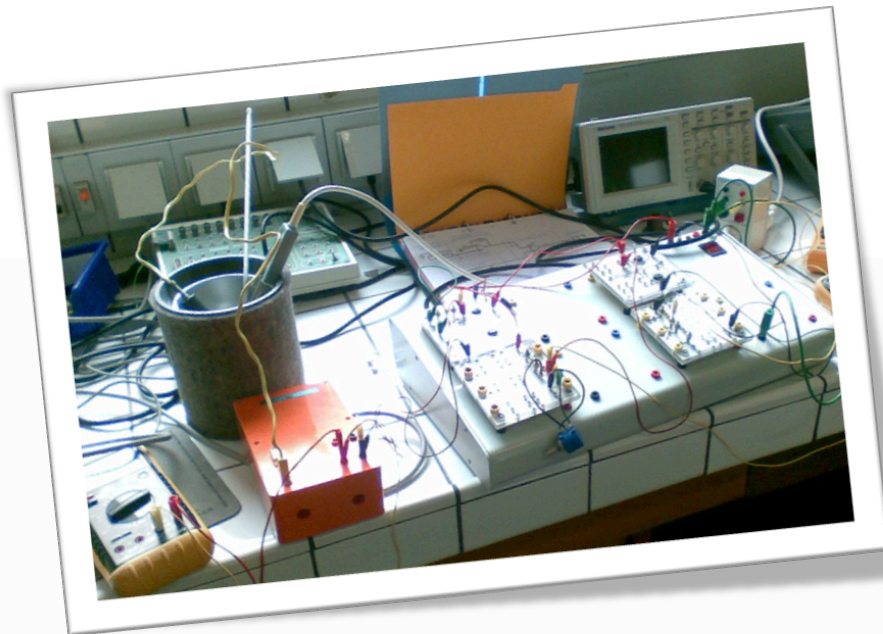


## MESURE DE LA TEMPERATURE AMBIANTE PAR UN ORDINATEUR



**Etudiants :**

**Marion PAYEN**

**Patrick CHEDID**

**Lionel TETSUEGNOUO**

**Luyi TANG**

**Gauthier MARQUAND**

**Enseignant-responsable du projet :**



**Date de remise du rapport** : lundi 22 juin 09

**Référence du projet** : STPI/P6-3/2009 – Projet N° 016

**Intitulé du projet :**

***MESURE DE LA TEMPERATURE AMBIANTE PAR ORDINATEUR***

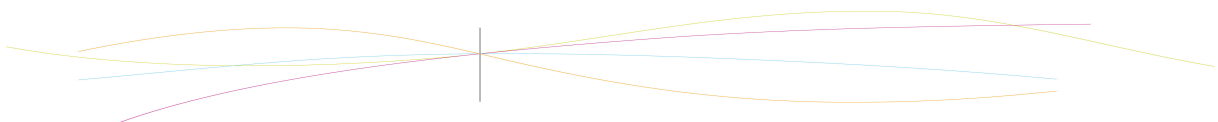
**Type de projet :** Expérimental

**Objectifs du projet :**

Le projet a pour objectif la réalisation d'un capteur de température ambiante à l'aide d'une conversion de l'analogique au numérique. Le travail doit être accompli grâce à un travail d'équipe, une bonne répartition des tâches, ainsi qu'à une bonne coordination.

## Table des matières

Intitulé du projet : .....	3
Type de projet : Expérimental .....	3
Table des matières .....	4
Notations, acronymes .....	5
Remerciements .....	6
1. Introduction .....	7
2. Méthodologie / Organisation du travail.....	8
3. Travail réalisé et résultats .....	10
3.1. Partie électrique/électronique : .....	10
3.1.1. Acquisition et mise en forme du signal.....	10
3.1.2. Conversion tension-fréquence .....	12
3.1.3. Intérêt du montage global .....	14
3.2. Partie informatique.....	14
3.2.1. Quelques notions sur le Delphi .....	14
3.2.2. La conception du logiciel.....	16
3.2.3. Le port parallèle .....	17
3.2.4. Le masque binaire.....	20
4. Conclusions et perspectives .....	22
4.1. Conclusions sur le travail réalisé .....	22
4.2. Conclusions sur l'apport personnel de cette U.V. projet.....	22
4.3. Perspectives pour la poursuite de ce projet.....	22
5. Bibliographie .....	23
Livres .....	23
Liens internet .....	23
6. Annexes (non obligatoire) .....	24
6.1. Documentation technique .....	<b>Erreur ! Signet non défini.</b>
6.2. Listings des programmes réalisés .....	24
6.3. Schémas de montages, plans de conception... ..	32
6.4. Propositions de sujets de projets (en lien ou pas avec le projet réalisé).....	<b>Erreur ! Signet non défini.</b>

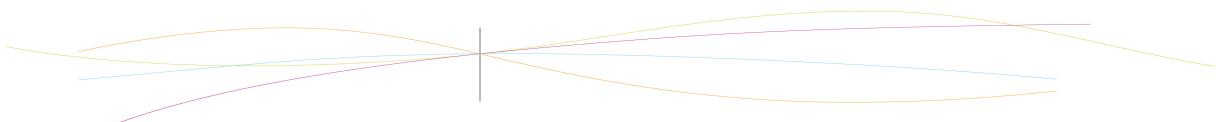




## Notations, acronymes

AO : Amplificateur Opérationnel

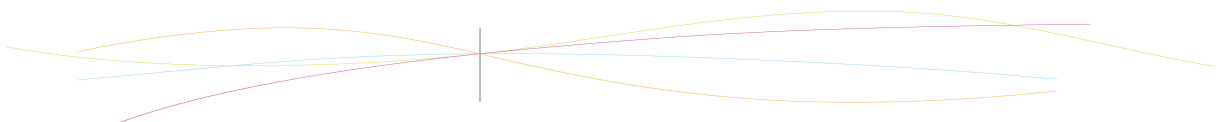
VCO : convertisseur tension fréquence



## Remerciements

Nous tenons à remercier l'ensemble du corps enseignant et du personnel de laboratoire qui nous ont permis de mener à bien notre projet.

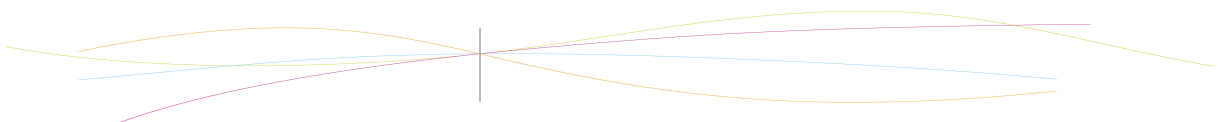
Nous voulons particulièrement remercier Mr Yon pour l'organisation et le bon déroulement de l'UV, et Mr Guillotin, professeur responsable de notre projet, pour ses explications, son soutien et son aide.



## 1. INTRODUCTION

Le but de notre projet était de réaliser un montage de captage de la température dans un certain milieu et de l'afficher sur un ordinateur.

Pour des raisons évidentes de répartition du travail, le projet a été réalisé par deux équipes travaillant en parallèle. Il se sera déroulé sur 10 semaines et s'est tenu dans les salles de TP de l'INSA, avec à disposition tout le matériel disponible en laboratoire.



## 2. METHODOLOGIE / ORGANISATION DU TRAVAIL

Dans un premier temps, le projet a consisté à analyser le sujet et à trouver toutes les composantes du problème. A cette fin, chacun a effectué des recherches sur les différents aspects qui apparaissaient au fur et à mesure de la compréhension du projet. La recherche a été menée non seulement sur internet, mais aussi grâce aux ressources de la bibliothèque et à des rapports de projets antérieurs similaires mis à disposition par Mr Guillotin. Il est très vite apparu que le projet se découperait en deux parties bien différentes. Effectivement, on distingue la partie d'acquisition et de mesure des données et d'un autre côté la partie concernant le traitement de ces données.

Pour commencer le projet, nous avons décidé de ne traiter que la température, pour pouvoir éventuellement généraliser notre travail et l'appliquer à d'autres grandeurs.

La méthode suivante a donc été adoptée par le groupe :

Un système électrique permettra d'acquérir les données et d'adapter le signal qu'il transmettra à l'ordinateur, par le biais du port parallèle (port de l'imprimante), qui est apparu comme le plus adapté au matériel et à nos connaissances en programmation.

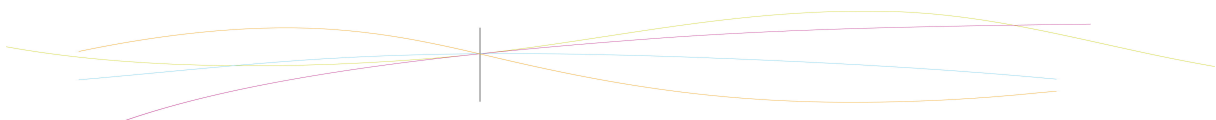
L'équipe s'est donc répartie en deux groupes afin de traiter indépendamment les deux parties principales avec les objectifs suivants :

- **Pole électrique/électronique**

1. Acquisition de données (thermocouple).
2. Mise en forme du signal mesuré afin d'assurer une certaine lisibilité des résultats.
3. Conversion tension/fréquence afin de pouvoir transmettre les données.

- **Pole informatique**

1. Programmation de l'interface avec le port parallèle (choix du langage de programmation adéquat).
2. Interprétation des données reçues.
3. Affichage par interface graphique



La répartition s'est faite de la manière suivante :

**Groupe 16 P6-3**



**Pôle électronique/électrique**

Marion PAYEN (calcul/acquisition)

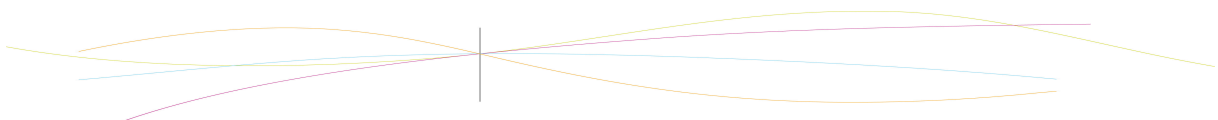
Luyi TANG (montage /acquisition)

Lionel TETSUEGNOUO(montage/acquisition)

**Pôle informatique**

Patrick CHEDID

Gauthier MARQUAND



### 3. TRAVAIL REALISE ET RESULTATS

#### 3.1. *Partie électrique/électronique :*

Cette partie a pour but d'expliquer l'existence des montages, la conception et la réalisation de ces montages. En conséquent, les données acquises à l'aide ces montages seront également expliquées, et surtout pour illustrer le fonctionnement des différents blocs.

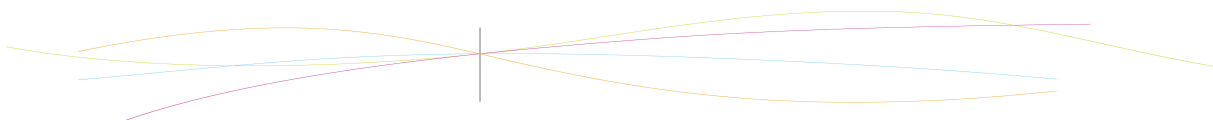
##### 3.1.1. *Acquisition et mise en forme du signal*

Le but du projet étant d'effectuer une mesure de température, il est évident que le montage électrique comporte une diode (diode faibles signaux 1n4148), représenté dans la *figure 1* par le bloc de l'AO A1. Ce bloc est l'interface avec le thermocouple et permet d'établir un lien entre la température et la tension en sortie. Nous avons effectué nos tests sur des températures variant de 0°C à 70°C (limitées par les capacités du bain thermostaté). La variation de tension des mesures était très faible et nous avons donc mis en place un autre bloc, permettant la mise en forme du signal.

Le but étant d'avoir une variation de tension d'environ 10V pour une variation théorique de 120°C, nous avons cherché les valeurs des résistances que nous devons prendre pour effectuer le montage du bloc A2 de la *figure 1*. Aucune valeur de résistance n'étant connue, nous avons premièrement tout fixé à 10kΩ, pour ne pas changer le signal. Connaissant les conditions limites voulues, nous avons établi deux équations avec les quatre résistances, à l'aide de ponts diviseurs de tension.

Pour pouvoir résoudre ces équations, nous avons fixé les valeurs de R1 et de R3 à 10kΩ. La résolution s'est faite grâce au logiciel *Maple 12*. Le détail de la résolution peut être vue en *Annexe 1*.

Le montage de bloc de gestion d'acquisition et de mise en forme du signal a pu être réalisé dans son intégralité. La tension repérée par  $U_s$  dans la *figure 1* est la tension directement acquise grâce au thermocouple, celle repérée par  $U_{\text{capt}}$  est la tension mise en forme qui sera mise en entrée du VCO, modélisée par la courbe dans la *figure 2*. Comme on peut le constater, le montage sort une tension comprise entre 1V et 9V.



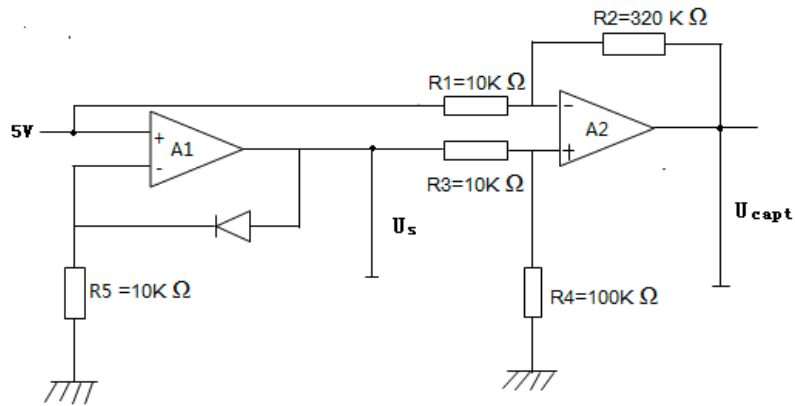


Figure 1 :  
Montage pour l'acquisition et la mise en forme de la température

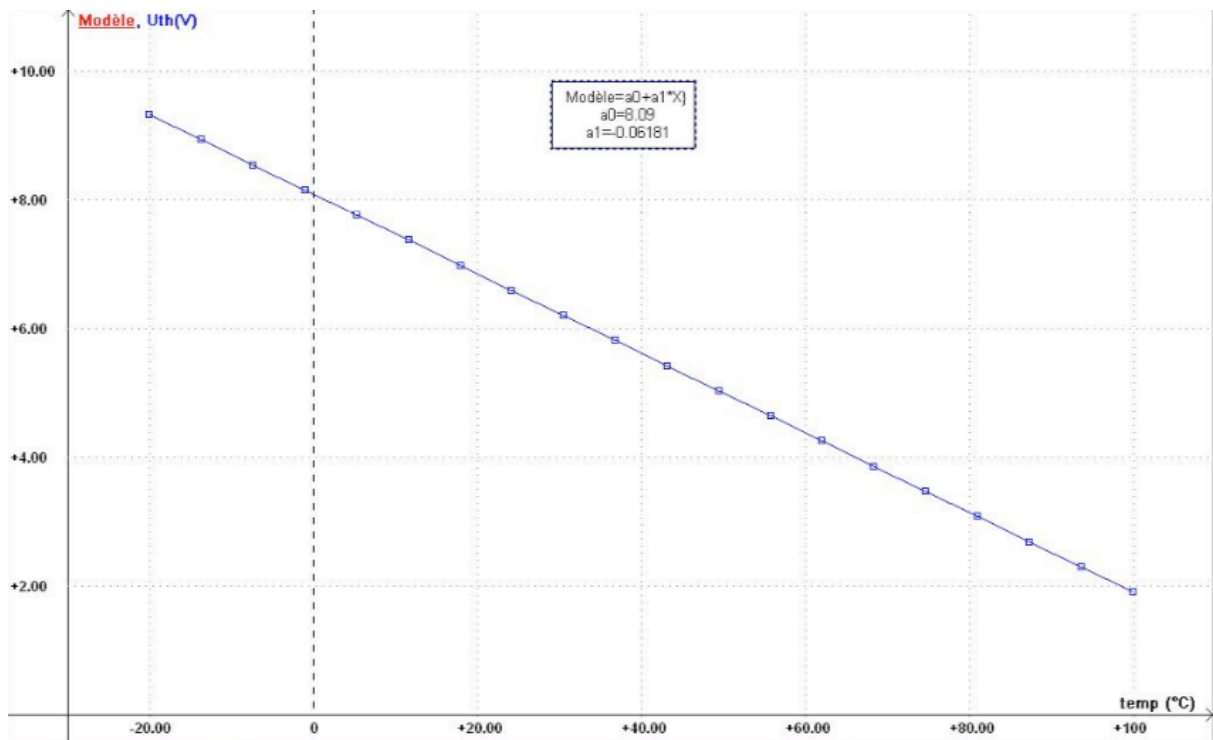
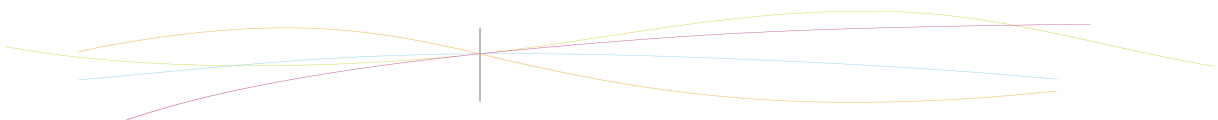


Figure 2 :  
Modélisation de la courbe obtenue expérimentalement



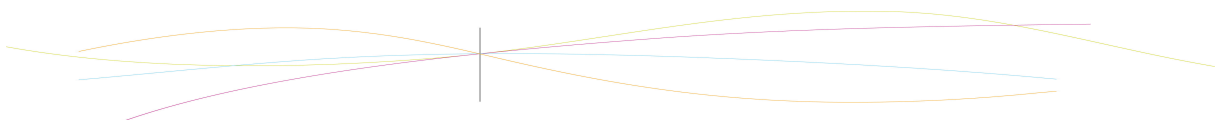
### 3.1.2. Conversion tension-fréquence

Le convertisseur de tension-fréquence est appelé un VCO et est un montage courant dans l'industrie. La tension appelée  $U_x$  dans la *figure 3*, est la tension obtenue après mise en forme du signal acquis.

Le convertisseur que nous avons monté se caractérise grâce à la capacité du condensateur utilisé, ici de 15nF. La particularité du convertisseur est qu'il présente un transistor qui fonctionne comme un interrupteur. Pour notre montage, nous avons utilisé un transistor BC337, qui est un transistor NPN, et il renvoie « tout ou rien ».

Pour expliquer l'évolution du signal dans le montage de la *figure 3*, on peut observer ce signal à différents endroits. La tension qui sortait du montage précédent entre en  $U_x$ . A la sortie de l'AO A1, la tension  $U_1$  retourne soit  $U_x$ , soit 0 (grâce au transistor), ce qui donne un signal carré à l'entrée de l'AO A2, monté un inverseur, qui lui ressort un signal triangulaire  $U_i$ . Cette tension entre dans l'AO A3, qui est un comparateur non-inverseur, qui ressort  $U_2$ , qui entre dans l'AO A4, qui est inverseur.  $U_c$  vaut donc  $-U_2$ .

On récupère la fréquence en  $U_c$  et on obtient la courbe modélisée de la *figure 4*. Nous pouvons observer qu'il y a une relation linéaire entre la fréquence récupérée et la tension en entrée.





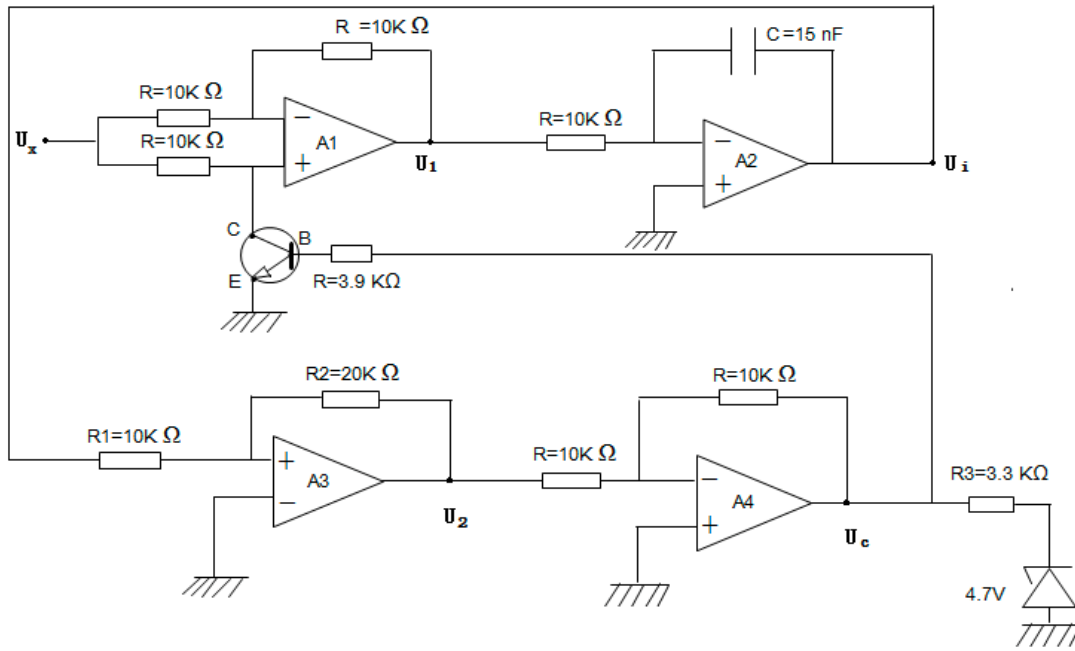


Figure 3 :  
Montage du VCO 15n : convertisseur tension-fréquence

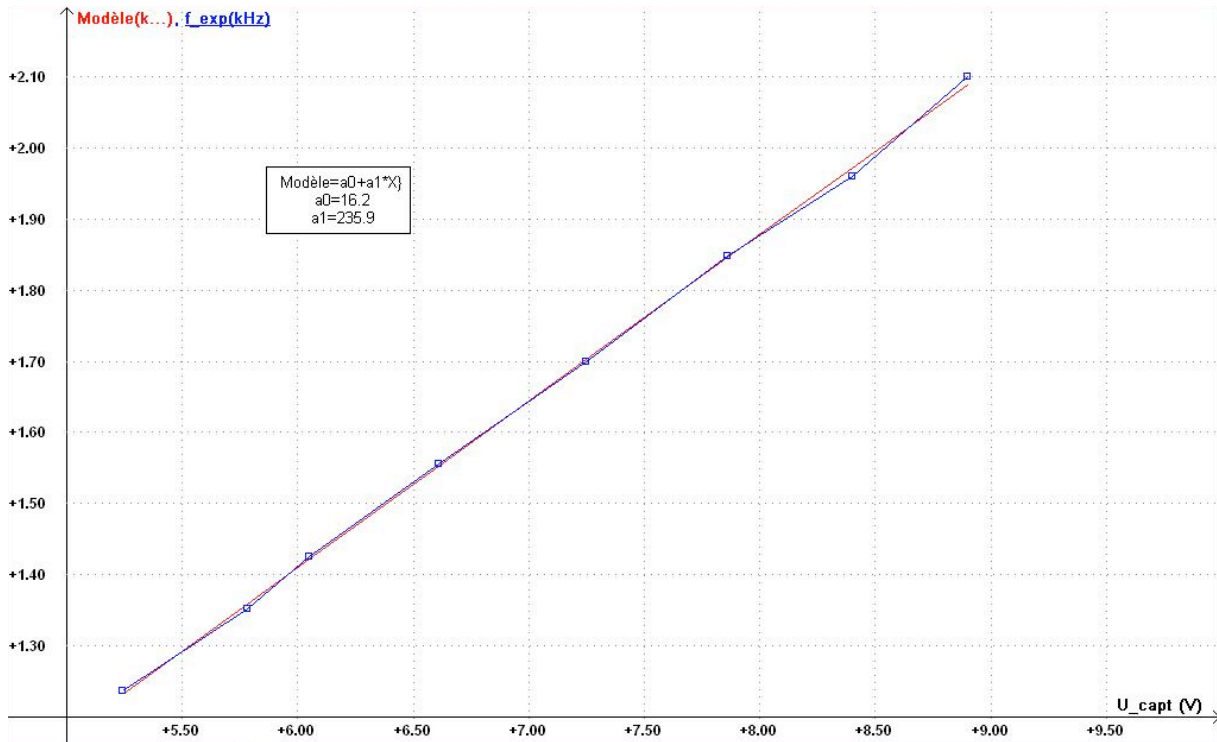
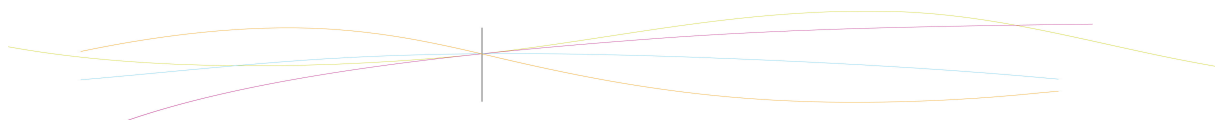


Figure 4 :  
Modélisation de la courbe obtenue expérimentalement



### 3.1.3. Intérêt du montage global

L'intérêt du montage global est que grâce aux équations des courbes obtenues, nous pouvons établir une relation directe entre la fréquence en sortie et la température captée grâce au thermocouple. C'est grâce à cette relation que les parties électronique et informatique pourront se rejoindre.

## 3.2. Partie informatique

### 3.2.1. Quelques notions sur le Delphi

#### Introduction

Delphi désigne à la fois un environnement de développement intégré et un langage de programmation orienté objet.

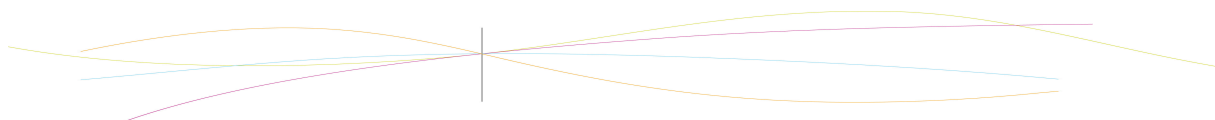
L'EDI Delphi est un EDI propriétaire fonctionnant sous Windows créé en 1995 et édité par Borland. À l'époque, créer des programmes graphiques sous Windows se faisait en grande majorité en utilisant soit la chaîne de compilation Visual C++, soit le RAD Visual Basic. Le premier outil étant excessivement complexe et le second assez peu structuré, Delphi apparut alors comme une alternative viable pour beaucoup de développeurs qui souhaitaient créer des programmes standards pour Windows.

En 2001, Borland édita une version Linux de Delphi appelé Kylix, qui ne connut pas le succès attendu par l'éditeur. Comme il s'agit d'un outil RAD (Rapid Application Development), Delphi est très simple à prendre en main et il est tout à fait aisé de créer toutes sortes d'applications.

Actuellement, Delphi reste l'un des rares outils avec lequel il est possible de produire des programmes Win32 ou .NET à partir du même langage et de la même bibliothèque de composants.

#### Le pascal objet

Delphi implémente une version orientée objet (POO) du langage Pascal, le Pascal Objet (Object Pascal), renommé Langage de programmation Delphi au fil des modifications apportées par Borland. Le Pascal Objet de Delphi possède plusieurs avantages qui améliorent la productivité du développeur par rapport au C++ : typage fort, déclarations de variables dans une section précise, contrôle strict du compilateur pour éviter les erreurs de mémoire, de



débordement, gestion intégrée des chaînes de caractères et des tableaux dynamiques, etc. La compilation ne se fait qu'en une seule passe et il n'y a pas de séparation entre l'implémentation et l'interface comme en C ou en C++ : la génération d'un projet Delphi est donc très rapide, ce qui a accru à sa sortie la popularité de l'outil vite réputé pour ses temps de compilation record.

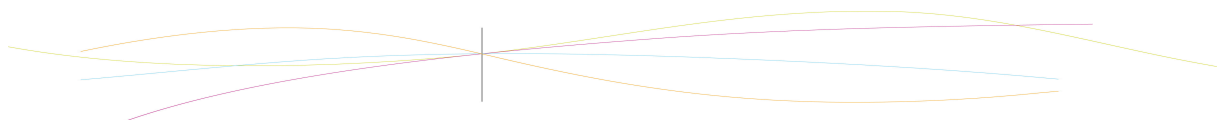
Bien que Pascal Objet introduise une multitude de concepts intéressants et novateurs (de nombreux ont été repris avec C#), il souffre de faiblesses connues. Il ne permet pas certaines fonctionnalités de POO tel que l'héritage multiple de classes (qui peut toutefois être contournée par l'utilisation des interfaces), la surcharge d'opérateurs n'a été introduite que tard (Delphi 2005), et il est impossible de reproduire le très utile système de "template" que l'on retrouve en C++. De plus, sa parenté avec le Pascal rebute de nombreux programmeurs plus habitués à des styles de programmation proches de Java ou C. Enfin, il n'est ni standardisé ni géré par un comité indépendant : propriété de Borland, l'éditeur est le seul à pouvoir décider de l'avenir et de l'ajout de nouvelles fonctionnalités du langage.

### **L'interface de développement:**

L'environnement de développement s'appuie sur un éditeur d'interface graphique associé à un éditeur de code source. Il doit son succès à sa facilité d'utilisation pour développer des applications graphiques et/ou liées aux bases de données. On l'a souvent comparé à Visual Basic de Microsoft pour cette facilité de développement. On peut même dire que par un amusant mouvement de balancier et de personne, le VB influença Delphi qui à son tour influença par la suite le VB (Anders Hejlsberg lors de son passage chez Microsoft).

Delphi (le programme) auto-génère du code pour faciliter le travail du programmeur. Il maintient une correspondance automatique entre la vue de conception (la fenêtre que le programmeur bâtit en déposant des composants graphiques) et l'éditeur de code (la vue affichant le code source qui créera ces composants à l'exécution). Les données spécifiques aux composants sont stockées dans des fichiers d'extension .DFM alors que le code source Pascal Objet est sauvegardé dans des fichiers d'extension .PAS.

L'interface de développement permet l'ajout de composants tiers (graphiques ou non) via un système de composants. La modularité est obtenue à la conception mais peut aussi être exploitée à l'exécution via un système de chargement dynamique de paquets d'exécution, Borland ayant étendu le concept de bibliothèques partagées et le format Windows DLL en introduisant un modèle propriétaire permettant d'enregistrer dynamiquement et d'exporter des classes entre modules. Le même système sera repris par Microsoft sous Visual Basic avec le format VBX, puis ensuite à l'échelle du système avec les composants COM et ActiveX.



### 3.2.2. La conception du logiciel

#### Les débuts

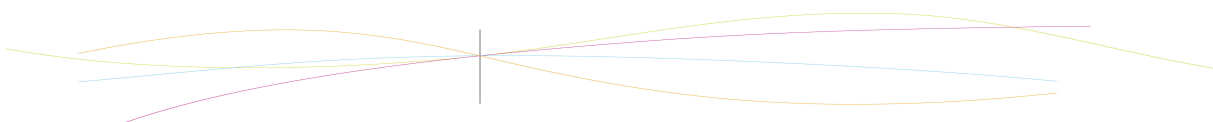
Avant de commencer le codage à proprement dit, nous avons évalué ce qui était attendu par rapport à nos connaissances. Nous ne nous sommes donc pas lancés directement dans la programmation du logiciel. Pour ce mettre dans le bain, on a fait ce que tout programmeur débutant aurait fait : commencer par concevoir un logiciel de type : « Hello world ». Ce programme a pour but d'afficher le message « Hello world » après un click sur un bouton. Notre deuxième logiciel était clairement plus sophistiqué :

- des boutons qui saisissent un texte (appelé chaîne de caractères) à partir d'un champ de texte (TextField) et le traduisent en réel pour en effectuer toutes sortes de calcul (addition, multiplication, ...).
- Des boutons qui changent la couleur du fond, du texte, des autres boutons...
- Des boutons qui ouvrent d'autres fenêtres.

#### La programmation

Une fois les bases acquises et l'aspect ludique terminé, on s'est lancé dans le vrai travail. Après deux trois heures de méditations et de contemplations inutiles, on s'est finalement rendu compte que concevoir ce logiciel requière beaucoup plus que nos simples et modestes bases en programmation : les notions d'assembleurs et la manipulation du port parallèle (entrées et sorties) nous paraissent totalement abstraites et inconcevables. Pour remédier à notre problème, il était indispensable de récupérer les fonctions et procédures de logiciels déjà conçus par des concepteurs ayant le même but que le notre : manipuler et contrôler les sorties et les entrées du port parallèle. Le plus efficace et le plus adapté fut le logiciel PORTIO®. Le travail le plus dur a été de « relier » le tout et le présenter sous une interface bien appropriée.

Tout au long de la programmation, un multimètre nous indiquait si tout ce passait comme on le souhaitait : on pouvait « écrire » un signal (c'est-à-dire envoyer un signal en sortie) - un signal de type BIT en décimal – et vérifier les différences de potentiels sur l'écran du voltmètre. Par exemple : l'« écriture » du signal représenté par 45 (en base 10) - ce qui représente 101101 en base 2 ( $45_{\text{base}10} = 2^5 + 2^3 + 2^2 + 2^0 = 101101_{\text{base}2}$ ) – conduira à une différence de potentiel de 3,64 sur les bornes D<sub>0</sub>, D<sub>2</sub>, D<sub>3</sub> et D<sub>5</sub>.



Quant à la lecture des signaux, ce fut un peu plus laborieux : le signal lu est converti en type WORD (ce qui occupe 4 BIT), et on étudie les fluctuations du premier BIT : en effet, une lecture du port ACK implique un « 3 » comme premier BIT du signal lu de type WORD, celle du port Select impliquant un « 6 », et quand deux signaux sont envoyés simultanément par ces deux entrées, le premier BIT sera « 2 ».

Une phase de vérification semblait primordiale : on a branché deux générateurs de fréquence, le premier de fréquence 1 Hz (signal carré) à l'entrée ACK, et le deuxième de fréquence 100 Hz à l'entrée Select. Le premier sert comme échelon de tension : une moitié de période (soit 0,5 seconde) en 1 (ou en créneau haut), et le reste de la période en 0 (ou créneau bas). Le deuxième représentera le signal reçu après la conversion tension-fréquence (fruit de la partie électronique du projet). Les résultats furent plus que satisfaisant : le logiciel marqua 48, donc 48 « pulses » pendant 1 seconde. On multiplie donc par 2, et on obtient la fréquence du signal qui traverse le port Select.

On a ainsi réussi à créer un « lecteur » de fréquence sur ordinateur (par l'intermédiaire du port parallèle).

### 3.2.3. Le port parallèle

#### Définition

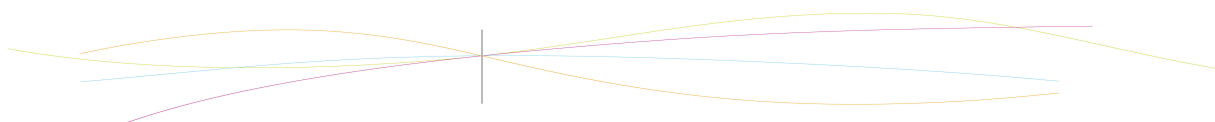
Le Port parallèle associé à l'interface parallèle Centronics (à l'origine : LPT pour Line Printing Terminal) est un connecteur situé à l'arrière des ordinateurs compatibles PC reposant sur la communication parallèle.

Elle a été conçue pour une imprimante texte employant un jeu de caractères de 8 bits : l'ASCII. Le nom vient des limitations de l'imprimante qui ne peut imprimer que du texte, une ligne après l'autre. Toutefois, les imprimantes graphiques (pouvant imprimer des images) se sont adaptées à ce système pour profiter des interfaces normalisées des ordinateurs de l'époque.



*Port parallèle*

Sa très grande robustesse mécanique a conduit à le faire utiliser aussi pour une multitude de branchements : scanner, lecteur de disques amovibles, lecteur de type Iomega Zip, caméra de vidéoconférence, lecteur de sauvegarde sur bandes, graveur de CD-Rom ou disques en rack.

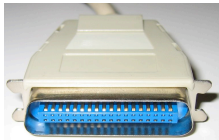


Le connecteur Centronics ne doit pas être confondu avec le connecteur DB25 sur lequel on branche le câble de l'imprimante côté PC, qui lui est bien plus fragile.

En fait il existe des câbles DB25 mâle vers Centronics pour les imprimantes parallèles, des câbles DB25 femelle ou DB9 femelle vers Centronics et, des câbles DB25 femelle ou DB9 femelle vers DB25 mâle pour les imprimantes série.

Ce type d'interface est aujourd'hui en cours de remplacement progressif par des interfaces de type USB et FireWire (IEEE 1394) qui permettent de plus grands débits, sont plus maniables, moins coûteuses et, surtout bien moins encombrantes.

Ce ne sont pas seulement les connecteurs qui changent, mais aussi la logique de flux sous-jacent, qui de parallèle devient série et multiplexée, ce qui demande davantage d'électronique, mais se traduit par une nette baisse de la quantité de matières premières nécessaires à la fabrication.



Centronic Male



Centronic Femelle

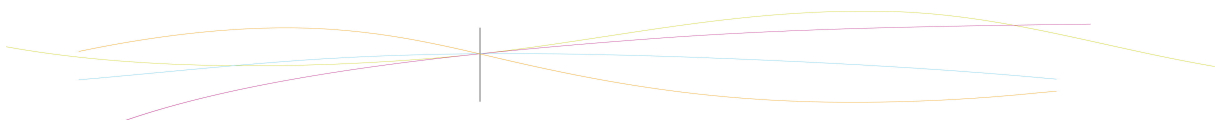


DB25 Femelle

## Utilisation

Tout d'abords, observons la connectique DB25 et centronics :

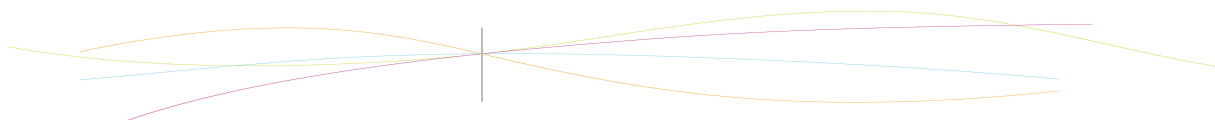
Broche		Nom	Type	Description
DB25	Centronics			
1	1	STROBE	S	Un 0 logique (impulsion) doit être envoyé sur cette ligne pour signaler à l'imprimante que des données valides sont présentes sur les lignes D0 à D7. Le signal doit normalement durer au moins 0,5 µs.
2	2	D0	S/(E)	C'est le bus de données sur lequel véhicule la valeur à imprimer. On ne peut écrire sur ce port, à moins d'avoir un port parallèle étendu (ECP/EPP).
3	3	D1	S/(E)	
4	4	D2	S/(E)	
5	5	D3	S/(E)	
6	6	D4	S/(E)	



7	7	D5	S/(E)	
8	8	D6	S/(E)	
9	9	D7	S/(E)	
10	10	ACK	E	L'imprimante envoie une impulsion negative sur cette ligne pour indiquer qu'elle a bien reçu le caractère transmis et que la transmission peut continuer.
11	11	BUSY	E	Cette ligne est mise à 0 par l'imprimante lorsque son buffer de réception est plein. L'ordinateur est ainsi averti que celle-ci ne peut plus recevoir de données. Il doit attendre que cette ligne revienne à 1 pour recommencer à émettre.
12	12	PE	E	Signifie "Paper End". L'imprimante indique par cette ligne que l'alimentation en papier a été interrompue.
13	13	SELECT	E	Cette ligne indique à l'ordinateur est "on line" ou "off line".
14	14	AUTOFEED	S	Lorsque ce signal est à 0, l'imprimante effectue un saut de ligne à chaque caractère CR (Carrier Return = #13) reçu. Dans le cas contraire, le saut de ligne doit être envoyé explicitement par l'ordinateur via le caractère LF (Line Feed = #10).
15	32	ERROR	E	Cette ligne indique à l'ordinateur que l'imprimante a rencontré une erreur.
16	31	INIT	S	Un 0 logique (impulsion) peut être envoyé sur cette ligne pour réinitialiser l'imprimante.
17	36	SELECT IN	S	L'ordinateur peut mettre l'imprimante hors ligne en plaçant un 0 sur cette ligne. Il peut aussi servir à sélectionner un deuxième périphérique si le port parallèle est partagé entre deux périphérique (cas d'un scanner par exemple).
18-25	16, 17, 19-30, 33	GND	-	C'est la masse du PC.
-	35	-	-	Mis au +5V par l'imprimante, par l'intermédiaire d'une résistance de 4,7 kohms.

E : entrée

S : sortie





Comme vous avez pu le constater, l'utilisation du port parallèle a comme première fonction de communiquer avec un élément extérieure. C'est une interface. On peut aussi remarquer qu'il possède 17 broches utilisables permettant d'envoyer ou de recevoir un signal. Les 18 restantes sont reliées à la masse. Si l'on regarde l'exemple ci-dessus, on peut s'apercevoir que chaque broche transmet un message par le biais de signaux lors d'un changement d'état (0 ou 1). On peut interpréter ces messages de la manière que l'on veut, par exemple, pour une imprimante, la ligne BUSY est mise à 0 par celle-ci lorsque son buffer de réception est plein. L'ordinateur est ainsi averti que celle-ci ne peut plus recevoir de données. Il doit attendre que cette ligne revienne à 1 pour recommencer à émettre. Pour notre projet, nous allons donner une toute autre interprétation de ces messages. Nous allons communiquer avec l'ordinateur afin de mesurer la fréquence d'un signal à l'aide du port parallèle. Nous possédons pour cela une boîte électronique qui nous permet d'envoyer un signal sur une piste précise, et un programme permettant de lire le signal reçu par l'ordinateur.

### 3.2.4. *Le masque binaire*

#### **Définition :**

Le masque binaire est une technique informatique utilisée très souvent en programmation. C'est une fonction informatique qui permet de lire un bit particulier très rapidement sur une variable Byte. Exemple d'utilisation du masque binaire : on veut connaître le 3<sup>ème</sup> bit de la variable 0100110.

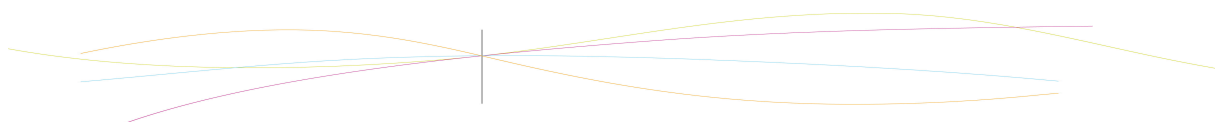
#### **Comment cela fonctionne-t-il?**

Cette fonction est très simple, elle repose sur la comparaison de deux variables Byte, l'une étant un témoin et l'autre celle que l'on veut tester. Tout d'abord, il faut savoir que cette comparaison se fait à l'aide d'une fonction (en pascal la fonction &). Cette fonction ressemble beaucoup à la fonction booléenne « et ». Elle compare selon le même principe deux bites (0 ou 1) donnée :

- Elle retourne 1 si les deux bites sont égales à 1
- Elle retourne 0 si l'un des deux bites est égal à 0.

Elle permet donc de savoir si le bit testé est à l'état 1 ou si il est à l'état 0.

En pratique un Byte est composé de plusieurs bites. Pour connaître le n-ième bite de celui si, on va prendre (0...010...0) comme Byte témoin.





1 n p

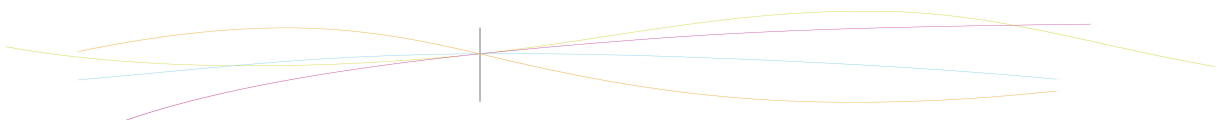
La fonction retournera donc :

- (0...010...0) si le bite testé est 1
- (0...000...0) si le bite testé est 0

1 n p

Prenons un exemple pour mieux comprendre: si l'on veut connaître la 3<sup>e</sup> bite de la variable 0100110, le masque binaire vous retournera 0000000, et si la variable vaut 0110110, il retournera 0010000.

Cette fonction est rapide et efficace puisqu'elle ne nécessite pas de boucle for ou while qui peuvent dans un programme ralentir celui-ci lors de son exécution.



## 4. CONCLUSIONS ET PERSPECTIVES

### 4.1. Conclusions sur le travail réalisé

La réalisation du projet a permis de montrer que certains instruments de mesure numérique qui nous paraissent très compliqués peuvent en fait être réalisés dans des conditions moindres que permettent les salles de TP. Evidemment la qualité n'est pas la même, mais les valeurs observées sont cohérentes avec la réalité et nous permettent de conclure à un certain succès du projet.

### 4.2. Conclusions sur l'apport personnel de cette U.V. projet

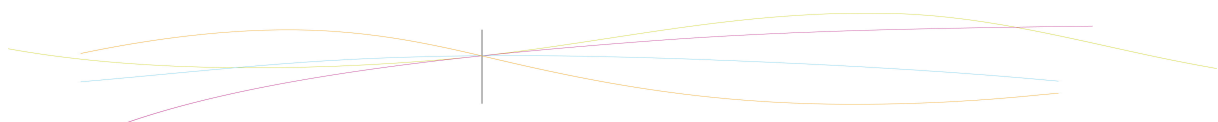
Dans un premier temps, il est évident que cette U.V projet nous apprend et nous habitue à travailler en équipe sur un sujet donné. La réalisation d'un tel projet demande de l'intérêt et de l'investissement personnel s'il veut être mené à bien. De plus, notre groupe étant composé de manière équivalente d'étudiants français et étrangers, et cela nous a obligé à comprendre les notions afin de pouvoir les réexpliquer, ce qui est très enrichissant pour tout le monde.

En outre, ce projet a permis de compiler et d'approfondir nos connaissances dans plusieurs domaines, l'électronique et l'informatique dans notre cas, et surtout d'appliquer ces connaissances théoriques dans un cadre concret, avec la satisfaction du résultat à la fin. Dans le cas de l'informatique, cela a permis d'apprendre la variante objet du Pascal, avec des notions encore inconnues, dans le domaine de l'électronique, le projet a permis de replacer des composants dans des circuits un peu moins pédagogiques que pendant les TP réalisés en premier cycle.

Même si le lancement a été laborieux et long, il ressort de ce projet un bilan très positif, tant au niveau du résultat que de l'apport personnel.

### 4.3. Perspectives pour la poursuite de ce projet

N'ayant couvert qu'une des deux grandeurs macroscopiques, il pourrait être intéressant de chercher à pouvoir faire la conversion de l'humidité. De plus, il serait intéressant de pouvoir exploiter le port USB, plus moderne que le port parallèle, et surtout plus commun sur les ordinateurs actuels.



## 5. BIBLIOGRAPHIE

### Livres

Kent Reisdorph, "Delphi 4", CampusPresse Edition.  
Rapport de projets antérieurs sur le même type de sujet

### Liens internet

#### *Partie informatique*

[http://www.pobot.org/spip.php?page=plan&tri=par\\_rubrique](http://www.pobot.org/spip.php?page=plan&tri=par_rubrique)

[http://www.aurel32.net/elec/port\\_parallele.php](http://www.aurel32.net/elec/port_parallele.php)

[http://www.sonelec-musique.com/logiciels\\_freewares\\_lptporttest.html](http://www.sonelec-musique.com/logiciels_freewares_lptporttest.html)

[http://fr.wikipedia.org/wiki/Delphi\\_\(informatique\)](http://fr.wikipedia.org/wiki/Delphi_(informatique))

<http://www.sixca.com/eng/articles/pardel/>

<http://www.abcelectronique.com/forum/archive/index.php/t-3399.html>

<http://www.abcelectronique.com/forum/archive/index.php/t-27524.html>

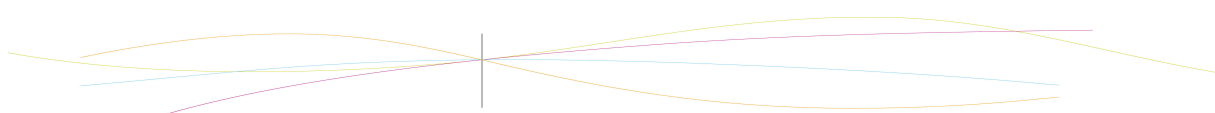
<http://www.astrosurf.org/buil/iris/32bits/winxp.htm>

[http://www.google.fr/search?hl=fr&q=input32.dll+delphi&meta=lr%3Dlang\\_fr](http://www.google.fr/search?hl=fr&q=input32.dll+delphi&meta=lr%3Dlang_fr)

#### *Partie électronique*

<http://www.eudil.fr/eudil/bbsc/bip/bip310.htm>

[http://www.datasheetcatalog.net/datasheets\\_pdf/T/I/P/1/TIP137.shtml](http://www.datasheetcatalog.net/datasheets_pdf/T/I/P/1/TIP137.shtml)



## 6. ANNEXES

### 6.1. Logiciels utilisés

Maple12 pour les résolutions

Synchronie 2003 pour les courbes

Delphi6 pour compiler le programme

### 6.2. Programme réalisé

```
{$A-,H+}
```

```
unit mainform;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, zportio, ComCtrls, shellapi{,rdtsc};
```

```
type
```

```
TMain = class(TForm)
```

```
lb1: TLabel;
```

```
ePort: TEdit;
```

```
lb2: TLabel;
```

```
eData: TEdit;
```

```
eRData: TEdit;
```

```
btnExit: TButton;
```

```
lb3: TLabel;
```

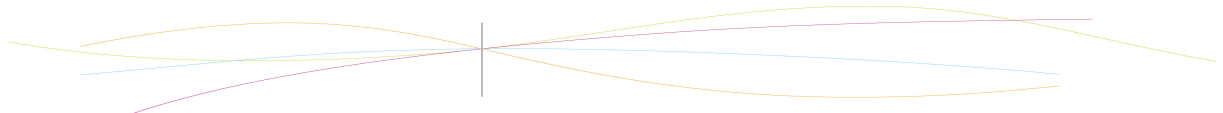
```
coDataType: TComboBox;
```

```
Lb4: TLabel;
```

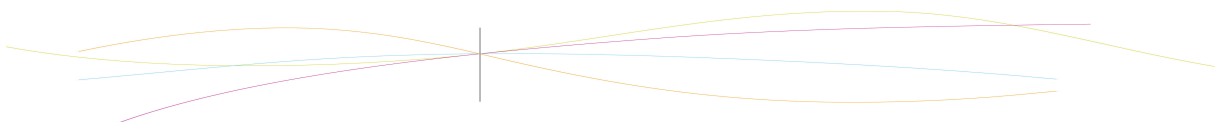
```
gb1: TGroupBox;
```

```
sbBar: TStatusBar;
```

```
lb5: TLabel;
```



```
btnWrite: TButton;
btnRead: TButton;
cbDirect: TCheckBox;
GroupBox1: TGroupBox;
Label1: TLabel;
Label2: TLabel;
GroupBox2: TGroupBox;
Label3: TLabel;
Label4: TLabel;
Label6: TLabel;
Label7: TLabel;
ack1: TEdit;
Label5: TLabel;
go: TButton;
Label8: TLabel;
Label9: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Label10: TLabel;
Label11: TLabel;
Edit3: TEdit;
Label12: TLabel;
Label13: TLabel;
Edit4: TEdit;
procedure btnExitClick(Sender: TObject);
procedure ePortKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure ePortKeyPress(Sender: TObject; var Key: Char);
procedure FormCreate(Sender: TObject);
procedure coDataTypeChange(Sender: TObject);
procedure ePortKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure btnWriteClick(Sender: TObject);
procedure eDataKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure btnReadClick(Sender: TObject);
procedure cbDirectClick(Sender: TObject);
```



```
procedure llbWWWClick(Sender: TObject);
procedure goClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
procedure setrdata(const Data:dword);
function str2int(const value:string;const HEX:boolean):integer;
end;

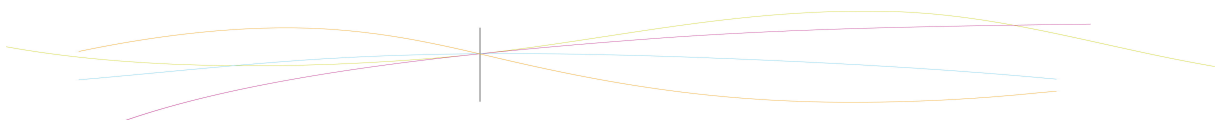
var
  Main: TMain;
implementation

{$R *.DFM}
{$R-}

function TMain.str2int(const value:string;const HEX:boolean):integer;
begin
  if HEX then
    result := strtoint('$' + value)
  else
    result := strtoint(value);
end;

procedure TMain.btnExitClick(Sender: TObject);
begin
  close;
end;

procedure TMain.ePortKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  try
    Case key of
      38: begin
```



```

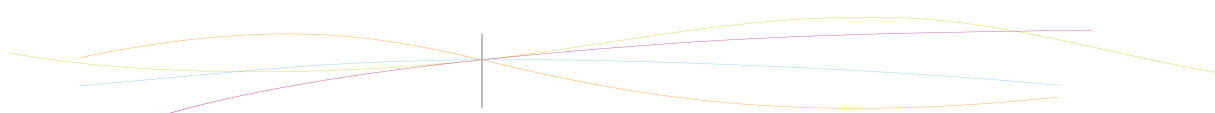
    key := 0;
    end;
40: begin
    key := 0;
    end
end;
except
end;
end;

procedure TMain.ePortKeyPress(Sender: TObject; var Key: Char);
begin
    // filter on hex
    if pos(key,#13#8'0123456789abcdefABCDEF')=0 then
        key := #0;
    end;

procedure TMain.FormCreate(Sender: TObject);
begin
    coDataType.itemindex :=0;
    coDataType.Tag := 2;
    if ZLIOShared then
        sbBar.SimpleText := 'Driver successfully started !'
    else
        sbBar.SimpleText := 'Couldnt start driver. Something wrong !';
    end;

procedure TMain.coDataTypeChange(Sender: TObject);
var s:string;
begin
    coDataType.Tag := 2 shl (coDataType.itemindex);
    erdata.MaxLength := coDataType.Tag;
    edata.MaxLength := coDataType.Tag;
    s := edata.Text;
    delete(s,1,length(s)-coDataType.Tag);
    edata.text := s;
end;

```



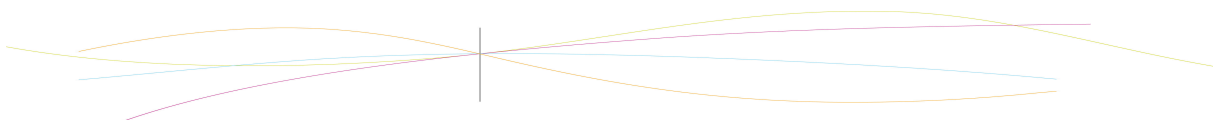
```

procedure TMain.ePortKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
var Port,data:dword;
begin
try
  try
    Port := str2int(ePort.Text,true);
  except
    Port := 0;
  end;
Case key of
  13: begin
    zlioportwrite(port,coDataType.itemindex,str2int(eData.Text, true));
    end;
  38: begin
    inc(Port);
    ePort.Text := inttohex(Port,3);
    key := 0;
    end;
  40: begin
    if port > 0 then
      dec(Port);
      ePort.Text := inttohex(Port,3);
      key := 0;
    end
end;
setthreadpriority(GetCurrentThread,THREAD_PRIORITY_TIME_CRITICAL);
data := zlioportread(Port,coDataType.itemindex);
setrdata(data);
except
end;

end;

procedure TMain.setrdata(const Data:dword);
begin

```

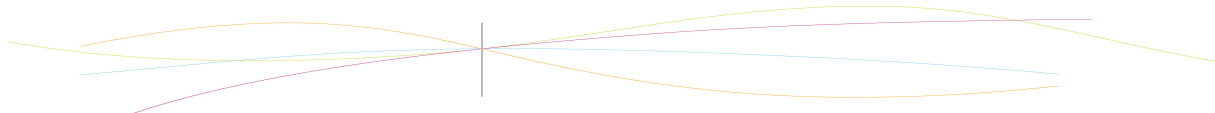




```
erData.Text := inttohex(Data,coDataType.Tag);  
end;
```

```
procedure TMain.btnWriteClick(Sender: TObject);  
var i:word;  
begin  
  i := 13;  
  ePortKeyDown( self,i,[])  
end;
```

```
procedure TMain.eDataKeyDown(Sender: TObject; var Key: Word;  
  Shift: TShiftState);  
var data:dword;  
begin  
  try  
    Data := str2int(eData.Text,true);  
    except Data := 0; end;  
  Case key of  
    13: begin  
      ePortKeyDown( self,key,[])  
      end;  
    38: begin  
      inc(Data);  
      eData.Text := inttohex(Data,coDataType.Tag);  
      key := 0;  
      end;  
    40: begin  
      dec(Data);  
      eData.Text := inttohex(Data,coDataType.Tag);  
      key := 0;  
      end  
  end;  
end;  
  
end;
```



```
procedure TMain.btnReadClick(Sender: TObject);
```

```
var k:word;
```

```
begin
```

```
  k := 0;
```

```
  ePortKeyDown(self, k,[]);
```

```
  // ack1.Text:='0' ;
```

```
  // EEdit1.Text:='0' ;
```

```
  // EEdit2.Text:='0' ;
```

```
  // pour le mask et tout
```

```
  // case eRData.Text[1] of
```

```
  // '3' : ack1.Text:='1' ;
```

```
  // '7' : EEdit1.Text:='1' ;
```

```
  // '6' : EEdit2.Text:='1' ;
```

```
  // end
```

```
end;
```

```
procedure TMain.cbDirectClick(Sender: TObject);
```

```
begin
```

```
  zliosetiopm(cbDirect.Checked);
```

```
end;
```

```
procedure TMain.llbWWWClick(Sender: TObject);
```

```
begin
```

```
  shellexecute( 0,'open','http://www.specosoft.com/update/portio_15.htm',nil,nil,SW_SHOW);
```

```
end;
```

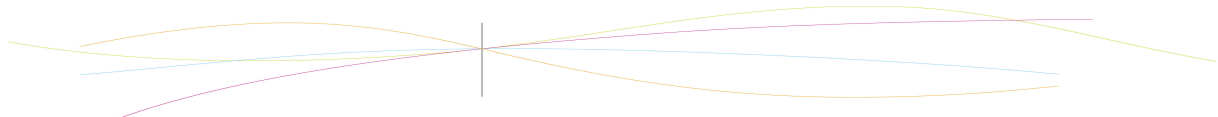
```
procedure TMain.goClick(Sender: TObject);
```

```
var
```

```
  compteur : integer ;
```

```
  temp : char ;
```

```
  {temperature : real ;    }
```



```

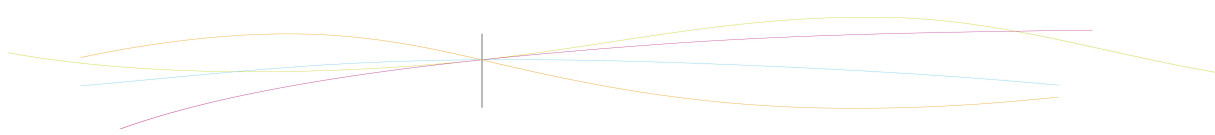
begin
  compteur := 0 ;
  temp:='2';
  repeat
    btnRead.click;
  until ((eRData.Text[1]='7') or (eRData.Text[1]='6'));

  repeat
    btnRead.click;
  until ((eRData.Text[1]='3') or (eRData.Text[1]='2'));
  while ((eRData.Text[1]='2') or (eRData.Text[1]='3')) do
  begin
    btnRead.click;
    if eRData.Text[1]<>temp then
      begin
        if(temp='2') then
          temp:='3'
        else
          begin
            temp:='2';
            compteur := compteur + 1 ;
          end;
          Edit3.text := inttostr(compteur);
        end
      end ;
    { temperature:= ((235.9*8.09+16.2)-compteur)/0.06; }
    { Edit4.Text:=floattostr(temperature); }

  end;

end.

```



### 6.3. Schémas de montages, plans de conception...

$$eq1 := 1 = \left( \frac{(10+x)}{10} \right) \cdot \left( \frac{y}{(10+y)} \right) \cdot (5.36) - \frac{x}{2};$$

$$1 = \frac{5.36 \left( 1 + \frac{1}{10} x \right) y}{10 + y} - \frac{1}{2} x$$

$$eq2 := 10 = \left( \frac{(10+x)}{10} \right) \cdot \left( \frac{y}{(10+y)} \right) \cdot (5.64) - \frac{x}{2};$$

$$10 = \frac{5.64 \left( 1 + \frac{1}{10} x \right) y}{10 + y} - \frac{1}{2} x$$

`solve({eq1, eq2}, {x, y});`

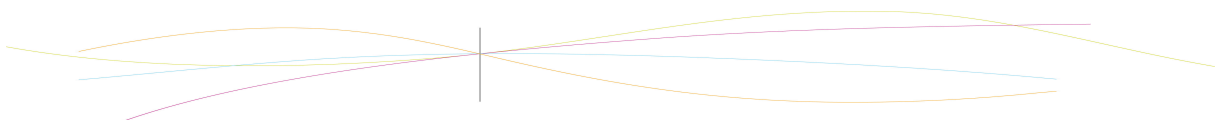
`{x = 342.5714286, y = 103.2110092}`

Résolution des équations pour trouver R2 (x) et R4 (y)

`Uo=0.6`  
`a=-0.002`

```
temp=RAMPE(-20,100)
Uak=a*temp+Uo
R1=10
R2=342.57
R3=10
R4=0.97
alpha=(1+(R2/R1))*(R3/(R3+R4))
beta=R2/R1
Uth=alpha*(Uak+5)-beta*5
```

Calcul pour avoir la tension comprise entre 1V et 10V



NotrePortIO

**OBJECT**

ce programme va nous permettre de manipuler le port parallèle en écriture (envoi d'un signal) et en lecture (reception d'un signal)

**Aide**

le programme peut accepter des valeur en base décimale ( 10 ) ou base hexadécimale ( 16 ).

exemple : pour le 7ieme port D6,  $2^7 = 64$  s'écrit aussi 40 en hexadécimal

conversion en hexa

16 → 10

32 → 20

64 → 40

128 → 80

numéro du port :

écriture :

lecture :

valeur port ack 3 :  (generateur 1 hz)

valeur port strobe 7:

valeur port select 6:  (convertisseur)

compteur

temperature

Interface obtenue grâce au programme, avec les résultats d'un essai

