

الجمهورية الجزائرية الديمقراطية الشعبية

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

وزارة التعليم العالي والبحث العلمي

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



## **MEMOIRE**

Présenté pour l'obtention du **diplôme** de **MASTER**

**En** : Electronique

**Spécialité** : Instrumentation

**Par** : Mahboub moustafa rafik & Ayad abdelhak

### **Sujet**

# **ÉTUDE, CONCEPTION ET RÉALISATION PRATIQUE D'UN MINI OSCILLOSCOPE BF NUMÉRIQUE A MÉMOIRE 4 VOIES**

Soutenu publiquement, le 11/06 /2024 , devant le jury composé de :

M. LACHACHI DJAMAL  
Mme GUEN AHLAM  
M NEMMICHE AHMED  
Mme GHEFIR FATIHA DJAZIA

MCA  
Professeur  
MCB  
MAA

Université de Tlemcen  
Université de Tlemcen  
Université de Tlemcen  
Université de Tlemcen

Président  
Examineur  
Encadreur  
Co-Encadreur

## Remerciements

Tout d'abord, nous remercions Dieu, « **ALLAH** », pour nous avoir donné la force physique et mentale nécessaires pour mener à bien ce modeste travail.

Ensuite, nous exprimons notre gratitude la plus sincère à notre encadrant, le **Dr. NEMMICHE AHMED**, et à notre co-encadrant, le **Dr. GHEFIR FATIHA DJAZIA**, pour leur soutien et leurs encouragements. Nous les remercions particulièrement pour la confiance qu'ils nous ont accordée, leur rigueur scientifique, leur patience et leurs conseils avisés, qui ont grandement contribué à la réalisation et à l'aboutissement de ce travail.

Nous tenons également à remercier l'ensemble des enseignants qui ont dirigé notre formation à l'Université Aboubakr Belkaid Tlemcen, ainsi que toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail.

Nous adressons un remerciement particulier à tous nos collègues de la promotion 2<sup>e</sup> année MASTER Instrumentation pour leur amitié et les moments très agréables que nous avons partagés.

Nous remercions également les membres du jury, **M. LACHACHI DJAMAL**, et **Mme GUEN AHLAM** pour avoir accepté d'évaluer et de juger notre travail.

Enfin, nos remerciements vont à tous ceux qui, de près ou de loin, ont apportés leur soutien et contribués à la réalisation de ce projet.

## **Dédicaces**

*Je dédie cet humble travail à mes parents, qui m'ont offert un soutien inconditionnel tout au long de mon parcours scolaire et universitaire. Papa, Maman, votre amour, vos sacrifices et vos encouragements constants ont été les piliers de ma réussite. Vous avez toujours été là pour moi, m'inspirant à persévérer et à donner le meilleur de moi-même. Vous serez toujours des modèles pour moi. J'espère que votre bénédiction continuera de m'accompagner à chaque étape de ma vie. Merci d'être tout simplement mes parents.*

*À mes grands-Pères,mes grands-mères, pour vos prières et votre amour. Puissiez-vous être bénies avec santé, bonheur et longue vie. Je prie pour ne jamais vous décevoir.*

*À mes frères et ma sœur, je vous dédie également ce succès. Votre soutien affectueux et vos encouragements ont été inestimables. Chacun de vous a contribué, par sa présence et son amour, à me donner la force et la motivation nécessaires pour avancer. Vous avez été une source constante de réconfort et d'inspiration. Merci pour chaque mot d'encouragement, chaque geste de soutien. C'est à vous que je dois cette réussite, et je suis immensément reconnaissant de vous avoir dans ma vie.*

*À mes amis chers, je dédie également cet accomplissement. Vous avez été à mes côtés, partageant les moments difficiles et les instants de joie. Merci pour votre soutien et votre encouragement continu, et pour les amitiés qui m'ont donné force et espoir. Votre présence dans ma vie a rendu cette aventure plus douce et plus agréable.*

*Enfin, je tiens à dédier ce travail à mon binôme, Mahboub Moustafa Rafik. Merci pour ta collaboration, ton dévouement et ton esprit d'équipe. Ta contribution a été essentielle pour mener à bien ce projet. Travailler avec toi a été une expérience enrichissante, et je suis reconnaissant pour ton soutien et ton amitié tout au long de ce parcours.*

**AYAD ABDELHAK**

*Je dédie ce modeste travail :*

*À ceux qui m'ont montré le bon chemin et qui attendent patiemment le fruit de leur éducation. À ceux qui ont toujours donné de leur personne, corps et âme, pour ma réussite.*

*À mes parents, pour leurs encouragements, leur affection, leurs conseils et leurs sacrifices. J'espère que vous trouverez dans ce travail l'expression de ma profonde reconnaissance et de mon grand respect pour vous.*

*À mon grand-père, pour vos prières et votre amour. Puissiez-vous être bénies avec santé, bonheur et longue vie. Je prie pour ne jamais vous décevoir.*

*À mes chers frères et sœur, pour leur précieuse assistance et leur soutien moral, toujours présents lorsque j'avais besoin d'encouragement.*

*Je dédie aussi ce travail à toute ma famille, mes oncles et tantes, cousins et cousines, pour leur amour et leur soutien indéfectible.*

*À tous mes amis, de près ou de loin, pour leur présence et leur amitié qui ont égayé ce parcours.*

*À tous ceux qui me connaissent et m'aiment, ainsi qu'à tous les enseignants et responsables du département de génie électrique, pour leur guidance et leur dévouement.*

*Enfin, je tiens à dédier ce travail à mon binôme, Ayad Abdelhak. Merci pour ta collaboration précieuse, ton engagement et ton esprit d'équipe. Ta contribution a été essentielle à la réussite de ce projet. Travailler avec toi a été non seulement productif, mais également très agréable. Ta persévérance et ton amitié ont rendu cette expérience plus enrichissante. Je te suis reconnaissant pour tout ce que nous avons accompli ensemble.*

***Mahboub Moustafa Rafik***

# Liste des Figures

FIGURE (I. 1):CLASSIFICATION MORPHOLOGIQUE DES SIGNAUX. [1] .....	5
FIGURE (I. 2):CLASSIFICATION SPECTRALE DES SIGNAUX. [2] .....	6
FIGURE (I. 3): SIGNAL TRANSITOIRE. [2].....	6
FIGURE (I. 4): SIGNAL PÉRIODIQUE. [3]      FIGURE (I. 5):SIGNAL ALÉATOIRE [3] .....	6
FIGURE (I. 6):REPRÉSENTATION TEMPORELLE. [4]      FIGURE (I. 7):REPRÉSENTATION SPECTRALE. [4] .....	7
FIGURE (I. 8):SYNTHÈSE D'UN SIGNAL. [5].....	7
FIGURE (I. 9):MODULATION D'UN SIGNAL. [5].....	8
FIGURE (I. 10):LE FILTRAGE DU BRUIT. [5].....	8
FIGURE (I. 11): LE CODAGE. [5].....	9
FIGURE I. 12: OSCILLOSCOPE ANALOGIQUE.      FIGURE I. 13: OSCILLOSCOPE NUMÉRIQUE. ....	9
FIGURE I. 14:FONCTIONNEMENT DE L'OSCILLOSCOPE NUMERIQUE. [6].....	10
FIGURE I. 15: LA BANDE PASSANTE. [6].....	11
FIGURE I. 16: LES ÉTAPES DE NUMÉRISATION D'UN SIGNAL. [7].....	12
FIGURE I. 17: L'ÉCHANTILLONNAGE. [7].....	13
FIGURE (I. 19): BROCHES DE L'ADC0831 [8] .....	15
FIGURE (I. 20): ARDUINO NANO .....	17
FIGURE (I. 21): BROCHES DE L'ARDUINO NANO [9] .....	18
FIGURE (I. 22): DISPOSITION DES BROCHES DE L'ARDUINO NANO[10].....	19
FIGURE II. 1: LOGO DE PYTHON. ....	23
FIGURE II. 2: CRÉATION D'UN CANVAS AVEC TKINTER .....	26
FIGURE II. 3: CODE TKINTER POUR LA CRÉATION D'UNE LIGNE SUR UN CANVAS .....	26
FIGURE III. 1: CIRCUIT DE MINI OSCILLOSCOPE .....	29
FIGURE III. 2: SHÉMA DE FONCTIONNEMENT D'UN MINI OSCILLOSCOPE NUMÉRIQUE. ....	30
FIGURE III. 3: ARDUINO NANO .....	31
FIGURE III. 4: ADC0831 .....	31
FIGURE III. 5: CHARGEMENT DES VALEURS ANALOGIQUES.....	32
FIGURE III. 6:DIAGRAMME SCHÉMATIQUE ILLUSTRANT LE FONCTIONNEMENT DE L'ADC0831 .....	33
FIGURE III. 7: LA VALEUR NUMÉRIQUE DE SORTIE DU CONVERTISSEUR EN FONCTION DE $V_{ref} + et V_{ref} -$ .....	36
FIGURE III. 8: INTERFACE DU MONITEUR SÉRIE ARDUINO AFFICHANT LES DONNÉES DE SIGNAL .....	37
FIGURE III. 9: DIAGRAMME DE FLUX POUR L'ENVOI DES VALEURS CONVERTIES PAR L'ARDUINO .....	38
FIGURE III. 10: SÉQUENCE DE CONVERSION ET DE RÉCEPTION DES DONNÉES ANALOGIQUES PAR L'ADC0831 .....	41
<b>FIGURE III. 11: FLUX DE CONFIGURATION ET D'AFFICHAGE DES SIGNAUX.....</b>	<b>43</b>
FIGURE III. 12: EXPLICATION DU MENU TKINTER.....	44
FIGURE III. 13: INTERFACE TKINTER.....	45
FIGURE III. 14: L'AFFICHAGE D'UN SIGNAL 1 PAR POTENTIOMÈTRE.....	46
FIGURE III. 15: SCHÉMA DE CIRCUIT SOUS ISIS PROTEUS AVEC UN SEUL PLAGE DE TENSION 0 À 5V.....	46
FIGURE III. 16: SCHÉMA DE CIRCUIT SOUS ISISPROTEUSÀ DES POTENTIOMÈTRE AVEC DES PLAGES DE TENSION ET DES RÉFÉRENCES SPÉCIFIQUES.....	48
FIGURE III. 17: VIRTUEL TERMINAL .....	48
FIGURE III. 18: SIMULATION DU MINI OSCILLOSCOPE BF NUMÉRIQUE AVEC ARDUINO NANO ET ADC0831 .....	49
FIGURE III. 19: SCHÉMA DE CIRCUIT DE SOURCE DE TENSION DE 5V.....	51
FIGURE III. 20: SCHÉMA DIVISEUR DE TENSION QUI DONNE DEUX TENSION DE RÉFÉRENCE 10V ET -5V.....	52
FIGURE III. 21: SCHÉMA DIVISEUR DE TENSION QUI DONNE DEUX TENSION DE RÉFÉRENCE 2V ET -1V.....	54
FIGURE III. 22: SCHÉMA DIVISEUR DE TENSION QUI DONNE DEUX TENSION DE REFERENCE 5V ET -2.5V.....	55
FIGURE III. 23: SCHÉMA DE CIRCUIT SOUS ISISPROTEUS .....	56
FIGURE III. 24: AFFICHAGE DES SIGNAUX SUR INTERFACE GRAPHIQUE TKINTER À 5456 HZ .....	57
FIGURE III. 25: AFFICHAGE DES SIGNAUX SUR INTERFACE GRAPHIQUE TKINTER À 7138 HZ .....	57

FIGURE III. 26: AFFICHAGE DES SIGNAUX SUR INTERFACE GRAPHIQUE TKINTER À 8393 HZ .....	57
FIGURE III. 27: AFFICHAGE DES SIGNAUX SUR INTERFACE GRAPHIQUE TKINTER À 103 KHZ.....	58
FIGURE III. 28: MONTAGE DE TEST ET MESURE DU SIGNAL POUR LE MINI OSCILLOSCOPE BF NUMÉRIQUE .....	58
FIGURE III. 29: AFFICHAGE DES SIGNAUX CARRÉ ET TRIANGLE SUR INTERFACE TKINTER À 6189 HZ ET 6087 HZ .....	58
FIGURE III. 30: VISUALISATION DE SIGNAUX SINUSOÏDAUX ET TRIANGULAIRES SUR UN OSCILLOSCOPE.....	59

# Liste des Tableaux

TABLEAU 1: SPÉCIFICATIONS TECHNIQUES DU CONVERTISEUR ADC0831.....	14
TABLEAU 2: BROCHAGE DE L'ADC .....	15
TABLEAU 4: BROCHAGE DE L'ARDUINO NANO.....	20
TABLEAU 5: CARACTÉRISTIQUES DES RÉSTANCES POUR DIFFÉRENTES TENSION DE RÉFÉRENCES <i>Vref</i> .....	56

## Nomenclature

CRT : tube à rayons cathodiques

GND: Ground

AC: Alternatif Current

DC: Direct Current

BNC : Bayonet Neill-Concelman

$x(t)$  : La fonction qui représente le signal à échantillonner.

$\delta(t)$  : la fonction du Dirac.

$X_e(t)$ : La fonction du signal échantillonné

$T_e$  : La période d'échantillonnage

$F_e$  : La fréquence d'échantillonnage

VPE : la tension analogique pleine échelle (Voltage Peak-to-Peak Equivalent)

LSB : Least Significant Bit

q: le quantum

DNL: Differential Non-Linearity.

INL: Integral Non-Linearity.

SAR: Successive Approximation Register.

CAN : convertisseur analogique-numérique

CNA : Convertisseur Numérique Analogique.

MSB: Most Significant Bit.

RAM: Random Access Memory.

SRAM: Static Random Access Memory.

ROM: Read-Only Memory.

EPROM: Erasable Programmable Read-Only Memory.

EEPROM: Electrically Erasable Programmable Read-Only Memory.

ADC: Analog-to-Digital Converter.

DAC: Digital-to-Analog Converter.

IRQ: Interrupt ReQuest.

IC: Integrated Circuit.

IDE: Integrated Development Environment.

FTDI: Future Technology Devices International.

TTL: Transistor-Transistor Logic.

PWM: Pulse Width Modulation.

SS: Slave Select.

MOSI: Master Output Slave Input.

MISO: Master Input Slave Output.

SCK: Serial Clock.

SPI: Serial Peripheral Interface.

TWI: Two-Wire Interface.

SDA: Serial Data Line.

SCL: Serial Clock Line.

UART: Universal Asynchronous Receiver/Transmitter.

ICSP: In-Circuit Serial Programming.

FTDI: Future Technology Devices International.

LED: Light Emitting Diode.

DTR: Data Terminal Ready.

DSO: Digital Storage Oscilloscope.

LCD : Liquid Crystal Display (affichage à cristaux liquides).

CRT: Cathode Ray Tube (tube cathodique).

FFT : Fast Fourier Transform (Transformée de Fourier rapide).

# Sommaire

<b>Remerciements</b> .....	<b>II</b>
<b>Dédicaces</b> .....	<b>III</b>
<b>Liste des Figures</b> .....	<b>V</b>
<b>Liste des Tableaux</b> .....	<b>VII</b>
<b>Nomenclature</b> .....	<b>VIII</b>
<b>Introduction Générale</b> .....	<b>1</b>
<b>CHAPITRE I : traitement du signal et introduction aux microcontrôleurs</b> .....	<b>3</b>
1. INTRODUCTION : .....	4
2. Introduction sur les signaux : .....	4
2.1 LE SIGNAL : .....	4
2.2 Classification : .....	4
2.2.1 Classification phénoménologique: .....	4
2.2.2 Classification morphologique: .....	5
2.2.3 Classification spectrale:.....	5
2.2.4 Classification énergétique : .....	6
3. Le traitement du signal:.....	7
3.1. Définition: .....	7
3.2. Etapes:.....	7
4. Définition de l'oscilloscope: .....	9
4.1. Les types d'oscilloscope: .....	9
4.1.1. L'oscilloscope analogique:.....	9
4.1.2. L'oscilloscope numérique .....	10
4.1.2.1. Principe de fonctionnement.....	10
4.1.2.2. Les caractéristiques:.....	10
5. Principes de base de la conversion analogique-numérique (CAN).....	12
5.1. Définition: .....	12
5.2. L'échantillonnage : .....	12
5.2.1. Le théorème de Shannon: .....	13
5.3. La quantification : .....	13
5.4. Le codage : .....	13
5.5. Les erreurs de conversion : .....	14
6. Présentation du ConvertisseurADC0831 .....	14
6.1. Spécifications techniques et fonctionnalités de l'ADC0831 :.....	14
6.2. Caractéristiques de l'ADC0831 :.....	15

6.3.	Applications de l'ADC0831 : .....	16
6.4.	Exemple d'utilisation de l'ADC0831 : .....	16
6.5.	Avantages et contraintes de l'utilisation de ADC0831 : .....	16
6.5.1.	Les avantages : .....	16
6.5.2.	Les contraintes: .....	16
7.	Introduction sur l'arduino: .....	17
7.1	Arduino nano : .....	17
7.1.1.	Alimentation : .....	18
7.1.2.	Mémoire : .....	18
7.1.3.	Les Entrées et sorties : .....	19
7.2.	Disposition des broches de l'Arduino Nano : .....	19
8.	CONCLUSION : .....	20
<b>CHAPITRE II: Conception et réalisation du mini oscilloscope BF numérique.....</b>		<b>21</b>
1.	Introduction : .....	22
2.	Architecture du mini oscilloscope BF numérique.....	22
2.1.	Vue d'ensemble de l'architecture du mini oscilloscope BF numérique à mémoire 4 voies : ..	22
2.2.	Fonctionnalité .....	22
3.	Programmation en Python : .....	23
3.1.	Définition : .....	23
3.2.	Avantages de Python:.....	23
4.	Utilisation de Tkinter pour le développement d'interfaces graphiques d'utilisateur (GUI) : .....	24
4.1.	Introduction à Tkinter: .....	24
4.2.	Les avantages de Tkinter: .....	24
5.	Utiliser Python et Tkinter pour afficher 4 signaux : .....	25
5.1.	Introduction : .....	25
5.2.	Bibliothèques utilisées : .....	25
5.3.	Explication du code : .....	25
6.	Conclusion : .....	26
<b>CHAPITRE III : Tests, performances et applications pratiques .....</b>		<b>27</b>
1.	INTRODUCTION : .....	28
2.	Principe de fonctionnement : .....	28
2.1.	Composants Principaux.....	28
2.2.	Schéma de Connexion.....	29
2.2.1.	Arduino Nano : .....	29
2.2.2.	ADC0831 (quatre ADC) : .....	29

3.	FonctionnementDétailé:	30
3.1.	Génération du Signal Analogique :	30
3.2.	Conversion Analogique-Numérique :	30
3.3.	Lecture des données par l'Arduino :	30
3.4.	Transmission des données à l'Ordinateur :	30
3.5.	Affichage des Signaux dans l'Interface Tkinter :	30
4.	Définition du Hardware et du Software :	30
4.1.	Hardware:	31
4.1.1.	Diagramme de Temps de Communication SPI pour un ADC0831 :	32
4.1.1.1.	Détails de la séquence :	32
4.1.2.	Le schéma de fonctionnement de l'ADC0831 :	33
4.1.2.1.	Les Composantsclés:	33
4.1.2.2.	Explication du diagramme :	34
4.1.2.3.	Fonctionnement:	34
4.1.3.	La Valeur Numérique de Sortie du Convertisseur :	35
4.1.4.	Points Importants pour le Hardware:	36
4.1.4.1.	Connectivité :	36
4.2.	Software:	37
4.2.1.	La Lecture des valeurs du potentiomètre via l'ADC0831 sur le Moniteur Série Arduino :	37
4.2.2.	L'organigrammed'arduino :	38
4.2.2.1.	Premièreétape :	38
4.2.2.2.	Deuxième étape :	40
4.2.3.	L'organigramme de Python.....	43
4.2.4.	Points Importants pour le Software:	44
4.2.4.1.	Lecture et Traitement des Données :	44
4.2.4.2.	Interface Utilisateur :	44
4.2.4.3.	Robustesse et Fiabilité :	44
4.2.5.	Interface Tkinter :	44
4.2.5.1.	Menu:	44
4.2.5.2.	Les Zones d'affichage :	45
5.	Le montage du circuit sous isis proteus .....	46
5.1.	Composants principaux :	47
5.2.	Connections détaillées :	47
5.3.	Affichage des Données Converties sur le Terminal Virtuel via Arduino :	48

5.3.1. Explication :	48
5.3.2. Explication des ValeursAffichéessur le Terminal Virtuel:	49
5.3.3. Calcul des résistances :	50
Pour calculer les résistances on aura cette équation :	50
6. Les Résultats :	57
7. Conclusion :	62
<b>CONCLUSION GENERALE</b>	<b>63</b>
<b>Conclusion Générale :</b>	<b>64</b>
<b>Bibliographie</b>	<b>65</b>
<b>Résumé</b>	<b>66</b>
<b>Abstract</b>	<b>66</b>

# **Introduction Générale**

## **Introduction générale**

L'évolution rapide des technologies en électronique et en informatique a ouvert de nouvelles perspectives pour la conception et la mise en œuvre de dispositifs de mesure et de contrôle de plus en plus avancés. Parmi ces dispositifs, l'oscilloscope numérique se distingue comme l'un des instruments les plus essentiels pour l'analyse et la visualisation des signaux électriques. Cet appareil permet de capturer et d'afficher des variations de signaux dans le domaine temporel, offrant ainsi une compréhension approfondie des phénomènes électriques complexes.

L'objectif de ce projet est de développer et de mettre en œuvre un oscilloscope numérique miniature à mémoire 4 voies. Ce dispositif sera conçu en utilisant un microcontrôleur Arduino, réputé pour sa flexibilité et son accessibilité, et des convertisseurs analogiques-numériques ADC0831, connus pour leur précision et leur fiabilité. Le choix de ces composants vise à équilibrer simplicité, coût et performance, rendant le projet accessible tant pour des applications pédagogiques que pratiques.

L'intégration de ces technologies permettra de créer un oscilloscope capable de répondre aux besoins des étudiants, des amateurs d'électronique, ainsi que des professionnels qui nécessitent un outil compact et efficace pour leurs travaux de mesure. En utilisant un microcontrôleur Arduino, le projet profite de l'écosystème vaste et bien documenté de la plateforme, facilitant ainsi le développement, le débogage et l'expansion future du dispositif. Les convertisseurs ADC0831, quant à eux, assureront une conversion fiable des signaux analogiques en données numériques, essentielle pour une analyse précise.

En somme, ce projet vise non seulement à fournir un outil pratique et éducatif, mais aussi à démontrer les capacités et les avantages de l'intégration des microcontrôleurs et des convertisseurs analogiques-numériques dans la conception d'instruments de mesure modernes.

# **CHAPITRE I : traitement du signal et introduction aux microcontrôleurs**

# I- Chapitre1 : traitement du signal et introduction aux microcontrôleurs

## 1. INTRODUCTION :

Dans ce chapitre, nous explorons les principes fondamentaux du traitement du signal et des microcontrôleurs. Après avoir défini et classé les signaux, nous abordons les méthodes de traitement du signal et l'utilisation des oscilloscopes. Ensuite, nous examinons les principes de la conversion analogique-numérique, avec le convertisseur ADC0831. Finalement, nous abordons la plateforme Arduino en mettant l'accent sur les caractéristiques et les applications de l'Arduino Nano dans le domaine du prototypage électronique.

## 2. Introduction sur les signaux :

### 2.1 LE SIGNAL :

Est une variation d'une quantité physique qui peut transporter de l'information. Il peut être électrique, comme une tension ou un courant, et il est utilisé pour transmettre, recevoir, et traiter des informations dans divers systèmes électroniques. Les signaux peuvent être analogiques ou numériques

### 2.2 Classification :

De la même manière que leurs différentes formes, les signaux possèdent des critères de classification variés. On peut les classer en fonction du temps, de la fréquence ou même de l'énergie.

#### 2.2.1 Classification phénoménologique:

##### a) Éléments déterminants:

- **Signaux périodiques** : Signaux dont la valeur peut être prédite avec certitude à tout moment, grâce à un modèle mathématique. On considère qu'un signal  $x(t)$  est périodique si  $T \in \mathbb{R}$ ,  $T > 0$ , ainsi que :  $x(t) = x(T + kT)$ ,  $\forall k \in \mathbb{Z}$ . Ils se répètent périodiquement et peuvent être rectangulaires, sinusoïdaux ou pseudo-aléatoires.
- **Signaux non-périodiques** : Signaux qui ne se répètent pas de manière régulière. Ils incluent des signaux quasi-périodiques et transitoires.

##### b) Événement imprévisibles:

- **Signaux stationnaires** : Signaux dont les propriétés statistiques restent constantes dans le temps.
- **Signaux non-stationnaires** : Signaux dont les propriétés changent au fil du temps, pouvant être cycliques ou d'autres formes.

### 2.2.2 Classification morphologique:

Dans cette classification, l'amplitude du signal est le paramètre pris en compte [Figure I.1].

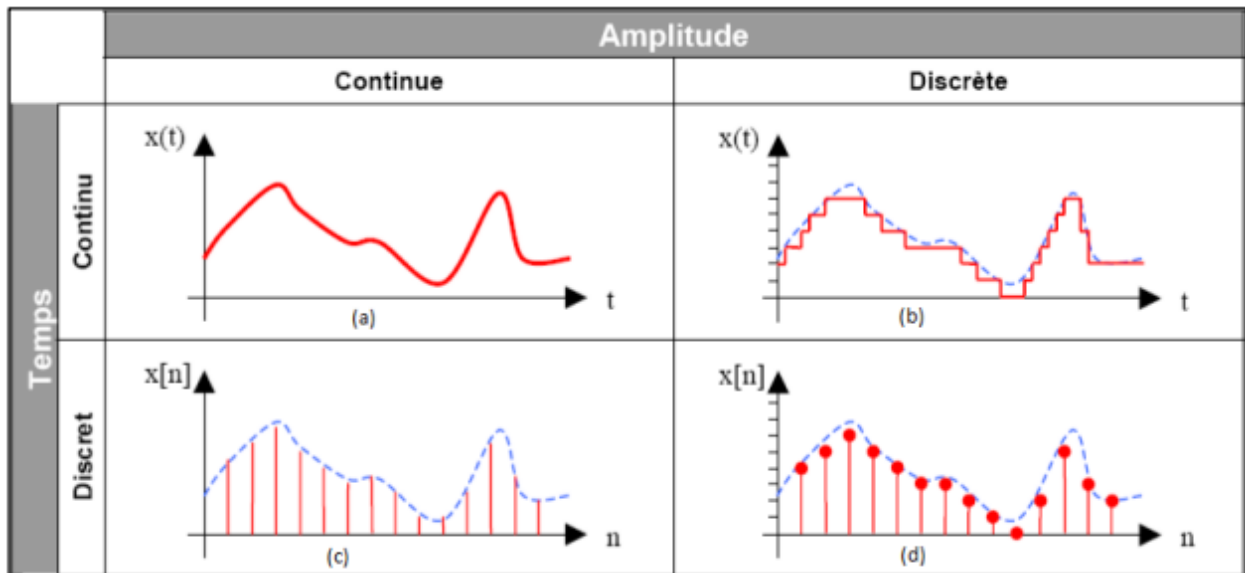


Figure (I. 1): Classification morphologique des signaux. [1]

- a) Signaux analogiques : Ce sont des signaux continus en temps et en amplitude.
- b) Signaux échantillonnés : Ce sont des signaux discrets en temps et continus en amplitude.
- c) Signaux quantifiés : Ce sont des signaux continus en temps et discrets en amplitude.
- d) Signaux numériques : Ce sont des signaux discrets en temps et en amplitude.

### 2.2.3 Classification spectrale:

Le spectre d'un signal représente l'amplitude, la phase, l'énergie ou la puissance du signal en fonction de sa fréquence (Hz). La largeur de bande, ou largeur spectrale, désigne le domaine des fréquences occupées par le spectre, définie comme  $\Delta F = F_{\max} - F_{\min}$ .

Selon ces critères, on distingue quatre catégories de signaux :

Signaux à bande étroite : Petit  $\Delta F / F_{\text{moy}}$  (soit  $F_{\max} \approx F_{\min}$ ).

Signaux à large bande : Grand  $\Delta F / F_{\text{moy}}$  (avec  $F_{\max}$  supérieur à  $F_{\min}$ ).

Signaux à basse fréquence (BF) : Largeur de bande limitée à des fréquences basses.

Signaux à haute fréquence (HF) : Largeur de bande limitée à des fréquences élevées.

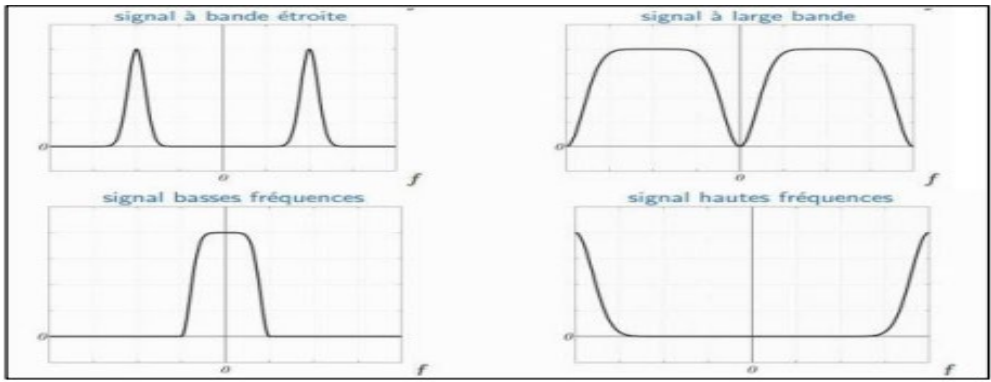


Figure (I. 2):Classification spectrale des signaux. [2]

### 2.2.4 Classification énergétique :

Lorsqu'on examine l'énergie et la puissance d'un signal, on distingue deux types :

- a) Signaux à puissance nulle : il s'agit de signaux à énergie finie. Les signaux transitoires et les signaux éphémères sont mentionnés [Figure I.3].

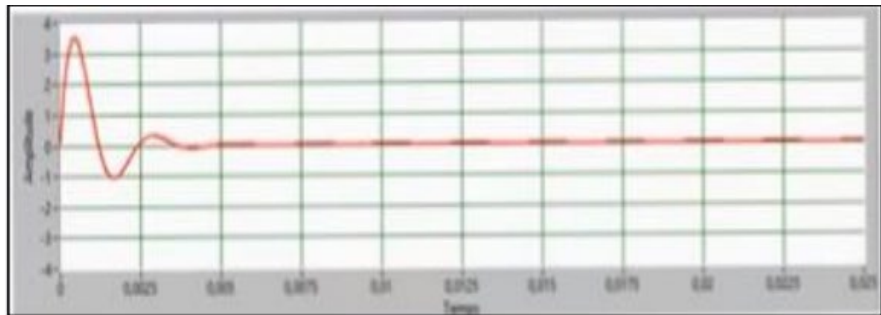


Figure (I. 3): Signal transitoire. [2]

- b) Les signaux à puissance infinie sont des signaux à énergie infinie. Les signaux périodiques [Figure I.4] et les signaux aléatoires [Figure I.5] peuvent être mentionnés.

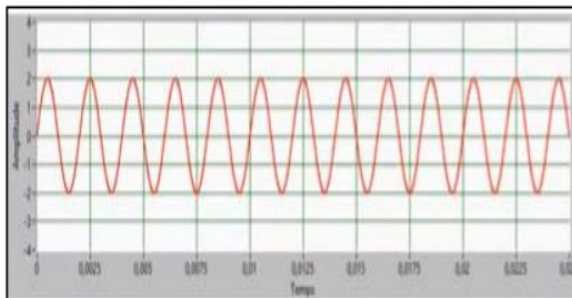


Figure (I. 4): Signal périodique. [3]

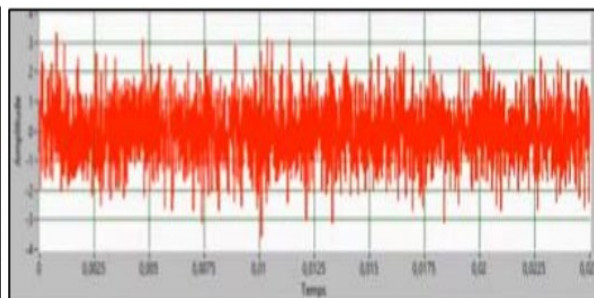


figure (I. 5):Signal aléatoire [3]

### 3. Le traitement du signal:

#### 3.1.Définition:

- a) Le traitement du signal est une discipline technique qui, en utilisant la théorie du signal, les techniques de l'électronique, de l'informatique et de la physique, cherche à générer, analyser et filtrer le bruit des signaux, quels qu'ils soient, afin d'en extraire le plus d'informations possible.
- b) L'étude de la théorie du signal consiste à représenter mathématiquement un signal dans les domaines temporels [Figure I.6] et spectraux [Figure I.7].

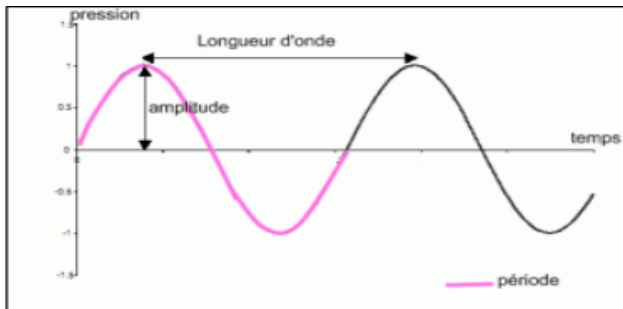


Figure (I. 6):Représentation temporelle. [4]

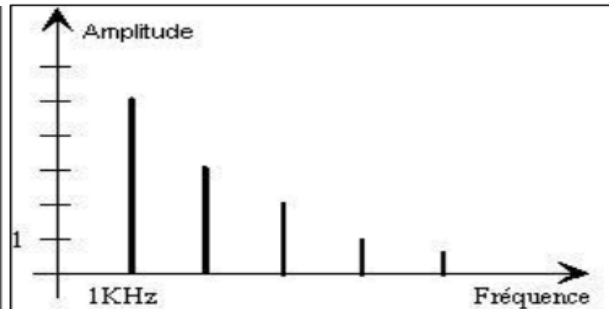


Figure (I. 7):Représentation spectrale. [4]

- c) Le bruit : est une perturbation aléatoire ou indésirable qui se superpose à un signal utile, altérant ainsi la clarté ou la qualité de l'information transmise.

#### 3.2.Etapes:

Le traitement du signal passe par trois grandes étapes et chacune a son propre déroulement.

a) **La création** : cette étape implique la conception du signal à transmettre, ce qui implique :

- La synthèse consiste à combiner des signaux élémentaires tels que des signaux sinusoïdaux, rectangulaires ou en dent de scie pour produire le signal. [Figure I.8]

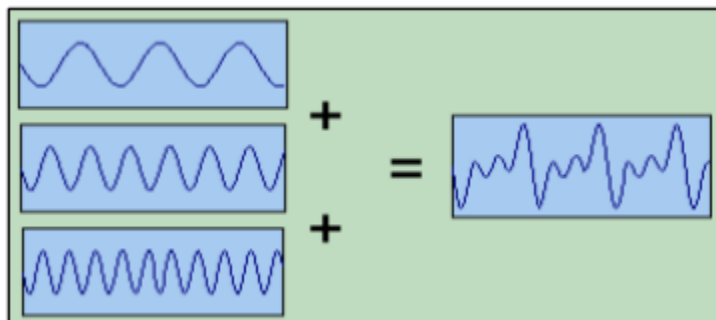


Figure (I. 8):Synthèse d'un signal. [5]

- La modulation est un processus de modification de fréquence qui permet d'ajuster le signal en fonction des caractéristiques fréquentielles du support de transmission ou d'un filtre d'analyse. [Figure I.9]

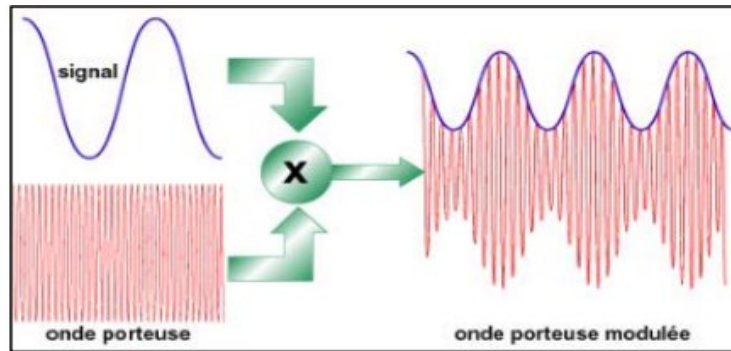


Figure (I. 9):Modulation d'un signal. [5]

d) **L'analyse:** À ce stade, l'analyse se concentre sur la compréhension de la nature et des origines du signal en utilisant :

- La détection : elle nous donne la possibilité d'extraire le signal souhaité d'un son de fond qui le perturbe.
- L'identification est essentielle pour le classer.

e) **La transformation** : elle nous permet de personnaliser un signal en fonction de nos besoins, en utilisant :

- Le filtrage est utilisé afin de supprimer certains effets indésirables, comme le craquement ou l'écho (pour le son), le bruit (pour les images), etc. [Figure I.10]

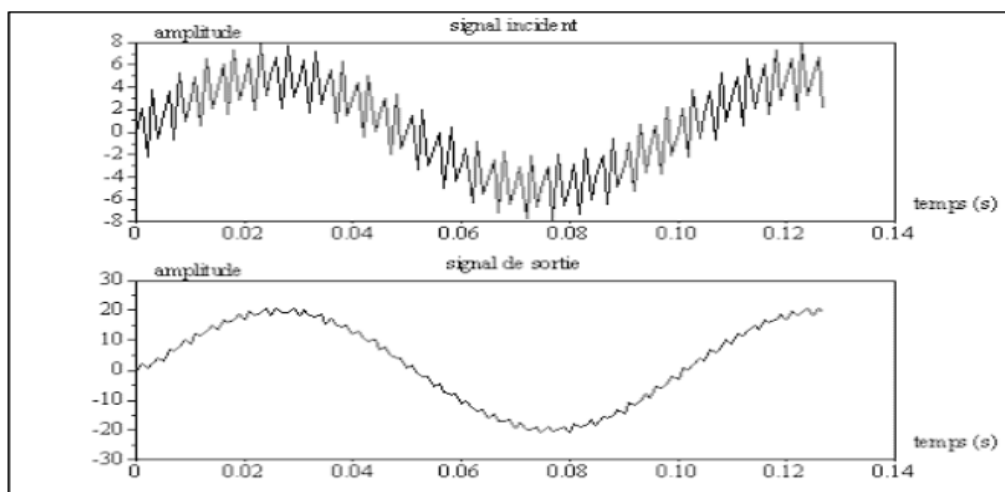


Figure (I. 10):Le filtrage du bruit. [5]

- **Codage** : est utilisé pour convertir un signal numérique afin de pouvoir l'analyser sur des appareils numériques tels qu'un ordinateur ou éviter le bruit de fond. [Figure I.11]

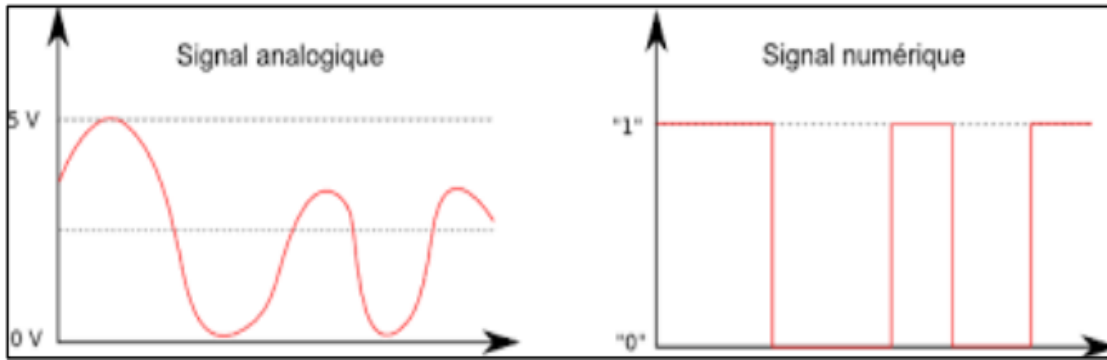


Figure (I. 11): Le codage. [5]

#### 4. Définition de l'oscilloscope:

L'oscilloscope analogique [Figure I.12] ou numérique [Figure I.13] est un instrument de mesure qui permet de représenter visuellement la variation dans le temps de grandeurs physiques préalablement converties en tension par un convertisseur ou des capteurs.



Figure I. 12: Oscilloscope Analogique.



Figure I. 13: Oscilloscope Numérique.

#### 4.1. Les types d'oscilloscope:

Il y'a deux type d'oscilloscope :

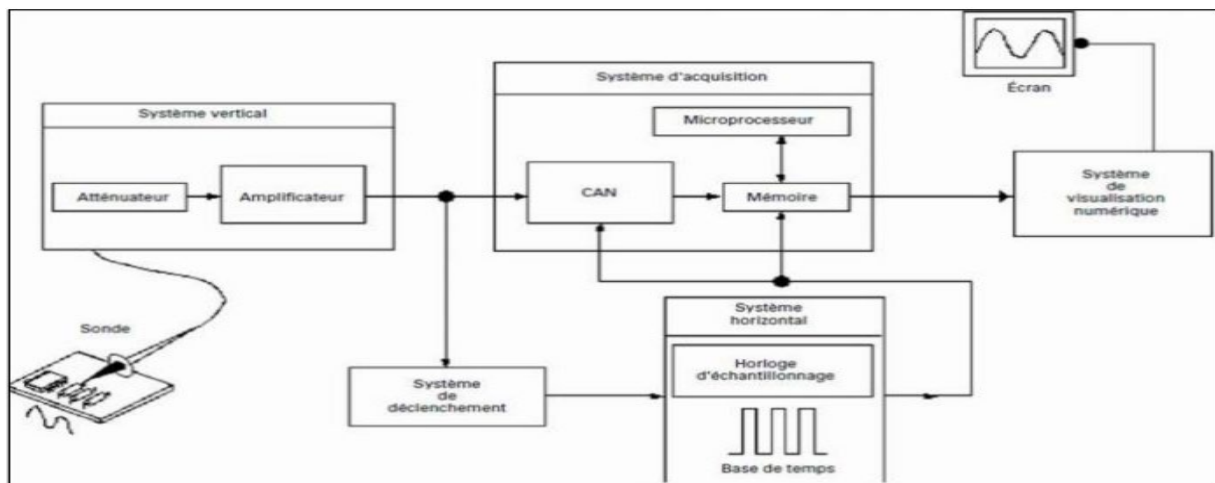
##### 4.1.1. L'oscilloscope analogique:

L'oscilloscope analogique est le plus ancien et est en voie de disparition, puisqu'il ne permet que l'observation de tensions périodiques. Il fonctionne par déflexion électrique d'un faisceau d'électrons en fonction de la tension mesurée. En effet, la tension dévie le faisceau verticalement (de haut en bas) et trace ainsi une courbe sur l'écran, qui représente le signal en temps réel.

## 4.1.2. L'oscilloscope numérique

### 4.1.2.1. Principe de fonctionnement

Les oscilloscopes numériques, similaires aux analogiques, disposent de systèmes de traitement de données supplémentaires. Lorsqu'ils sont connectés à un circuit, ils ajustent l'amplitude du signal et un convertisseur analogique-numérique (CAN) enregistre le signal à des intervalles précis, convertissant les tensions en valeurs numériques appelées échantillons. Le système horizontal utilise une horloge pour définir ces moments d'échantillonnage. Les données sont stockées en mémoire pour former une courbe complète, avec la longueur d'enregistrement correspondant au nombre de points requis. Le système de déclenchement détermine le début et la fin de l'enregistrement. L'écran affiche les points, et des traitements supplémentaires peuvent améliorer la résolution et permettre de voir les événements précédant le déclenchement.



**Figure I. 14:FONCTIONNEMENT DE L'OSCILLOSCOPE NUMERIQUE. [6]**

L'oscilloscope numérique est un appareil moderne utilisant des méthodes de traitement numérique du signal, offrant plusieurs avantages qui ont entraîné la disparition progressive des oscilloscopes analogiques. Ces avantages incluent l'étude en temps réel des signaux, la conservation des informations numériques pour une visualisation ultérieure, des outils d'analyse et de traitement pour évaluer les propriétés des signaux, des filtres performants pour améliorer la clarté des détails, et un écran à cristaux liquides pour économiser de l'énergie.

### 4.1.2.2. Les caractéristiques:

#### a) La bande passante :

La fréquence maximale que l'oscilloscope peut capturer et analyser est indiquée par la bande passante. Lorsque la fréquence se rapproche de la fréquence maximale, la précision de

l'oscilloscope diminue. La fréquence à laquelle le signal affiché est réduit à 70,7% de l'amplitude à l'entrée est également indiquée (cette atténuation de 70,7% est connue sous le nom d'atténuation à -3dB). [Figure I.15]

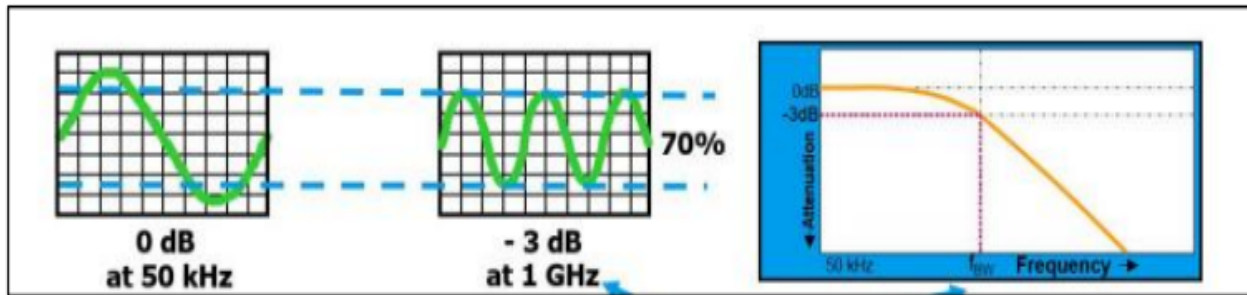


Figure I. 15: la bande Passante. [6]

#### b) Le temps de montée :

La durée nécessaire pour que le signal passe de 10% à 90% de sa valeur est appelée temps de montée de la forme d'onde, ce qui est une autre manière de décrire la fréquence utile d'un oscilloscope.

#### c) L'échelle horizontale de l'oscillogramme :

La base de temps (ou balayage ou sensibilité horizontale) permet, grâce aux deux bases principales (P) et retardées (R), de sélectionner un détail du signal à une vitesse P puis de zoomer sur l'ensemble de l'écran à une vitesse  $R > P$ .

#### d) La sensibilité verticale :

L'échelle verticale de l'oscillogramme correspond à la capacité de l'amplificateur vertical à amplifier un signal faible à l'entrée. La tension minimale détectée par un oscilloscope d'usage général est d'environ 2 mV par division. Il est nécessaire d'ajuster la sensibilité verticale de manière à ce que la courbe couvre l'ensemble de l'écran.

#### e) La résolution verticale:

Résolution d'un instrument en termes de mesure, est le plus petit incrément que l'instrument indique ou affiche. La résolution verticale d'un oscilloscope numérique est liée au nombre de bits du CAN.

#### f) Le couplage:

Permet à l'utilisateur de sélectionner si le signal entrant est filtré ou non. Trois types de couplage d'entrée (modes d'entrée) sont disponibles : AC, DC et GND.

## 5. Principes de base de la conversion analogique-numérique (CAN)

### 5.1. Définition:

La numérisation, ou conversion analogique-numérique, consiste à transformer un signal analogique en un signal numérique grâce à un convertisseur électronique appelé ADC (Analog to Digital Converter). Cette technique permet de traiter le signal avec des appareils numériques comme les ordinateurs et offre plusieurs avantages, notamment une meilleure résistance aux perturbations sonores, une reproduction facilitée et la possibilité d'adapter les traitements. Les trois étapes clés de ce processus sont l'échantillonnage, la quantification et le codage.

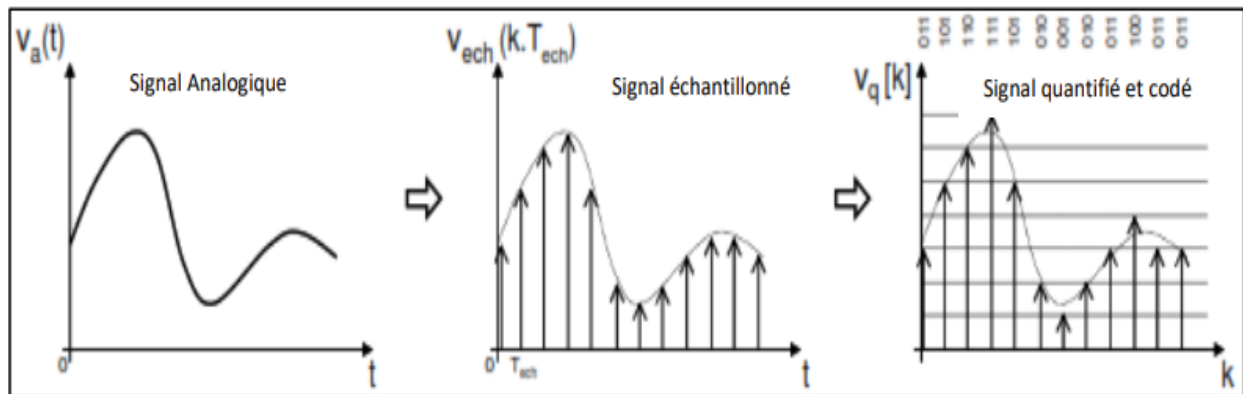


Figure I. 16: Les étapes de numérisation d'un signal. [7]

### 5.2. L'échantillonnage :

L'échantillonnage nous permet d'extraire des portions du signal à des intervalles de temps réguliers, appelés périodes d'échantillonnage  $T_e$ . Mathématiquement, cela implique la multiplication de la fonction représentant le signal par une série de pics espacés dans le domaine temporel, et une convolution dans le domaine spectral/fréquentiel. Soient :

$x(t)$  : la fonction qui représente le signal à échantillonner.

$\delta(t)$  : la fonction du Dirac.

$X_e(t)$  : La fonction du signal échantillonné

On aura alors :

$$X_e(k.T_e) = x(t) \cdot \sum_k \delta(t - k.T_e)$$

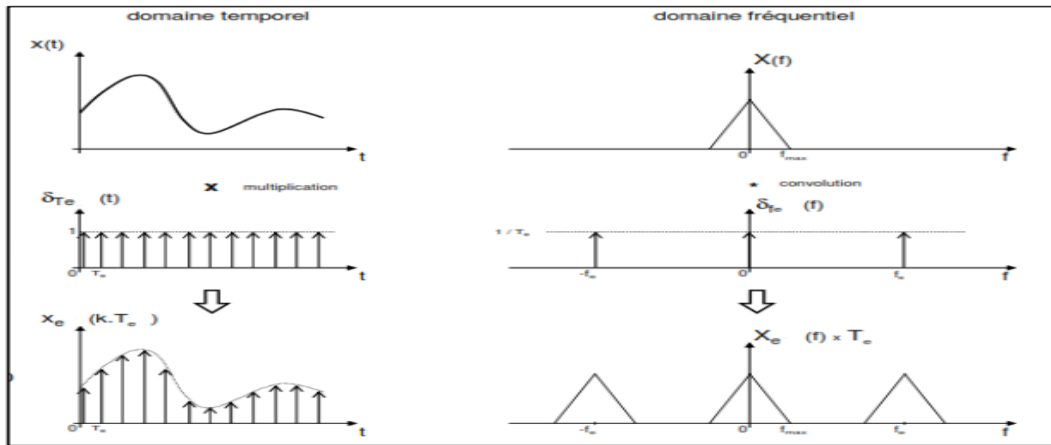


Figure I. 17: L'échantillonnage. [7]

### 5.2.1. Le théorème de Shannon:

Il est nécessaire de sélectionner une fréquence d'échantillonnage deux fois supérieure à la fréquence maximale, présente dans le signal d'entrée, afin de reconstruire fidèlement un signal de sortie :  $F_e$  est supérieur à  $2f_{max}$ . [7] Par exemple, les CD utilisent une fréquence d'échantillonnage de 44.1kHz, car la plage de fréquence audio à laquelle l'ouïe humaine est sensible se situe entre 20 Hz et 20kHz.

### 5.3. La quantification :

Pour traiter un signal avec précision sur des appareils numériques, la quantification est essentielle. Chaque valeur échantillonnée est représentée en binaire sur  $n$  bits, définissant un niveau de quantification  $N=2^n$ . En plus de ce niveau, des paramètres comme la tension analogique pleine échelle ( $V_{PE}$ ) et le pas de quantification ( $q$ ) sont déterminés. Le pas de quantification, divise la pleine échelle en fonction du niveau de quantification.

$$q = \frac{V_{PE}}{2^n}$$

. Si le pas est constant, c'est une quantification uniforme; s'il est variable, c'est non-uniforme. Cette méthode vise à améliorer la précision de certains échantillons par rapport à d'autres.

Si le quantum est constant, on parle de quantification uniforme, tandis que s'il est variable, on parle de quantification non-uniforme. Cette méthode est utilisée pour améliorer la précision de certains échantillons par rapport à d'autres.

### 5.4. Le codage :

Il s'agit de la phase finale de la numérisation. La conversion en code binaire implique la conversion des valeurs quantifiées du signal échantillonné. Le convertisseur a son propre nombre de bits pour le codage.

## 5.5. Les erreurs de conversion :

La conversion analogique-numérique sélectionne des valeurs du signal d'entrée à intervalles de temps constants, entraînant des pertes d'informations et des erreurs.

Parmi ces erreurs, l'erreur de gain, représentant l'écart entre valeurs théoriques et réelles lors de la dernière transition, et l'erreur d'offset, mesurant la différence initiale, sont traduites en LSB.

### ❖ Erreur de linéarité :

La variation de résolution des convertisseurs induit une erreur de linéarité, évaluée après correction des erreurs précédentes. Elle se divise en non-linéarité différentielle (DNL), mesurant la disparité entre l'écart mesuré et le LSB théorique, et non-linéarité totale (INL), désignant la différence entre valeurs réelles et théoriques.

### ❖ Temps de conversion et Précision :

Le temps de conversion correspond au délai entre l'ordre de conversion et la sortie des valeurs binaires. La précision du convertisseur, calculée en pourcentage de la pleine échelle ou en fraction du quantum, combine l'impact de toutes les erreurs précédentes.

## 6. Présentation du Convertisseur ADC0831

### 6.1. Spécifications techniques et fonctionnalités de l'ADC0831 :

Microchip Technology a développé l'ADC0831, un convertisseur analogique-numérique (CAN) intégré (IC) de 8 bits. C'est un CAN très consommateur, très rapide et très abordable qui est couramment employé dans différentes applications, telles que les microcontrôleurs, les systèmes d'acquisition de données et les systèmes de contrôle industriel.

Spécifications clés de l'ADC0831:

Spécification	Description
Résolution	8 bits (256 niveaux de quantification)
Temps de conversion	10 $\mu$ s (maximum)
Plage de tension d'entrée	0V à VREF (mode mono-extrémité) ou -VREF/2 à VREF/2 (mode différentiel)
Tension de référence	Tension de référence externe (VREF) requise
Alimentation	2,7 V à 5,5 V
Température de fonctionnement	-40°C à +85°C
Boîtier	DIP-8, SOIC-8

**Tableau 1: Spécifications techniques du Convertisseur ADC0831**

## 6.2. Caractéristiques de l'ADC0831 :

- La consommation d'énergie est faible. En moyenne, il ne consomme que 150  $\mu\text{A}$  en mode actif et 1  $\mu\text{A}$  en mode veille.
- Haute vitesse : La conversion rapide de 10  $\mu\text{s}$  permet d'obtenir rapidement des données.
- Des modes d'entrée souples : Supporte les entrées mono-extrémité et différentielles.
- La tension de référence externe permet d'obtenir des mesures précises de tension.
- Un large éventail de températures de fonctionnement : Il peut être employé dans différents contextes.

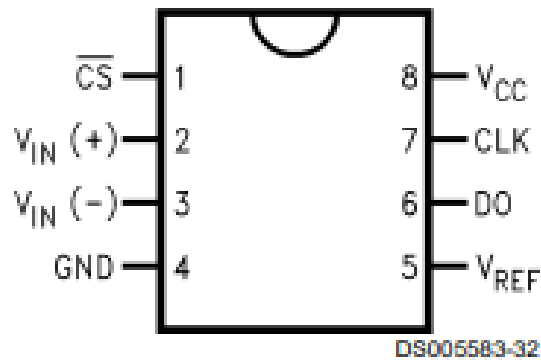


Figure (I. 18): broches de L'ADC0831 [8]

### Résumé des pins de l'ADC0831 :

Nom	Pin	Description
$\overline{CS}$	1	Sélection du circuit (Chip Select)
$V_{IN+}$	2	Entrée analogique positive
$V_{IN-}$	3	Entrée analogique négative
GND	4	La Masse ( <b>GrouND</b> )
$V_{ref}$	5	Tension de référence
$D_0$	6	Données de sortie
CLK	7	Horloge
$V_{CC}$	8	Alimentation positive : Généralement connectée à une tension de 5V.

Tableau 2: Brochage de l'ADC

### 6.3.Applications de l'ADC0831 :

- Les Convertisseurs sont utilisés pour convertir les données des capteurs analogiques en valeurs numériques afin de les traiter.
- Systèmes de collecte des données : Effectuer la collecte et la numérisation de signaux analogiques afin d'analyser ou de stocker des informations.
- Systèmes d'inspection industrielle : Observation et régulation des paramètres analogiques dans les processus industriels
- Appareils médicaux : Évaluation des indicateurs physiologiques tels que la fréquence cardiaque ou la tension artérielle.

### 6.4.Exemplé d'utilisation de l'ADC0831 :

Une application courante de l'ADC0831 est de mesurer la tension d'un potentiomètre. Le potentiomètre est connecté aux broches d'entrée de l'ADC, et l'ADC convertit la tension en une valeur numérique. Cette valeur numérique peut ensuite être lue par un microcontrôleur et utilisée pour contrôler un appareil ou afficher la tension mesurée.

### 6.5.Avantages et contraintes de l'utilisation de ADC0831 :

#### 6.5.1. Les avantages :

- ❖ **Faible Consommation d'Énergie** : Idéal pour les applications à faible consommation d'énergie ou alimentées par batterie.
- ❖ **Vitesse de Conversion Élevée** : Temps de conversion rapide de 10  $\mu$ s, adapté aux applications nécessitant une grande rapidité.
- ❖ **Mesures de Tension Précises** : Permet des mesures précises et indépendantes de la tension d'alimentation grâce à une référence externe.
- ❖ **Large Plage de Température** : Fonctionne dans des environnements allant de  $-40^{\circ}\text{C}$  à  $+85^{\circ}\text{C}$ , adapté à diverses conditions.
- ❖ **Facile à Utiliser** : Interface simple et nécessite un nombre minimal de composants externes.

#### 6.5.2. Les contraintes:

- ❖ **Faible Résolution** : Une résolution de 8 bits peut ne pas suffire pour des applications nécessitant une grande précision.

- ❖ **Sensibilité au Bruit** : La précision des mesures peut être affectée par le bruit lors de la conversion, surtout dans des environnements bruyants.
- ❖ **Non-linéarité** : La relation entre la tension d'entrée et la valeur numérique peut ne pas être parfaitement linéaire, introduisant une certaine non-linéarité.
- ❖ **Plage de Tension Limitée** : La plage de tension d'entrée est restreinte de 0V à VREF, nécessitant parfois une mise à l'échelle ou un décalage pour certains signaux.
- ❖ **Boîtier DIP-8** : Moins pratique pour les applications nécessitant un montage en surface ou une miniaturisation.

## 7. Introduction sur l'arduino:

L'interface Arduino, commandée par un logiciel open source semblable aux langages C et C++, a été introduite en 2005. Conçue pour être abordable et facile à utiliser, la plate-forme permet aux amateurs, étudiants et professionnels de créer des projets interactifs avec des capteurs et des actionneurs. Basée sur un microcontrôleur, elle sert à construire et programmer des appareils électroniques à l'aide de cartes de développement Arduino et du logiciel Arduino IDE.

### 7.1 Arduino nano :

C'est une carte miniature [Figure I.20], complète et compatible avec les planches à pain basé sur l'ATmega328 (Arduino Nano 3.0) ou ATmega168 (Arduino Nano 2.x). Elle a plus ou moins les mêmes fonctionnalités que l'Arduino Du emilanove, mais le boîtier est différent, elle n'y pas équipée d'une prise d'alimentation de courant continu et la communication ce fait par un câble USB Mini-Bau lieu standard. Le Nano a été conçu et produit par Gravitech [1].

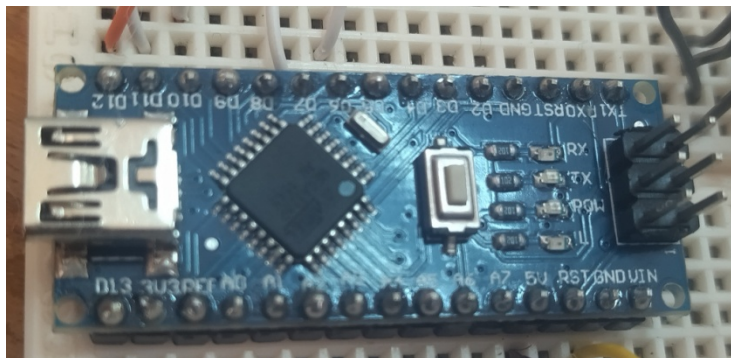
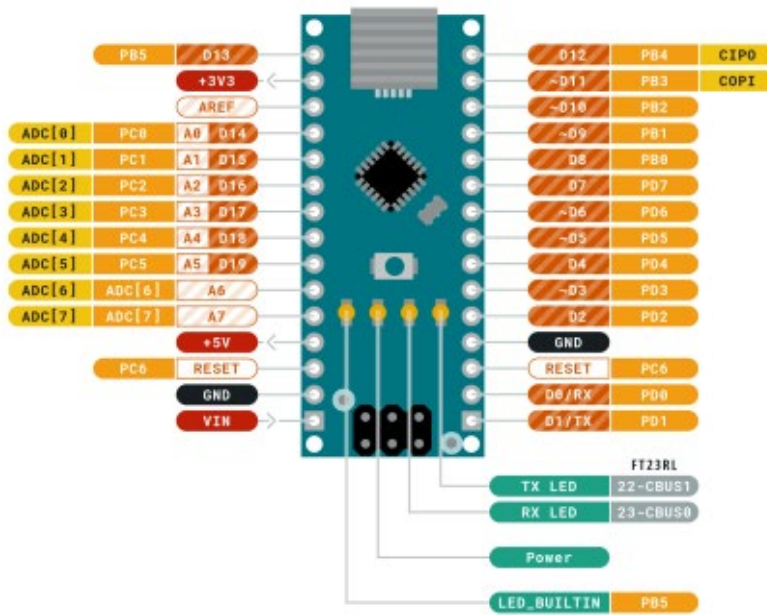


Figure (I. 19): arduino nano



Power Tree of Arduino Nano

Figure (I. 20): broches de l'Arduino Nano [9]

### 7.1.1. Alimentation :

- L'Arduino Nano peut être alimenté via USB Mini-B, une alimentation externe de 6 à 20V ou une alimentation externe régulée de 5V.
- La puce FTDI FT232RL est alimentée uniquement via USB, désactivée avec une alimentation externe non USB.

### 7.1.2. Mémoire :

- ATmega168 a 16 Ko de mémoire flash, dont 2 Ko pour le chargeur de démarrage ; ATmega328 a 32 Ko, également avec 2 Ko pour le chargeur de démarrage.
- ATmega168 a 1 Ko de SRAM et 512 octets d'EEPROM ; ATmega328 a 2 Ko de SRAM et 1 Ko d'EEPROM.

### 7.1.3. Les Entrées et sorties :

Chacune des 14 broches numériques de l'Arduino Nano peut être configurée en entrée ou sortie avec `pinMode()`, `digitalWrite()`, et `digitalRead()`, fournissant 5 volts avec une capacité maximale de 40 mA. Certaines broches ont des fonctions spécialisées comme la communication série, les interruptions externes, PWM, SPI, et une LED intégrée. Le Nano dispose également de 8 entrées analogiques offrant une résolution de 10 bits, et d'autres broches pour I2C, AREF, et RESET.

### 7.1.4. Communication :

- Communication série UART TTL (5V) sur RX et TX, gérée par la puce FTDI FT232RL via USB.
- Utilisation des FTDI drivers pour la communication série sur l'ordinateur.

### 7.1.5. Programmation :

- Programmation via le logiciel Arduino avec sélection du type de carte (ATmega168 ou ATmega328).
- Préchargé avec un bootloader pour télécharger un nouveau code sans programmeur externe.
- Possibilité de programmer via ICSP pour contourner le bootloader.

## 7.2. Disposition des broches de l'Arduino Nano :

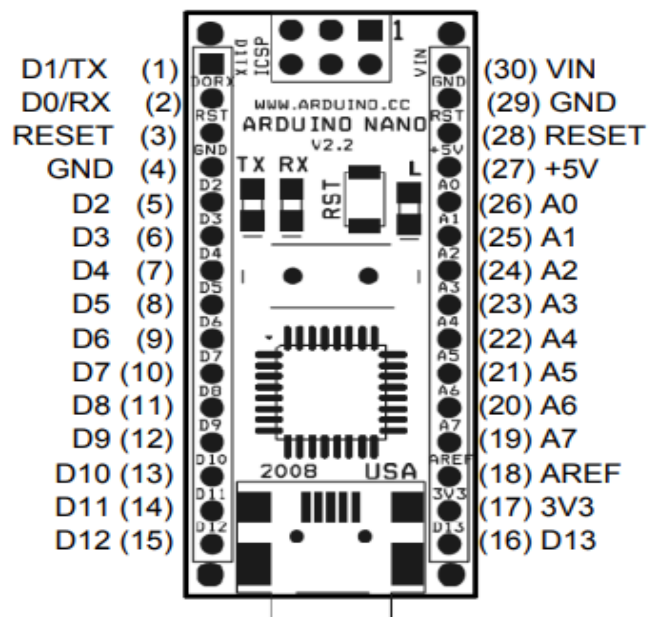


Figure (I. 21): Disposition des broches de l'Arduino nano[10]

Pin No.	Name	Type	Description
1-2, 5-16	D0,D13	I/O	Digital input/output port 0 to 13
3,28	RESET	Input	Reset (active low)
4,29	GND	PWR	Supplyground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19,26	A7,A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

**Tableau 3: Brochage de L'arduino Nano**

## 8. CONCLUSION :

Dans ce chapitre, nous avons étudié les principes fondamentaux du traitement du signal et des microcontrôleurs, en définissant et classifiant les différents types de signaux. Nous avons examiné l'utilisation des oscilloscopes, analogiques et numériques, pour analyser ces signaux. Les concepts de conversion analogique-numérique ont été explorés, avec un focus sur le convertisseur ADC0831. La plateforme Arduino, en particulier l'Arduino Nano, a été présentée, en détaillant ses caractéristiques et applications. Ce chapitre établit les bases essentielles pour comprendre et manipuler les signaux et microcontrôleurs, ouvrant la voie à des applications innovantes en électronique et informatique, notamment dans le développement de mini oscilloscopes numériques à 4 voies pour des applications de basse fréquence.

## **CHAPITRE II: Conception et réalisation du mini oscilloscope BF numérique**

## II-Conception et réalisation du mini oscilloscope BF numérique

### 1. Introduction :

Ce chapitre explore la conception et la réalisation d'un mini oscilloscope BF numérique à quatre voies, essentiel pour l'observation des signaux électroniques en temps réel. En utilisant Python et la bibliothèque Tkinter pour développer une interface graphique intuitive, le projet combine composants matériels et logiciels accessibles. Python est choisi pour sa simplicité et sa polyvalence, tandis que Tkinter facilite la création d'interfaces utilisateur. L'intégration de l'Arduino Nano et des convertisseurs ADC0831 permet de capturer et d'afficher les signaux analogiques en temps réel.

### 2. Architecture du mini oscilloscope BF numérique

#### 2.1.Vue d'ensemble de l'architecture du mini oscilloscope BF numérique à mémoire 4 voies :

Un oscilloscope numérique à mémoire (DSO) capte, traite et présente les formes d'ondes des signaux électriques. Utilisé dans l'électronique, l'ingénierie et la science, il permet de représenter et analyser des phénomènes électriques. Un DSO classique à quatre voies de mémoire comprend plusieurs éléments essentiels.

- **Générateurs de fonctions** :Utilisés pour générer des signaux de test de différentes fréquences et d'amplitudes ne dépassant pas 5 V, pour que l'oscilloscope puisse les mesurer.
- **Convertisseur Analogique-Numérique (ADC0831)** : Utilisé pour numériser les signaux analogiques reçus des générateurs de fonctions.
- **Microcontrôleur** : Le microcontrôleur sur la plaque d'essai (Arduino Nano) reçoit les données numérisées de l'ADC0831 et les transmet à l'ordinateur portable pour l'affichage et l'analyse.
- **Interface Tkinter**:L'interface graphique Tkinter est utilisée par l'ordinateur portable afin d'afficher les formes d'onde capturées et de faciliter la visualisation et l'analyse des signaux.

#### 2.2. Fonctionnalité

- **Génération et mesure des signaux**:Les générateurs de fonctions génèrent des signaux mesurés par l'oscilloscope. En ajustant la fréquence et en maintenant l'amplitude de ces signaux en dessous de 5 V, on peut tester les performances et la précision de l'oscilloscope.
- **Numérisation des signaux**:L'ADC0831 numérise les signaux analogiques des générateurs de fonctions.

- **Traitement des données:** Les données numérisées par l'ADC0831 sont envoyées au microcontrôleur (arduino nano)
- **Transmission des données :**Les données numérisées sont transmises par le microcontrôleur à l'ordinateur portable.
- **Visualisation et analyse :**L'interface Tkinter sur le PC portable affiche les signaux capturés, permettant une surveillance et une analyse en temps réel.

### 3. Programmation en Python :

#### 3.1. Définition :

Python est un langage de programmation interprété de haut niveau, apprécié pour sa simplicité et sa lisibilité. Créé par Guido van Rossum et lancé en 1991, il privilégie une syntaxe claire et expressive, ce qui le rend idéal pour les débutants et les programmeurs expérimentés.



**Figure II. 1: logo de Python.**

Python prend en charge plusieurs paradigmes de programmation, notamment procédural, orienté objet et fonctionnel. Il inclut une bibliothèque standard complète pour des tâches variées comme l'I/O de fichier, la mise en réseau et la manipulation de données, réduisant le besoin de bibliothèques externes. Sa polyvalence et son adoption généralisée couvrent le développement web, l'informatique scientifique, l'analyse de données, l'IA, l'apprentissage automatique, l'automatisation et le scripting. Des bibliothèques comme NumPy, Pandas, TensorFlow, Django et Flask étendent ses capacités. Python est très populaire grâce à sa simplicité, sa lisibilité et son soutien communautaire, étant utilisé par développeurs, chercheurs, scientifiques de données et éducateurs, et figurant parmi les langages les plus populaires dans l'industrie et les universités.

#### 3.2. Avantages de Python:

##### Simplicité et clarté :

- Syntaxe simple et lisible, idéale pour débutants et professionnels.
- Code concis accélérant le développement.

### **Versatilité:**

- Utilisé dans le développement web, l'analyse de données, le machine learning, l'automatisation, etc.
- Large éventail de bibliothèques et cadres pour diverses applications.

### **Grande communauté:**

- Communauté active offrant soutien, ressources de formation et bibliothèques tierces.

### **Écosystème de bibliothèques riches:**

- Bibliothèques comme NumPy, pandas, scikit-learn, Django, Flask facilitent le développement d'applications complexes.

### **Facilité d'apprentissage:**

- Syntaxe simple et documentation complète, recommandé comme premier langage de programmation.

### **Interopérabilité:**

- Intégration facile avec d'autres langages (C/C++, Java, .NET), exploitant les caractéristiques de chacun.

### **Source ouverte et communauté active:**

- Open-source avec une licence libre, favorisant la collaboration et l'innovation.

## **4. Utilisation de Tkinter pour le développement d'interfaces graphiques d'utilisateur (GUI) :**

### **4.1. Introduction à Tkinter:**

Tkinter est un kit d'outils GUI (Graphical User Interface) standard fourni avec Python pour la construction d'applications de bureau. Il est basé sur le kit d'outils de l'interface graphique Tk, qui est né dans le cadre du langage de script Tcl (Tool Command Language). Tkinter fournit un ensemble de modules Python qui permettent aux développeurs de créer des applications GUI simples et complexes avec facilité.

### **4.2. Les avantages de Tkinter:**

- **Facilité d'utilisation:** Tkinter est connu pour sa simplicité et la facilité d'utilisation, ce qui le rend adapté à la fois pour les développeurs débutants et expérimentés.
- **Plateforme croisée :** Les applications Tkinter peuvent être exécutées sur divers systèmes d'exploitation sans modification, assurant la compatibilité entre les plateformes.

- **Documentation détaillée:** Tkinter a une documentation complète et une grande communauté, fournissant des ressources et un soutien suffisants pour les développeurs.

## 5. Utiliser Python et Tkinter pour afficher 4 signaux :

### 5.1. Introduction :

Dans notre code, nous utilisons Python avec la bibliothèque Tkinter pour créer une interface utilisateur pour afficher quatre signaux sur un graphique en temps réel. Les données du signal sont obtenues à partir d'un Arduino Nano connecté au port USB de l'ordinateur.

### 5.2. Bibliothèques utilisées :

**time:** La bibliothèque de temps est utilisée pour fournir des fonctions liées au temps, telles que le temps.sleep(), qui est utilisé pour attendre une courte période avant de commencer à lire les signaux.

**numpy :** Cette bibliothèque est utilisée pour des calculs numériques efficaces. Bien qu'il soit importé, il n'est pas utilisé dans ce code spécifique.

**os:** La bibliothèque os fournit des fonctions pour l'interaction avec le système d'exploitation. Dans ce code, il n'est pas utilisé.

**serial:** Cette bibliothèque est utilisée pour établir une connexion série avec l'Arduino Nano pour recevoir des données de signal.

**sys :** Cette bibliothèque permet d'accéder à certaines variables utilisées ou maintenues par l'interprète Python et à des fonctions qui interagissent fortement avec l'Interprète. Dans ce code, il n'est pas utilisé.

**random :** Cette bibliothèque est utilisée pour générer des nombres pseudo-random. Dans ce code, il n'est pas utilisé.

**tkinter :** Tkinter est la bibliothèque Python standard pour la création d'interfaces graphiques. Il est utilisé pour créer la fenêtre principale de l'interface utilisateur et afficher le graphique du signal.

### 5.3. Explication du code :

- ❖ Le code commence par importer les bibliothèques nécessaires et initialiser les variables globales.
- ❖ Création Canvas : Le composant Canvas est créé à l'aide de la ligne suivante :

```
1 C = Canvas(top,width=1000,borderwidth=0,highlightthickness=0, height=haut*24/32)
```

**Figure II. 2: Création d'un Canvas avec Tkinter**

‘C’ est le nom de variable attribué au widget `Canvas`. Le Canvas est ajouté à la fenêtre ‘top’ avec une largeur et une hauteur spécifiques.

❖ **Éléments de dessin :** Divers éléments graphiques tels que des lignes et du texte sont dessinés sur le Canvas en utilisant ses méthodes. Par exemple :

```
1 C.create_line(x1, y1, x2, y2, fill='blue')
2
```

**Figure II. 3: Code Tkinter pour la Création d'une Ligne sur un Canvas**

Ce code crée une ligne droite entre le point ‘(x1, y1)’ et le point ‘(x2, y2)’ avec une couleur bleue.

❖ **Éléments de manipulation:** Le Canvas permet aux développeurs de modifier les propriétés des éléments dessinés de manière dynamique. Dans le code, la méthode ‘`itemconfigure()`’ est utilisée pour modifier l'état (normal ou caché) des lignes de signal en fonction de l'entrée utilisateur.

❖ **Événements contraignants:** Tkinter permet aux développeurs de lier des événements (tels que des événements de clavier ou de souris) à des fonctions spécifiques. Dans le code, les touches du clavier sont liées à des fonctions pour contrôler l'affichage des signaux.

## 6. Conclusion :

En conclusion, ce chapitre a démontré la réalisation d'un mini oscilloscope BF numérique efficace et pratique. Utilisant Python et Tkinter pour une interface utilisateur interactive, et des composants comme l'Arduino Nano pour la collecte de données, le projet a prouvé sa fiabilité. Il offre une solution accessible pour l'analyse des signaux électroniques, avec des possibilités d'améliorations futures. Ce mini oscilloscope représente une avancée significative dans l'acquisition et l'analyse de données analogiques pour divers utilisateurs.

## **CHAPITRE III : Tests, performances et applications pratiques**

### III. Chapitre3 : Tests, performances et applications pratiques

#### 1. INTRODUCTION :

Dans ce chapitre Nous allons traite des tests, performances et applications pratiques du mini oscilloscope numérique à mémoire 4 voies basé sur un Arduino Nano et des ADC0831. Nous y explorerons le principe de fonctionnement du système, comprenant les composants principaux et le schéma de connexion.

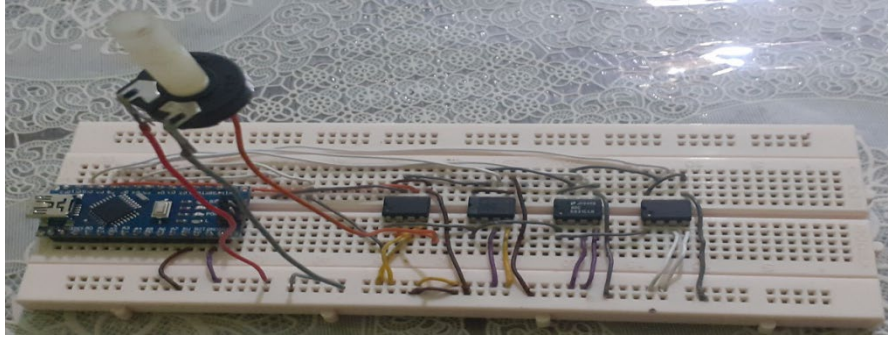
Les aspects matériels et logiciels seront détaillés, notamment les diagrammes de communication et le fonctionnement des ADC0831 et des potentiomètres. La conversion analogique-numérique, la transmission des données à l'ordinateur, et leur affichage via l'interface Tkinter en Python seront également expliqués.

Enfin, le chapitre couvre la connectivité, l'alimentation et l'étalonnage des potentiomètres, ainsi que le développement du code Arduino et du programme Python/Tkinter. Les résultats obtenus en laboratoire illustreront les performances et les applications pratiques du système conçu.

#### 2. Principe de fonctionnement :

##### 2.1.Composants Principaux

- 1) **ArduinoNano:**Le microcontrôleur Arduino Nano est responsable de la lecture des signaux analogiques convertis en numérique par les ADC et de leur transmission à l'interface graphique..
- 2) **ADC0831 :** Les convertisseurs analogiques-numériques (ADC) à 8 bits sont des dispositifs qui transforment les signaux analogiques en valeurs numériques qui peuvent être interprétées par l'Arduino.
- 3) **Résistances Variables (Potentiomètres):** Utilisées pour ajuster les signaux analogiques appliqués aux entrées des ADC0831.
- 4) **Interface Tkinter:** bibliothèque Python qui permet de concevoir l'interface graphique qui présente les signaux.



**Figure III. 1: Circuit de mini oscilloscope**

## 2.2.Schéma de Connexion

### 2.2.1. Arduino Nano :

- **Pin 7 (D7)** : Connecté au pin **1 (CS)** de tous les ADC0831 pour la sélection de l'ADC actif.
- **Pin 6 (D6)** : relié aux broches **7 (CLK)** de tous les ADC0831 afin de fournir le signal horaire.
- **Pins D12, D11, D9, D8** : sont connectées respectivement aux broches **6 (DO)** des ADC0831 afin de recevoir les données numériques.
- **5V** et **GND** pour l'alimentation.

### 2.2.2. ADC0831 (quatre ADC) :

- **Pin 1 (CS)** : Connecté au **pin 7 (D7)** de l'Arduino.
- **Pin 2 (VIN+)** : connecté à la source du signal analogique. C'est cette dernière qui fera l'objet de la conversion A/D.
- **Pin 3 (VIN-)** : Connecté à la tension de référence minimale. Le signal analogique ne doit pas être au dessous de cette tension.
- **Pin 4 (GND)** : Connecté au GND de l'alimentation.
- **Pin 5 (Vref)** : Connecté à la tension de référence maximale. Le signal analogique ne doit pas être au dessus de cette tension.
- **Pin 6 (DO)** : Connecté aux pins D12, D11, D9, et D8 de l'Arduino pour les quatre ADC respectivement.
- **Pin 7 (CLK)** : Connecté au **pin 6 (D6)** de l'Arduino pour le signal d'horloge.
- **Pin 8 (VCC)** : Connecté au VCC de l'alimentation (5V).

### 3. Fonctionnement Détaillé:

#### 3.1. Génération du Signal Analogique :

En ajustant leur wiper 'S', les potentiomètres génèrent des signaux analogiques. On applique ces signaux analogiques aux entrées VIN+ des ADC0831. En ajustant ces potentiomètres, le niveau de tension appliqué à l'ADC est modifié, ce qui entraîne une modification du signal analogique à observer.

#### 3.2. Conversion Analogique-Numérique :

Les ADC0831 convertissent les signaux analogiques en données numériques allant de (0 à 255). Le pin CS (connecté au D7 de l'Arduino) choisit chaque ADC et reçoit un signal d'horloge via le pin CLK (connecté au D6 de l'Arduino).

#### 3.3. Lecture des données par l'Arduino :

Les données numériques des ADC sont accessibles par l'Arduino Nano grâce aux pins D12, D11, D9 et D8, qui représentent respectivement les quatre ADC. L'Arduino lit les données à intervalles réguliers et les envoie à un ordinateur via la connexion série (USB).

#### 3.4. Transmission des données à l'Ordinateur :

L'ordinateur reçoit les données numériques et exécute un programme Python utilisant Tkinter afin d'afficher les signaux sous forme d'oscillogrammes.

#### 3.5. Affichage des Signaux dans l'Interface Tkinter :

Les données série de l'Arduino sont reçues par l'interface Tkinter, traitées et présentées sous forme de graphes ou d'oscillogrammes en temps réel. Le programme Python analyse les données, les interprète et met régulièrement à jour l'affichage afin de représenter les signaux analogiques.

### 4. Définition du Hardware et du Software :

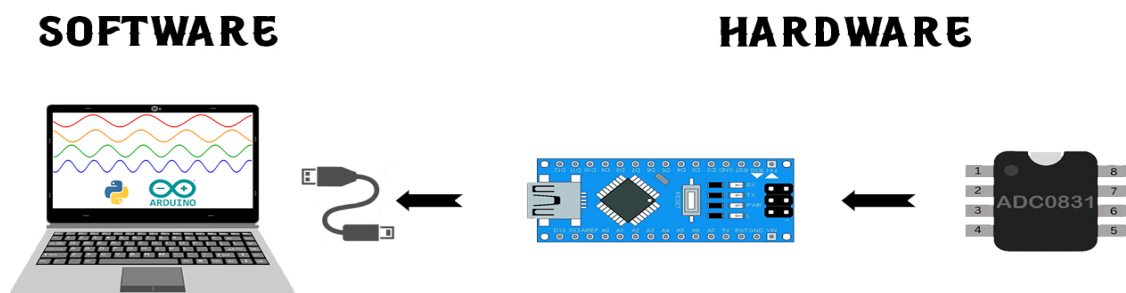
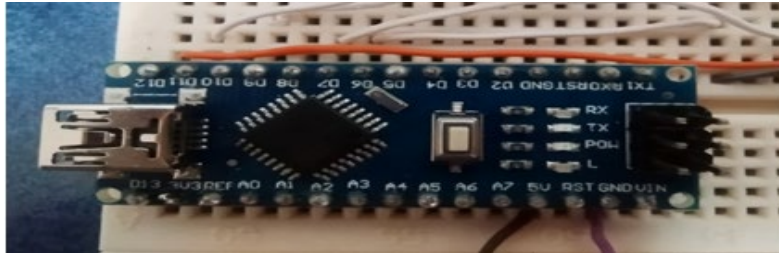


Figure III. 2: Schéma de fonctionnement d'un Mini oscilloscope Numérique.

## 4.1. Hardware:

Le mot "hardware" fait référence à toutes les parties physiques et matérielles d'un système informatique. Pour votre projet, l'ensemble des composants électroniques et mécaniques que vous employez pour fabriquer l'oscilloscope numérique sont inclus dans le hardware. Cela implique :

- **Arduino Nano:** L'objectif de ce microcontrôleur est de lire les données des ADC et de les envoyer à l'ordinateur.



**Figure III. 3: Arduino Nano**

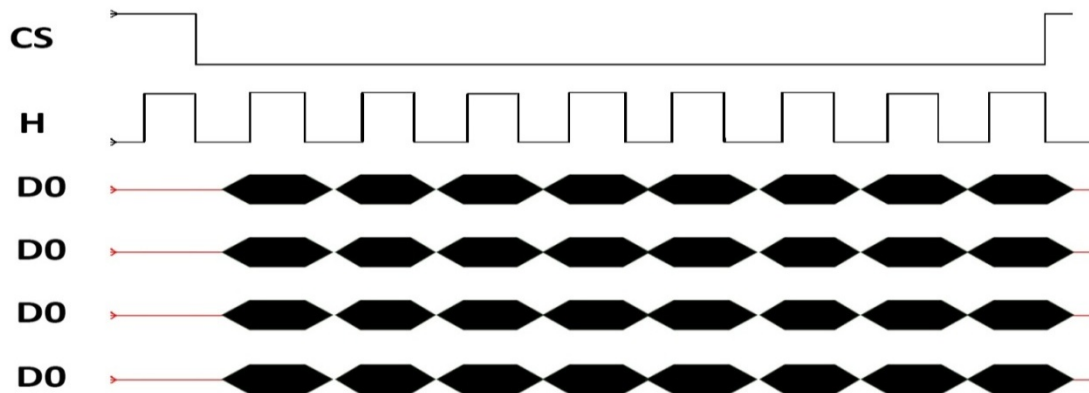
- **ADC0831:** Ces convertisseurs analogiques-numériques sont utilisés pour convertir les signaux analogiques des potentiomètres en données numériques.



**Figure III. 4: ADC0831**

- **Potentiomètres:** L'utilisation de résistances variables pour générer des signaux analogiques.
- **Câblage et alimentation :** Les câbles électriques et les sources d'alimentation qui connectent et alimentent les différents composants.

### 4.1.1. Diagramme de Temps de Communication SPI pour un ADC0831 :



**Figure III. 5: chargement des valeurs analogiques**

La figure (III.5) représente les signaux de contrôle et de données dans une communication SPA (Serial Peripheral Interface) entre un microcontrôleur (Arduino) et plusieurs convertisseurs analogique-numérique (CAN) ADC0831.

**CS:** Sélection du circuit (Chip Select) : Doit être basse (LOW) pour activer l'ADC et permettre la conversion.

**H (CLK) :** Horloge (Reçoit le signal d'horloge externe pour synchroniser les opérations de conversion.

**DO (Data Out):** Données de sortie (Fournit les données numériques série résultant de la conversion de l'entrée analogique).

#### 4.1.1.1. Détails de la séquence :

- a) **Initialisation :** CS est haut, indiquant que les ADC0831 ne sont pas encore sélectionnés.
- b) **Sélection:** CS passe de haut à bas, sélectionnant un ADC0831 pour la communication.
- c) **Transmission des données :**
  - L'horloge H commence à pulser.
  - À chaque front montant de l'horloge, un bit de données est transmis sur la ligne DO.
  - Cette séquence continue jusqu'à ce que tous les bits nécessaires soient transmis.
- d) **Désélection :** CS revient à haut, désélectionnant l'ADC0831, et mettant fin à la communication.

## 4.1.2. Le schéma de fonctionnement de l'ADC0831 :

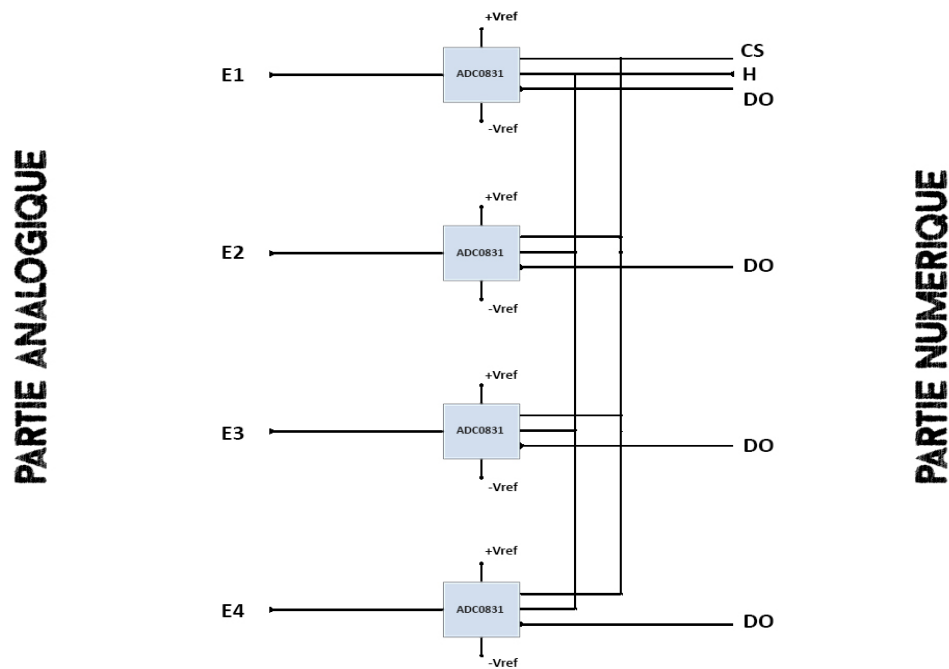


Figure III. 6:diagramme schématique illustrant le fonctionnement de l'ADC0831

La figure (III.6) fournie est un diagramme schématique illustrant le fonctionnement de l'ADC0831, qui est un convertisseur analogique à numérique de 8 bits. (ADC). Le diagramme montre la connexion et le flux des signaux entre les parties analogiques et numériques d'un système. Voici une explication détaillée:

### 4.1.2.1. Les Composants clés:

- **ADC0831:** Le composant principal du diagramme, ce ADC convertit les valeurs analogiques en valeurs numériques.
- **E1, E2, E3, E4:** Ceux-ci représentent les signaux d'entrée analogiques.
- **+Vref et -Vref:** Ce sont les entrées de tension de référence pour l'ADC. La tension de référence définit la plage d'entrée de l'ADC. Par exemple, si +Vref est 5V et -Vref 0V, l'ADC convertira les signaux d'entrée dans la plage de 0-5V en valeurs numériques correspondantes.
- **CS:** Ce signal est utilisé pour activer l'ADC. Lorsque le CS est faible, l'ADC est sélectionné et prêt à effectuer des conversions.
- **H:** Ceci représente le signal d'horloge requis pour le fonctionnement de l'ADC.
- **DO (sortie numérique):** Il s'agit de la sortie de l'ADC où la représentation numérique des entrées analogiques est disponible.

#### 4.1.2.2. Explication du diagramme :

- **Inputs analogiques (E1 à E4) :**

La figure montre quatre signaux d'entrée analogiques (E1 à E4) connectés à quatre puces ADC0831.

Ces entrées sont les signaux qui doivent être convertis de la forme analogique à la forme numérique.

- **Voltage de référence (+Vref et -Vref):**

Chaque puce ADC0831 est connectée à une tension de référencement (+ Vref et -Vref). Ces tensions définissent la plage de conversion des signaux analogiques.

Les entrées de tension de référence sont communes à l'ensemble des quatre ADC dans ce diagramme, ce qui suggère une référence partagée pour la cohérence dans les conversions.

- **Signaux de contrôle (CS et H):**

Le signal CS (Chip Select) est connecté à chaque puce ADC0831. Ce signal contrôle quel ADC est actif pour la conversion.

Le signal H est l'entrée de l'horloge, nécessaire pour synchroniser le processus de conversion. Il est également connecté à chaque ADC.

- **Sorties numériques (DO):**

Chaque ADC0831 fournit une sortie numérique (DO) correspondant à l'entrée analogique qu'il reçoit.

- Les sorties numériques des quatre ADC (DO de E1, E2, E3 et E4) sont affichées sur le côté droit du diagramme.

#### 4.1.2.3. Fonctionnement:

- Lorsque le signal CS est activé (bas), le signal ADC0831 correspondant est sélectionné et prêt à être converti.
- Le signal d'entrée analogique à E1, E2, E3 ou E4 est converti en signal numérique par l'ADC0831.
- Le signal d'horloge (H) conduit le processus de conversion.
- La sortie numérique (DO) fournit une représentation numérique de 8 bits du signal d'entrée analogique.

### 4.1.3. La Valeur Numérique de Sortie du Convertisseur :

La valeur numérique (N) en fonction de la valeur analogique d'entrée (x), des tensions de référence supérieure ( $V_{ref+}$ ) et inférieure ( $V_{ref-}$ ), et de la résolution du convertisseur (n bits). Elle exprime à l'équation suivant :

$$N = \frac{(2^n - 1) (x - V_{ref-})}{(V_{ref+} - V_{ref-})}$$

- N : est la valeur numérique de sortie du convertisseur.
- $2^n$  : représente le nombre de niveaux de quantification du convertisseur, où n est le nombre de bits.
- x : est la tension analogique d'entrée.
- $V_{ref+}$  : est la tension de référence supérieure.
- $V_{ref-}$  : est la tension de référence inférieure.

On aura 8 bits dans L'ADC 0831 donc n=8 :

Alors :

$$N = \frac{(2^8 - 1) (x - V_{ref-})}{(V_{ref+} - V_{ref-})} = \frac{255 \times (x - V_{ref-})}{(V_{ref+} - V_{ref-})}$$

➤ Si  $x = V_{ref-}$  donc : N=0

$$N = \frac{255 \times (V_{ref-} - V_{ref-})}{(V_{ref+} - V_{ref-})} = 0$$

➤ Si  $x = V_{ref+}$  donc : N=255

$$N = \frac{255 \times (V_{ref+} - V_{ref-})}{(V_{ref+} - V_{ref-})} = 255$$

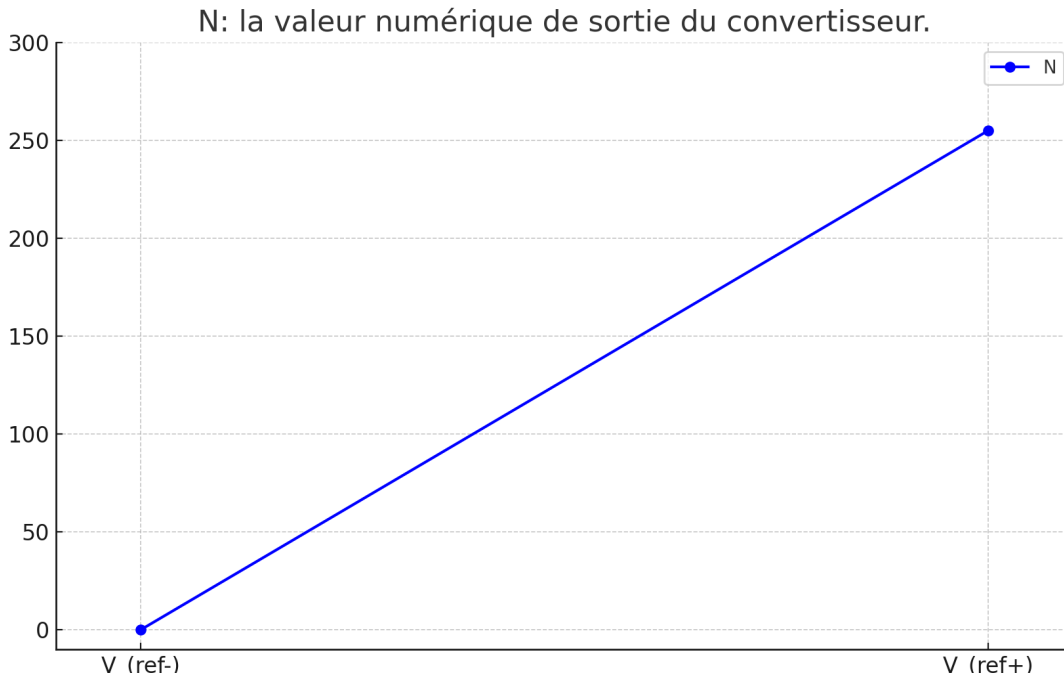


Figure III. 7: La valeur numérique de sortie du convertisseur en fonction de  $V_{ref+}$  et  $V_{ref-}$

#### 🔧 Explication de la courbe:

- Lorsque  $x = V_{ref-} = 0V$ ,  $N$  est également 0. C'est le point (0, 0) sur la courbe.
- La courbe est une droite passant par l'origine, indiquant une relation linéaire directe entre la tension d'entrée  $x$  et la valeur numérique  $N$ .
- Lorsque  $x = V_{ref+} = 5V$ ,  $N$  atteint sa valeur maximale de 255. C'est le point (5, 255) sur la courbe.
- Pour toute valeur intermédiaire de  $x$  entre 0V et 5V,  $N$  varie proportionnellement entre 0 et 255.

#### 4.1.4. Points Importants pour le Hardware:

##### 4.1.4.1. Connectivité :

- Vérifiez la conformité et la sécurité de toutes les connexions entre l'Arduino, les ADC0831 et les potentiomètres.
- Veillez à utiliser des câbles et des connecteurs adaptés afin d'éviter les contacts erronés et les interférences.

## 4.2. Software:

Software est le nom donné à tous les programmes et instructions qui contrôlent le hardware et exécutent les tâches spécifiques. Lors de votre projet, le software comprend le code écrit pour l'Arduino Nano ainsi que le programme Python qui utilise Tkinter pour l'affichage des signaux. Cela comprend :

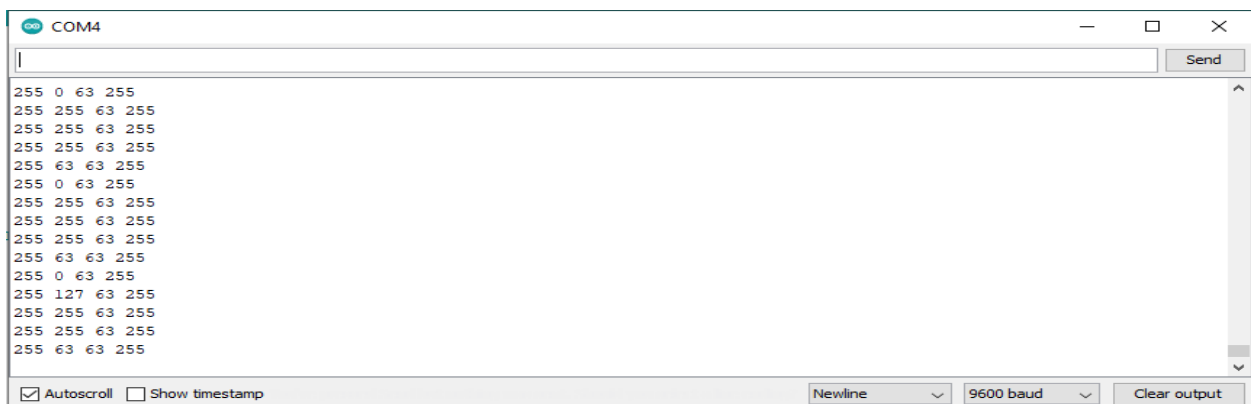
### ➤ Code Arduino:

Programme chargé de configurer les pins, de lire les données des ADC et de les envoyer à l'ordinateur via une connexion série.

### ➤ Programme Python/Tkinter:

Selon une interface graphique, un script lit les données série provenant de l'Arduino, les traite et les affiche sous forme de graphes ou d'oscillogrammes.

### 4.2.1. La Lecture des valeurs du potentiomètre via l'ADC0831 sur le Moniteur Série Arduino :



**Figure III. 8: Interface du Moniteur Série Arduino Affichant les Données de Signal**

Les valeurs lues à partir de potentiomètres connectés à un convertisseur analogique-numérique (ADC0831) sont affichées sur le Moniteur Série Arduino, qui est connecté à une carte Arduino via le port COM4. Lorsque les potentiomètres sont tournés, les tensions analogiques fluctuent. Ces tensions analogiques sont converties en valeurs numériques (de 0 à 255) par l'ADC0831, que l'Arduino lit ensuite via une interface série. Ensuite, les valeurs sont transmises et présentées dans le Moniteur Série, comme le montrent les lignes de données ("255 0 63 255"). Cela permet de visualiser en temps réel les variations de potentiomètres.

## 4.2.2. L'organigramme d'arduino :

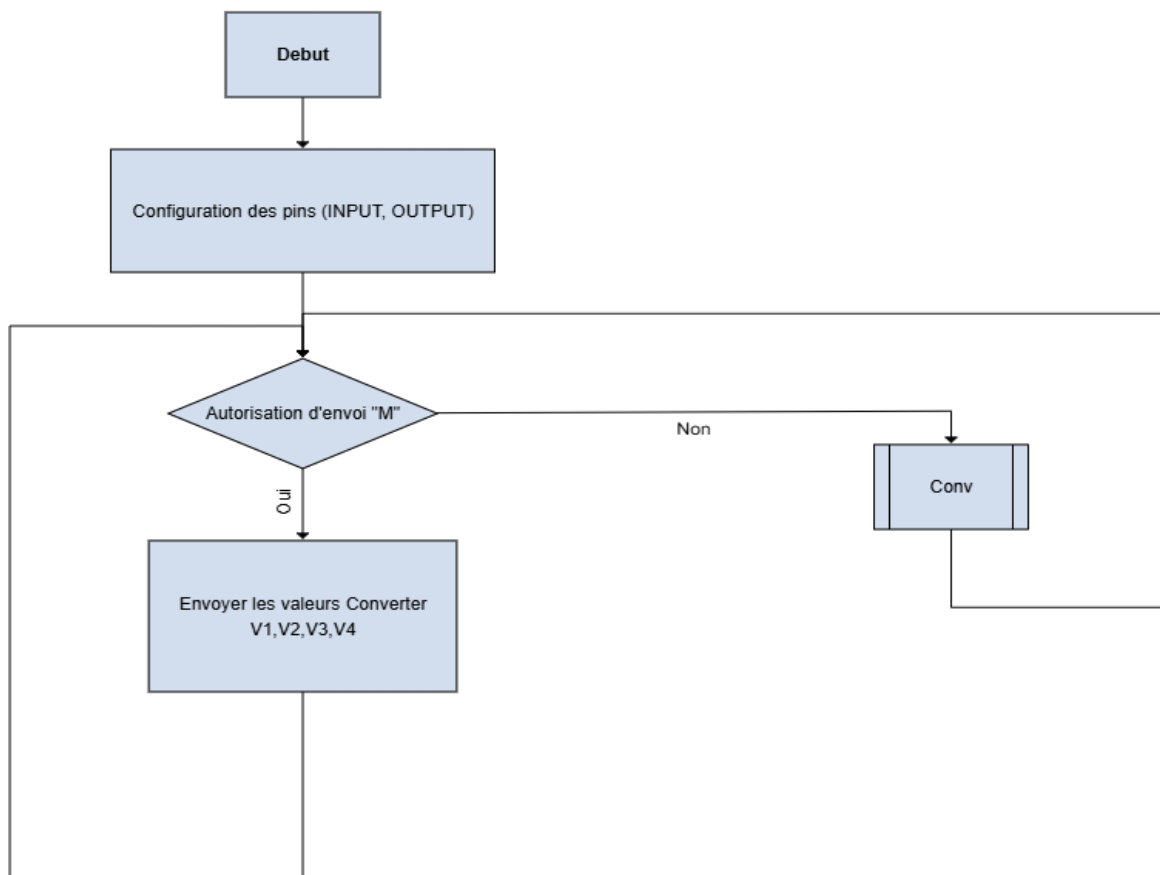


Figure III. 9: Diagramme de Flux pour l'Envoi des Valeurs Converties par l'Arduino

### 4.2.2.1. Première étape :

Le flux de travail d'un programme Arduino est illustré dans cet organigramme. Il configure les pins, attend une autorisation d'envoi, puis envoie des valeurs converties. Voici une explication approfondie de chaque étape de l'organigramme :

- **Début :**

Le programme commence ici.

- **Configuration des pins (INPUT, OUTPUT) :**

Les pins de l'Arduino sont configurés comme entrées ou sorties. Cette étape est cruciale pour préparer l'Arduino à lire les signaux entrants et à envoyer des signaux sortants.

- Les Pins 12, 11, 9, 8 comme entrées.
- Les Pins 7, 6 comme sorties
- **Autorisation d'envoi "M" :**

Le programme vérifie si l'autorisation d'envoyer des données (représentée par "M") est reçue. Cette vérification se fait généralement via une entrée numérique ou une commande reçue par communication série.

analogiques-numériques (CAN) V1, V2, V3, et V4. Ces valeurs sont généralement les tensions lues sur des entrées analogiques, converties en valeurs numériques.

- **Conv (Conversion) :**

Si l'autorisation d'envoi n'est pas donnée (Non), le programme effectue une autre action ou attend (Conv). Cette étape pourrait impliquer la conversion des valeurs analogiques en numériques ou toute autre opération de préparation avant une nouvelle tentative d'envoi.

- **Boucle de retour :**

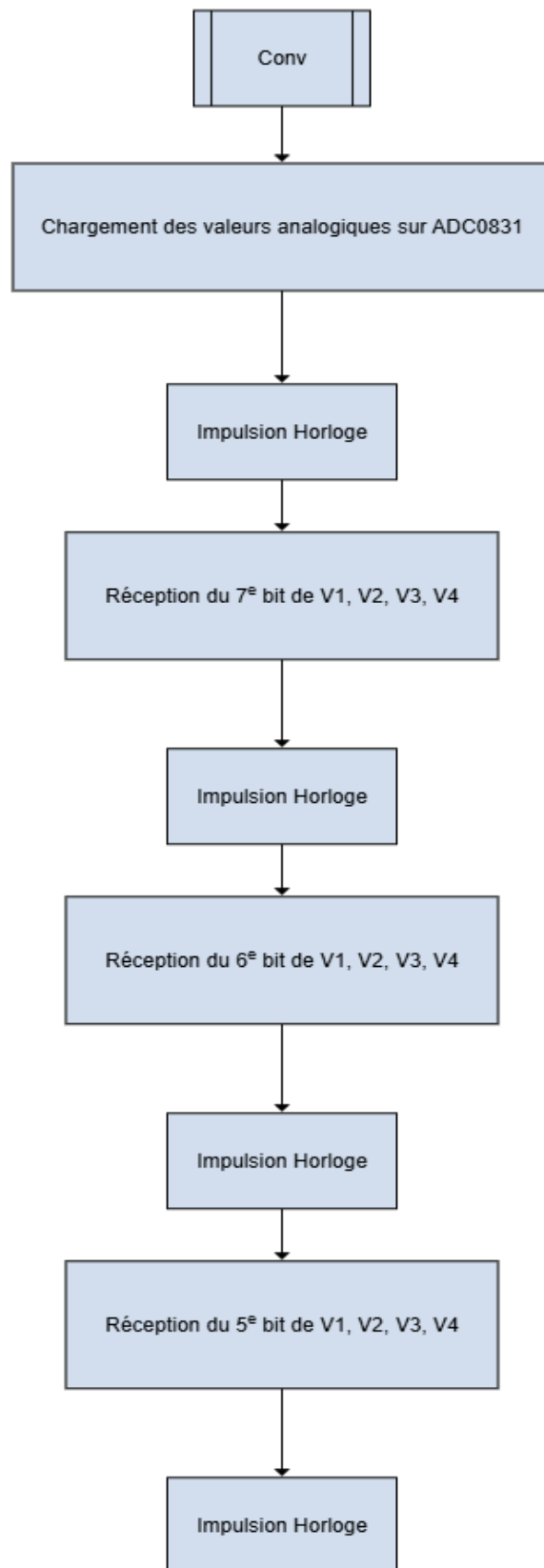
Après avoir envoyé les valeurs ou effectué la conversion, le programme retourne à la vérification de l'autorisation d'envoi "M". Cela crée une boucle continue, permettant au programme de rester actif et de répéter l'envoi des valeurs tant que le système fonctionne.

- **Résumé**

L'organigramme illustre un processus itératif où l'Arduino :

- Configure ses pins en entrées ou sorties.
- Attend une autorisation pour envoyer des données.
- Envoie les valeurs converties des entrées analogiques lorsqu'il reçoit l'autorisation.
- Effectue des conversions ou d'autres actions si l'autorisation n'est pas encore donnée.
- Retourne à la vérification de l'autorisation, créant une boucle de fonctionnement continue.

#### 4.2.2.2. Deuxième étape :



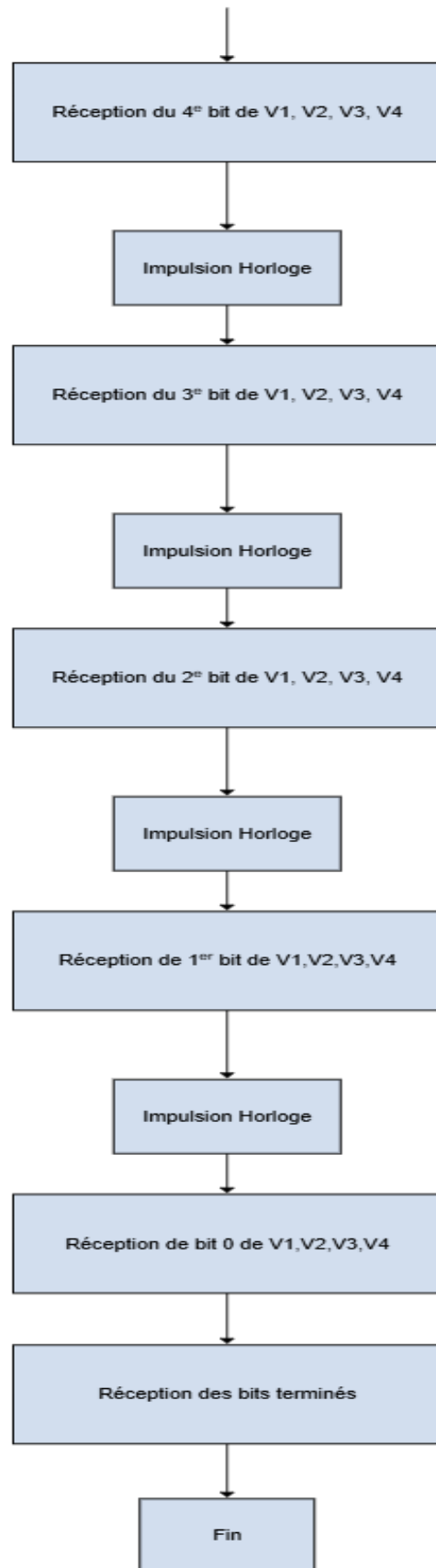


Figure III. 10: Séquence de Conversion et de Réception des Données Analogiques par l'ADC0831

Cet organigramme décrit le processus de conversion et de réception des valeurs analogiques sur un convertisseur analogique-numérique (ADC) de type ADC0831. Voici une explication étape par étape de chaque bloc de l'organigramme :

Le processus de conversion et de réception des bits est terminé.

Chaque étape "Impulsion Horloge" permet de lire successivement chaque bit des valeurs converties par l'ADC0831, de manière synchrone. L'ADC0831 convertit les valeurs analogiques en valeurs numériques bit par bit, et ces valeurs numériques sont ensuite réceptionnées et traitées.

### 4.2.3. L'organigramme de Python

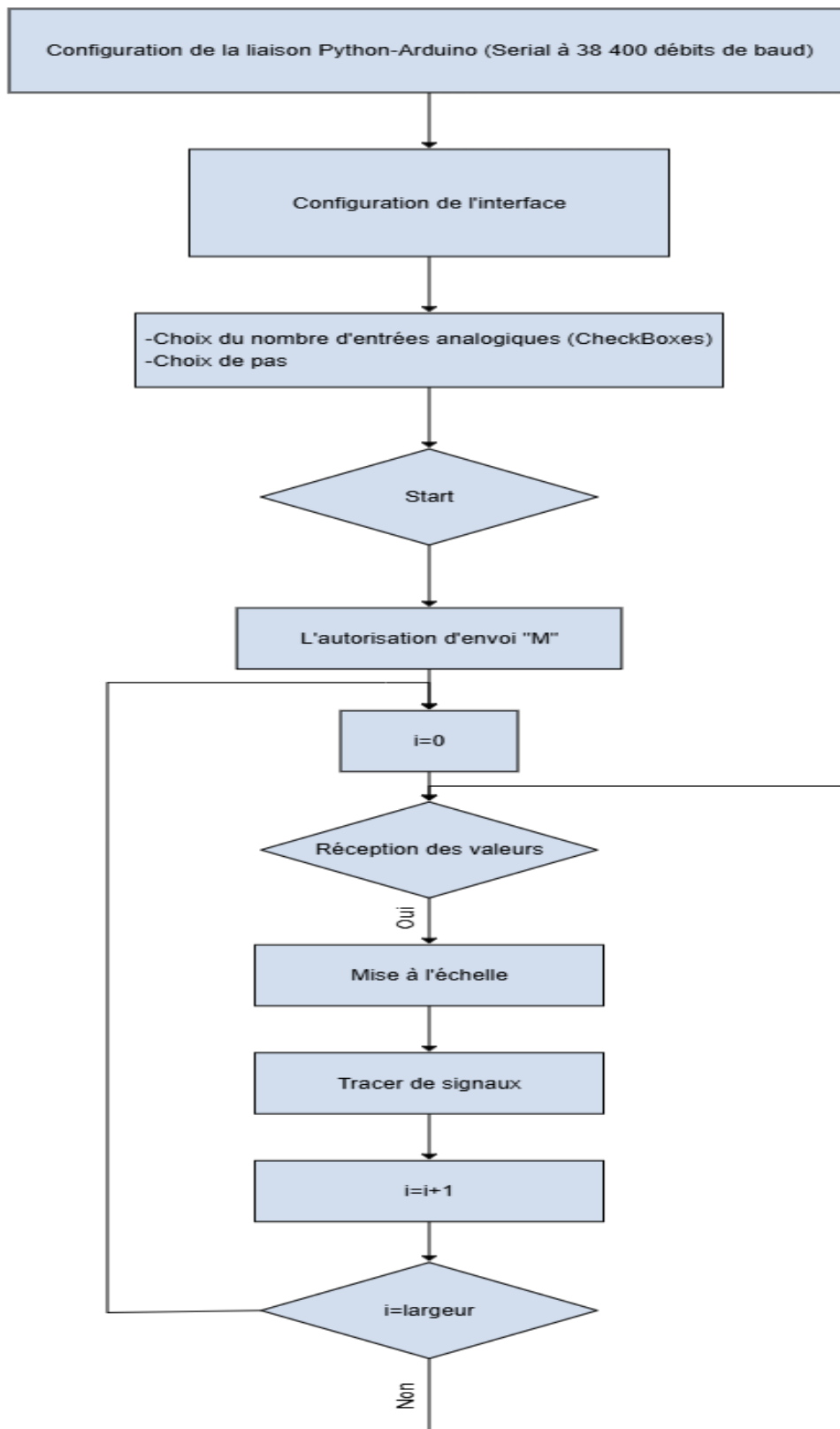


Figure III. 11: Flux de Configuration et d'Affichage des Signaux

#### 4.2.4. Points Importants pour le Software:

##### 4.2.4.1. Lecture et Traitement des Données :

- Il est nécessaire que le code Arduino puisse lire les données des ADC de manière efficace et les envoyer en temps réel à l'ordinateur.
- Vérifier que la configuration de la communication série est adéquate afin d'éviter les pertes de données ou les décalages.

##### 4.2.4.2. Interface Utilisateur :

- L'interface utilisateur du programme Python/Tkinter doit être claire et intuitive afin d'afficher les signaux analogiques.
- S'assurer de mettre à jour régulièrement l'affichage afin de fournir une visualisation en temps réel des signaux.

##### 4.2.4.3. Robustesse et Fiabilité :

- Le logiciel doit avoir une solidité suffisante pour faire face aux erreurs de communication ou aux fluctuations des signaux sans être endommagé.
- Mettre en place des dispositifs de gestion des erreurs et de récupération afin de garantir une expérience utilisateur sans problème.

#### 4.2.5. Interface Tkinter :

##### 4.2.5.1. Menu:



Figure III. 12: Explication du menu Tkinter

### 1) Menu principal :

Un sous-menu contenant des cases à cocher permet de choisir les signaux à afficher.

- Les nombres de signaux affichés : indique le nombre de signaux actuellement affichés.
- Liste des noms des signaux affichés : Montre les noms des signaux actuellement affichés.

### 2) Contrôle des signaux :

- **Start** : Commence à afficher les signaux.
- **Stop** : Arrête l'affichage des signaux.
- **Suit** : Reprend l'affichage des signaux après une pause.

### 3) Contrôle du Pas :

Les boutons + et - permettent d'augmenter et de diminuer la fréquence d'affichage des signaux.

Affichage du Pas : Montre la valeur actuelle du pas.

### 4) Sauvegarde et Lecture :

Save : Stocke les informations et les valeurs numériques des signaux dans un document.

Read : lit et présente les informations des signaux sauvegardés.

#### 4.2.5.2. Les Zones d'affichage :

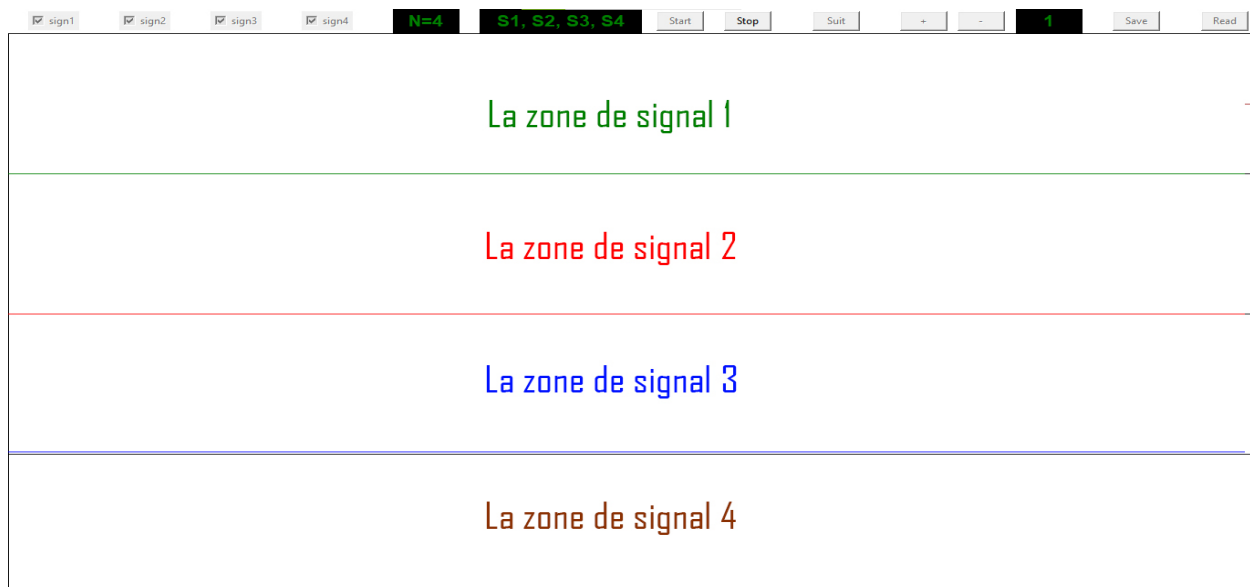


Figure III. 13: Interface Tkinter

Il y a quatre zones dans l'interface, chacune affichant un signal particulier avec une couleur différente :

- **Zone 1** : le signal 1 est affiché en vert.
- **Zone 2** : le signal 2 est affiché en rouge.

- **Zone 3** : le signal 3 est affiché en bleu.
- **Zone 4** : le signal 4 est affiché en couleur marron.

Voici un exemple sur l'affichage de signal 1 :



Figure III. 14: L'affichage d'un signal 1 par potentiomètre

## 5. Le montage du circuit sous isis proteus

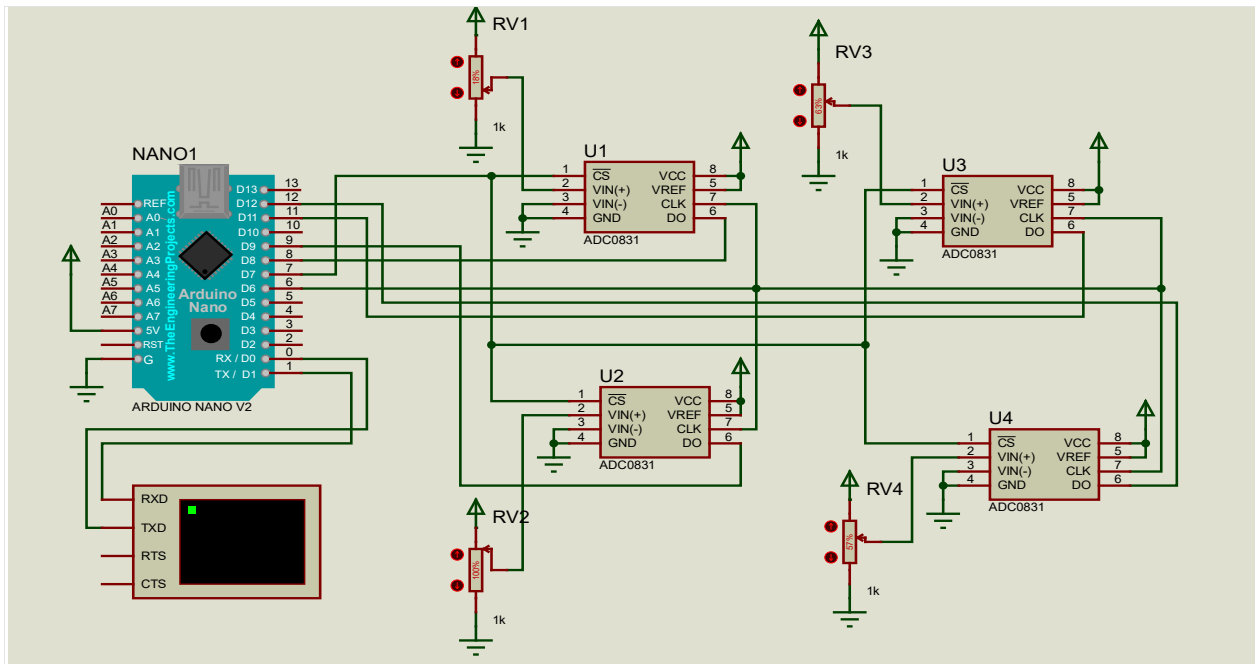


Figure III. 15: Schéma de circuit sous isis proteus avec un seul plage de tension 0 à 5V

## 5.1.Composants principaux :

### ➤ **Arduino Nano (NANO V2):**

Un microcontrôleur chargé de lire les valeurs analogiques via les convertisseurs CAN.

### ➤ **Convertisseurs ADC0831 (U1, U2, U3, U4):**

Il s'agit de convertisseurs analogique-numérique de 8 bits disponibles.

Pour chaque ADC0831, une tension analogique est convertie en un signal numérique que l'Arduino peut analyser.

### ➤ **Potentiomètres (RV1, RV2, RV3, RV4):**

Les ADC0831 sont alimentés par des tensions analogiques variables afin de simuler des signaux analogiques à mesurer.

## 5.2.Connections détaillées :

### ➤ **Alimentation:**

Les broches VCC des ADC0831 (broche 8) sont reliées à l'Arduino à une tension de 5V.

Les broches GND des ADC0831 (broche 4) sont connectées à la masse (GND) de l'Arduino.

### ➤ **Entrées Analogiques (VIN+ ET VIN-):**

Pour chaque ADC0831, le potentiomètre correspondant (RV1, RV2, RV3, RV4) est relié au VIN+ (broche 2).

On relie les broches VIN- (broche 3) à la masse.

### ➤ **Tension de tension ( $V_{ref}$ ):**

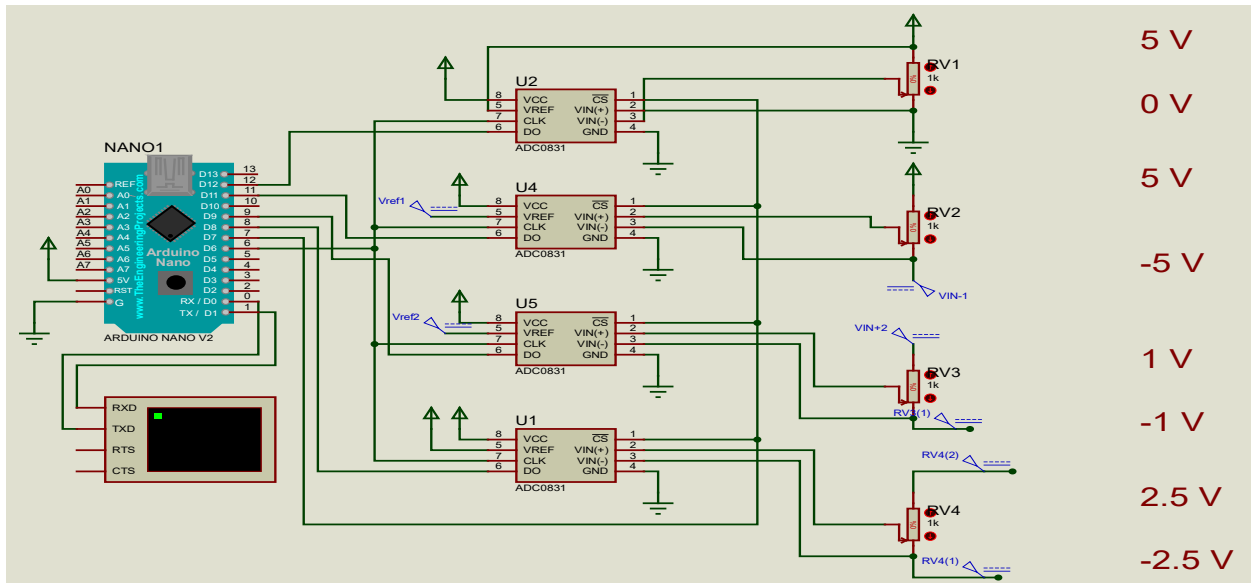
Les broches VREF (broche 5) des ADC0831 sont reliées à une tension de 5V, ce qui leur permet de servir de point de référence pour les conversions analogique-numérique.

### ➤ **Communication SPI:**

Tous les ADC0831 ont des broches CLK (broche 7) qui sont reliées à une broche numérique de l'Arduino (Broche D6).

Chaque ADC0831 est connecté à différentes broches numériques de l'Arduino, telles que U1 à D8, U2 à D9, U3 à D11, U4 à D12.

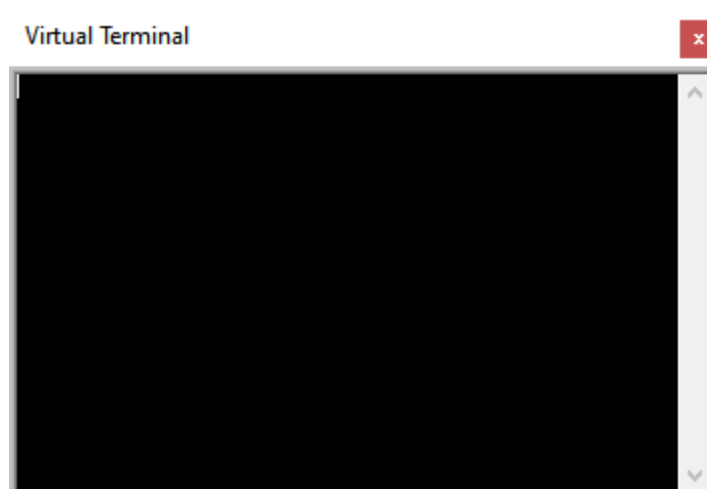
Chaque ADC0831 possède des broches CS (broche 1) qui sont connectées à des broches numériques de l'Arduino (broches D7).



**Figure III. 16: Schéma de circuit sous isis proteus à des potentiomètres avec des plages de tension et des références spécifiques**

Ce montage sur ISIS Proteus montre un Arduino Nano connecté à quatre ADC0831, chacun relié à un potentiomètre avec des plages de tension et des références spécifiques. Les valeurs analogiques des potentiomètres sont converties en valeurs numériques par les ADC et envoyées à l'Arduino. Ces valeurs sont ensuite affichées en temps réel sur un terminal virtuel, permettant de visualiser les variations de tension mesurées par chaque potentiomètre.

### 5.3. Affichage des Données Converties sur le Terminal Virtuel via Arduino :



**Figure III. 17: Virtuel terminal**

#### 5.3.1. Explication :

Le terminal virtuel est utilisé pour afficher les valeurs numériques converties envoyées par l'Arduino Nano via la communication série. Initialement, l'écran du terminal virtuel est vide.

Lorsqu'on appuiesur la touche 'M', l'Arduino envoie les valeurs converties des quatre potentiomètres, affichées en hexadécimal (HEX), permettant de visualiser en temps réel les variations de tension des potentiomètres sur le terminal virtuel.

### 5.3.2. Explication des Valeurs Affichées sur le Terminal Virtuel:

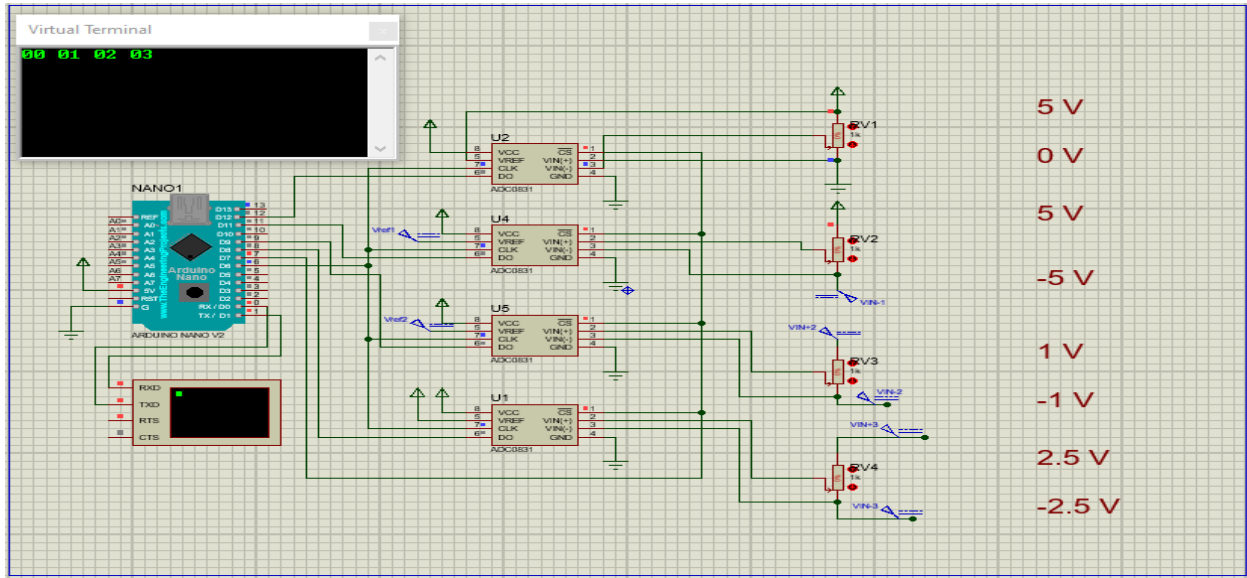


Figure III. 18: Simulation du Mini Oscilloscope BF Numérique avec Arduino Nano et ADC0831

Après avoir appuyé sur la touche 'M', les valeurs affichées sont "4D 00 01 02 03".

4D : Ce code ASCII pour la lettre 'M' (4D en hexadécimal équivaut à 77 en décimal) confirme que l'appui sur la touche 'M' a été enregistré correctement par le système.

00, 01, 02, 03 : Ces valeurs représentent les données numériques converties des potentiomètres. En détaillant chaque octet (byte) en bits pour mieux comprendre :

- 00 en hexadécimal est égal à 0000 0000 en binaire.

Les six premiers bits (000000) indiquent que le potentiomètre est à un niveau bas pour la plage de -2.5V à 2.5V.

Les deux derniers bits (00) indiquent que cette valeur provient du premier signal du premier ADC.

- 01 en hexadécimal est égal à 0000 0001 en binaire.

Les six premiers bits (000000) indiquent un niveau bas pour la plage de -1V à 1V.

Les deux derniers bits (01) indiquent que cette valeur provient du deuxième signal du deuxième ADC.

- 02 en hexadécimal est égal à 0000 0010 en binaire.

Les six premiers bits (000000) indiquent un niveau bas pour la plage de -5V à 5V.

Les deux derniers bits (10) indiquent que cette valeur provient du troisième signal du troisième ADC.

- 03 en hexadécimal est égal à 0000 0011 en binaire.

Les six premiers bits (000000) indiquent un niveau bas pour la plage de 0V à 5V.

Les deux derniers bits (11) indiquent que cette valeur provient du quatrième signal du quatrième ADC.

### 5.3.3. Calcul des résistances :

Pour calculer les résistances on aura cette équation :

$$N = \frac{(2^n - 1) (x - V_{ref-})}{(V_{ref+} - V_{ref-})}$$

- **N** : est la valeur numérique de sortie du convertisseur.
- **2<sup>n</sup>** : représente le nombre de niveaux de quantification du convertisseur, où n est le nombre de bits.
- **x** : est la tension analogique d'entrée (pin 2 d'ADC0831 (Vin+)).
- **V<sub>ref+</sub>** : est la tension de référence supérieure (pin 5 d'ADC0831 (Vref)).
- **V<sub>ref-</sub>** : est la tension de référence inférieure (pin 3 d'ADC0831 (Vin-)).

#### 0V à 5V :

On a:

$$x=V_{in(+)}, 2^n - 1 = 2^8 - 1 = 255 \text{ et } V_{ref-} = V_{in(-)}$$

Alors:

$$N = \frac{V_{in(+)} - V_{in(-)}}{V_{ref}} \times 255$$

Si N=0 et Vin(+)=0V:

Donc :

$$0 = \frac{0 - \text{Vin}(-)}{\text{Vref}} \times 255$$
$$\text{Vin}(-) = 0\text{V}$$

Si  $N=255$ ,  $\text{Vin}(-)=0\text{V}$  et  $\text{Vin}(+)=5\text{V}$  :

Donc :

$$255 = \frac{5 - 0}{\text{Vref}} \times 255$$
$$\text{Vref} = 5\text{V}$$

Pas besoin pas des résistances

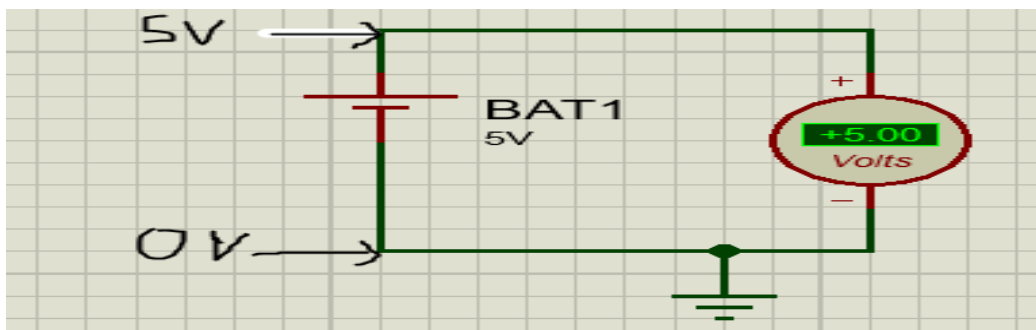


Figure III. 19: Schéma de Circuit de Source de Tension de 5V

-5V à 5V :

On a:

$$N = \frac{\text{Vin}(+) - \text{Vin}(-)}{\text{Vref}} \times 255$$

Si  $N=0$  et  $\text{Vin}(+)=-5\text{V}$ :

Donc:

$$0 = \frac{-5 - \text{Vin}(-)}{\text{Vref}} \times 255$$
$$\text{Vin}(-) = -5\text{V}$$

Si  $N=255$ ,  $\text{Vin}(-)=-5\text{V}$  et  $\text{Vin}(+)=5\text{V}$  :

Donc :

$$255 = \frac{5 - (-5)}{\text{Vref}} \times 255$$
$$\text{Vref} = 10\text{V}$$

Pour calculer la résistance on fixe l'une et on calcule l'autre :

On fixe

$$R_2 = 1\text{K ohm}$$

Par diviseur de tension on a :

$$\frac{12}{R_1 + R_2} \times R_2 = 10V$$

Alors :

$$\frac{12}{R_1 + 1} = 10$$

$$10R_1 + 10 = 12$$

Donc:

$$R_1 = 200\text{ohm}$$

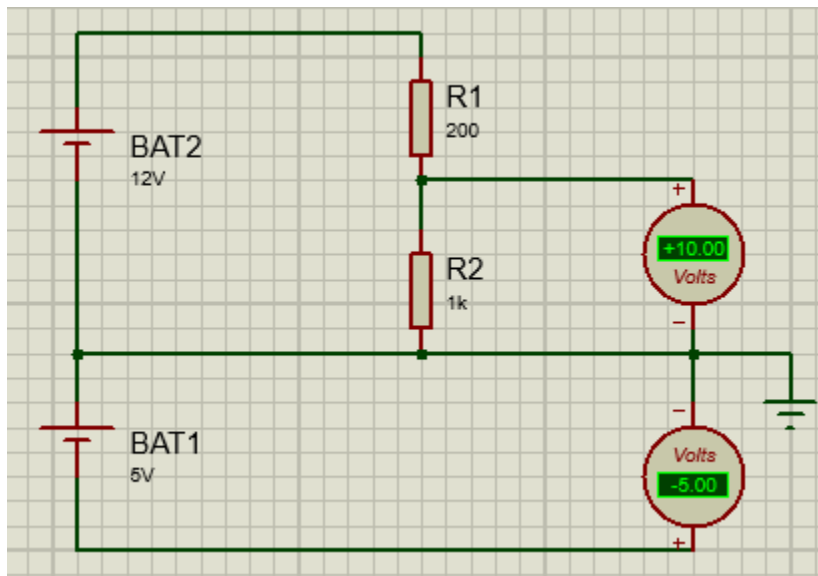


Figure III. 20: Schéma Diviseur de Tension qui donne deux tension de référence 10V et -5V

-1V à 1V :

On a:

$$N = \frac{V_{in(+)} - V_{in(-)}}{V_{ref}} \times 255$$

Si N=0 et  $V_{in(+)} = -1V$ :

Donc :

$$0 = \frac{-1 - V_{in(-)}}{V_{ref}} \times 255$$

$$V_{in(-)} = -1V$$

Si N=255,  $V_{in(-)} = -1V$  et  $V_{in(+)} = 1V$  :

Donc :

$$255 = \frac{1 - (-1)}{V_{\text{ref}}} \times 255$$

$$V_{\text{ref}} = 2 \text{ V}$$

Pour calculer la résistance on fixe l'une et on calcule l'autre :

On fixe  $R_2$

$$R_2 = 1 \text{ K ohm}$$

Par diviseur de tension on a :

$$\frac{12}{R_1 + R_2} \times R_2 = 2 \text{ V}$$

Alors :

$$\frac{12}{R_1 + 1} = 2$$

$$2R_1 + 2 = 12$$

Donc :

$$R_1 = 5 \text{ K ohm}$$

En fixe  $R_4$ :

$$R_4 = 1 \text{ K ohm}$$

Par diviseur de tension On a :

$$\frac{5}{R_3 + R_4} \times R_4 = 1 \text{ V}$$

Alors :

$$\frac{5}{R_3 + 1} = 1$$

$$R_3 + 1 = 5$$

Donc :

$$R_3 = 4 \text{ K ohm}$$

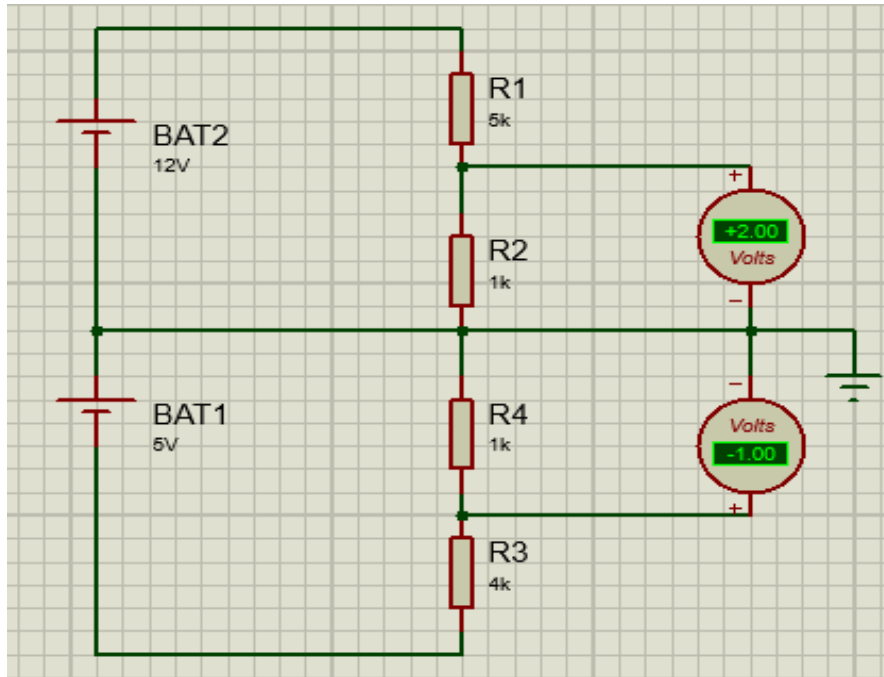


Figure III. 21: Schéma Diviseur de Tension qui donne deux tension de référence 2V et -1V

-2.5V à 2.5V :

On a:

$$N = \frac{V_{in(+)} - V_{in(-)}}{V_{ref}} \times 255$$

Si  $N=0$  et  $V_{in(+)}=-2.5V$ :

Donc:

$$0 = \frac{-2.5 - V_{in(-)}}{V_{ref}} \times 255$$

$$V_{in(-)} = -2.5V$$

Si  $N=255$ ,  $V_{in(-)}=-2.5V$  et  $V_{in(+)}=2.5V$  :

Donc :

$$255 = \frac{2.5 - (-2.5)}{V_{ref}} \times 255$$

$$V_{ref} = 5V$$

Pour calculer la résistance on fixe l'une et on calcule l'autre :

On fixe  $R_2$  :

$$R_2 = 1K\ ohm$$

Par diviseur de tension On a :

$$\frac{12}{R_1 + R_2} \times R_2 = 5V$$

Alors :

$$\frac{12}{R_1 + 1} = 5$$

$$5R_1 + 5 = 12$$

Donc :  **$R_1 = 1.4 \text{ K ohm}$**

En fixe  $R_4$ :

$$R_4 = 1 \text{ K ohm}$$

Par diviseur de tension On a :

$$\frac{5}{R_3 + R_4} \times R_4 = 2.5V$$

Alors :

$$\frac{5}{R_3 + 1} = 2.5$$

$$2.5R_3 + 2.5 = 5$$

Donc :

$$R_3 = 1 \text{ K ohm}$$

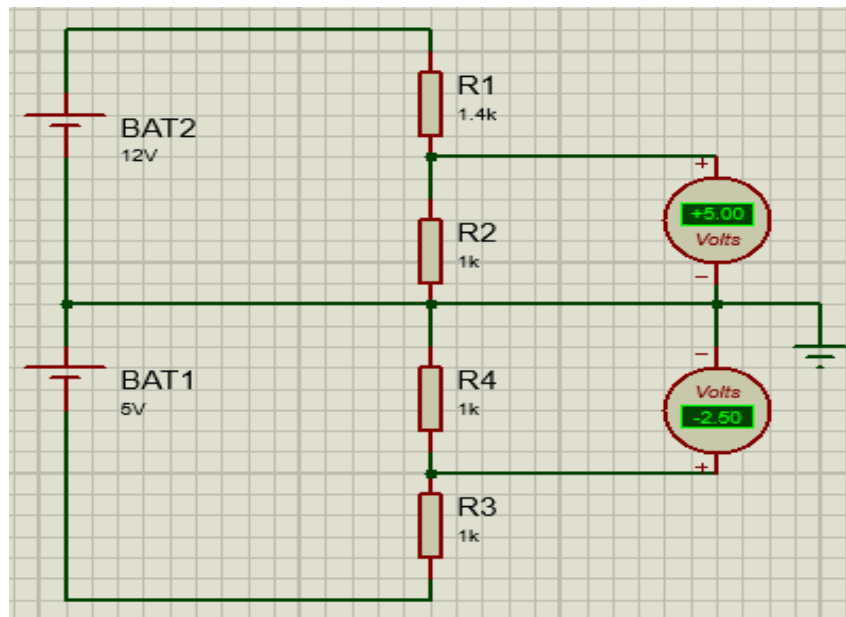


Figure III. 22: Schéma Diviseur de Tension qui donne deux tension de reference 5V et -2.5V

Résumez ce calculs dans un tableau :

$V_{ref}$ (Volts)	$R_1$ (KOhms)	$R_2$ (KOhms)	$R_3$ (KOhms)	$R_4$ (KOhms)
10	0.2	1	-	-
5	1.4	1	-	-
2	5	1	-	-
-1	-	-	4	1
-2.5	-	-	1	1

Tableau 4: Caractéristiques des Résistances pour Différentes Tension de Références  $V_{ref}$

Les tirets ("-") indiquent les valeurs non applicables.

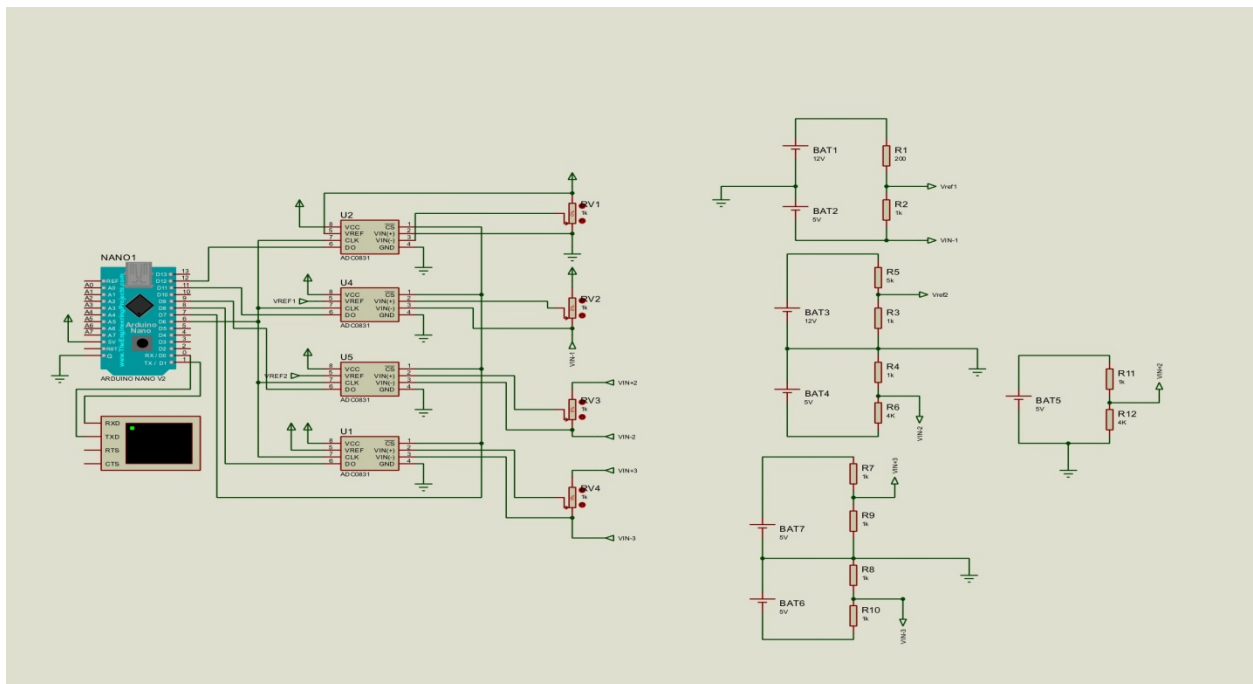


Figure III. 23: Schéma de circuit sous isisproteus

## 6. Les Résultats :

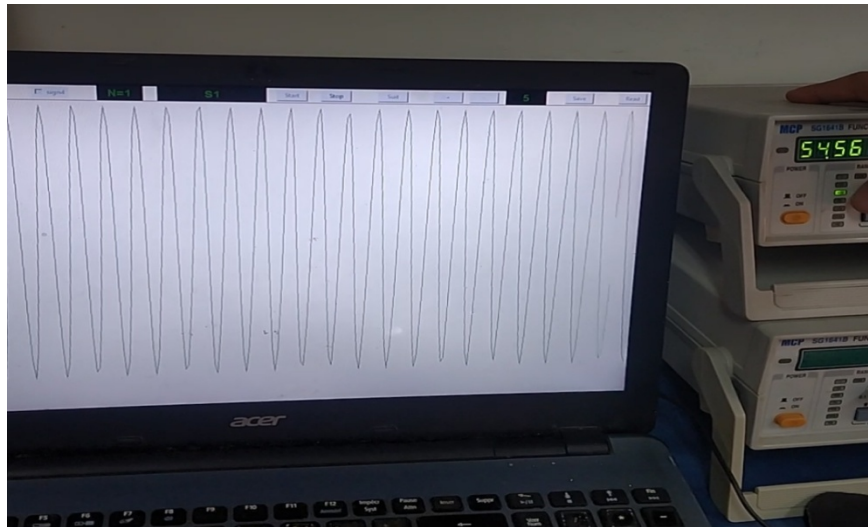


Figure III. 24: Affichage des Signaux sur Interface Graphique Tkinter à 5456 Hz

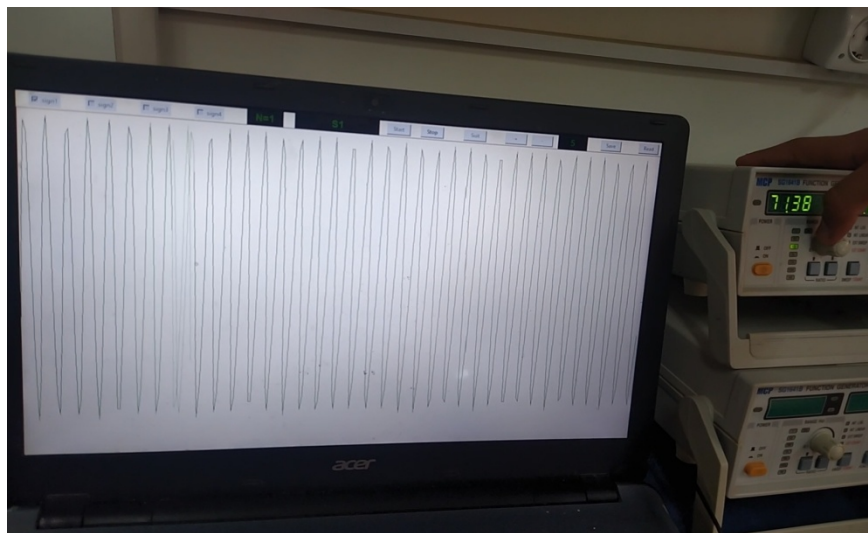


Figure III. 25: Affichage des Signaux sur Interface Graphique Tkinter à 7138 Hz

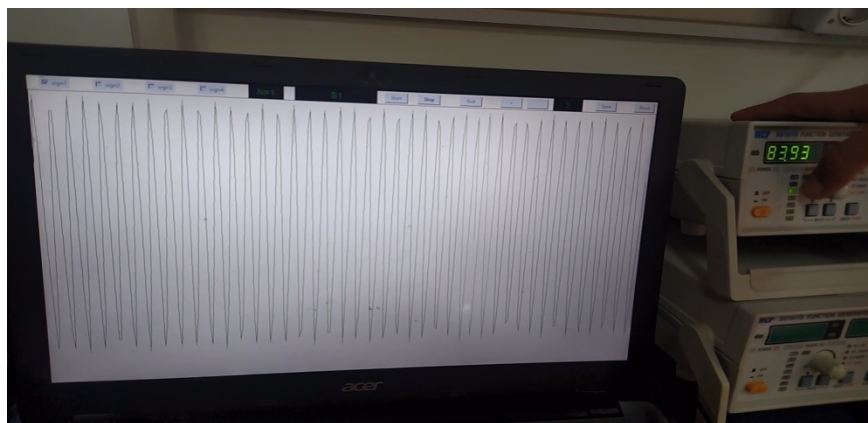


Figure III. 26: Affichage des Signaux sur Interface Graphique Tkinter à 8393 Hz

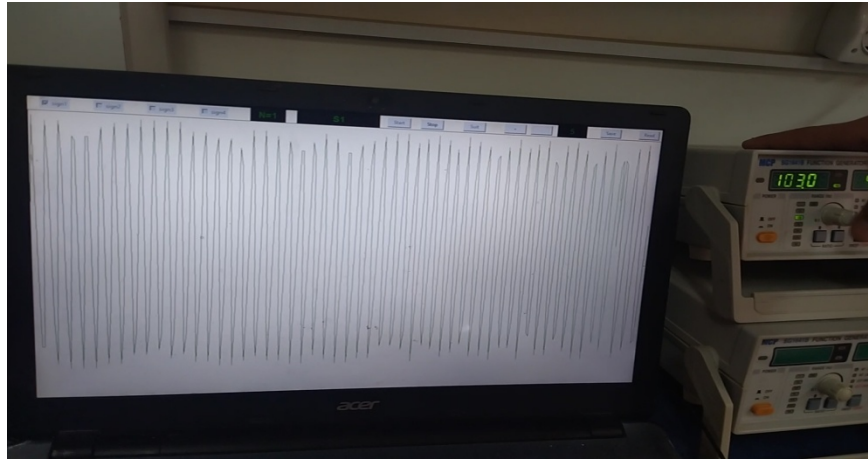


Figure III. 27: Affichage des Signaux sur Interface Tkinter à 103 KHz



Figure III. 28: Montage de Test et Mesure du Signal pour le Mini Oscilloscope BF Numérique



Figure III. 29: Affichage des Signaux Carré et Triangle sur Interface Tkinter à 6189 Hz et 6087 Hz

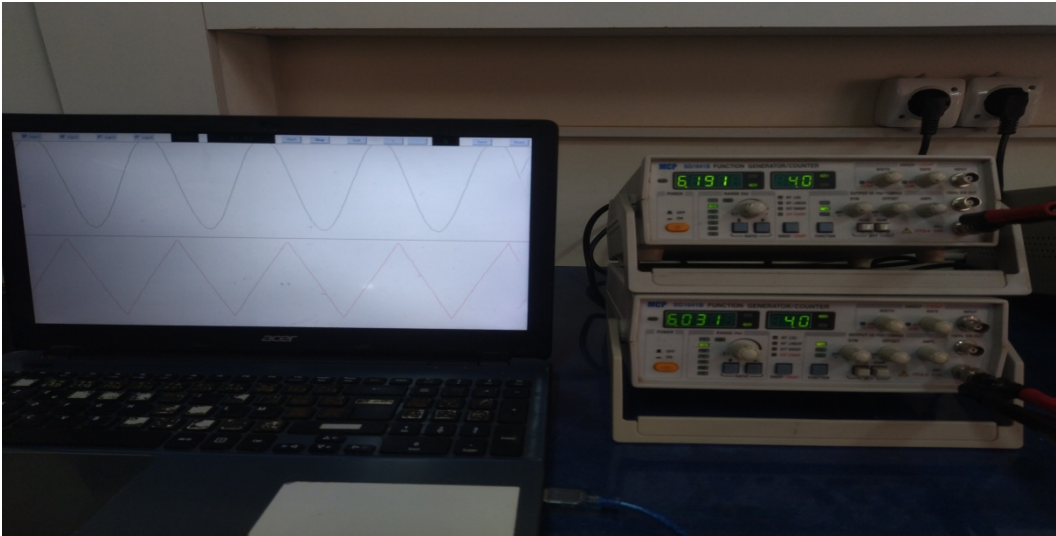
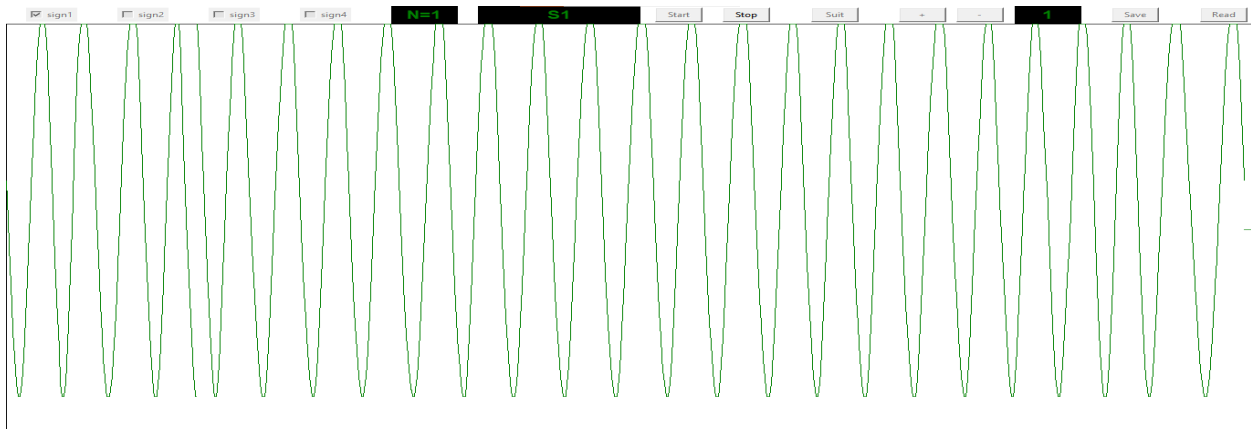


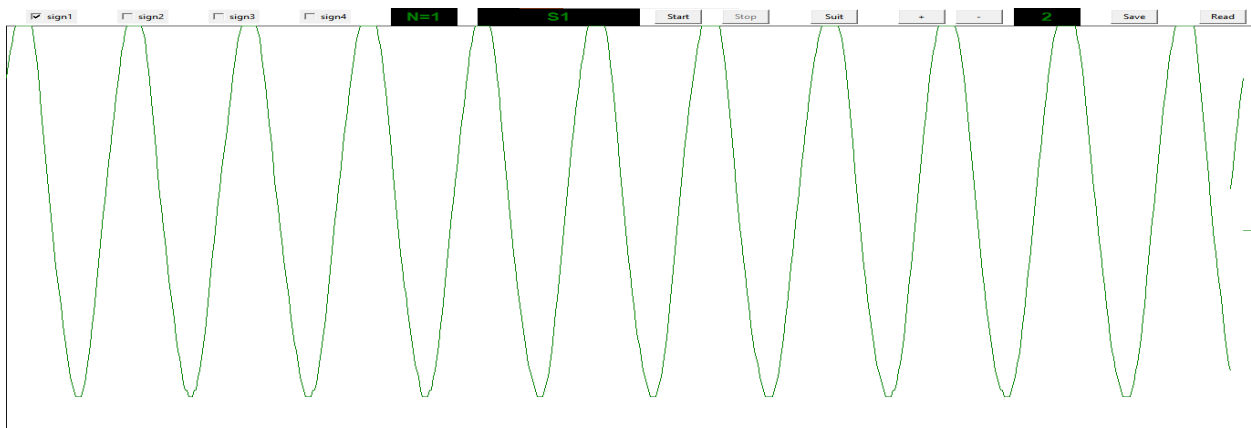
Figure III. 30: Visualisation de Signaux Sinusoïdaux et Triangulaires sur un Oscilloscope

a) Signal 1 Sinusoïdal, Pas= 1 à 5, Fréquence = 6Hz :

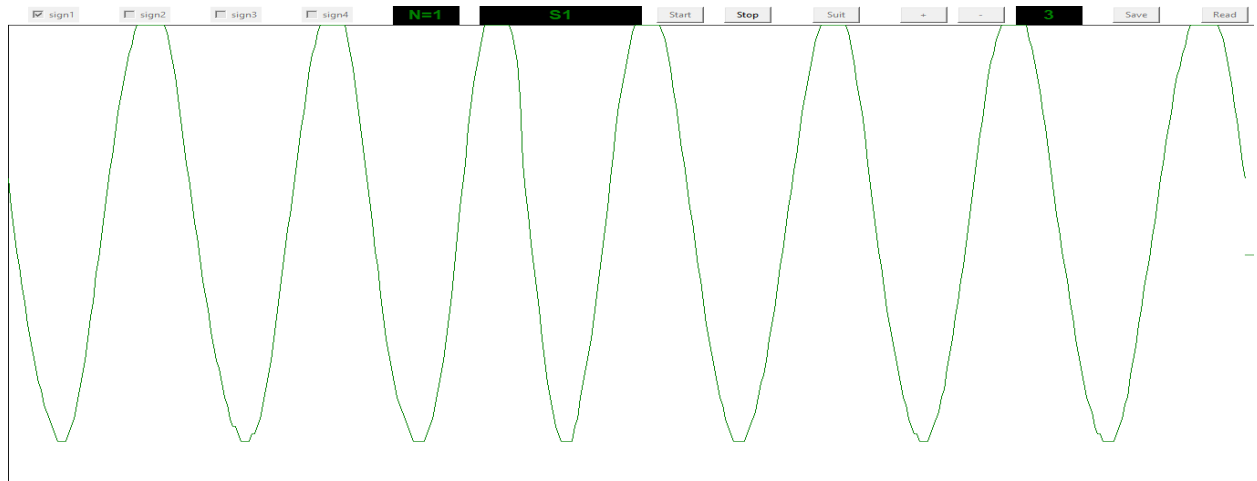
➤ Pas=1



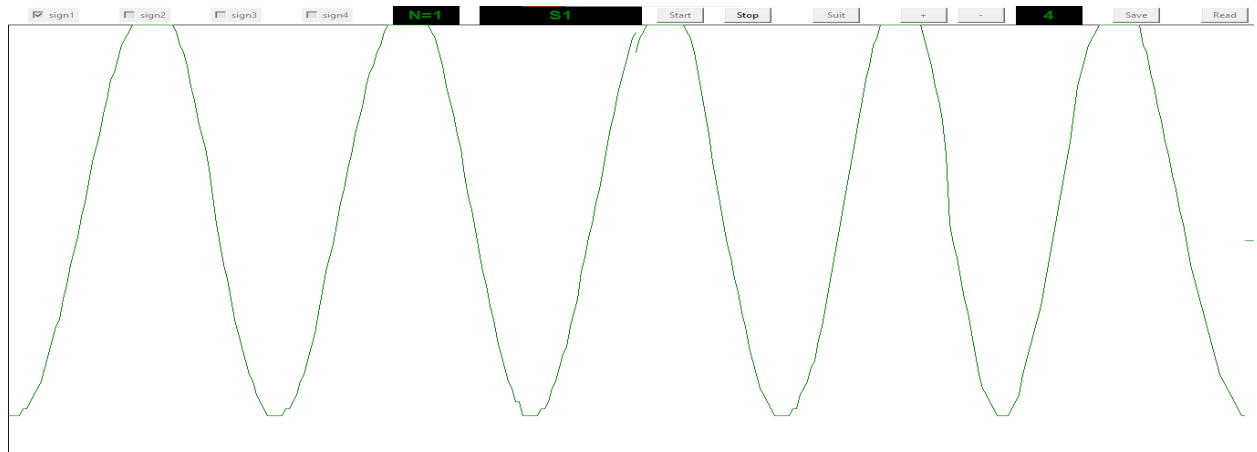
➤ Pas=2



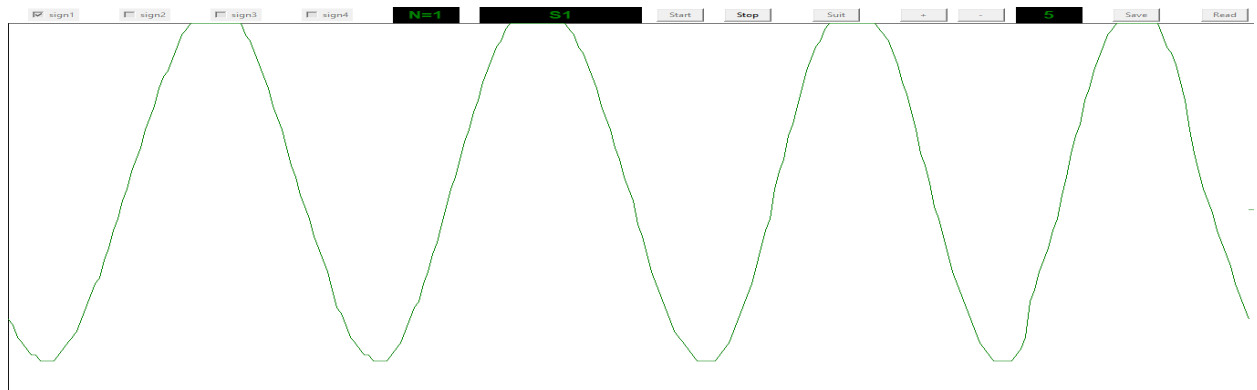
➤ **Pas=3**



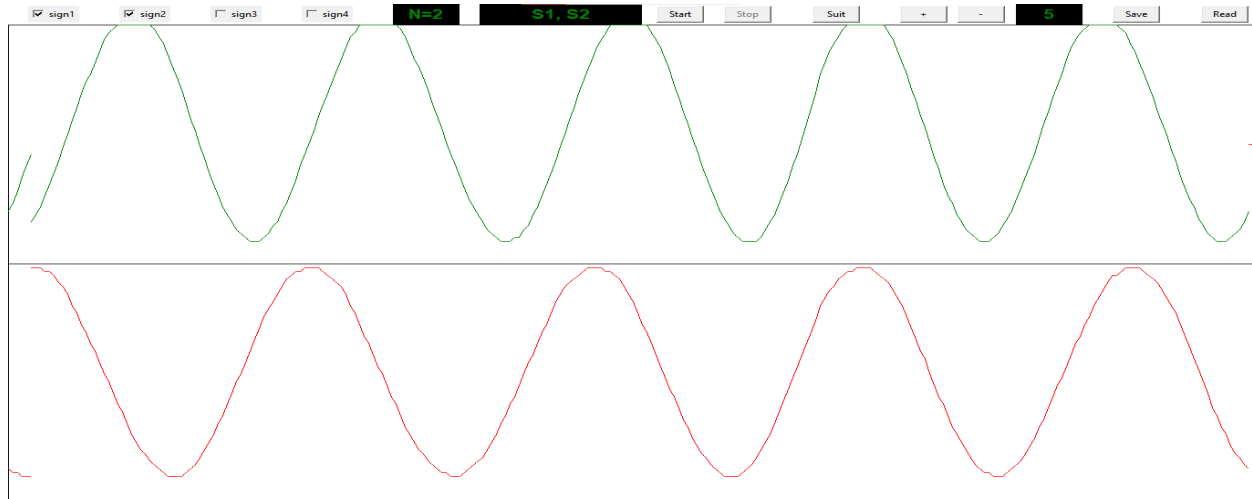
➤ **Pas=4**



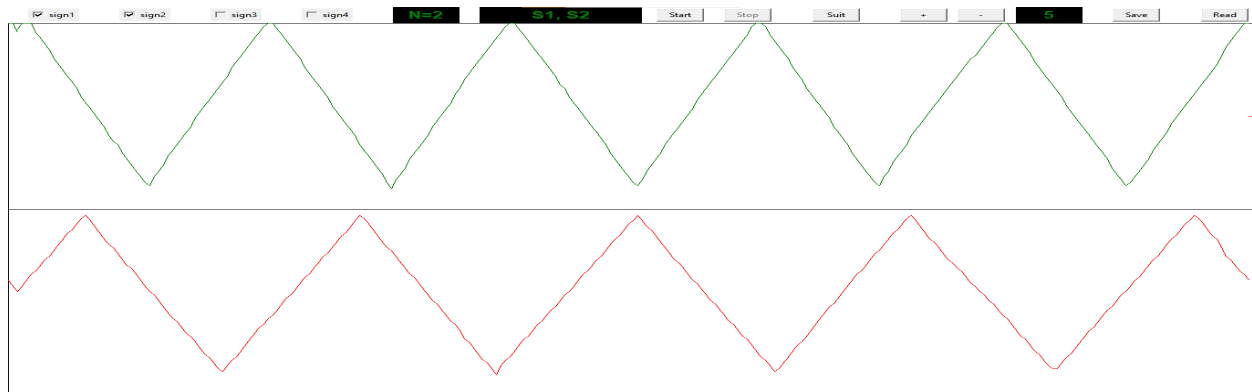
➤ **Pas=5**



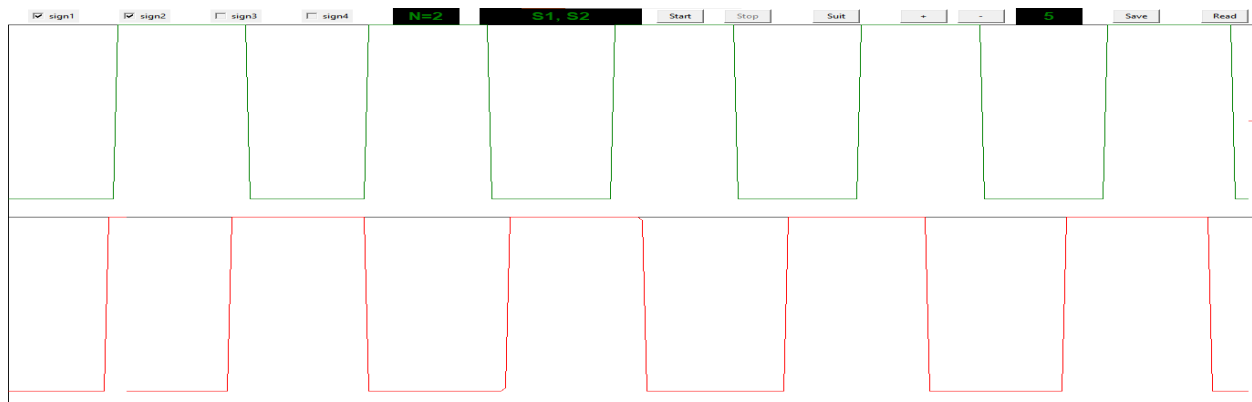
**b) Signal 1 et Signal 2 Sinusoïdal, Pas= 5, Fréquence = 6Hz :**



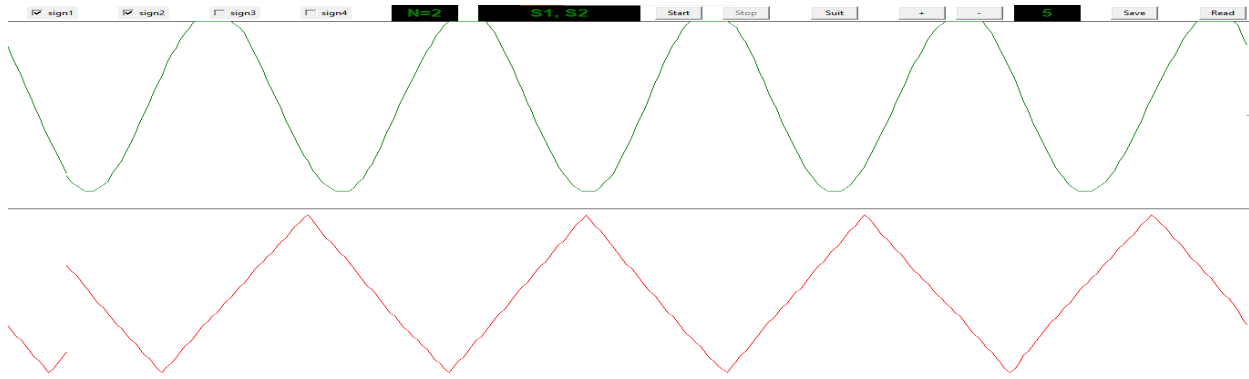
**c) Signal 1 et Signal 2 Triangulaire, Pas= 5, Fréquence = 6Hz :**



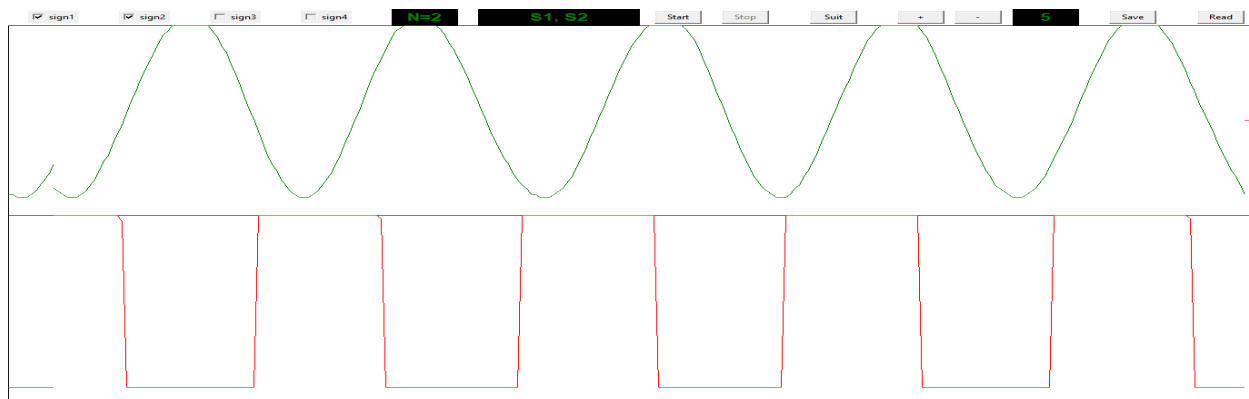
**d) Signal 1 et Signal 2 Rectangulaire, Pas= 5, Fréquence = 6Hz :**



**e) Signal 1 Sinusoïdale Signal 2 Triangulaire, Pas= 5, Fréquence = 6Hz :**



**f) Signal 1 Sinusoïdale Signal 2 Rectangulaire, Pas= 5 , Fréquence = 6Hz :**



**7. Conclusion :**

Dans ce chapitre, nous avons exploré les tests, les performances et les applications pratiques du mini oscilloscope numérique à 4 voies basé sur un Arduino Nano et un ADC0831. Nous avons examiné en détail les principes opérationnels du système, y compris ses principaux composants et son schéma de connexion.

Les aspects matériels et logiciel sont été détaillés, couvrant les diagrammes de communication et le fonctionnement de l'ADC0831 et des potentiomètres. La conversion analogique-numérique, la transmission des données vers l'ordinateur et l'affichage via l'interface Tkinter en Python ont également été expliqués.

De plus, le chapitre abordé la connectivité, l'alimentation électrique et l'étalonnage du potentiomètre, ainsi que le développement du code Arduino et du programme Python/Tkinter. Les résultats en laboratoire ont illustré les performances du système et ses applications pratiques. Des principes de fonctionnement aux diagrammes de circuits détaillés et aux explications d'interface, ce chapitre offre une compréhension complète de la fonctionnalité et de l'utilisation du mini oscilloscope numérique. En suivant les procédures et directives décrites, les utilisateurs peuvent mettre en œuvre et utiliser efficacement cet outil polyvalent pour divers projets et expériences électroniques.

## **CONCLUSION GENERALE**

## **Conclusion Générale :**

En conclusion, ce projet intitulé "ÉTUDE, CONCEPTION ET RÉALISATION PRATIQUE D'UN MINI OSCILLOSCOPE BF NUMÉRIQUE A MÉMOIRE 4 VOIES" a permis de développer et de mettre en œuvre un mini oscilloscope numérique multifonctionnel. Ce dispositif, capable de capturer et d'afficher des signaux analogiques sur quatre voies, repose sur les convertisseurs analogiques-numériques ADC0831 et une interface utilisateur interactive développée avec Tinter.

Le projet a suivi une approche méthodique et structurée, couvrant les aspects de la conception matérielle, le développement logiciel et les tests de performance. La première phase a impliqué l'étude approfondie des ADC et la configuration matérielle nécessaire pour garantir une conversion précise des signaux analogiques en données numériques. Ensuite, la phase de conception logicielle a intégré la création d'une interface utilisateur intuitive, permettant aux utilisateurs de sélectionner les signaux à afficher, d'ajuster les paramètres d'échantillonnage et de visualiser les données en temps réel.

Les tests réalisés ont confirmé l'efficacité et la fiabilité du mini oscilloscope, démontrant sa capacité à surveiller et enregistrer des signaux analogiques avec une haute précision. Ce projet se révèle particulièrement utile pour diverses applications pratiques, telles que la surveillance des signaux en temps réel, l'enseignement des concepts de conversion analogique-numérique, et le prototypage de systèmes embarqués.

De plus, ce travail ouvre des perspectives intéressantes pour des améliorations futures. Les extensions possibles incluent l'augmentation du nombre de canaux ADC, l'optimisation des algorithmes de traitement des signaux, et l'enrichissement des fonctionnalités de l'interface utilisateur pour offrir une meilleure expérience utilisateur et des capacités d'analyse avancées.

En somme, ce projet constitue une réalisation concrète et aboutie, offrant une base solide pour des développements futurs et des innovations dans le domaine de l'acquisition et du traitement des données analogiques, positionnant ainsi ce mini oscilloscope numérique comme un outil versatile et indispensable pour les professionnels et les chercheurs en électronique et en instrumentation.

## Bibliographie

- [1] M.Sabrina, and DamyaOumessaoud. *Conception et Réalisation d'un Oscilloscope Numérique Avec le microcontrôleur STM32*. Diss. Université Mouloud Mammeri, 2020.
- [2] Jutten, C. (2018). Cours théorie du signal. GIPSA-lab, Grenoble INP. Retrieved from [http://www.gipsalab.grenoble-inp.fr/~christian.jutten/mescours/Cours\\_Theorie\\_Signal\\_2018.pdf](http://www.gipsalab.grenoble-inp.fr/~christian.jutten/mescours/Cours_Theorie_Signal_2018.pdf)
- [3] Benkouider, S. (2021, January 18). Cours traitement du signal C01 classification de signaux [Video]. YouTube. [https://www.youtube.com/watch?v=5ppzxUuh\\_rQ](https://www.youtube.com/watch?v=5ppzxUuh_rQ)
- [4] Cottet, Francis. *Aide-mémoire-traitement du signal*. Dunod, 2017.
- [5] INSA Rouen Normandie. (s. d.). Cours 1. Récupéré de [https://moodle.insa-rouen.fr/pluginfile.php/6090/mod\\_resource/content/0/cours1.pdf](https://moodle.insa-rouen.fr/pluginfile.php/6090/mod_resource/content/0/cours1.pdf)
- [6] Rohde& Schwarz. (n.d.). Fondamentaux d'un oscilloscope II. Retrieved from [https://cdn.rohdeschwarz.com/fr/general\\_37/local\\_webpages/Fondamentaux\\_dun\\_oscilloscopeII.pdf](https://cdn.rohdeschwarz.com/fr/general_37/local_webpages/Fondamentaux_dun_oscilloscopeII.pdf)
- [7] Dutertre, D. (s. d.). Cours sur les convertisseurs. École des Mines de Saint-Étienne. [https://www.emse.fr/~dutertre/documents/cours\\_convertisseurs.pdf](https://www.emse.fr/~dutertre/documents/cours_convertisseurs.pdf)
- [8] Microchip. (s. d.). ADC0831: 8-bit serial I/O A/D converter with multiplexer options. Récupéré de <https://www.microchip.com/developmenttools/listing/aa513ec9-5066-47e2-ac2d-2e80a2afa8d1>
- [9] T. ABDENNACER & B. OUASSINI. "Conception et réalisation d'un système de contrôle d'accès et éclairage de la cage d'escalier d'un immeuble à base de technologie RFID." [1682238.pdf](#). Juin 2022.
- [10] [Lampiran.pdf](#)

## Résumé

Le mini oscilloscope BF numérique à mémoire 4 voies utilise un Arduino Nano et un ADC0831 pour l'acquisition des signaux. L'affichage est réalisé sur une interface Tkinter(Python). La liaison entre le PC et le circuit se fait via USB, offrant une connexion stable et une alimentation pratique. Compact et portable, il est idéal pour les applications sur le terrain et offre une solution abordable pour les besoins de mesure en basse fréquence.

**Mot clés:** Microcontrôleur, Python, CAN,Echantillonnage.

## Abstract

The mini 4-channel digital BF oscilloscope utilizes an Arduino Nano and ADC0831 for signal acquisition. Display is done on a Tkinter interface (Python). The connection between the PC and the circuit is via USB, providing a stable connection and convenient power supply. Compact and portable, it is ideal for field applications and offers an affordable solution for low-frequency measurement needs.

**Keywords:** Microcontroller, Python, ADC, Sampling

## ملخص

يستخدم الجهاز الصغير المحمول لقياس الموجات الصوتية رقمياً بذاكرة 4 قنوات أردوينو نانو و ADC0831 لاستقبال الإشارات. يتم عرض البيانات على واجهة tkinter(Python). يتم الاتصال بين الكمبيوتر والدارة عبر منفذ USB، مما يوفر اتصالاً مستقرًا وتيارًا كهربائيًا مريحًا. صغير ومحمول، فهو مثالي للاستخدام في الميدان ويوفر حلاً ميسور التكلفة لاحتياجات قياس الترددات الصوتية الباطنية.

**كلمات مفتاحية:** متحكم دقيق، بايثون، محول تناظري رقمي، أخذ عينات