13.086	13.098	13.111	13.123	13.135	13.147	13.159
°C	0	1	2	3	4	5	6	7	8	9	10
1300	13.159	13.171	13.183	13.195	13.208	13.220	13.232	13.244	13.256	13.268	13.280
1310	13.280	13.292	13.305	13.317	13.329	13.341	13.353	13.365	13.377	13.390	13.402
1320	13.402	13.414	13.426	13.438	13.450	13.462	13.474	13.487	13.499	13.511	13.523
1330	13.523	13.535	13.547	13.559	13.572	13.584	13.596	13.608	13.620	13.632	13.644
1340	13.644	13.657	13.669	13.681	13.693	13.705	13.717	13.729	13.742	13.754	13.766
1350	13.766	13.778	13.790	13.802	13.814	13.826	13.839	13.851	13.863	13.875	13.887
1360	13.887	13.899	13.911	13.924	13.936	13.948	13.960	13.972	13.984	13.996	14.009
1370	14.009	14.021	14.033	14.045	14.057	14.069	14.081	14.094	14.106	14.118	14.130
1380	14.130	14.142	14.154	14.166	14.178	14.191	14.203	14.215	14.227	14.239	14.251
1390	14.251	14.263	14.276	14.288	14.300	14.312	14.324	14.336	14.348	14.360	14.373
°C	0	1	2	3	4	5	6	7	8	9	10
1400	14.373	14.385	14.397	14.409	14.421	14.433	14.445	14.457	14.470	14.482	14.494
1410	14.494	14.506	14.518	14.530	14.542	14.554	14.567	14.579	14.591	14.603	14.615
1420	14.615	14.627	14.639	14.651	14.664	14.676	14.688	14.700	14.712	14.724	14.736
1430	14.736	14.748	14.760	14.773	14.785	14.797	14.809	14.821	14.833	14.845	14.857
1440	14.857	14.869	14.881	14.894	14.906	14.918	14.930	14.942	14.954	14.966	14.978
1450	14.978	14.990	15.002	15.015	15.027	15.039	15.051	15.063	15.075	15.087	15.099
1460	15.099	15.111	15.123	15.135	15.148	15.160	15.172	15.184	15.196	15.208	15.220
1470	15.220	15.232	15.244	15.256	15.268	15.280	15.292	15.304	15.317	15.329	15.341
1480	15.341	15.353	15.365	15.377	15.389	15.401	15.413	15.425	15.437	15.449	15.461
1490	15.461	15.473	15.485	15.497	15.509	15.521	15.534	15.546	15.558	15.570	15.582
°C	0	1	2	3	4	5	6	7	8	9	10
1500	15.582	15.594	15.606	15.618	15.630	15.642	15.654	15.666	15.678	15.690	15.702
1510	15.702	15.714	15.726	15.738	15.750	15.762	15.774	15.786	15.798	15.810	15.822
1520	15.822	15.834	15.846	15.858	15.870	15.882	15.894	15.906	15.918	15.930	15.942
1530	15.942	15.954	15.966	15.978	15.990	16.002	16.014	16.026	16.038	16.050	16.062
1540	16.062	16.074	16.086	16.098	16.110	16.122	16.134	16.146	16.158	16.170	16.182
1550	16.182	16.194	16.205	16.217	16.229	16.241	16.253	16.265	16.277	16.289	16.301
1560	16.301	16.313	16.325	16.337	16.349	16.361	16.373	16.385	16.396	16.408	16.420
1570	16.420	16.432	16.444	16.456	16.468	16.480	16.492	16.504	16.516	16.527	16.539
1580	16.539	16.551	16.563	16.575	16.587	16.599	16.611	16.623	16.634	16.646	16.658
1590	16.658	16.670	16.682	16.694	16.706	16.718	16.729	16.741	16.753	16.765	16.777
°C	0	1	2	3	4	5	6	7	8	9	10
1600	16.777	16.789	16.801	16.812	16.824	16.836	16.848	16.860	16.872	16.883	16.895
1610	16.895	16.907	16.919	16.931	16.943	16.954	16.966	16.978	16.990	17.002	17.013
1620	17.013	17.025	17.037	17.049	17.061	17.072	17.084	17.096	17.108	17.120	17.131
1630	17.131	17.143	17.155	17.167	17.178	17.190	17.202	17.214	17.225	17.237	17.249
1640	17.249	17.261	17.272	17.284	17.296	17.308	17.319	17.331	17.343	17.355	17.366
1650	17.366	17.378	17.390	17.401	17.413	17.425	17.437	17.448	17.460	17.472	17.483
1660	17.483	17.495	17.507	17.518	17.530	17.542	17.553	17.565	17.577	17.588	17.600
1670	17.600	17.612	17.623	17.635	17.647	17.658	17.670	17.682	17.693	17.705	17.717
1680	17.717	17.728	17.740	17.751	17.763	17.775	17.786	17.798	17.809	17.821	17.832
1690	17.832	17.844	17.855	17.867	17.878	17.890	17.901	17.913	17.924	17.936	17.947
°C	0	1	2	3	4	5	6	7	8	9	10
1700	17.947	17.959	17.970	17.982	17.993	18.004	18.016	18.027	18.039	18.050	18.061

FEATURES

- Microprocessor Compatible (6800, 8085, Z80, Etc.)
- TTL/CMOS Compatible Inputs
- On-Chip Data Latches
- Endpoint Linearity
- Low Power Consumption
- Monotonicity Guaranteed (Full Temperature Range)
- Latch Free (No Protection Schottky Required)

APPLICATIONS

- Microprocessor Controlled Gain Circuits
- Microprocessor Controlled Attenuator Circuits
- Microprocessor Controlled Function Generation
- Precision AGC Circuits
- Bus Structured Instruments

GENERAL DESCRIPTION

The AD7524 is a low cost, 8-bit monolithic CMOS DAC designed for direct interface to most microprocessors.

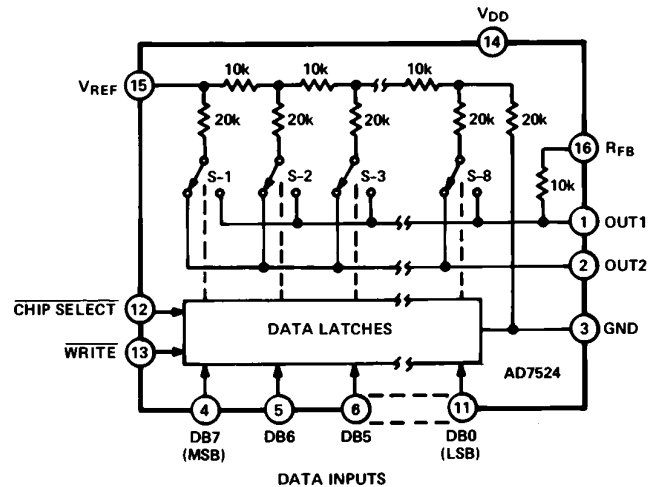
Basically an 8-bit DAC with input latches, the AD7524's load cycle is similar to the "write" cycle of a random access memory. Using an advanced thin-film on CMOS fabrication process, the AD7524 provides accuracy to 1/8 LSB with a typical power dissipation of less than 10 milliwatts.

A newly improved design eliminates the protection Schottky previously required and guarantees TTL compatibility when using a +5 V supply. Loading speed has been increased for compatibility with most microprocessors.

Featuring operation from +5 V to +15 V, the AD7524 interfaces directly to most microprocessor buses or output ports.

Excellent multiplying characteristics (2- or 4-quadrant) make the AD7524 an ideal choice for many microprocessor controlled gain setting and signal control applications.

FUNCTIONAL BLOCK DIAGRAM



ORDERING GUIDE

Model ¹	Temperature Range	Nonlinearity (V _{DD} = +15 V)	Package Option ²
AD7524JN	-40°C to +85°C	±1/2 LSB	N-16
AD7524KN	-40°C to +85°C	±1/4 LSB	N-16
AD7524LN	-40°C to +85°C	±1/8 LSB	N-16
AD7524JP	-40°C to +85°C	±1/2 LSB	P-20A
AD7524KP	-40°C to +85°C	±1/4 LSB	P-20A
AD7524LP	-40°C to +85°C	±1/8 LSB	P-20A
AD7524JR	-40°C to +85°C	±1/2 LSB	R-16A
AD7524AQ	-40°C to +85°C	±1/2 LSB	Q-16
AD7524BQ	-40°C to +85°C	±1/4 LSB	Q-16
AD7524CQ	-40°C to +85°C	±1/8 LSB	Q-16
AD7524SQ	-55°C to +125°C	±1/2 LSB	Q-16
AD7524TQ	-55°C to +125°C	±1/4 LSB	Q-16
AD7524UQ	-55°C to +125°C	±1/8 LSB	Q-16
AD7524SE	-55°C to +125°C	±1/2 LSB	E-20A
AD7524TE	-55°C to +125°C	±1/4 LSB	E-20A
AD7524UE	-55°C to +125°C	±1/8 LSB	E-20A

NOTES

¹To order MIL-STD-883, Class B processed parts, add/883B to part number. Contact your local sales office for military data sheet. For U.S. Standard Military Drawing (SMD) see DESC drawing #5962-87700.

²E = Leadless Ceramic Chip Carrier; N = Plastic DIP; P = Plastic Leaded Chip Carrier; Q = Cerdip; R = SOIC.

REV. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

AD7524–SPECIFICATIONS ($V_{REF} = +10\text{ V}$, $V_{OUT1} = V_{OUT2} = 0\text{ V}$, unless otherwise noted)

Parameter	Limit, $T_A = +25^\circ\text{C}$		Limit, T_{MIN}, T_{MAX}^1		Units	Test Conditions/Comments
	$V_{DD} = +5\text{ V}$	$V_{DD} = +15\text{ V}$	$V_{DD} = 5\text{ V}$	$V_{DD} = +15\text{ V}$		
STATIC PERFORMANCE						
Resolution	8	8	8	8	Bits	
Relative Accuracy						
J, A, S Versions	$\pm 1/2$	$\pm 1/2$	$\pm 1/2$	$\pm 1/2$	LSB max	
K, B, T Versions	$\pm 1/2$	$\pm 1/4$	$\pm 1/2$	$\pm 1/4$	LSB max	
L, C, U Versions	$\pm 1/2$	$\pm 1/8$	$\pm 1/2$	$\pm 1/8$	LSB max	
Monotonicity	Guaranteed	Guaranteed	Guaranteed	Guaranteed		
Gain Error ²	$\pm 2\ 1/2$	$\pm 1\ 1/4$	$\pm 3\ 1/2$	$\pm 1\ 1/2$	LSB max	
Average Gain TC ³	± 40	± 10	± 40	± 10	ppm/ $^\circ\text{C}$	Gain TC Measured from $+25^\circ\text{C}$ to T_{MIN} or from $+25^\circ\text{C}$ to T_{MAX} $\Delta V_{DD} = \pm 10\%$
DC Supply Rejection, ³ $\Delta\text{Gain}/\Delta V_{DD}$	0.08 0.002	0.02 0.001	0.16 0.01	0.04 0.005	% FSR/% max % FSR/% typ	
Output Leakage Current						
I_{OUT1} (Pin 1)	± 50	± 50	± 400	± 200	nA max	DB0-DB7 = 0 V; $\overline{\text{WR}}, \overline{\text{CS}} = 0\text{ V}$; $V_{REF} = \pm 10\text{ V}$
I_{OUT2} (Pin 2)	± 50	± 50	± 400	± 200	nA max	DB0-DB7 = V_{DD} ; $\overline{\text{WR}}, \overline{\text{CS}} = 0\text{ V}$; $V_{REF} = \pm 10\text{ V}$
DYNAMIC PERFORMANCE						
Output Current Settling Time ³ (to 1/2 LSB)	400	250	500	350	ns max	OUT1 Load = 100 Ω , $C_{EXT} = 13\text{ pF}$; $\overline{\text{WR}}, \overline{\text{CS}} = 0\text{ V}$; DB0-DB7 = 0 V to V_{DD} to 0 V.
AC Feedthrough ³						
at OUT1	0.25	0.25	0.5	0.5	% FSR max	$V_{REF} = \pm 10\text{ V}$, 100 kHz Sine Wave; DB0-DB7 = 0 V; $\overline{\text{WR}}, \overline{\text{CS}} = 0\text{ V}$
at OUT2	0.25	0.25	0.5	0.5	% FSR max	
REFERENCE INPUT						
R_{IN} (Pin 15 to GND) ⁴	5 20	5 20	5 20	5 20	k Ω min k Ω max	
ANALOG OUTPUTS						
Output Capacitance ³						
C_{OUT1} (Pin 1)	120	120	120	120	pF max	DB0-DB7 = V_{DD} ; $\overline{\text{WR}}, \overline{\text{CS}} = 0\text{ V}$
C_{OUT2} (Pin 2)	30	30	30	30	pF max	
C_{OUT1} (Pin 1)	30	30	30	30	pF max	DB0-DB7 = 0 V; $\overline{\text{WR}}, \overline{\text{CS}} = 0\text{ V}$
C_{OUT2} (Pin 2)	120	120	120	120	pF max	
DIGITAL INPUTS						
Input HIGH Voltage Requirement V_{IH}	+2.4	+13.5	+2.4	+13.5	V min	
Input LOW Voltage Requirement V_{IL}	+0.8	+1.5	+0.5	+1.5	V max	
Input Current I_{IN}	± 1	± 1	± 10	± 10	μA max	$V_{IN} = 0\text{ V}$ or V_{DD}
Input Capacitance ³						
DB0-DB7	5	5	5	5	pF max	$V_{IN} = 0\text{ V}$
$\overline{\text{WR}}, \overline{\text{CS}}$	20	20	20	20	pF max	$V_{IN} = 0\text{ V}$
SWITCHING CHARACTERISTICS						
Chip Select to Write Setup Time ⁵ t_{CS}						See Timing Diagram $t_{WR} = t_{CS}$
AD7524J, K, L, A, B, C	170	100	220	130	ns min	
AD7524S, T, U	170	100	240	150	ns min	
Chip Select to Write Hold Time t_{CH}						
All Grades	0	0	0	0	ns min	
Write Pulse Width t_{WR}						$t_{CS} \geq t_{WR}$, $t_{CH} \geq 0$
AD7524J, K, L, A, B, C	170	100	220	130	ns min	
AD7524S, T, U	170	100	240	150	ns min	
Data Setup Time t_{DS}						
AD7524J, K, L, A, B, C	135	60	170	80	ns min	
AD7524S, T, U	135	60	170	100	ns min	
Data Hold Time t_{DH}						
All Grades	10	10	10	10	ns min	
POWER SUPPLY						
I_{DD}	1 100	2 100	2 500	2 500	mA max μA max	All Digital Inputs V_{IL} or V_{IH} All Digital Inputs 0 V or V_{DD}

NOTES

¹Temperature ranges as follows: J, K, L versions: -40°C to $+85^\circ\text{C}$
A, B, C versions: -40°C to $+85^\circ\text{C}$
S, T, U versions: -55°C to $+125^\circ\text{C}$

²Gain error is measured using internal feedback resistor. Full-Scale Range (FSR) = V_{REF} .

³Guaranteed not tested.

⁴DAC thin-film resistor temperature coefficient is approximately $-300\text{ ppm}/^\circ\text{C}$.

⁵AC parameter, sample tested @ $+25^\circ\text{C}$ to ensure conformance to specification.

Specifications subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*

(T_A = +25°C, unless otherwise noted)

V _{DD} to GND	-0.3 V, +17 V
V _{RFB} to GND	±25 V
V _{REF} to GND	±25 V
Digital Input Voltage to GND	-0.3 V to V _{DD} +0.3 V
OUT1, OUT2 to GND	-0.3 V to V _{DD} +0.3 V

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Power Dissipation (Any Package)

To +75°C	450 mW
Derates above 75°C by	6 mW/°C
Operating Temperature		
Commercial (J, K, L)	-40°C to +85°C
Industrial (A, B, C)	-40°C to +85°C
Extended (S, T, U)	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Lead Temperature (Soldering, 10 secs)	+300°C

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the AD7524 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.



TERMINOLOGY

RELATIVE ACCURACY: A measure of the deviation from a straight line through the end points of the DAC transfer function. Normally expressed as a percentage of full scale range. For the AD7524 DAC, this holds true over the entire V_{REF} range.

RESOLUTION: Value of the LSB. For example, a unipolar converter with n bits has a resolution of (2⁻ⁿ) (V_{REF}). A bipolar converter of n bits has a resolution of [2⁻⁽ⁿ⁻¹⁾] [V_{REF}]. Resolution in no way implies linearity.

GAIN ERROR: Gain Error is a measure of the output error between an ideal DAC and the actual device output. It is measured

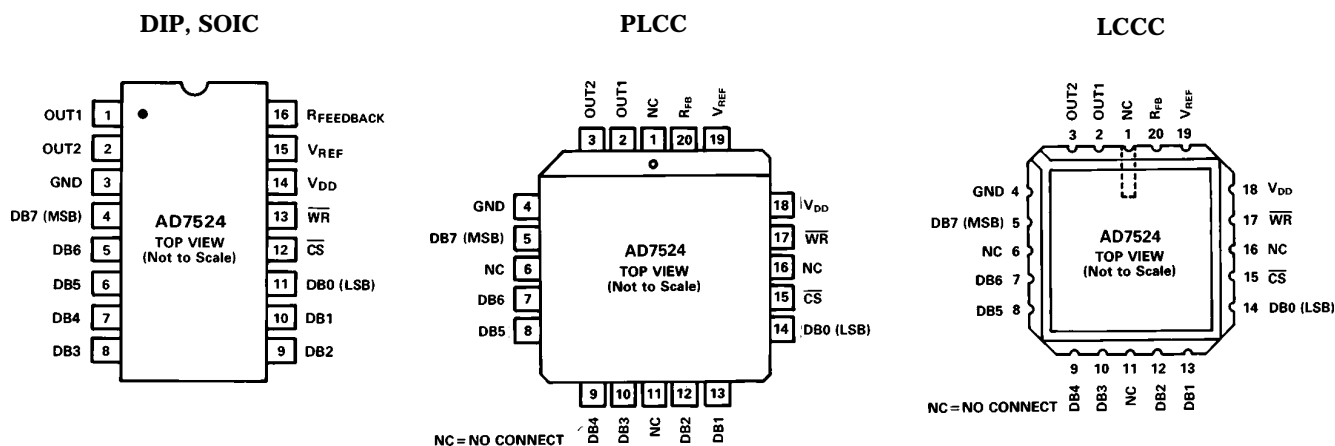
with all 1s in the DAC after offset error has been adjusted out and is expressed in LSBs. Gain Error is adjustable to zero with an external potentiometer.

FEEDTHROUGH ERROR: Error caused by capacitive coupling from V_{REF} to output with all switches OFF.

OUTPUT CAPACITANCE: Capacity from OUT1 and OUT2 terminals to ground.

OUTPUT LEAKAGE CURRENT: Current which appears on OUT1 terminal with all digital inputs LOW or on OUT2 terminal when all inputs are HIGH. This is an error current which contributes an offset voltage at the amplifier output.

PIN CONFIGURATIONS



AD7524

CIRCUIT DESCRIPTION

CIRCUIT INFORMATION

The AD7524, an 8-bit multiplying D/A converter, consists of a highly stable thin film R-2R ladder and eight N-channel current switches on a monolithic chip. Most applications require the addition of only an output operational amplifier and a voltage or current reference.

The simplified D/A circuit is shown in Figure 1. An inverted R-2R ladder structure is used—that is, the binary weighted currents are switched between the OUT1 and OUT2 bus lines, thus maintaining a constant current in each ladder leg independent of the switch state.

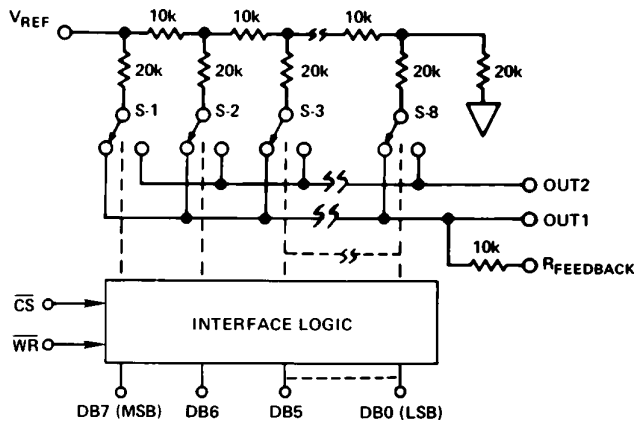


Figure 1. Functional Diagram

EQUIVALENT CIRCUIT ANALYSIS

The equivalent circuit for all digital inputs LOW is shown in Figures 2. In Figure 2 with all digital inputs LOW, the reference current is switched to OUT2. The current source $I_{LEAKAGE}$ is composed of surface and junction leakages to the substrate

while the $\frac{1}{256}$ current source represents a constant 1-bit current drain through the termination resistor on the R-2R ladder. The “ON” capacitance of the output N-channel switches is 120 pF, as shown on the OUT2 terminal. The “OFF” switch capacitance is 30 pF, as shown on the OUT1 terminal. Analysis of the circuit for all digital inputs high is similar to Figure 2 however, the “ON” switches are now on terminal OUT1, hence the 120 pF appears at that terminal.

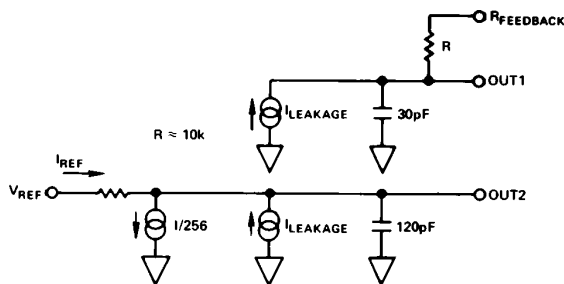


Figure 2. AD7524 DAC Equivalent Circuit—All Digital Inputs Low

INTERFACE LOGIC INFORMATION

MODE SELECTION

AD7524 mode selection is controlled by the \overline{CS} and \overline{WR} inputs.

WRITE MODE

When \overline{CS} and \overline{WR} are both LOW, the AD7524 is in the WRITE mode, and the AD7524 analog output responds to data activity at the DB0–DB7 data bus inputs. In this mode, the AD7524 acts like a nonlatched input D/A converter.

HOLD MODE

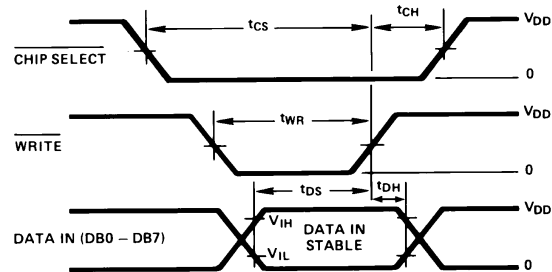
When either \overline{CS} or \overline{WR} is HIGH, the AD7524 is in the HOLD mode. The AD7524 analog output holds the value corresponding to the last digital input present at DB0–DB7 prior to \overline{WR} or \overline{CS} assuming the HIGH state.

MODE SELECTION TABLE

\overline{CS}	\overline{WR}	Mode	DAC Response
L	L	Write	DAC responds to data bus (DB0–DB7) inputs.
H	X	Hold	Data bus (DB0–DB7) is Locked Out:
X	H	Hold	DAC holds last data present when \overline{WR} or \overline{CS} assumed HIGH state.

L = Low State, H = High State, X = Don't Care.

WRITE CYCLE TIMING DIAGRAM



- NOTES:
- All input signal rise and fall times measured from 10% to 90% of V_{DD} . $V_{DD} = +5V$, $t_r = t_f = 20ns$; $V_{DD} = +15V$, $t_r = t_f = 40ns$.
 - Timing Measurement Reference level is $\frac{V_{IH} + V_{IL}}{2}$.
 - $t_{DS} + t_{DH}$ is approximately constant at 145ns min at $+25^\circ C$, $V_{DD} = +5V$ and $t_{WR} = 170ns$ min. The AD7524 is specified for a minimum t_{DH} of 10ns, however, in applications where $t_{DH} > 10ns$, t_{DS} may be reduced accordingly up to the limit $t_{DS} = 65ns$, $t_{DH} = 80ns$.

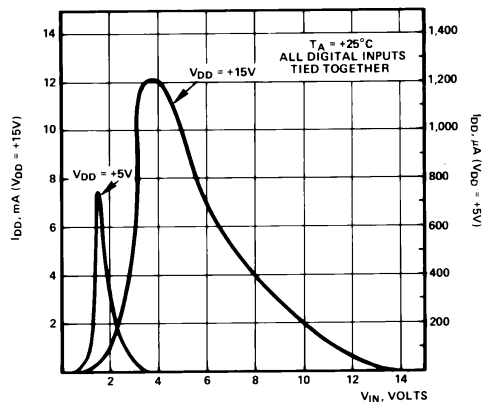


Figure 3. Supply Current vs. Logic Level
Typical plots of supply current, I_{DD} , versus logic input voltage, V_{IN} , for $V_{DD} = +5V$ and $V_{DD} = +15V$ are shown above.

ANALOG CIRCUIT CONNECTIONS

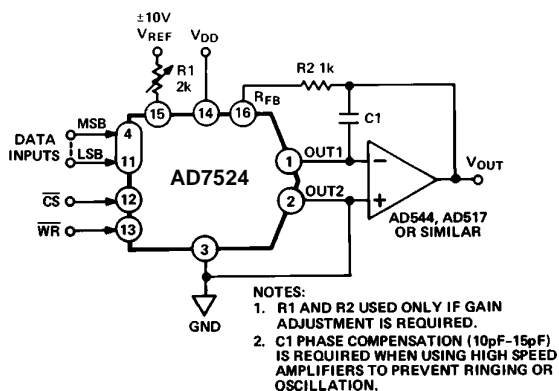


Figure 4. Unipolar Binary Operation (2-Quadrant Multiplication)

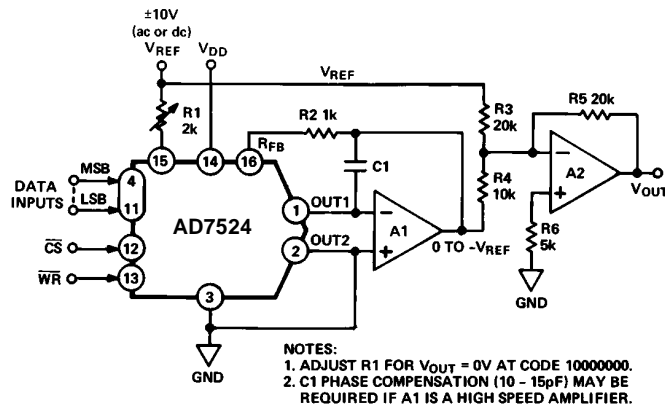


Figure 5. Bipolar (4-Quadrant) Operation

Table I. Unipolar Binary Code Table

Digital Input MSB LSB	Analog Output
1111 1111	$-V_{REF}$ (255/256)
1000 0001	$-V_{REF}$ (129/256)
1000 0000	$-V_{REF}$ (128/256) = $-V_{REF}/2$
0111 1111	$-V_{REF}$ (127/256)
0000 0001	$-V_{REF}$ (1/256)
0000 0000	$-V_{REF}$ (0/256) = 0

Note: 1 LSB = $(2^{-8})(V_{REF}) = 1/256 (V_{REF})$

Table II. Bipolar (Offset Binary) Code Table

Digital Input MSB LSB	Analog Output
1111 1111	$+V_{REF}$ (127/128)
1000 0001	$+V_{REF}$ (1/128)
1000 0000	0
0111 1111	$-V_{REF}$ (1/128)
0000 0001	$-V_{REF}$ (127/128)
0000 0000	$-V_{REF}$ (128/128)

Note: 1 LSB = $(2^{-7})(V_{REF}) = 1/128 (V_{REF})$

MICROPROCESSOR INTERFACE

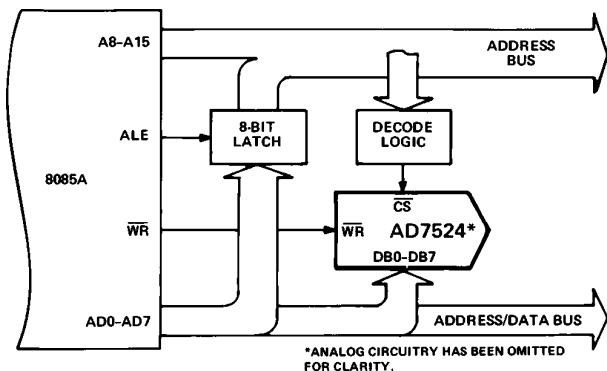


Figure 6. AD7524/8085A Interface

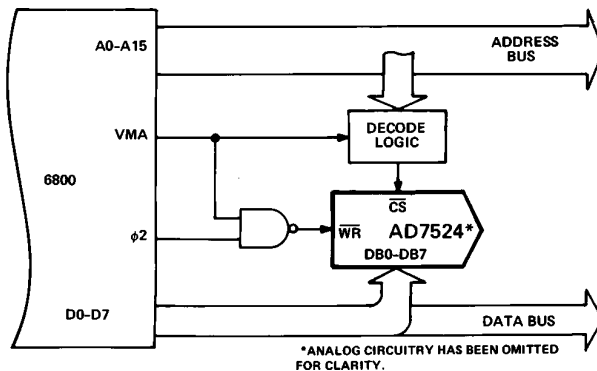


Figure 7. AD7524/MC6800 Interface

POWER GENERATION

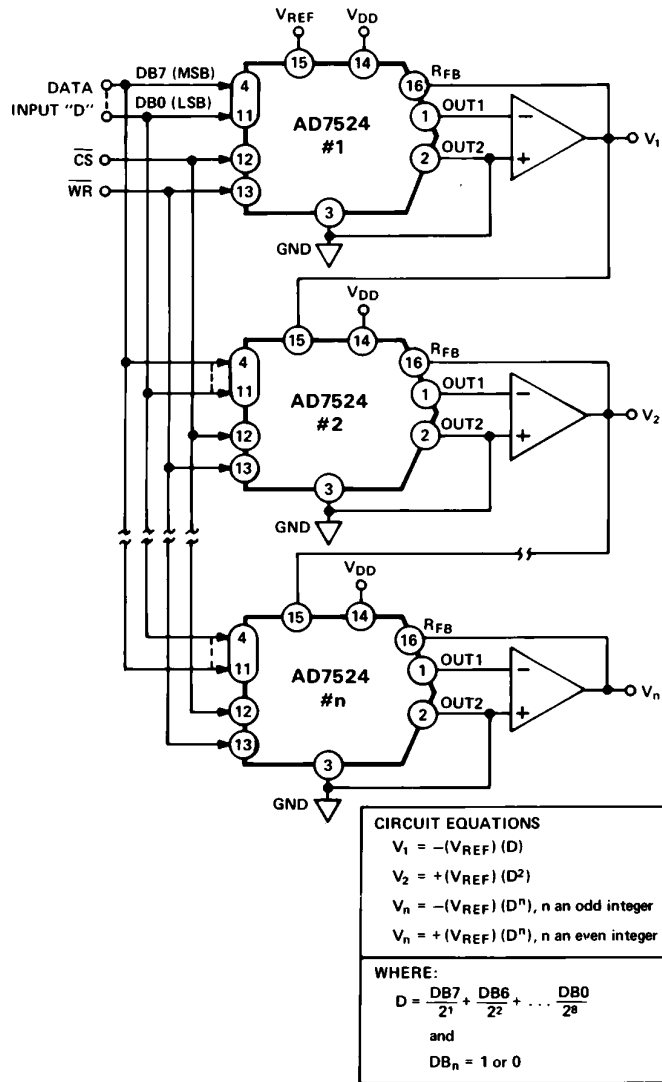


Figure 8.

LE BUS I²C

Introduction

Le bus I²C (Inter Integrated Circuit) fait partie des bus série : 3 fils pour faire tout passer. Il a été développé au début des années 1980, par Philips pour minimiser les liaisons entre les circuits intégrés numériques de ses produits (Téléviseurs, éléments HiFi, magnétoscopes, ...).

Aujourd'hui, Philips a dans son catalogue plus de 150 circuits intégrés CMOS et bipolaires qui sont compatibles I²C. D'autres fabricants ont aussi développé des circuits intégrés qui peuvent être connectés au bus I²C. Le protocole utilisé est simple et rapide.

Que demander de plus ? Multimaster ? Pas de problèmes, c'est l'une des caractéristiques standard du bus I²C.

Caractéristiques

Le bus I²C permet de faire communiquer entre eux des composants électroniques très divers grâce à seulement trois fils : un signal de données (SDA), un signal d'horloge (SCL), et un signal de référence électrique (masse).

Il s'agit d'une liaison en mode série, ce qui signifie que la vitesse de transfert sera plus faible qu'avec un bus de type parallèle. Le bus I²C permet cependant des échanges à la vitesse de 100 kbits par seconde. Certes, la vitesse de transfert du bus I²C n'est pas fulgurante, mais dans bien des cas, la vitesse n'est pas l'élément prédominant.

L'utilisation d'un bus I²C permet de réduire la complexité des circuits imprimés à réaliser. Par exemple, pour connecter une EEPROM ou une RAM à un microcontrôleur classique, il faut relier entre eux les bits de données et les bits d'adresses des différents composants, et, en plus, il faut bien souvent ajouter une logique de sélection. Avec des composants prévus pour se connecter au bus I²C, il suffit de les relier par deux pistes seulement. Si, par la suite, on souhaite ajouter des composants sur le circuit, le nombre de pistes à ajouter sera vraiment plus réduit (essayez d'ajouter une EEPROM sur un circuit existant pour voir).

De nombreux fabricants ayant adopté le système la variété des systèmes disponibles disposant d'un port I²C est énorme :

- microcontrôleurs
- expandeurs de bus (entrée/sortie 8 bits)
- convertisseurs A/N et N/A
- mémoires (RAM, EPROM, EEPROM, etc.)
- récepteurs infra-rouge (télécommande RC5)
- capteurs de température
- circuits audio (égaliseur, contrôle de volume, etc.)
- drivers d'affichage LCD ou à LEDs
- décodeurs télétexte
- chargeurs de batterie
- etc.

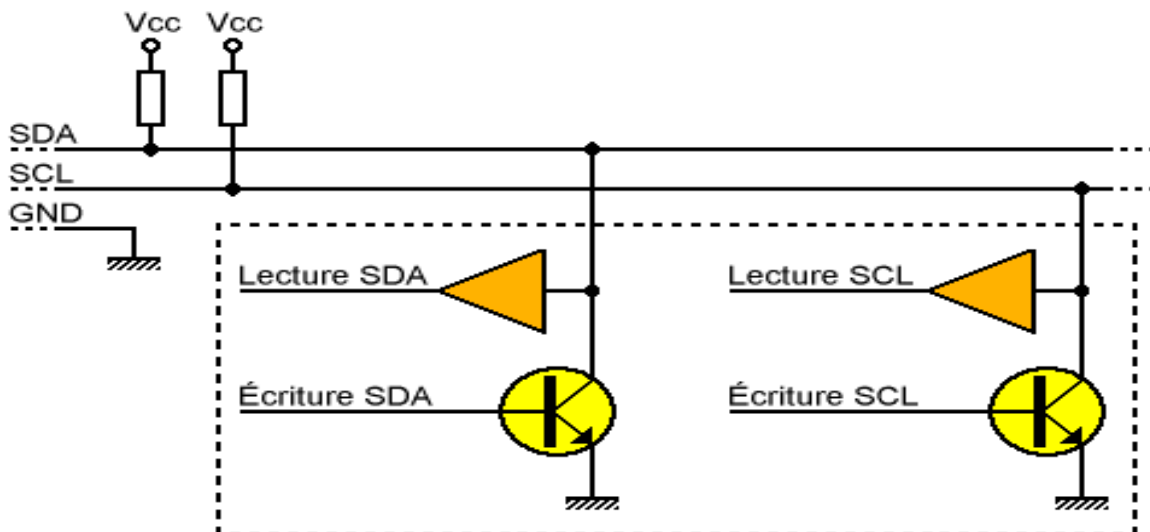
Le nombre de composants qu'il est possible de relier est essentiellement limité par la charge capacitive des lignes SDA et SCL : 400pF.

Le bus I²C a encore le vent en poupe car il est de plus en plus utilisé dans l'électronique grand-public, parfois déguisé sous une norme pour des besoins particuliers comme le SMBus qui est implanté dans tous les nouveaux PC, ou encore le fameux DDC qui équipe tous les moniteurs et cartes vidéos récents.

Le support physique utilisé

Comme indiqué précédemment, pour se connecter à un bus I²C il faut une masse, et deux fils de communication. Le premier fil, SDA (Signal Data), est utilisé pour transmettre les données. L'autre fil, SCL (Signal Clock) est utilisé pour transmettre un signal d'horloge synchrone (signal qui indique le rythme d'évolution de la ligne SDA). Les tensions associées aux niveaux logiques vont dépendre de la technologie des circuits en présence (CMOS, TTL). Il faudra que tous les circuits connectés au bus I²C utilisent les mêmes potentiels pour définir les niveaux haut et bas. En définitive, cela implique que tous les composants connectés à un même bus soient alimentés de façon identique. Cela ne signifie pas que les composants doivent utiliser la même source pour s'alimenter ; il suffit que la tension d'alimentation soit à la même valeur pour tous les composants, le fil de masse permettant d'unifier les références.

Il reste maintenant un problème crucial. Comment permettre à plusieurs circuits logiques de connecter leurs sorties ensemble, sachant que certains circuits voudront imposer un niveau haut tandis que d'autres voudront imposer un niveau bas ? La réponse est connue depuis longtemps. Il faut utiliser des sorties à collecteur ouvert (ou à drain ouvert pour des circuits CMOS). Le niveau résultant sur la ligne est alors une fonction «ET» de toutes les sorties connectées.



Les résistances de rappel au potentiel VCC permettent aux signaux SDA et SCL d'être à 1 si toutes les sorties à collecteurs ouverts sont aussi au niveau 1 (résultat de la fonction «ET»). Si une ou plusieurs sorties tentent d'imposer un niveau bas sur une ligne, le ou les transistors associés vont conduire, ce qui entraîne un niveau bas sur la ligne correspondante (ce qui est conforme au résultat de la fonction « ET »).

En ce qui concerne la lecture des signaux SDA et SCL, cela ne pose pas de problème. Les signaux peuvent être lus en permanence sans risque d'interférer sur le niveau de la ligne.

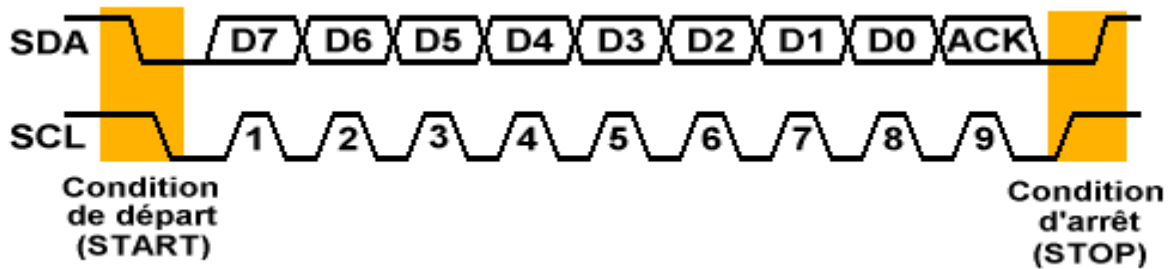
Au repos, tous les circuits connectés doivent imposer un niveau haut sur leurs sorties respectives. Si les lignes SDA et SCL sont au niveau haut dans ces conditions, cela signifie qu'aucun circuit ne tente de prendre le contrôle du bus. Si une des lignes SDA ou SCL passe à un niveau bas dans les mêmes conditions, c'est qu'un des circuits désire prendre le contrôle du bus. Mais il peut aussi y avoir deux circuits qui tentent de prendre le contrôle du bus en même temps (ou à quelques nanosecondes d'écart près). Il faut donc mettre en place un protocole pour gérer les conflits possibles.

Le protocole I²C

Le protocole du bus I²C définit la succession des états possibles sur les lignes SDA et SCL, et comment doivent réagir les circuits en cas de conflit.

La prise de contrôle du bus

Pour transmettre des données sur le bus I²C, il faut surveiller deux conditions particulières : la condition de départ et la condition d'arrêt.



Avant de tenter de prendre le contrôle du bus, un circuit doit vérifier que les lignes SDA et SCL sont au repos, c'est-à-dire à l'état haut. Si c'est le cas, le circuit indique qu'il prend le contrôle du bus en mettant la ligne SDA à 0. A partir de ce moment là, les autres circuits savent que le bus est occupé et ils ne devraient pas tenter d'en prendre contrôle. Le circuit qui vient de prendre le contrôle du bus en devient le **maître** (en anglais « master »). C'est lui qui génère le signal d'horloge, quel que soit le sens du transfert.

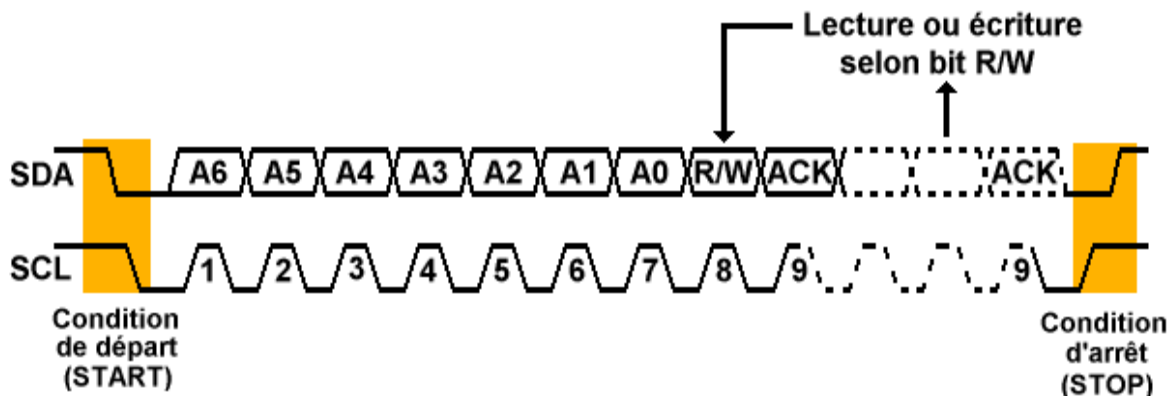
La transmission d'un octet

Avant de placer les bits qui forment l'octet à transmettre sur le bus, le maître doit placer la ligne d'horloge SCL à 0. Tant que la ligne SCL est au niveau haut, la ligne SDA ne doit pas changer d'état, sinon cette condition sera interprétée comme la condition d'arrêt. La condition arrêt peut survenir même au milieu de la transmission d'un octet, pour abandonner la transmission et libérer le bus pour les autres circuits. Pour transmettre correctement les bits sur la ligne SDA, le maître doit donc tout d'abord placer la ligne SCL à 0. Ensuite, le maître peut placer la ligne SDA au niveau correspondant au bit à transmettre et replacer la ligne SCL au niveau 1 pour indiquer que le bit est présent sur la ligne SDA. La même opération va se répéter autant de fois que nécessaire pour transmettre les 8 bits de données. Notez que c'est le bit de poids fort qui est transmis en premier.

Une fois les 8 bits transmis, le circuit qui vient de recevoir les données doit imposer un bit d'acquiescement ACK sur la ligne SDA. Pour cela, pendant que la ligne SCL est au niveau bas, le maître place sa propre sortie au niveau haut, tandis que le récepteur (auss appelé l'esclave) place sa sortie au niveau bas. Puisque les sorties sont à collecteur ouvert, la ligne SDA restera au niveau bas à cause de l'esclave. Le maître relit ensuite la ligne SDA une fois qu'il a passé la ligne SCL au niveau haut. Si la valeur lue pour le bit ACK est 0, c'est que l'esclave s'est bien acquiescé de l'octet reçu, sinon c'est qu'il y a une erreur et le maître doit générer la condition arrêt.

La transmission d'une adresse

Le nombre de composants qu'il est possible de connecter sur un bus I²C étant largement supérieur à deux, le maître doit pouvoir choisir quel esclave est censé recevoir les données. Dans ce but, le premier octet que transmet le maître n'est pas une donnée mais une adresse. Le format de l'octet d'adresse est un peu particulier puisque le bit D0 est réservé pour indiquer si le maître demande une lecture à l'esclave ou bien au contraire si le maître impose une écriture à l'esclave



Chaque circuit connecté au bus I²C possède une adresse, qui doit être unique. L'adresse associée à un composant est définie en partie par l'état de broches de sélections et d'autre part par sa fonction. Par exemple, le circuit PCF8574, qui est un port d'entrées/sorties bidirectionnel 8 bits, décompose son adresse de la façon suivante :

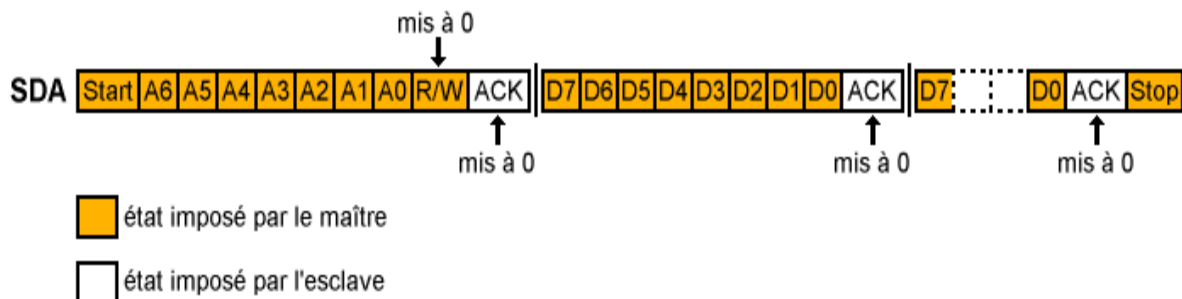
[0] [1] [0] [0] [A2] [A1] [A0] [R/W]. Les bits A2, A1 et A0 reflètent l'état des broches 1, 2 et 3 du circuit. Cela permet de placer 8 circuits PCF8574 sur le bus I²C. Lors de la conception d'un système, il faut donc veiller à l'unicité des adresses attribuées aux différents composants.

Une fois l'adresse envoyée sur le bus, l'esclave concerné doit répondre en plaçant le bit ACK à 0. Si le bit ACK vaut 1, le maître comprend qu'il y a une erreur de sélection et il génère la condition arrêt. En revanche, si le bit ACK vaut 0, le maître peut continuer les opérations.

Note : Les adresses 0000 0xxx et 1111 11xx sont réservées à des modes de fonctionnement particuliers.

Écriture d'une donnée

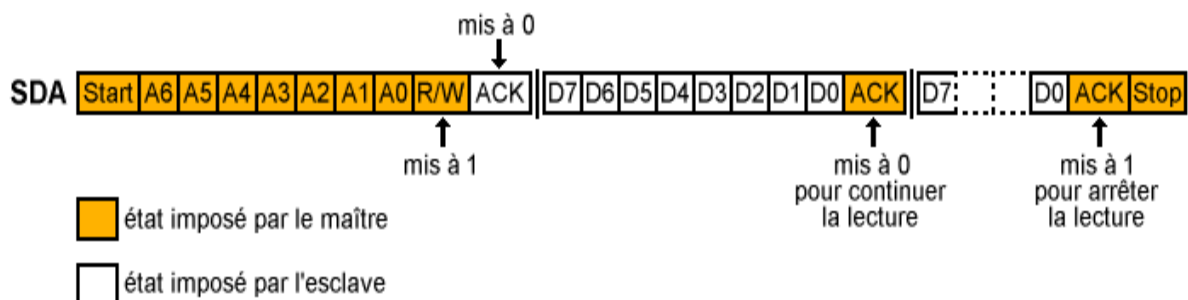
Si le bit R/W précédemment envoyé était à 0, cela signifie que le maître doit transmettre un ou plusieurs octets de données. Après chaque bit ACK valide, le maître peut continuer d'envoyer des octets à l'esclave ou bien il peut décider de terminer le dialogue par une condition d'arrêt.



Lecture d'une donnée

Si le bit R/W transmis en même temps que l'adresse est à 1, cela signifie que le maître veut lire des données issues de l'esclave. C'est toujours le maître qui va générer le signal d'horloge SCL. En revanche, après le bit ACK de l'adresse, c'est l'esclave qui va garder le contrôle de la ligne SDA. Pour cela, le maître va placer sa propre sortie SDA au niveau haut pour permettre à l'esclave de prendre le contrôle de la ligne SDA. L'esclave doit alors scruter la ligne SCL et attendre le niveau bas pour changer l'état de la ligne SDA, faute de quoi le maître détectera une condition arrêt et abandonnera le transfert (l'électronique intégrée dans l'esclave se doit de détecter aussi qu'il y a eu une condition arrêt, bien entendu).

Après que l'esclave a transmis les 8 bits de données, c'est le maître, cette fois-ci, qui va générer un bit d'acquittement. Si le maître désire lire des octets supplémentaires, il placera le bit d'acquittement à 0. En revanche, si le maître décide que la lecture est terminée, il placera le bit ACK au niveau 1. L'esclave comprendra alors que le transfert est terminé. Cette fois-ci, bien que le bit ACK soit au niveau 1, cela ne correspond pas à une condition d'erreur mais à une fin de transfert.



Restart

Le protocole du bus I²C ne s'arrête pas là. Il est possible d'enchaîner écriture et lecture de l'esclave sans avoir à passer par une condition arrêt. Par exemple, dans le cas de la lecture d'une RAM, le maître commence par envoyer l'adresse du composant avec le bit R/W positionné sur l'écriture. La RAM adressée, en esclave renvoie ACK = 0. Ensuite, le maître transmet l'adresse interne de la case mémoire demandée. Une fois encore, l'esclave répond par

ACK = 0. Le maître envoie alors à nouveau une condition de départ (sans passer par une condition d'arrêt), puis de nouveau l'adresse du composant sélectionné mais en plaçant le bit R/W sur la position lecture. L'esclave va répondre par ACK = 0 et enchaîner par la transmission du contenu de la case mémoire demandée. C'est toujours le maître qui impose l'horloge SCL mais c'est l'esclave, en l'occurrence la RAM, qui contrôle la ligne SDA. Une fois les 8 bits de données transmis par la RAM, si le maître veut lire le contenu de la case mémoire suivante, il placera le bit ACK au niveau 0. Dans ce cas, la RAM recommence la lecture avec la case mémoire suivante. En revanche, si le maître souhaite en terminer avec la lecture, il placera le bit ACK au niveau 1 et il générera ensuite la condition arrêt.

Le contenu des octets de données lus ou écrits aura une signification qui dépend du composant sélectionné. Mais le protocole reste le même.

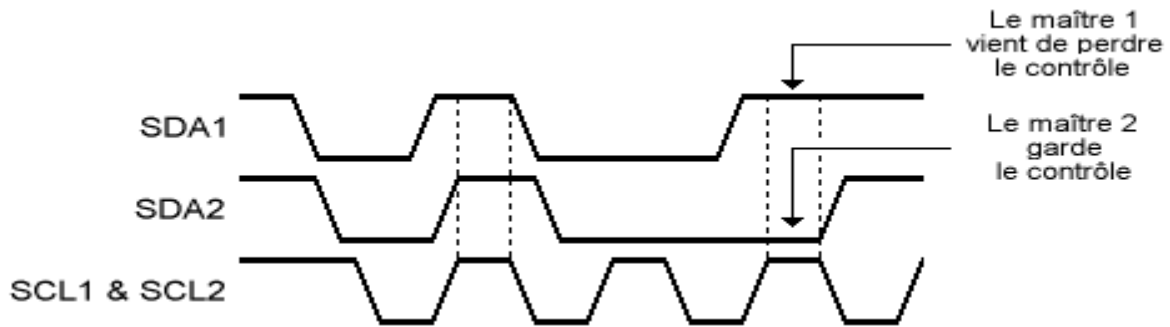
La gestion des conflits

La structure même du bus I²C a été conçue pour pouvoir y accueillir plusieurs maîtres. Se pose alors le problème commun à tout les réseaux utilisant un canal de communication unique : la gestion des conflits. En effet, chaque maître pouvant prendre possession du bus dès que celui-ci est libre, c'est à dire tant que les lignes SDA et SCL sont au niveau haut depuis suffisamment de temps, (4,7 µs), il existe la possibilité que deux maîtres prennent le contrôle du bus en même temps. Si cela ne pose pas de problème sur le plan électrique grâce à l'utilisation de collecteurs ouverts, il faut pouvoir détecter cet état de fait pour éviter la corruption des données transmises. Comment cela se passe-t-il ?

Une fois qu'il est établi que le bus est libre, le circuit qui souhaite prendre le contrôle du bus place la ligne SDA à 0 puis il relit l'état réel de la ligne SDA pour le comparer avec l'état qu'il souhaite imposer. Si dans un laps de temps très proche un autre maître décide aussi de prendre le contrôle du bus, il placera lui aussi la ligne SDA à 0. Du fait de la configuration à collecteur ouvert, le niveau de la ligne SDA sera à un niveau résultant d'une fonction "ET" entre les niveaux demandés par chaque maître. Pour l'instant, le résultat sera bien un niveau 0, de sorte qu'aucun maître ne peut savoir qu'il y a un conflit.

Chaque circuit maître va donc commencer normalement à générer le signal d'horloge, de sorte que l'état de la ligne SCL sera aussi le résultat d'une fonction « ET » des différents états imposés par chaque maître.

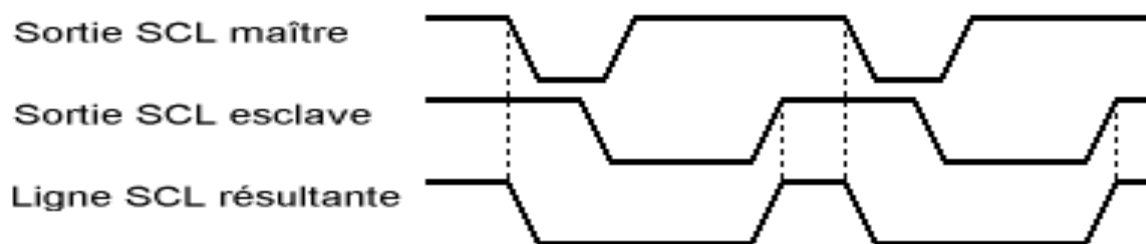
Pendant l'état bas de la ligne SCL, chaque maître va modifier sa propre sortie SDA. Si tous les maîtres qui ont pris le contrôle du bus placent le même état sur la ligne SDA, le conflit n'étant pas visible, la transmission se poursuit normalement, comme si chacun était seul. En revanche, dès qu'un maître place un niveau différent des autres sur la ligne SDA, il y aura forcément un état bas sur la ligne SDA, tandis qu'un ou plusieurs maîtres souhaitaient imposer un niveau haut. Tous les maîtres qui ont demandé un niveau haut, tandis que la ligne SDA reste à 0, vont perdre immédiatement le contrôle du bus. A partir de cet instant, le ou les maîtres qui viennent de perdre le contrôle du bus vont continuer à lire les états successifs de la ligne SDA. Ils vont continuer la lecture comme ils l'auraient fait en tant qu'esclaves, pour le cas où le maître qui reste encore en course souhaite justement adresser l'un d'eux.



Si les différents maîtres en compétitions tentent d'adresser le même composant, la ligne SDA sera toujours au bon niveau. La procédure d'arbitration va alors continuer avec les données à transmettre. Si les différents maîtres qui adressaient le même composant écrivent la même donnée, le conflit n'a pas lieu d'exister puisque tout le monde veut faire la même chose. La procédure d'arbitration va alors se poursuivre jusqu'à ce que l'un des maîtres demande un niveau différent des autres. En poussant le raisonnement à l'extrême, on peut imaginer le cas où tous les maîtres vont demander les mêmes niveaux jusqu'à la condition d'arrêt. Mais la probabilité d'un tel cas est bien faible. Quoi qu'il en soit, si le cas se produit, la procédure d'arbitration garantit que le résultat final est correct, chaque maître ayant réalisé la même opération exactement en même temps.

Ralentissement de la vitesse

Il est possible de ralentir la vitesse de transmission du bus I²C. Si un circuit esclave a besoin de ralentir les échanges sur le bus, il lui suffit de maintenir la ligne SCL à état bas. Le circuit maître scrute en permanence la ligne SCL pour la comparer avec l'état qu'il souhaite lui-même imposer. Quand le circuit maître détecte un niveau bas tandis qu'il vient de placer sa sortie SCL à l'état haut, il passe dans une boucle d'attente. L'attente se poursuivra jusqu'à ce que la ligne SCL aisse au niveau haut.



Les nouvelles spécifications

Face à l'explosion du nombre de circuits I²C disponibles en très forte augmentation, Philips a publié en 1993 les nouvelles spécifications de l'I²C :

- ✓ compatibilité totale avec l'ancien I²C (qui date de 1982)
- ✓ vitesse de 400 kbit/s
- ✓ adressage étendu sur 10 bits (jusqu'à 1024 circuits) répartis dans deux octets de la façon suivante :



- ✓ utilisation d'entrées à trigger de Schmitt afin de limiter la sensibilité au bruit
- ✓ diminution du temps entre une condition de stop et une condition de départ à 1,3 μ s
- ✓ mise en haute impédance d'un circuit non alimenté afin d'éviter de bloquer le bus si un circuit n'est pas alimenté

Très récemment, Philips a introduit une nouvelle extension de la norme I²C qui étend la vitesse à 3,4 Mbits/s, mais je n'ai pas beaucoup d'informations sur le sujet.

Les adresses réservées

Les adresses 0000 0xxx ne sont pas utilisées pour l'adressage de composants. Elles ont été réservées par Philips pour effectuer certaines fonctions spéciales.

Adresse d'appel général : 0000 0000

Après l'émission d'un appel général, les circuits ayant la capacité de traiter ce genre de demande d'appel émettent un acquiescement.

Le deuxième octet permet de définir le contenu de l'appel :

0000 0110

RESET. Remet tous les registres de circuits connectés dans leur état initial (mise sous tension). Les circuits qui le peuvent rechargent leur adresse esclave.

0000 0100

Les circuits définissant leur adresse de façon matérielle réinitialisent leur adresse esclave. Cela ne réinitialise pas les circuits.

0000 0000

Interdit

xxxx xxx1

Cette commande joue le rôle d'interruption. xxxx xxx peut être l'adresse du circuit qui a généré l'interruption.

Les autres valeurs du second octets ne sont pas définies et sont tout simplement ignorées.

Octet de start : 0000 0001

Cet octet est utilisé pour synchroniser les périphériques lents avec les périphériques rapides.

Début d'adressage CBUS : 0000 001x

L'émission de cet octet permet de rendre sourd tous les circuits I²C présents sur le bus. À partir de ce moment, on peut transmettre ce que l'on désire sur le bus, en utilisant par exemple un autre protocole. Le bus repasse en mode normal lors de la réception d'une condition d'arrêt.

Autre : 0000 0110 à 0000 1111

Ces adresses ne sont pas définies et sont ignorées par les circuits I²C. Elles peuvent être utilisées pour déboguer un réseau multimaster

Quelques composants I²C

Le PCF8583



Le PCF8583 est une horloge avec calendrier combinée à une RAM de 240 octets. Les années sont gérées sur 4 ans, ce qui permet de tenir compte des années bisextiles. Il possède aussi une fonction alarme avec une sortie d'interruption. L'horloge de référence peut être soit un quartz d'horloger cadencé à 32,768 kHz soit une entrée TTL cadencée à 50 Hz (pour la synchronisation sur le secteur).