



UNIVERSITE D'ANTANANARIVO

-----

ECOLE SUPERIEURE POLYTECHNIQUE

D'ANTANANARIVO

-----

DEPARTEMENT METEOROLOGIE

-----



**MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLOME D'INGENIEUR**

***Spécialité* : SYSTEME D'INSTRUMENTS METEOROLOGIQUES**

***CONCEPTION, SIMULATION ET REALISATION  
D'UN SYSTEME D'INSTRUMENTS  
METEOROLOGIQUES NUMERIQUES POUR LA  
MESURE DE VENT ET DE TEMPERATURE  
(ff, dd, TT) AVEC INTERFACE USB 2.0***

***Présenté par:***

ANDRIAMAOLISON Herman

*Soutenu publiquement le 30 Mai 2015*



UNIVERSITE D'ANTANANARIVO

-----  
ECOLE SUPERIEURE POLYTECHNIQUE D'ANTANANARIVO



-----  
DEPARTEMENT METEOROLOGIE

-----  
MEMOIRE DE FIN D'ETUDES POUR L'OBTENTION DU DIPLOME D'INGENIEUR

**CONCEPTION, SIMULATION ET REALISATION  
D'UN SYSTEME D'INSTRUMENTS  
METEOROLOGIQUES NUMERIQUES POUR LA  
MESURE DE VENT ET DE TEMPERATURE  
(ff, dd, TT) AVEC INTERFACE USB 2.0**

**Spécialité** : SYSTEME D'INSTRUMENTS METEOROLOGIQUES

**Présenté par** : ANDRIAMAOLISON Herman

**Membres de Jury**

**Président de Jury** : **Monsieur RAKOTOVAZAHA Olivier**, Chef de Département de la  
Météorologie

**Examineurs** : **Monsieur RANDRIANASOLO Léon**, Enseignant Chercheur au  
sein du Département de la Météorologie

**Monsieur RASTEFANO Elisée**, Enseignant Chercheur au sein du  
Département de l'Electronique

**Monsieur RAKOTONDRA SOA Jean Claude**, Enseignant  
Chercheur au sein du Département de la Météorologie

**Directeur de mémoire** : **Monsieur RAKOTOARIMANANA Nirison**, Chef de Département  
de la Navigation Aérienne à l'Aviation Civile de Madagascar

*A ma Petite Fille...*

*A ma Femme ...*

*A ma Mère...*

*A toute ma Famille...*

*“Un obstacle n’est pas un obstacle”*

## REMERCIEMENTS

Le présent Mémoire est dédié à la mémoire de notre cher Père.

Cette page de remerciement est dédiée à toutes les personnes qui nous ont aidés et favorisés, professionnellement ou personnellement, à l'élaboration de ce mémoire de fin d'études.

Nous adressons donc toute notre gratitude et toute notre sympathie à :

Monsieur ANDRIANARY Philippe Antoine, Directeur de l'Ecole Supérieure Polytechnique d'Antananarivo qui a permis grâce aux différents organismes d'assurer le bon déroulement de nos années d'études.

Monsieur RAKOTOVAZAHA Olivier, Chef du Département de la Météorologie qui malgré ses multiples tâches et obligations, nous fait un grand honneur de présider la séance de notre soutenance.

Monsieur RANDRIANASOLO Léon, Enseignant Chercheur au sein du Département de la Météorologie. Je suis très touché et honoré qu'il ait accepté d'examiner ce travail.

Monsieur RASTEFANO Elisée, Enseignant Chercheur à l'Ecole Supérieure Polytechnique d'Antananarivo au sein du Département de l'Electronique, qui donne encore plus de valeur à notre mémoire pour avoir bien voulu consacrer à notre travail une partie de leur temps extrêmement précieux.

Monsieur RAKOTONDRA SOA Jean Claude, Enseignant Chercheur à l'Ecole Supérieure Polytechnique d'Antananarivo au sein du Département de la Météorologie, qui nous a familiarisé avec les instruments météorologiques, et qui a accepté de juger ce travail.

Monsieur RAKOTOARIMANANA Nirison, Chef de Département de la Navigation Aérienne à l'Aviation Civile de Madagascar et Directeur de ce Mémoire à qui nous a apportés une aide efficace et des conseils durant l'élaboration de notre tâche.

A tous nos Professeurs de l'Ecole Supérieure Polytechnique d'Antananarivo, qui nous ont transmis leurs savoirs.

A tout le Personnel Administratif et Technique de l'ESPA, de la Direction de la Météorologie Ampandrianomby, de l'ACM et de l'ASECNA.

A notre mère pour son affection, soutien moral, patience et sa sacrifice qu'elle a fait preuve tout au long de notre études : « Merci maman ».

A notre famille, nos amis, qui nous ont apporté leurs aides et leurs conseils.

## TABLE DES MATIERES

<b>REMERCIEMENTS.....</b>	<b>i</b>
<b>TABLE DES MATIERES .....</b>	<b>ii</b>
<b>LISTE DES ABREVIATIONS ET SIGLES.....</b>	<b>vii</b>
<b>LISTE DES FIGURES.....</b>	<b>ix</b>
<b>LISTE DES TABLEAUX .....</b>	<b>xi</b>
<b>INTRODUCTION .....</b>	<b>1</b>
<b>PARTIE I .....</b>	<b>2</b>
<b>GENERALITES SUR LES INSTRUMENTS METEOROLOGIQUES .....</b>	<b>2</b>
<b>I.1 Stations météorologiques d’observations [16] .....</b>	<b>3</b>
<b>I.2 Classification des stations .....</b>	<b>3</b>
I.2.1 Stations synoptiques .....	3
I.2.2 Stations climatologiques.....	4
I.2.3 Stations de météorologie aéronautique.....	4
I.2.4 Stations de météorologie agricole.....	4
I.2.5 Stations spéciales.....	4
<b>I.3 Types d’observations.....</b>	<b>4</b>
I.3.1 Observations synoptiques.....	4
I.3.1.1 Observations sensorielles.....	4
I.3.1.2 Observations instrumentales .....	4
I.3.2 Observations climatologiques .....	5
I.3.2.1 Observations sensorielles.....	5
I.3.2.2 Observations instrumentales .....	5
I.3.3 Observations aéronautiques (sensorielles et instrumentales).....	5
I.3.4 Observations agrométéorologiques .....	6
I.3.4.1 Observations sensorielles.....	6
I.3.4.2 Observations instrumentales .....	6
I.3.5 Observations pluviométriques .....	6
I.3.6 Observations spéciales.....	7
<b>I.4 Heures des observations.....</b>	<b>7</b>
I.4.1 Heures d’observations synoptiques de surface .....	7
I.4.2 Heures des observations climatologiques .....	7
<b>I.5 Ordre chronologique des observations météorologiques .....</b>	<b>8</b>
<b>I.6 Emplacement et exposition des instruments.....</b>	<b>10</b>
I.6.1 Site de la station .....	10
I.6.2 Parc d’instruments.....	10
<b>I.7 Aspects spécifiques des mesures météorologiques [1].....</b>	<b>11</b>
<b>I.8 Mesures météorologiques directes et indirectes .....</b>	<b>12</b>
<b>I.9 Diagramme d’un instrument météorologique .....</b>	<b>12</b>
I.9.1 Instruments de mesure analogique .....	12
I.9.2 Instruments de mesure numérique .....	13

<b>I.10 Comportement dynamique des instruments de mesure - Instrument de mesure de premier ordre - Instrument de mesure de second ordre - Caractéristiques général des instruments météorologiques [16]</b>	<b>15</b>
I.10.1 Temps de stabilisation .....	15
Les instruments (ou systèmes) de mesure de premier ordre.....	15
Les instruments (ou systèmes) de mesure de second ordre .....	16
I.10.2 La précision .....	18
I.10.3 La sensibilité.....	18
I.10.4 Spécificité de réponse.....	18
I.10.5 Linéarité de réponse : .....	18
I.10.6 Régularité .....	18
I.10.7 La simplicité et la commodité d'utilisation et d'entretien .....	19
I.10.8 La solidité de construction.....	19
I.10.9 La fiabilité .....	19
<b>PARTIE II.....</b>	<b>21</b>
<b>CONCEPTION ET SIMULATION DU SYSTEME .....</b>	<b>21</b>
<b>II.1 Contexte du mémoire .....</b>	<b>22</b>
<b>II.2 Cahier des charges .....</b>	<b>22</b>
<b>II.3 L'anémomètre .....</b>	<b>23</b>
II.3.1 La force du vent – Caractéristiques – Mesure. [16] .....	23
<b>Calculs d'un anémomètre à coupelles .....</b>	<b>26</b>
II.3.2 Principe de fonctionnement de l'anémomètre .....	29
<b>II.4 La girouette.....</b>	<b>30</b>
<b>II.5 Le thermomètre.....</b>	<b>31</b>
I.5.1 Généralités sur la chaleur et la température.....	31
La chaleur .....	31
La température.....	32
I.5.2 Echelles de température.....	32
II.5.3 Mesure de température en météorologie .....	33
II.5.4 Le capteur électronique du thermomètre [21] .....	33
<b>II.6 La carte d'acquisition [17].....</b>	<b>36</b>
II.6.1 Généralité sur les microcontrôleurs PIC .....	36
Différentes familles des PIC .....	36
Identification d'un Pic.....	36
<b>II.6.2 Le microcontrôleur PIC18F4550 [10] .....</b>	<b>37</b>
II.6.2.1 Vue d'ensemble .....	37
Les caractéristiques principales du 18F4550 .....	37
Aspect externe du PIC18F4550 .....	37
Structure interne du PIC.....	38
II.6.2.2 Organisation des mémoires.....	40
Mémoire programme.....	40
Mémoire de Données RAM .....	40
Mémoire de Données EEPROM .....	41
II.6.2.3 Le reset du microcontrôleur .....	41
II.6.2.4 Les bits de configuration du microcontrôleur [8] .....	43
II.6.2.5 Le circuit de génération d'horloge .....	44
II.6.2.6 Fonctionnement des entrées/sorties du Pic18f4550 .....	45
II.6.2.7 Les Timers ou Compteur .....	47
Le Timer0 .....	47
Le Timer1 .....	48
Le Timer2 .....	48
II.6.2.8 Le Watch Dog Timer .....	48
II.6.2.9 Les interruptions .....	49

Généralité sur les interruptions .....	49
Les interruptions sur le PIC18F4550 .....	49
II.6.3 Le Convertisseur Analogique Numérique ou ADC [18] .....	50
II.6.3.1. Généralité sur l'ADC (Analogic Digital Converter ADC en anglais).....	50
Echantillonnage/Blocage .....	51
La quantification .....	51
Le codage.....	51
II.6.3.2 Le CAN du microcontrôleur .....	52
II.6.4 Le port USB et sa fonctionnalité [2] .....	54
II.6.4.1 Généralité sur l'USB.....	54
II.6.4.2 Notions importantes pour commencer avec l'USB [4] .....	56
Le câble USB .....	56
Choix « Low » ou « Full » USB .....	57
Temps de connexion et de déconnexion .....	57
Le codage NRZI.....	58
Alimentation USB.....	59
Différents types de transfert .....	60
L'énumération .....	60
II.6.4.3 Les descripteurs [5].....	61
Device Descriptor .....	62
Configuration descriptor .....	64
Interface descriptor .....	65
Endpoint descriptor .....	66
HID descriptor .....	67
II.6.4.4 Le bus USB.....	68
Principe du bus USB .....	68
Topologie du Bus USB .....	68
Protocole USB .....	69
Type de paquet USB .....	71
<b>II.7 Programmation du PIC 18F4550 [8] .....</b>	<b>72</b>
II.7.1 Les étapes de la programmation de pic .....	72
II.7.2 Programmation en langage C .....	72
II.7.2.1 L'environnement MPLAB .....	73
II.7.2.2 Le programmeur PIC Pickit3 .....	74
<b>II.8 Dimensionnement de l'alimentation .....</b>	<b>74</b>
<b>II.9 Programmation host [11].....</b>	<b>75</b>
II.9.1 Operating System (OS) .....	75
II.9.2 Les fichiers .INF [12].....	77
Règles à respecter et recommandation pour concevoir un fichier INF .....	77
Applications des fichiers INF .....	78
Rôle du fichier INF .....	78
II.9.3 Les drivers .SYS [12].....	78
Définitions d'un driver .SYS .....	78
Création d'un driver avec le DDK [13].....	79
<b>II.10 Simulation du système .....</b>	<b>79</b>
<b>PARTIE III .....</b>	<b>81</b>
<b>REALISATION, ETALONNAGE ET INTERPRETATION DES RESULTATS .....</b>	<b>81</b>
<b>III.1 Réalisation du système .....</b>	<b>82</b>
III.1.1 L'anémomètre.....	82
La partie électrique de l'anémomètre .....	82
Réalisation de la partie mécanique .....	83
III.1.2 La girouette.....	84
La partie électrique de la girouette .....	84

Réalisation de la partie mécanique .....	84
III.1.3 Le thermomètre [21] .....	85
III.1.3 Réalisation de la carte d'acquisition [24] .....	85
III.1.3 Réalisation de l'alimentation .....	87
<b>III.2 Etalonnage et Interprétation des résultats .....</b>	<b>88</b>
III.2.1 L'étalonnage .....	88
Etalonnage de la girouette .....	89
Etalonnage de l'anémomètre .....	89
Etalonnage du thermomètre .....	89
III.2.2 Interpretation des résultats .....	90
III.2.2.1 La carte d'acquisition .....	90
III.2.2.2 L'anémomètre .....	90
III.2.2.3 La girouette .....	90
III.2.2.3 Le thermomètre .....	91
<b>CONCLUSION.....</b>	<b>92</b>
<b>ANNEXES.....</b>	<b>I</b>
<b>Annexe A : Le microcontrôleur PIC18F4550 .....</b>	<b>II</b>
A1. Spécificité du PIC18F4550 .....	II
A2. Paramètres de configuration de l'horloge et le registre OSCCON .....	III
A2.1 Detail des 4 paramètres de configuration de l'horloge du PIC18F4550 .....	III
A2.2 Détail du registre OSCCON ou Oscillator Control Register .....	III
<b>Annexe B : La carte d'acquisition .....</b>	<b>V</b>
B1. Schéma électronique de la carte .....	V
B2. Liste des composants électroniques de la carte d'acquisition .....	V
B3. Tracés des circuits imprimés (typons) .....	VI
B4. Implantation des composants .....	VII
<b>Annexe C : L'anémomètre .....</b>	<b>VIII</b>
C1. Schéma électronique .....	VIII
C2. Listes des composants tête anémomètre .....	VIII
C3. Typons tête anémomètre .....	VIII
C4. Implantation des composants .....	IX
C5. Disque de codage .....	IX
<b>Annexe D : La girouette.....</b>	<b>X</b>
D1. Schéma électronique .....	X
D2. Listes des composants tête de la girouette .....	X
D3. Typons de la girouette .....	X
D4. Implantation des composants .....	XI
D5. Disque de codage .....	XI
<b>Annexe E : Le thermomètre .....</b>	<b>XII</b>
E1. Schéma électronique .....	XII
E2. Liste des composants .....	XII
E3. Typons du thermomètre .....	XII
E4. Implantation des composants .....	XIII
<b>Annexe F : L'alimentation 5V.....</b>	<b>XIV</b>
F1. Schéma électronique .....	XIV
F2. Liste des composants .....	XIV
F3. Typons de l'alimentation 5V .....	XIV
F4. Implantation des composants pour l'alimentation .....	XV
<b>Annexe G : Quelque capture d'écran lors de la simulation.....</b>	<b>XVI</b>
G1. Capture de l'afficheur LCD .....	XVI



G2. Capture du console.....	XVII
G3. Capture de la fenêtre “ Périphériques et imprimantes” sous Windows 7 .....	XVII

## **REFERENCE BIBLIOGRAPHIQUE .....XVIII**

### **A. BIBLIOGRAPHIE ..... XVIII**

### **B. WEBOGRAPHIE ..... XVIII**

## LISTE DES ABREVIATIONS ET SIGLES

<b>ADC</b>	Analog to Digital Converter ou CAN (Convertisseur Analogique Numérique) en français
<b>API</b>	Application Programming Interface
<b>CISC</b>	Complex Instruction Set Computer
<b>DDK</b>	Driver Developer's Kit
<b>DOS</b>	Disk Operating System
<b>EEPROM</b>	Electrically Erasable Programmable Read Only Memory
<b>EPROM</b>	Erasable Programmable Read Only Memory
<b>GMT</b>	Greenwich Mean Time
<b>GND</b>	Ground
<b>HID</b>	Human Interface Device
<b>IBJT</b>	Insulated Gate Bipolar Transistor
<b>IRQ</b>	Interrupt ReQuest
<b>LCD</b>	Liquid Cristal Display
<b>LED</b>	Light Emitting Diode
<b>LSB</b>	Least Significant Bit
<b>Mbps</b>	Méga Bits Par Seconde
<b>MSB</b>	Most Significant Bit
<b>MTBF</b>	Mean Time Between failures
<b>NRZI</b>	Non Return to Zero Inverted
<b>NT</b>	Nouvelle Technologie
<b>OACI</b>	Organisation de l'Aviation Civile Internationale
<b>OMM</b>	Organisation Météorologique Mondiale
<b>PIC</b>	Programmable Interrupt Controller
<b>PID</b>	Product IDentifier
<b>PnP</b>	Plug-and-Play

<b><i>PROM</i></b>	Programmable Read Only Memory
<b><i>RAM</i></b>	Random Access Memory
<b><i>RISC</i></b>	Reduced Instruction Set Computer
<b><i>ROM</i></b>	Read Only Memory
<b><i>USB</i></b>	Universal Serial Bus
<b><i>VID</i></b>	Vendor Identifier
<b><i>WDK</i></b>	Windows Driver Kit
<b><i>WDM</i></b>	Windows Driver Models

## **LISTE DES FIGURES**

<i>Figure 1 : Diagramme d'un instrument de mesure analogique</i> .....	13
<i>Figure 2: Schéma de fonctionnement de l'anémomètre analogique</i> .....	13
<i>Figure 3 : Diagramme d'un instrument de mesure numérique</i> .....	14
<i>Figure 4 : Schéma de fonctionnement de l'anémomètre numérique</i> .....	14
<i>Figure 5 : Réponse d'un instrument météorologique du premier ordre</i> .....	16
<i>Figure 6 : Réponse d'un instrument de mesure du second ordre</i> .....	17
<i>Figure 7 : Taux de défaillance en fonction du temps</i> .....	20
<i>Figure 8 : Schema fonctionnel du système</i> .....	23
<i>Figure 9 : Diagramme des forces d'un anémomètre à coupelles</i> .....	27
<i>Figure 10 : Chronogramme de mise à jour des données anémométrique</i> .....	30
<i>Figure 11 : Montage utilisant la Diode comme capteur de température</i> .....	34
<i>Figure 12 : Montage utilisant un BJT comme capteur de température</i> .....	34
<i>Figure 13 : Boitier du capteur LM35DZ</i> .....	35
<i>Figure 14 : Brochage du capteur LM35CZ/LM35DZ</i> .....	35
<i>Figure 15 : Aspect externe du PIC18F4550</i> .....	38
<i>Figure 16 : Structure interne du 18F4550.</i> .....	39
<i>Figure 17 : Mapping du mémoire programme du 18F4550</i> .....	40
<i>Figure 18 : Mapping du mémoire de données RAM du 18F4550</i> .....	41
<i>Figure 19 : Schéma bloc simplifié du circuit de reset du 18F4550</i> .....	42
<i>Figure 20 : Circuit de génération d'horloge du 18F4550</i> .....	44
<i>Figure 21 : Schéma simplifié des Entrées/Sorties du pic 18F4550</i> .....	45
<i>Figure 22 : schéma interne d'une borne d'entrée / sortie</i> .....	46
<i>Figure 23 : Schéma bloc du Timer0</i> .....	47
<i>Figure 24 : Schéma bloc du Timer1</i> .....	48
<i>Figure 25 : Schéma bloc du Timer2</i> .....	48
<i>Figure 26 : Symbole de la fonction Conversion Analogique/Numérique</i> .....	50
<i>Figure 27 : Principe d'un échantillonneur bloqueur</i> .....	51
<i>Figure 28 : Les différents signaux d'un CAN</i> .....	52
<i>Figure 29 : Schéma bloc du CAN intégré dans le PIC18F4550</i> .....	53
<i>Figure 30 : Brochage des connecteurs USB de type A et B</i> .....	56
<i>Figure 31 : Aspect des connecteurs USB de type A et B</i> .....	56
<i>Figure 32 : Choix Low ou Full USB – Résistance de Pull Up</i> .....	57
<i>Figure 33 : Principe du Codage NRZI</i> .....	58
<i>Figure 34 : Diagramme hiérarchique des descripteurs</i> .....	62
<i>Figure 35 : Topologie du bus USB</i> .....	69
<i>Figure 36 : Protocole USB</i> .....	70
<i>Figure 37 : Transaction USB</i> .....	71
<i>Figure 38 : Architecture de Windows NT</i> .....	76
<i>Figure 39 : Capteur infrarouge EX-SX1115 de l'anémomètre</i> .....	82
<i>Figure 40 : Tête de l'anémomètre sous Proteus</i> .....	83

<i>Figure 41 : Roue de codage de l'anémomètre .....</i>	<i>83</i>
<i>Figure 42 : Schéma de câblage du capteur de la girouette .....</i>	<i>84</i>
<i>Figure 43 : Schéma du disque de codage de la girouette .....</i>	<i>84</i>
<i>Figure 44 : Tête du thermomètre .....</i>	<i>85</i>
<i>Figure 45 : Schéma électrique de la carte .....</i>	<i>86</i>
<i>Figure 46 : Photos de la carte finie .....</i>	<i>87</i>
<i>Figure 47 : Schéma de l'alimentation linéaire 5V à point milieu.....</i>	<i>88</i>
<i>Figure 48 : Représentation des 16 directions pouvant désigner la girouette .....</i>	<i>89</i>

## **LISTE DES TABLEAUX**

<i>Tableau 1 : Tableau de l'ordre chronologique des observations météorologique .....</i>	<i>9</i>
<i>Tableau 2 : Tableau des tâches de l'observateur météorologique .....</i>	<i>10</i>
<i>Tableau 3 : Relation entre les différentes unités de vitesse du vent .....</i>	<i>24</i>
<i>Tableau 4 : Echelle de Beaufort .....</i>	<i>25</i>
<i>Tableau 5 : Equivalence entre binaire naturel et binaire réfléchi sur 4bits .....</i>	<i>31</i>
<i>Tableau 6 : Echelle de température .....</i>	<i>32</i>
<i>Tableau 7 : Bits de configuration du 18F4550 .....</i>	<i>43</i>
<i>Tableau 8 : Identification des descripteurs .....</i>	<i>61</i>
<i>Tableau 9 : Tableau récapitulatif des différents champs des Device Descriptor .....</i>	<i>63</i>
<i>Tableau 10 : Tableau récapitulatif des différents champs des Configuration Descriptor .....</i>	<i>65</i>
<i>Tableau 11 : Tableau récapitulatif des différents champs des Interfaces Descriptor .....</i>	<i>66</i>
<i>Tableau 12 : Tableau récapitulatif des différents champs des Endpoint Descriptor .....</i>	<i>67</i>
<i>Tableau 13: Tableau récapitulatif des différents champs des HID Descriptors .....</i>	<i>68</i>
<i>Tableau 14 : Différentes couches de l'échange via USB .....</i>	<i>70</i>

# INTRODUCTION

Un nombre appréciable d'activités humaines sont affectées par le temps. De façon à minimiser les effets défavorables des phénomènes atmosphériques sur ces activités, l'homme a depuis les premiers temps étudié les lois qui gouvernent le temps.

L'acquisition des connaissances relatives au temps est un objectif de la météorologie. Les phénomènes météorologiques sont étudiés à partir des observations, des expériences et des méthodes scientifiques d'analyse.

Une observation météorologique est une évaluation ou une mesure d'un ou des plusieurs éléments météorologiques.

Les observations météorologiques sont soit sensoriels, c'est-à-dire acquises par l'observateur sans le secours d'instruments de mesure, soit instrumentales, qui sont généralement appelées mesures météorologiques et faites à l'aide d'instruments météorologiques.

Bon nombre d'instruments sont utilisés dans les diverses stations météorologiques pour mesurer les variables météorologiques : la pression atmosphérique, la température de l'air, l'humidité atmosphérique, la direction et la force du vent, les précipitations, l'évaporation, la durée d'isolation, le rayonnement solaire, la hauteur de la base des nuages, la visibilité...

Ces données d'observations sont la base de toute étude antérieure. Il est donc impératif de prendre soins à toutes activités relatives à leur recueil. Parmi ces activités, la conception des appareils de mesure est vitale. Nous y sommes ! Dieu merci !

Ce mémoire présente donc « *Conception, simulation et réalisation d'un système d'instruments météorologiques pour la mesure de vent et de température avec interface USB 2.0* ».

Notre travail se compose en trois parties :

**PARTIE I : Généralités sur les instruments météorologiques**

**PARTIE II : Conception et simulation du système**

**PARTIE III : Réalisation, étalonnage et interprétation des résultats**

**PARTIE I**

**GENERALITES SUR LES INSTRUMENTS  
METEOROLOGIQUES**



## I.1 Stations météorologiques d'observations [16]

Les stations météorologiques d'observation sont les lieux où l'on mesure ou évalue un ou plusieurs éléments météorologiques

Chaque pays devrait établir sur son territoire un réseau de station synoptiques, climatologiques, aéronautiques, agrométéorologiques, etc.

L'emplacement de chaque station doit être choisi de sorte que les instruments puissent y être exposé d'une manière correcte et que les observations faites sans instruments puissent y être exécutées d'une manière satisfaisante.

Chaque station synoptique doit être située de manière à fournir des données météorologiques représentatives de la région dans laquelle elle se trouve. Chaque station climatologique doit être située en un lieu et dans des conditions telles que son fonctionnement soit assuré pendant une période d'au moins dix ans et que son exposition ne soit pas modifiée pendant une période prolongée, à moins que cette station ne soit destinée à une fin spéciale justifiant son fonctionnement pendant un laps de temps plus court.

Les alentours de la station ne doivent jamais, au cours des années, subir des modifications qui puissent altérer l'homogénéité des séries d'observations.

Chaque station de la météorologie doit être située en un lieu représentatif des conditions naturelles et des caractéristiques de l'agriculture dans la région où elle se trouve.

## I.2 Classification des stations

Les stations météorologiques sont classées dans les principales catégories suivantes :

### I.2.1 Stations synoptiques

Ces sont des stations où l'on effectue des observations pour les besoins de la météorologie synoptique. La météorologie synoptique est la partie de la météorologie qui se rapporte à l'étude du temps réel basée sur des observations reportées sur des cartes géographiques. Son but est de prévoir l'évolution du temps.

#### En surface :

- |                                      |  |
|--------------------------------------|--|
| - stations terrestres avec personnel | } principales<br>} complémentaires<br>} à position fixe<br>} ou mobile |
| - Stations terrestres automatique    |  |
| - stations en mer avec personnel     |  |
| - Stations maritimes automatiques    |  |

#### En altitude :

- |  |  |
|--|--|
| - Stations terrestres                  | } de radiosondage et de radiovent<br>} ou d'observation par ballon sonde |
| - Stations en mer                      |  |
| - Stations météorologiques d'aéronefs. |  |

L'intervalle entre les stations synoptiques ne devrait pas dépasser 150km dans le cas des stations terrestre.

### **I.2.2 Stations climatologiques**

Une station climatologique est une station où l'on effectue des observations pour les besoins d'étude climatologique. Le climat correspond aux conditions météorologiques considérées sur une longue période. Les données météorologiques provenant des stations synoptiques peuvent aussi être utilisées à de fins climatologiques.

### **I.2.3 Stations de météorologie aéronautique**

Ses stations sont situées aux aéroports. Elles sont créées pour répondre aux besoins particuliers de l'aviation. On peut également y effectuer des observations synoptiques et climatologiques.

### **I.2.4 Stations de météorologie agricole**

Ces stations sont créées dans le but de fournir une assistance à l'agriculture. On y fait des observations particulières sur le milieu physique ainsi que des observations de caractère biologique.

### **I.2.5 Stations spéciales**

Ces stations sont créées dans le but d'observer des phénomènes météorologiques particuliers : observations des parasites atmosphériques, détections par radar des nuages et des hydrométéores, hydrologies, mesure de rayonnement ou de l'ozone, microclimatologie, chimie atmosphérique, électricité atmosphérique.

## **I.3 Types d'observations**

### **I.3.1 Observations synoptiques**

Dans toutes les stations du réseau synoptique, les observations et les mesures portent sur les éléments suivants :

#### ***I.3.1.1 Observations sensorielles***

- Nébulosité (qualité, identification, et hauteur des nuages)
- Visibilité
- Temps présent
- Phénomènes spéciaux
- Etat du sol
- Temps passé

#### ***I.3.1.2 Observations instrumentales***

- Température de l'air
- Température extrême (maxima et minima)

- Température du sol (aux profondeurs de 5, 10, 20, 50 et 100)
- Température minima du sol gazonneux (5cm au-dessus du sol)
- Précipitation, quantité et intensité
- Humidité
- Direction et vitesse du vent
- Pression atmosphérique
- Tendance de la pression atmosphérique
- Caractéristique de la tendance de la pression atmosphérique
- Evaporation
- Insolation
- Rayonnement

Une station synoptique peut être chargée d'exécuter d'autres observations et mesures en cas de nécessité.

### **I.3.2 Observations climatologiques**

Les observations faites par une station climatologiques portent sur les éléments suivants :

#### *I.3.2.1 Observations sensorielles*

- Nébulosité
- Visibilité
- Temps
- Phénomènes spéciaux

#### *I.3.2.2 Observations instrumentales*

- Température de l'air
- Température extrême
- Température du sol
- Précipitation
- Evaporation
- Humidité
- Insolation
- Pression
- Direction et vitesse du vent

Une station climatologique peut être chargée d'effectuer d'autres observations et mesures en cas de nécessité.

### **I.3.3 Observations aéronautiques (sensorielles et instrumentales)**

- Direction et vitesse du vent
- Visibilité
- Temps
- Nébulosité

- Pression atmosphérique
- Température

Les procédures de travail dans une station aéronautique seront conformes aux normes et pratiques recommandées par l'OACI.

#### I.3.4 Observations agrométéorologiques

L'ordre chronologique d'observation d'une station agrométéorologique est le suivant :

##### *I.3.4.1 Observations sensorielles*

- Nébulosité (quantité des nuages)
- Visibilité
- Temps
- Phénomènes spéciaux

##### *I.3.4.2 Observations instrumentales*

- Température de l'air
- Température extrême
- Températures du sol
- Précipitation
- Evaporation
- Humidité
- Insolation
- Pression
- Direction et vitesse du vent
- Humidité du sol (teneurs en différentes profondeurs)
- Turbulence de l'air dans une station agrométéorologie
- Hydrométéores et autres éléments du bilan hydrique
- Insolation et radiation
- Mesure des gradients des éléments météorologiques
- Observation biologique
- Observation phénologiques
- Observations concernant les rendements animaux et végétaux
- Observations des dommages causés par les maladies et parasites sur les animaux, les végétaux et les récoltes.

Les procédures de travail des stations agrométéorologiques sont semblables à celles des stations climatologiques ; en outre, il pourra y être exécuté des programmes spéciaux selon les besoins pour lesquels la station a été créée.

#### I.3.5 Observations pluviométriques

Le programme d'observations d'un poste pluviométrique est le suivant :

- Hauteur de la pluie tombée (en millimètres et dixièmes)

- Nombre de jours avec pluie
- Heure du début et de la fin de la pluie
- Direction de la provenance, intensité et dégâts causés par les phénomènes atmosphériques (pluie, orages, brouillards, rosée et grêle).

### I.3.6 Observations spéciales

Le programme d'observations d'une station spéciale dépend des buts pour lesquelles la station a été créée. Toutefois, à titre indicatif on peut citer

- Mesure de rayonnement
- Microclimat
- Pollution atmosphérique
- Mesure des parasites atmosphériques
- Détection par radar des nuages et des hydrométéores
- Mesure de l'électricité atmosphérique
- Mesure de l'ozone

## I.4 Heures des observations

### I.4.1 Heures d'observations synoptiques de surface

Pour les observations synoptiques de surface, l'heure réelle d'observation est celle de la lecture du baromètre.

L'heure officielle d'observation est l'heure TU. Pour des raisons pratiques, il est d'usage, en météorologie, d'utiliser le cycle de 24 heures. Minuit où le commencement d'un jour est signalé par 0000, pour six heures du matin, on indique 0600, alors que 1800 correspond à six heures de l'après-midi. On notera que l'on n'utilise pas 2400 pour indiquer minuit.

Les observations synoptiques devraient être faites dans le monde entier en se référant au temps universel.

On admet que le temps universel correspond au temps local moyen le long du méridien de Greenwich (c'est-à-dire à la longitude 0°) qui est également connu sous le nom de temps moyen de Greenwich (GMT). Les observations synoptiques en surface devraient être effectuées à 0000, 0600, 1200 et 1800 GMT, avec des observations intermédiaires à 0300, 0900, 1500 et 2100 GMT. C'est donc à ces heures précises que devrait se faire la lecture du baromètre. L'observation des éléments autres que la pression atmosphérique devrait se faire au cours des dix minutes que précèdent ces heures.

### I.4.2 Heures des observations climatologiques

Les observations climatologiques doivent être effectuées à 0700, 1200 et 1700 GMT. L'observation de la pression atmosphérique devrait donc se faire à ces heures.

## I.5 Ordre chronologique des observations météorologiques

L'observation du temps doit être assurée en permanence dans toute la durée d'ouverture de la station.

Cette continuité est particulièrement nécessaire pour pouvoir donner des renseignements relatifs aux phénomènes significatifs.

Le plan typique de l'ordre des observations météorologiques effectuées selon le programme complet est indiqué dans le **Tableau 1**.

Certaines stations soumises à un programme d'observation réduit sont dotées d'appareils enregistreurs. Ces données doivent être inscrites à l'encre rouge dans les documents quotidiens rédigés au niveau de la station. Les mesures de l'évaporation seront effectuées à 07h20 et 17h20.

Heures	Minutes	Appareils	observations
05	20 à 39	Contrôle de l'état de fonctionnement des appareils météorologiques de surface et du sol	
	40 à 49	Thermomètre au sol et dans le sol	Lecture des thermomètres
06	H <sup>(2)</sup>		
	30 à 35	Héliographe	Remise de la bande d'enregistrement
07	20	Bac d'évaporation	Mesure de l'évaporation
08	10 à 20	Enregistreurs	Changement des diagrammes : tous les lundis pour les hebdomadaires, tous les jours pour le quotidien
09 <sup>(1)</sup>	H <sup>(2)</sup>		
11	20 à 39	Contrôle de l'état de fonctionnement des appareils météorologiques de surface et du sol	
	40 à 49	Thermomètre au sol et dans le sol	Lecture des thermomètres
12	H <sup>(2)</sup>		
15 <sup>(1)</sup>	H <sup>(2)</sup>		
17	20	Bac d'évaporation	Mesure de l'évaporation
	21 à 39	Contrôle de l'état de fonctionnement des appareils météorologiques de surface et du sol	
	40 à 49	Thermomètre au sol et dans le sol	Lecture des thermomètres
18	H <sup>(2)</sup>		
19	30 à 35	Héliographe	Retrait de la bande du jour
21 <sup>(1)</sup>	H <sup>(2)</sup>		
00	H <sup>(2)</sup>		
03 <sup>(1)</sup>	H <sup>(2)</sup>		

*Tableau 1 : Tableau de l'ordre chronologique des observations météorologique*

Note : (1) signifie heure d'observation synoptique secondaire  
(2) se rapporte au **Tableau 2** suivant

Heure	minutes	appareils	observations
H (06, 09, 12, 15, 18, 21, 00, 03)	<b>-10</b>	<b>Tour d'horizon (nuages, visibilité et phénomènes)</b>	
	<b>-8</b>	<b>Abri météo</b>	<b>Lecture des appareils sous abri</b>
	<b>-6</b>	<b>Pluviomètre et pluviographe</b>	<b>lecture</b>
	<b>-4</b>	<b>Contrôle du tour d'horizon</b>	
	<b>-2</b>	<b>Appareil à vent et barographe</b>	<b>Exécution des observations</b>
	<b>00</b>	<b>Baromètre</b>	<b>lecture</b>

*Tableau 2 : Tableau des tâches de l'observateur météorologique*

## I.6 Emplacement et exposition des instruments

### I.6.1 Site de la station

La manière dont les instruments sont exposés constitue un facteur déterminant dans la mesure de certains éléments météorologiques.

L'emplacement d'une station météorologique exige les conditions suivantes :

Le site doit être choisi de telle façon que les valeurs observées et mesurées représentent météorologiquement le temps réel et le climat considérés pendant une longue période.

- 1) Le site doit être suffisamment dégagé pour permettre une libre circulation de l'air et une bonne détermination du tour d'horizon.
- 2) La nature du sol de la station doit être représentative de celle des sols de la région.
- 3) Autant que possible, la station ne devrait pas être située sur des pentes raides, de crêtes, des falaises ou dans des dépressions, ni à leur proximité.
- 4) Aucune construction ne devrait être exigée dans les environs immédiats de la station.
- 5) Le choix de l'emplacement d'une station doit être fait par des spécialistes
- 6) Aucun instrument météorologique ne doit être transféré sans avis préalable écrit de la Direction Générale du service.
- 7) Le chef d'une station météorologique est tenu d'informer la Direction Générale à temps de toutes activités tendant à modifier la représentativité des conditions locales de sa station.

### I.6.2 Parc d'instruments

L'emplacement et l'exposition des instruments météorologiques dans une station sont conçus avec attention.



Pour que l'exposition des instruments dans chaque station de même catégorie soit similaire, il est nécessaire d'adopter un plan de disposition des instruments que soit valable pour toutes les stations d'un même type.

Le parc d'instruments de la station météorologique doit avoir les dimensions 26m×26m. La porte d'accès doit se trouver du côté nord. La position des instruments et aussi les distances horizontales entre eux doivent respecter des règles strictes.

La clôture sera constituée de préférence par un grillage. Le terrain doit être bien nivelé et recouvert d'un gazon entretenu. Les allées doivent être couvertes par du gravier.

On peut trouver différents instruments dans un parc météorologique synoptique et climatologique :

- 1) Pylône de girouette et anémomètre à distance
- 2) Pylône d'anémomètre de réserve
- 3) Abri météo pour psychromètre et thermomètre maxi et mini
- 4) Abri météo pour enregistreurs (thermo-hygro)
- 5) Pluviomètre association
- 6) Pluviomètre enregistreur
- 7) Théodolite
- 8) Héliographe
- 9) Evaporomètre classe A
- 10) Actinomètre ou actinographe
- 11) Drossomètre
- 12) Thermomètre dans le sol nu
- 13) Thermomètre dans le sol gazonneux
- 14) Thermomètre minimum au sol

## **I.7 Aspects spécifiques des mesures météorologiques [1]**

La physique a pour objet la mesure de variables physiques. Afin de découvrir la valeur d'une quantité avec une précision prédéterminée, il faut que soient remplies un certain nombre de conditions telles que l'exposition de l'instrument à la variable mesurée, l'étalonnage de l'instrument, et l'habileté professionnelle de l'opérateur.

Lorsqu'on mesure des variables météorologiques, dont la plupart sont essentiellement des variables physiques, les problèmes sont rendus encore plus compliqués du fait des aspects spécifiques des phénomènes météorologiques. Les processus atmosphériques se déroulent sur les échelles de temps et d'espace complètement différents de celles des phénomènes observés en laboratoire. Pour étudier des phénomènes se déroulant sur d'immenses zones du globe, il est hors de question de n'avoir qu'un seul point de mesure ; c'est pourquoi les météorologistes utilisent des systèmes d'observation météorologique et des réseaux de stations. De façon à obtenir des résultats comparables à partir des points de mesure du réseau météorologique, en plus d'un programme d'observation unifié, une certaine uniformité dans les paramètres des instruments de mesure est nécessaire. Il faut employer des instruments de mesure ayant une

précision et des caractéristiques opérationnelles similaires et procéder à la maintenance et à l'étalonnage selon une méthode uniforme.

Il faut tenir compte de l'étendue dynamique des variables météorologiques. Les instruments doivent fonctionner aussi bien à l'équateur qu'aux pôles.

Les heures d'observations devront être les mêmes sur tout le globe, utilisant la même échelle de temps, celle du temps universel TU.

On voit donc qu'il est nécessaire d'observer strictement les règles énoncées dans les différents guides de l'OMM ayant trait aux observations météorologiques et aux instruments

## **I.8 Mesures météorologiques directes et indirectes**

On effectue des mesures météorologiques directes au moyen d'instruments de mesure ayant leur capteur au point de mesure. Les instruments eux-mêmes peuvent être appelés instruments météorologiques directs ; ils comprennent les plus conventionnels des instruments météorologiques de surface, comme le cas des appareils conçus dans ce projet de mémoire, tels que thermomètre, anémomètre, girouette, hygromètre, baromètre, etc. Les instruments de météorologiques directs peuvent être employés pour la mesure à distance des variables météorologiques ; on dit alors qu'il s'agit d'instruments de lecture à distance.

Si l'on utilise une liaison radio comme relais entre la sortie des données météorologiques et l'observateur, ces instruments sont appelées instruments de télémessure.

On effectue des mesures météorologiques indirectes en utilisant soit les ondes acoustiques, soit les ondes électromagnétique. Les instruments employés, dénommés instruments météorologiques indirects, mesurent généralement à distance. On peut donner par exemple l'instrument de mesure indirecte (ou plutôt de systèmes de mesure indirecte) les radars acoustiques, les lidars (radars à faisceaux lumineux), les radiomètres à infrarouge et les radars à impulsions et à effet Doppler.

## **I.9 Diagramme d'un instrument météorologique**

Tous les instruments de mesure directs, selon la façon dont ils indiquent l'information, peuvent être classés dans les deux catégories suivantes :

### **I.9.1 Instruments de mesure analogique**

Les instruments de mesure analogiques donnent une indication de la quantité mesurée sous une forme analogique continue (analogique = sortie sous forme similaire- ou analogue – à l'entrée). La valeur de la variable mesurée se lit sur l'échelle convenablement graduée d'un indicateur.

On peut donner une représentation pratique d'un instrument de mesure météorologique analogique à l'aide d'un diagramme montrant que l'instrument est formé de trois blocs fonctionnels, comme sur la figure suivante : le capteur, le convertisseur de signal, l'indicateur/enregistreur. (*Figure 1*)



Figure 1 : Diagramme d'un instrument de mesure analogique

Un anémomètre utilisant un moteur à courant continu est analogique type. En effet, si l'on applique une tension continue à ses bornes, le moteur rentre en rotation. Mais si l'on fait tourner son axe avec une force mécanique, la force du vent par exemple, on obtient une tension à ses bornes dont la valeur est directement liée à celle de la vitesse de rotation de son axe, et donc du vent. (**Figure 2**)

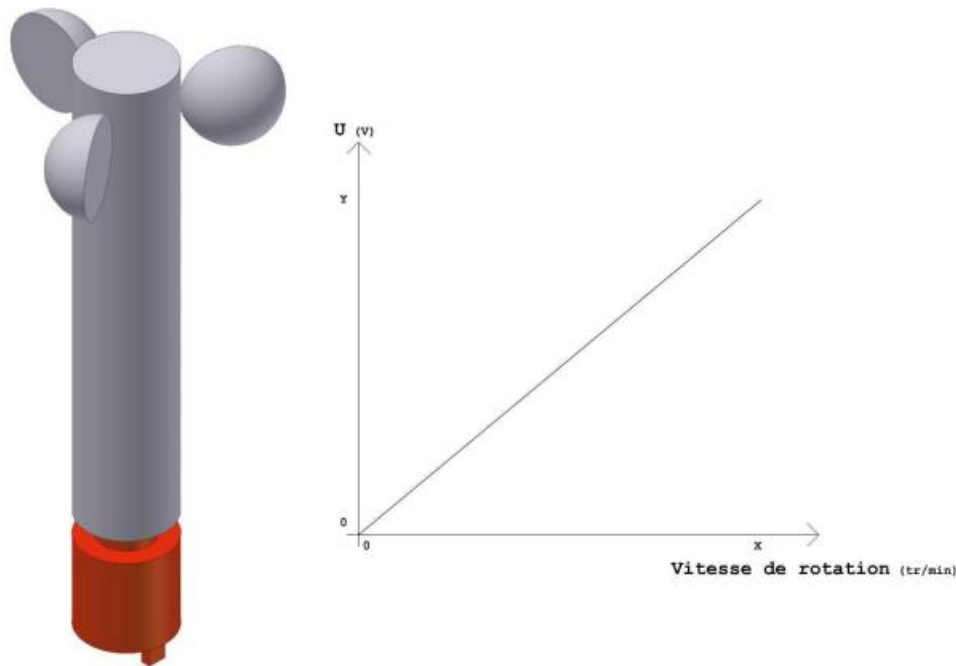


Figure 2: Schéma de fonctionnement de l'anémomètre analogique

L'opérateur lit la valeur de la quantité à mesurée avec une précision dépendant de la graduation de l'échelle de l'indicateur. En pratique, on considère que la précision de lecture est limitée à une demi-division de d'échelle, bien qu'un opérateur expérimenté soit capable de lire l'échelle de l'instrument avec une plus grande précision.

### I.9.2 Instruments de mesure numérique

L'instrument météorologique numérique affiche la valeur de la quantité mesurée sous une forme numérique discrète. La valeur numérique de la variable mesurée est indiquée sur un affichage numérique. Elle peut être imprimée sur une bande de papier par une imprimante

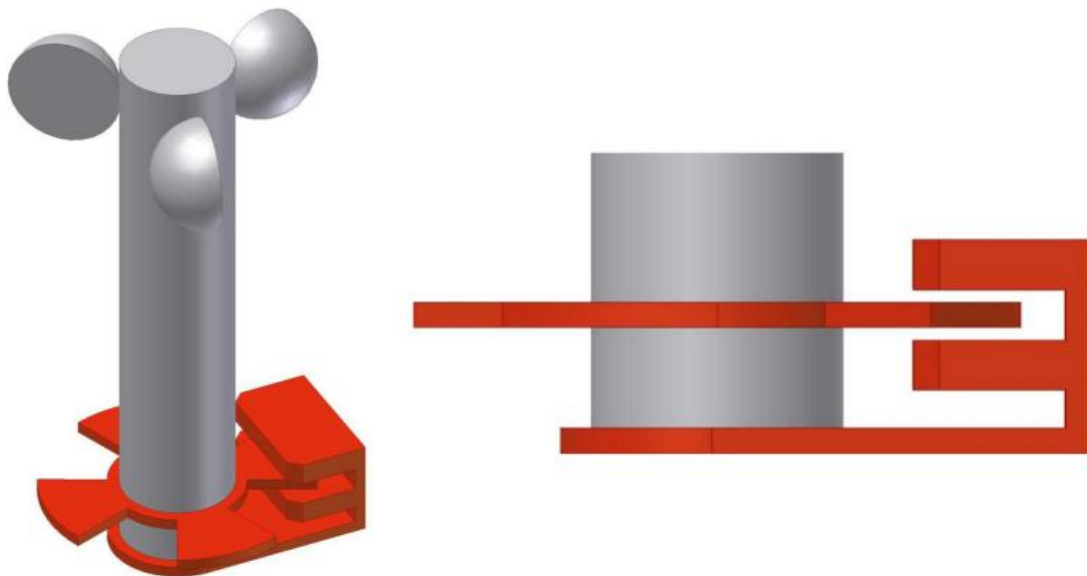
numérique. Dans la représentation en diagramme d'un instrument météorologique numérique (**Figure 3**), l'instrument se compose de trois blocs fonctionnels : le capteur, le convertisseur de signal analogique numérique, l'affichage (ou l'imprimante).



*Figure 3 : Diagramme d'un instrument de mesure numérique*

Le principe de fonctionnement de l'anémomètre numérique diffère complètement de celui de l'anémomètre analogique puisqu'il s'agit cette fois-ci de compter des impulsions, plus ou moins nombreuses en un laps de temps donné proportionnelles à la vitesse du vent.

Pour cela il faut utiliser un capteur optique de type fourche, ainsi qu'un disque comportant plusieurs zones opaques et transparentes, ce qui fournira autant d'impulsions par tour que nécessaire. Il suffit alors de récupérer en sortie le signal et de compter le nombre d'impulsions (**Figure 4**). Nous allons adopter ce procédé pour le capteur de notre futur anémomètre.



*Figure 4 : Schéma de fonctionnement de l'anémomètre numérique*

## I.10 Comportement dynamique des instruments de mesure - Instrument de mesure de premier ordre - Instrument de mesure de second ordre - Caractéristiques général des instruments météorologiques [16]

### I.10.1 Temps de stabilisation

La réponse des instruments météorologiques à un changement soudain de la variable mesurée n'est pas instantanée. D'une façon générale, un instrument a besoin d'un certain « *temps de stabilisation* » pour indiquer la nouvelle valeur d'une variable météorologique qui a subi une variation d'échelon. Cette variation est connue sous le nom de « *fonction échelon* » et la réponse de l'instrument de mesure durant son temps de stabilisation est appelée sa « *réponse transitoire* ».

Selon leur comportement dynamique et leur réponse transitoire, les instruments météorologiques rentrent dans deux catégories :

- ✚ Les instruments (ou systèmes) de mesure de premier ordre
- ✚ Et les instruments (ou systèmes) de mesure de second ordre.

#### Les instruments (ou systèmes) de mesure de premier ordre

Les instruments de mesure du premier ordre, lorsqu'ils sont soumis à une fonction échelon, voient leur sortie (indication de la variable mesurée) croître vers la valeur de la fonction appliquée, comme une fonction de l'entrée (la valeur de la fonction appliquée) et de la dérivée première de la sortie (la vitesse de variation de la sortie). La traduction simple en langage mathématique s'écrit

$$S = f\left(E, \frac{dS}{dt}\right)$$

La réponse transitoire de cette catégorie d'instruments est illustrée à la (**Figure 5**) : la sortie de l'instrument tend vers la valeur de la fonction appliquée en suivant une loi exponentielle – rapidement au départ, mais ralentissant graduellement, elle s'approche asymptotiquement de la valeur de la fonction appliquée. La coordonnée de temps est graduée en unités de constante de temps, une constante de temps étant le temps nécessaire pour que la sortie atteigne 63% de la valeur de la fonction appliquée.

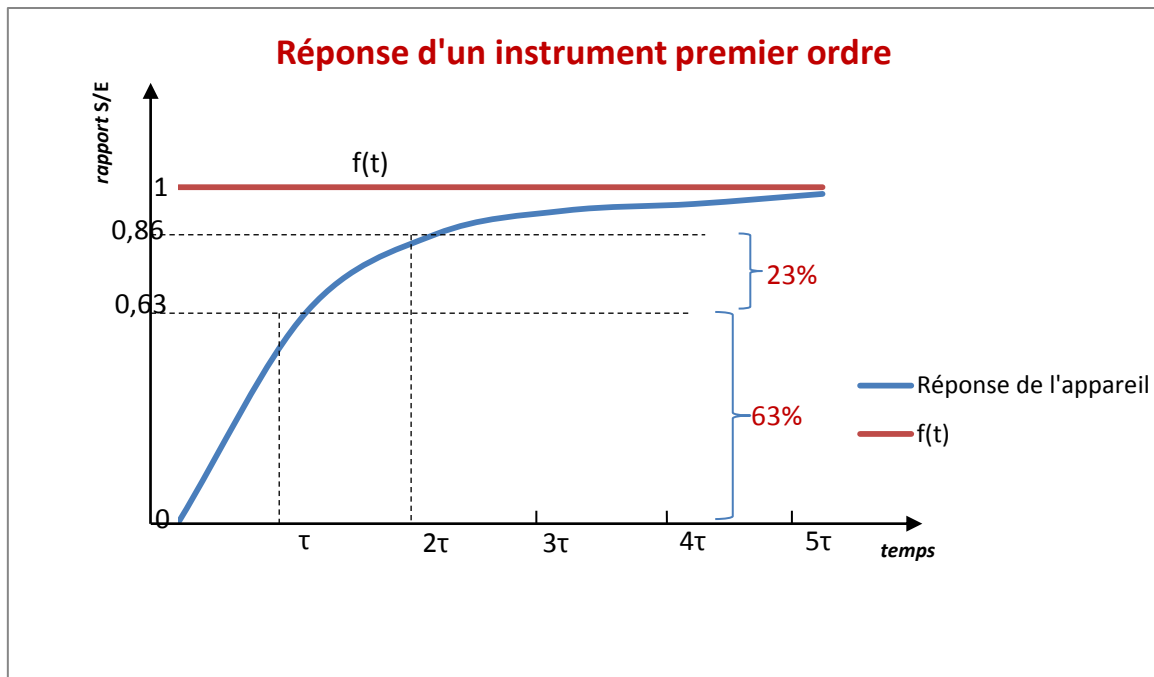


Figure 5 : Réponse d'un instrument météorologique du premier ordre

#### Les instruments (ou systèmes) de mesure de second ordre

Dans les instruments appartenant à la catégorie des systèmes de mesure du second ordre, la sortie tend vers la valeur de la fonction appliquée, proportionnellement à l'entrée et aux dérivées première et seconde de la sortie (vitesse de variation et accélération). Mathématiquement, cela se traduit comme suit :

$$S = f\left(E, \frac{dS}{dt}, \frac{d^2S}{dt^2}\right)$$

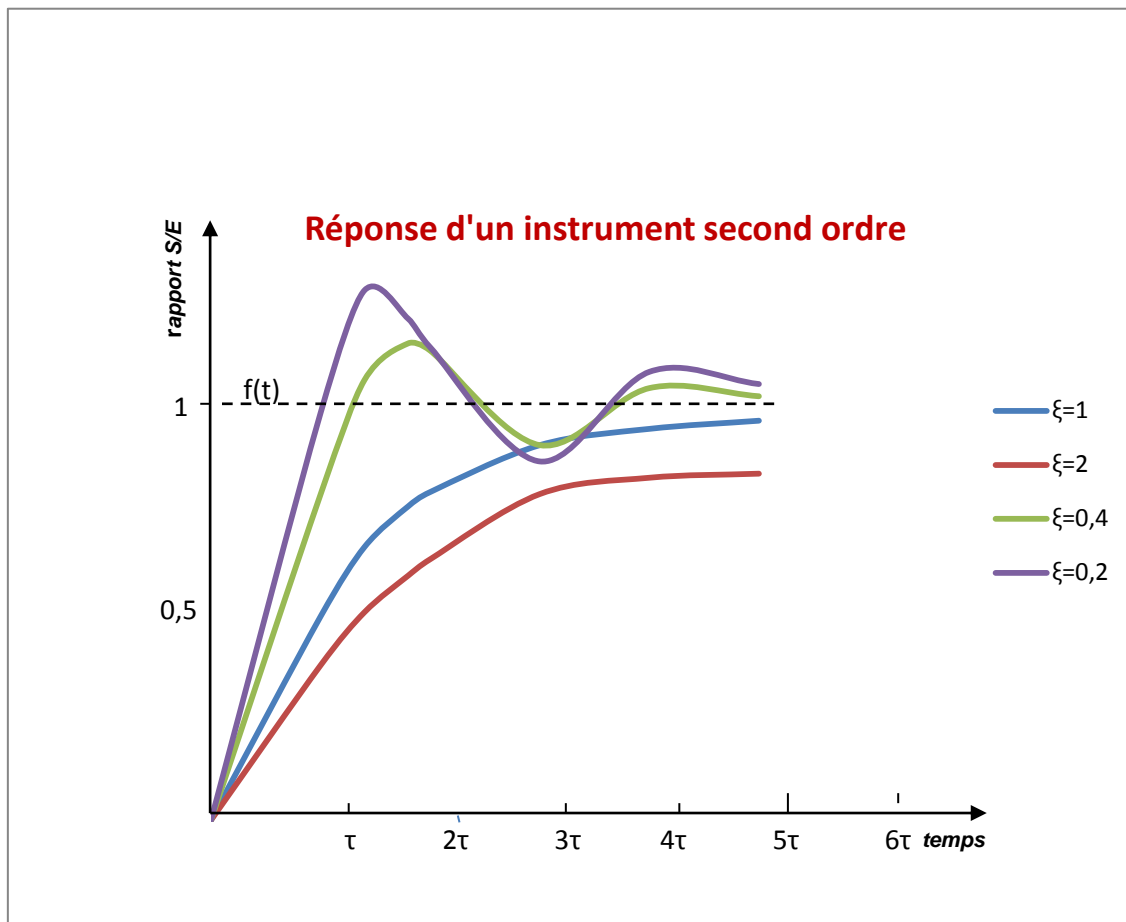
Avec de tels systèmes de mesure, il existe une tendance à la présence d'oscillations amorties autour de la valeur finale de la fonction (**Figure 6**). La fréquence et l'amplitude de ces oscillations sont fonction de l'amortissement du système de mesure. Le comportement dynamique des systèmes du second ordre est décrit par une entité appelée le « rapport d'amortissement »  $\xi$  qui est le rapport de l'amortissement réel présent dans le système à l'amortissement critique (qui ne produit aucun dépassement).

$$\xi = \frac{\text{amortissement réel}}{\text{amortissement critique}}$$

Si  $\xi < 1$ , le système tend à osciller avec une amplitude décroissante avec le temps quand il est soumis à une fonction échelon.

Si  $\xi = 1$ , le système se comporte comme les systèmes du premier ordre : il y a réponse apériodique.

Pour  $\xi > 1$ , le système est suramorti et a une réponse apériodique, lente, dépendant de la valeur réelle du rapport d'amortissement.



*Figure 6 : Réponse d'un instrument de mesure du second ordre*

L'amortissement de système mécanique est dû au frottement. La réponse transitoire du second ordre peut avoir un problème dans le cas de certains instruments météorologiques, par exemple les girouettes. Un amortissement proche de la valeur critique accroît la précision de la girouette.

On a souligné le temps de stabilisation des instruments météorologiques, à cause de son importance dans la mesure. Les instruments possèdent cependant d'autres aspects et caractéristiques importants que nous allons brièvement présenter et définir ci-dessous :

### I.10.2 La précision

Un instrument est précis si sa réponse est compatible avec l'échelle de sa référence d'étalonnage et si la réponse est reproductible. Plus un instrument est précis, plus les résultats des mesures sont proches de la « vraie valeur » de la quantité mesurée.

### I.10.3 La sensibilité

On peut dire que, par définition, la sensibilité d'un instrument de mesure est le rapport existant entre la variation de la sortie à la variation de l'entrée (la variable mesurée). Il y a des qualités requises déterminées selon les différentes catégories d'instruments météorologiques. Un instrument ayant une sensibilité inférieure à celle requise perdra un détail important de la mesure, alors qu'un instrument hypersensible enregistrera un détail inutile (par exemple un barographe hypersensible enregistrera des fluctuations de pression causées par l'ouverture ou la fermeture d'une porte).

### I.10.4 Spécificité de réponse

Un instrument a une bonne spécificité de réponse s'il est insensible à toute variable exceptée celle pour la mesure de laquelle il a été conçu. S'il est protégé contre les précipitations, le thermomètre à mercure est virtuellement insensible aux variations de pression ou d'humidité de l'air. Sa réponse spécifique est due aux variations de température, alors que les lectures de pression sur le barographe anéroïde sont affectées par les variations de la température. Ce dernier instrument a une spécificité de réponse inférieure à celle du thermomètre sous abri.

### I.10.5 Linéarité de réponse :

Un instrument a une réponse linéaire si sa sortie est une fonction linéaire de l'entrée (la courbe d'étalonnage est une ligne droite). La linéarité est une caractéristique souhaitable de l'instrument météorologique, car elle permet une interpolation rapide et précise entre deux divisions adjacentes de l'échelle.

Les exigences concernant la précision, la sensibilité, la linéarité de réponse et la réponse transitoire des instruments météorologiques font l'objet d'une normalisation dans le but d'obtenir des mesures comparables avec des instruments de type et de fabrication différents.

### I.10.6 Régularité

Les instruments utilisés de façon opérationnelle (les instruments du réseau météorologique) doivent être conçus de manière à être durables et pratiques en exploitation.



### I.10.7 La simplicité et la commodité d'utilisation et d'entretien

Elles sont importantes, étant donné que la plupart des instruments météorologiques sont utilisés continuellement, année après année, et qu'ils peuvent être situés loin d'un bon atelier de réparation.

### I.10.8 La solidité de construction

Une construction robuste est particulièrement souhaitable pour les instruments qui sont entièrement ou partiellement exposés à toutes les conditions de temps.

### I.10.9 La fiabilité

La fiabilité est un facteur important. Les instruments météorologiques devront être conçus de façon à fonctionner avec un *taux de défaillance* aussi faible possible. Une méthode statistique donne une représentation intéressante du taux de défaillance  $\lambda$ , qui est un paramètre de la fiabilité d'un instrument. La courbe représentant le taux de défaillance en fonction du temps, sur la **Figure 7**, révèle trois périodes bien définies ; dans chacune d'elles le taux de défaillance,  $\lambda$ , a une caractéristique différente :

- (I) : période des « défauts innés »,  $\lambda$  décroît avec le temps ;
- (II) : période à taux de défaillance constant ;
- (III) : période de défaillance due à l'usure,  $\lambda$  croît avec le temps.

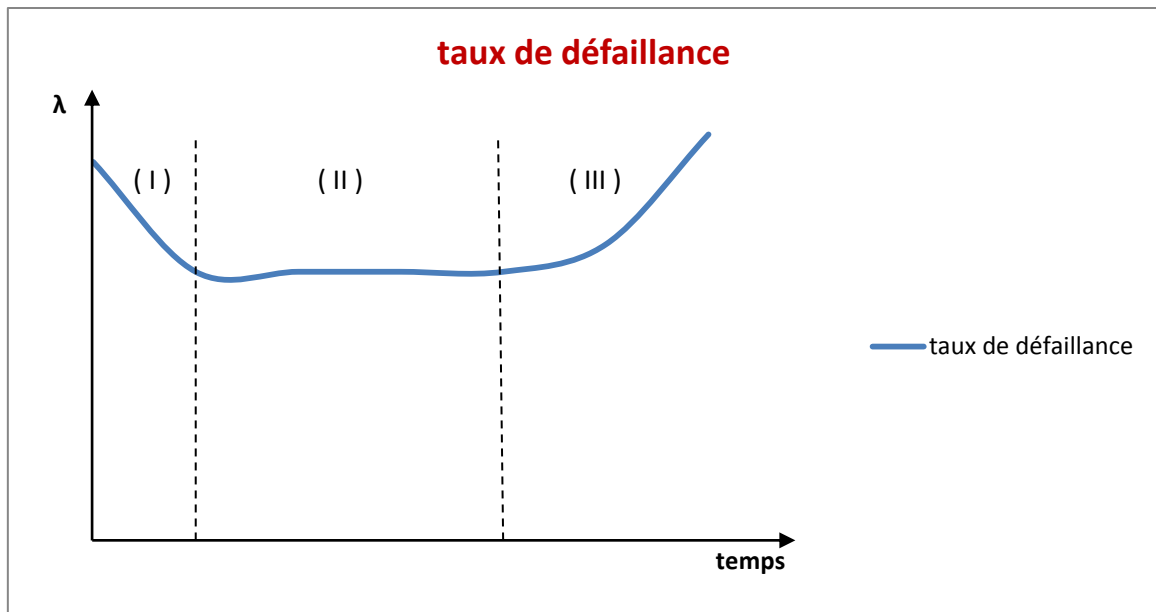
L'intervalle (II) est la durée de vie opérationnelle de l'instrument ou de l'équipement. La fiabilité  $R$  de l'instrument, tant que l'on se trouve dans la période (II), s'exprime en fonction du taux de défaillance et du temps par la relation :

$$R(t) = e^{-\lambda t}$$

Le temps moyen entre défaillance (MTBF) est aussi une fonction de  $\lambda$  :

$$MTBF = \frac{1}{\lambda}$$

Les paramètres et les courbes de fiabilité d'un équipement complexe peuvent être obtenus en fonction des paramètres correspondants des composants de l'équipement.



*Figure 7 : Taux de défaillance en fonction du temps*

## **PARTIE II**

### **CONCEPTION ET SIMULATION DU SYSTEME**

## II.1 Contexte du mémoire

A Madagascar, comme dans tous les pays sous-développés, l'accès aux instruments de mesures météorologiques sophistiqués est presque impossible vu leurs prix très élevés. Par conséquent, on utilise peu d'instruments pour le relevé des grandeurs météorologiques. Les observateurs ne font que les estimer. Les données d'observations varient donc en fonction du personnel, de sa santé et même de son sentiment. Le manque d'entretien et le non-responsabilité du Gouvernement aggravent également le problème.

Notre projet essaie de résoudre ces problèmes par la réalisation d'un système d'instruments météorologiques numériques comportant un *anémomètre*, une *girouette* et un *thermomètre*. La carte d'acquisition est basée autour d'un microcontrôleur PIC 18F4550 ayant une sortie USB. Pourquoi le 18F4550? Parce qu'il est un PIC accessible à Madagascar et il répond bien à notre besoin. Il est un composant intellectuel.

## II.2 Cahier des charges

Ce projet vise à réaliser un système d'instruments incluant un anémomètre, une girouette et un thermomètre :

### L'anémomètre

- Il est à capteur rotatif : on va utiliser trois coupelles de Robinson.
- Il est censé de mesurer une vitesse du vent qui peut entraîner une révolution de 10 tours par seconde.
- Le transducteur est à découpeur optique
- Il doit avoir un affichage numérique à deux chiffres pour la visualisation (vitesse en  $m.s^{-1}$ )

### La girouette

- Il est à codeur optique comportant un disque de Gray
- Il peut indiquer 16 directions du rose des vents de 0 à 360°

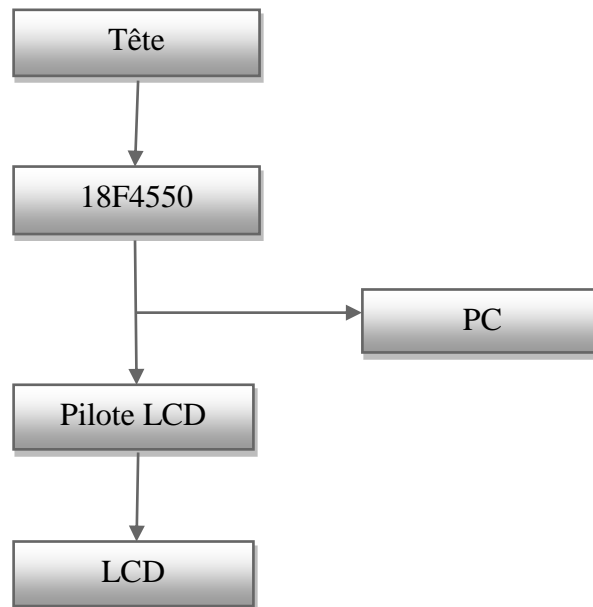
### Le thermomètre

- Il est à capteur électronique : on va utiliser LM35 de Texas Instruments
- Il est censé de mesurer une gamme de température de -5 à 70°C

Ce système doit :

- Avoir un affichage numérique LCD pour la visualisation à l'œil nu
- Avoir un système d'acquisition de données tous les 15s via le port **USB** 2.0 d'un microordinateur.
- Avoir une interface utilisateur

Le principe de fonctionnement du système est montré par la **Figure 8**.



*Figure 8 : Schema fonctionnel du système*

## II.3 L'anémomètre

### II.3.1 La force du vent – Caractéristiques – Mesure. [16]

Rappelons que le vent est une grandeur vectorielle possédant une direction et une grandeur scalaire dénommée vitesse.

Le vent est mesuré en dehors de la zone de turbulence due aux obstacles à une hauteur de 10m au-dessus de sol. La vitesse et la direction relevées lors d'une observation synoptique représente la vitesse et la direction moyennes déterminées 10min avant l'observation.

La vitesse du vent se décompose en trois éléments dont deux se trouvent dans un plan parallèle à la surface terrestre, le troisième étant perpendiculaire à ce plan. Dans la plupart des problèmes de météorologie opérationnelle, la composante verticale n'est pas prise en compte, et le vent de surface est considéré comme une grandeur vectorielle à deux dimensions.

La vitesse du vent est sujette à des variations de période et d'amplitude, du fait que l'écoulement d'air dénommé vent n'est pas laminaire. Le vent à la surface de la Terre est un flux turbulent, comportant des turbulences de tailles et paramètres physiques différents se déplaçant avec le flux.

Le relief terrestre est la cause principale de la structure turbulente du vent, laquelle se manifeste sous forme de rafales de vent, c'est-à-dire par des fluctuations des paramètres du vent de surface.

Dans la plupart des cas, la vitesse moyenne du vent est égale à la moyenne des valeurs instantanées relevées sur un intervalle de 10 minutes.

Grace au facteur G, facteur d'intensité des rafales, la variabilité de la vitesse du vent peut s'écrire :

$$G = \frac{v_{max} - v_{min}}{v_{moy}}$$

Où :

$v_{max}$  = vitesse maximale du vent ;

$v_{min}$  = vitesse minimale du vent ;

$v_{moy}$  = vitesse moyenne du vent ;

Ces trois grandeurs scalaires se rapportent à un intervalle de temps de 10 minutes.

La vitesse du vent s'exprime dans l'une quelconque des unités suivantes :

- a) Nœuds (milles marins) par heure, en abrégé : kt
- b) Mètre par seconde :  $m s^{-1}$
- c) Kilomètre par heure :  $k h^{-1}$
- d) Mille par heure : m.p.h
- e) Pieds par seconde :  $ft s^{-1}$

Dans les messages météorologiques internationaux la vitesse du vent est indiquée en mètre par seconde ( $m s^{-1}$ ) ou en nœuds (kt).

Le **Tableau 3** donne la relation existant entre les différentes unités de vitesse du vent

kt	$m s^{-1}$	$Km h^{-1}$	m.p.h	$Ft s^{-1}$
<b>1,000</b>	0,515	1,853	1,152	1,689
<b>1,943</b>	1,000	3,600	2,237	3,281
<b>0,540</b>	0,278	1,000	0,621	0,911
<b>0,868</b>	0,447	1,609	1,000	1,467
<b>0,592</b>	0,305	1,097	0,682	1,000

*Tableau 3 : Relation entre les différentes unités de vitesse du vent*

Beaufort	Grandeur	Description	Vitesses limites		
			<i>Nœuds</i>	<i>km h<sup>-1</sup></i>	<i>m s<sup>-1</sup></i>
<b>0</b>		Calme. La fumée s'élève verticalement	< 1	< 2	0 à 0,2
<b>1</b>	Très légère brise	La direction du vent est indiquée par les colonnes de fumée et non par les girouettes	1-3	2-6	0,3-1,5
<b>2</b>	Légère brise	Le vent est sensible sur le visage, les feuilles frémissent, les girouettes sont mises en mouvement	4-6	7-12	1,6-3,3
<b>3</b>	Petite brise	Les feuilles et les petites plantes oscillent, le vent déploie les drapeaux légers	7-10	13-18	3,4-5,4
<b>4</b>	Jolie brise	Le vent soulève la poussière et les papiers, les petites branches sont agitées	11-16	20-30	5,5-7,5
<b>5</b>	Bonne brise	Les arbustes oscillent, de petites vagues se forment sur les eaux intérieures	17-21	31-39	8,0-10,7
<b>6</b>	Vent frais	Les grosses branches remuent, les fils téléphoniques vibrent et sifflent	22-27	40-50	10,8-13,8
<b>7</b>	Grand frais	Les arbres sont agités en entier, la marche contre le vent est pénible	28-33	51-61	13,9-17,1
<b>8</b>	Coup de vent	Le vent casse des branches; la marche contre le vent est en général impossible	34-40	62-74	17,2-20,7
<b>9</b>	Fort coup de vent	Le vent occasionne de légers dommages aux habitations (arrachement de cheminées)	41-47	75-87	20,8-24,4
<b>10</b>	Tempête	Rare sur la terre : arbre déracinés, dégâts importants aux habitations.	48-55	88-102	24,5-28,4
<b>11</b>	Violente tempête	Très rarement observé; s'accompagne de ravages étendus	56-63	103-117	28,5-32,6
<b>12</b>	Ouragan	Idem que 11.	> 64	>118	>32,7

*Tableau 4 : Echelle de Beaufort*

En complétant des unités indiquées dans le **Tableau 3**, la vitesse du vent s'exprime aussi en unité Beaufort. L'échelle Beaufort utilise des phénomènes naturels familiers en rapport avec la vitesse du vent.

Le **Tableau 4** montre l'échelle de Beaufort qui a trait à l'estimation de la vitesse du vent en surface.

Dans le cas général, on utilise un anémomètre pour mesurer la vitesse du vent. Il y a différents type d'anémomètre utilisé en météorologie mais notre étude se focalise sur le type d'anémomètre à coupelles.

### Calculs d'un anémomètre à coupelles

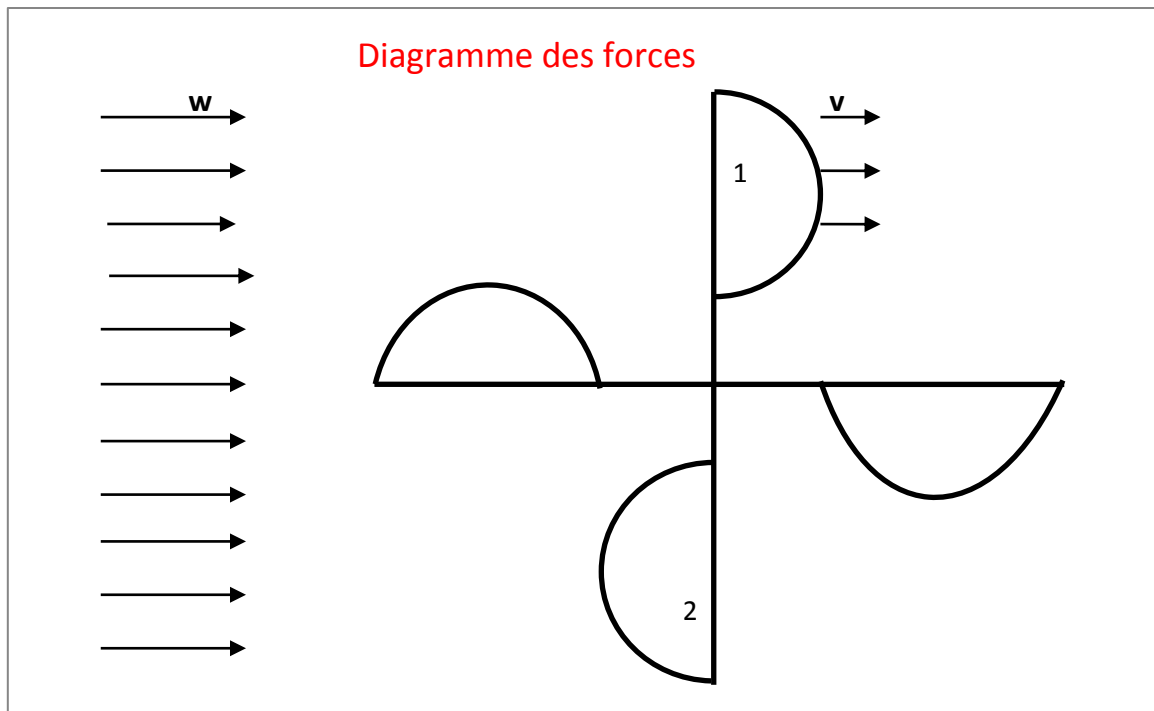
Notre étude se base sur un anémomètre à coupelles pour la mesure de la vitesse du vent.

L'anémomètre à coupelles (**Figure 9**) utilise ce qu'on appelle un moulinet. Trois coupelles en matière plastique sont installées soigneusement par rapport à un axe vertical à l'extrémité de trois bras émergeant du moyeu du moulinet. Comme le vent exerce une force plus grande sur la partie concave que sur la partie convexe des coupelles, l'écoulement de l'air va faire tourner le moulinet. Pour tout anémomètre à coupelles, il existe une valeur minimale de vitesse du vent à partir de laquelle le moulinet se met à tourner, selon le frottement au niveau des engrenages de l'axe du moulinet et les caractéristiques techniques de l'anémomètre.

La vitesse de rotation du moulinet ne dépend pas de la direction du vent ni de façon sensible de la densité de l'air. Robinson a conçu l'anémomètre à moulinet à partir de la théorie suivante :

La résultante des forces entraînant la rotation est due, comme cela a été mentionné ci-dessus, à la différence des valeurs des coefficients de résistance aérodynamique  $c_1$  et  $c_2$  d'une coupelle selon qu'elle expose au vent sa partie concave ou sa partie convexe.





*Figure 9 : Diagramme des forces d'un anémomètre à coupelles*

Dans le cas d'un moulinet à 4 coupelles (**Figure 9**) on a les relations suivantes :

$$F_1 = C_1 \rho S (w - v)^2 \quad (1)$$

$$F_2 = C_2 \rho S (w + v)^2 \quad (2)$$

Où :

$F_1$  = force qui s'exerce sur la coupelle n°1 ;

$F_2$  = force qui s'exerce sur la coupelle n°2 ;

$w$  = vitesse du vent ;

$v$  = vitesse de rotation de la coupelle ;

$\rho$  = densité de l'air ;

$S$  = section droite de la coupelle ;

$C_1$  = coefficient d'aérodynamisme de la face concave ;

$C_2$  = coefficient d'aérodynamisme de la face convexe ;

Si pour une vitesse de vent  $w$  la vitesse de rotation  $v$  est atteinte ( $F_1 = F_2$ ), les équations (1) et (2) donnent :

$$C_1 (w - v)^2 = C_2 (w + v)^2 \quad (3)$$

Soit :

$$\frac{C_1}{C_2} = \frac{(w+v)^2}{(w-v)^2} \quad (4)$$

Le rapport  $C_1/C_2$  mesuré expérimentalement avec une précision de 2% donne, après substitution dans l'équation (4) :

$$w = w_0 + Av + Bv^2 \quad (\text{en ms}^{-1})$$

Où :

$w_0$  = valeur minimale du vent nécessaire à la mise en mouvement de la coupelle ;

$v$  = vitesse de rotation du moulinet ;

$A$  = coefficient fonction du modèle de moulinet utilisé ;

$B$  = caractéristique de construction,  $B = 0,0001A$  ;

Dans le cas de vent bien établi la coupelle a de bonnes performances pour des valeurs du vent comprises entre 0,5 et 60 ms<sup>-1</sup>. Dans le cas de rafales de vent, l'anémomètre a tendance à surévaluer la valeur moyenne de la vitesse du vent. Ceci provient du fait que le moulinet, à cause de son inertie, a une plus grande accélération quand la vitesse augmente et une décélération moins grande quand cette vitesse diminue.

Le comportement de moulinet dans les rafales a été étudié de façon approfondie. Schrenk a suggéré l'existence d'un paramètre sans dimension  $K$  pour tenir compte de la surestimation de la vitesse du vent par un capteur à coupelles :

$$K = \frac{0,55\rho R^2 r^2 T v}{I}$$

Où :

$T$  = période de variation de la vitesse du vent

$V$  = vitesse moyenne du vent

$\rho$  = densité de l'air

$R$  = rayon du cercle décrit par le centre des coupelles

$r$  = rayon des coupelles

$I$  = moment d'inertie des parties mobiles du moulinet ;

La surestimation du vent est proportionnelle à  $1/K$ .

La relation précédente montre d'évidence qu'une très surestimation de la force moyenne du vent dépend tout autant des caractéristiques de l'instrument que du vent. On a démontré expérimentalement que la réponse du moulinet à une brusque saute de vent n'est pas exponentielle. En général, la réponse est d'autant plus rapide que la vitesse du vent est élevée. Les causes d'erreurs sont les suivantes :

- Augmentation graduelle du frottement au niveau des paliers avec le temps.
- Variation des propriétés aérodynamiques et du poids par suite de givrage ;
- Déformation du moulinet.

### II.3.2 Principe de fonctionnement de l'anémomètre

Dans ce projet l'anémomètre comporte une tête qui produit de train d'impulsions dont la fréquence est proportionnelle à la vitesse de vent.

On supposera que la gamme des vitesses de vent à mesurer entraîne une rotation du rotor de l'anémomètre dans la gamme de 0 à 10 révolutions par seconde.

Chaque révolution produit 4 impulsions électriques de sorte que la fréquence du train d'impulsions soit 0 à 40 Hertz.

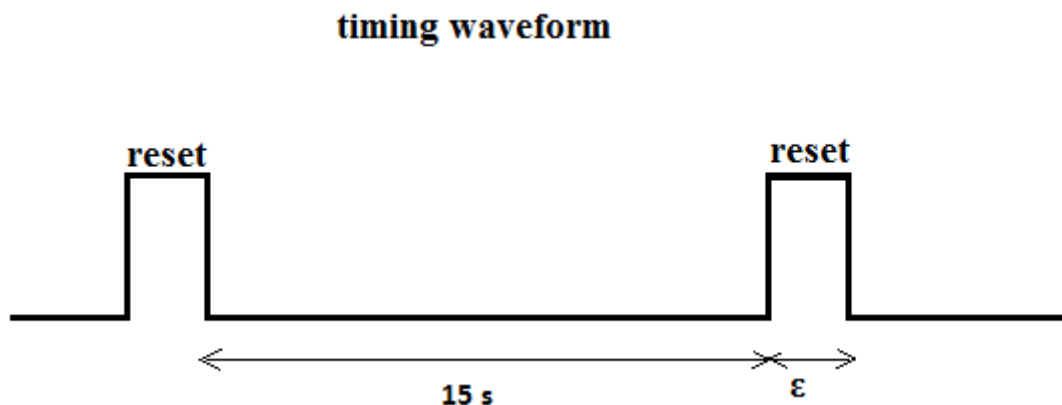
Afin de réduire la variation courte en termes de vitesse du vent, la valeur à afficher est moyennée dans 15 secondes.

Faire la moyenne implique tout simplement de compter les impulsions pour 15s, et de montrer le résultat après multiplication par le coefficient anémométrique qui dépend de la géométrie de l'anémomètre et la fonction de l'étalonnage.

L'afficheur de vitesse se compose de deux chiffres décimaux et ne peut pas donc indiquer 0 à 600 ( $4 \times 15 \times 10$ ) directement. Un traitement préalable est nécessaire pour ramener le compte de 0 à  $100 \text{ m.s}^{-1}$  qui peut alors être affiché par deux chiffres (00-99). On pourrait produire une seule impulsion par révolution du rotor de l'anémomètre, mais cette approche présente un certain inconvénient.

L'affichage de la vitesse de vent mesuré pendant une période de 15s doit continuer tandis que des comptes sont accumulés dans la période suivante.

La forme d'onde qui est employée pour remettre à zéro le compteur, pour mettre à jour l'affichage et les données envoyées vers le PC, est produite au sein du microcontrôleur 18F4550 en utilisant un quartz de 20Mhz. La valeur affichée est renouvelée à chaque 15,0077 secondes et indique la vitesse du vent moyenne dans 15 secondes selon le chronogramme de la *Figure 10*.



*Figure 10 : Chronogramme de mise à jour des données anémométrique*

**Note de figure :** La durée  $\epsilon$  est négligeable, de l'ordre de quelques  $\mu s$ . Ce chronogramme est utilisé par le système entier.

## II.4 La girouette

En météorologie la direction du vent représente la direction d'où vient le vent. Elle est exprimée en degrés, comptés dans le sens des aiguilles d'une montre, à partir du nord géographique.

Cette direction est repérée d'après un ensemble de direction divisant en angles égaux les quadrants déterminés par les 4 points cardinaux : Nord, Sud, Est et Ouest. Ce que l'on appelle habituellement la « Rose des Vents ». Il y a différents nombres de Rose des Vents : rose de 8, rose de 16, rose de 32, rose de 36 directions....

La rose de 8 directions comprend les directions des 4 points cardinaux et les bissectrices des angles formés par ces directions.

La rose de 16 directions est obtenue en traçant les bissectrices des angles formés par les directions de la rose de 8 directions.

Notre future girouette est basée sur une rose de 16 directions.

Le principe de fonctionnement de la girouette a le même squelette que celle de l'anémomètre du point de vue capteur. Il consiste en la réalisation d'un codeur optique associé à l'axe pivotant de la girouette. Le codage est obtenu grâce à 4 capteurs infrarouges à réflexion TCRT5000 et un disque formant un code de gray de 4 bits, c'est-à-dire du binaire réfléchi.

Valeur décimale	Binaire naturel	Binaire réfléchi
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

*Tableau 5 : Equivalence entre binaire naturel et binaire réfléchi sur 4bits*

En effet, à la différence du binaire naturel, le binaire réfléchi n'a qu'un seul bit qui change entre le nombre  $n$  et le nombre  $n+1$ , ce qui se traduit en pratique par une plus grande fiabilité du résultat (**Tableau 5**). Par exemple si le codeur optique est juste à la limite entre le nombre 7 et 8 (0111 et 1000 en binaire naturel) sa sortie pourra prendre toutes les valeurs possibles (de 0000 à 1111), en revanche si l'on utilise le binaire réfléchi (0100 et 1100) sa sortie ne pourra obtenir que ces deux mêmes valeurs.

## II.5 Le thermomètre

### I.5.1 Généralités sur la chaleur et la température

#### *La chaleur*

La chaleur est une forme d'énergie échangée entre deux corps. Ce transfert s'effectue sous forme d'énergie mécanique microscopique, correspondant au degré d'agitation des molécules. Ainsi, lorsqu'un corps reçoit de la chaleur, l'agitation de ses molécules s'intensifie, ce qui se traduit la plupart du temps par une augmentation de la température de ce corps. Cependant, un apport de chaleur peut également provoquer un changement d'état : un glaçon chauffé fond progressivement tout en restant à la température de  $0\text{ }^{\circ}\text{C}$ . Entre deux corps, la chaleur se propage spontanément du corps ayant la température la plus élevée vers celui ayant la température la plus basse, élevant donc la température de ce dernier, tout en abaissant la température du premier.

Il existe trois modes de transfert de chaleur : la conduction, le rayonnement et la convection. La conduction implique un contact physique entre les corps ou les parties des corps échangeant de la chaleur. Le transfert par rayonnement ne nécessite ni contact ni présence d'aucune matière entre les deux corps. La convection se produit lorsqu'un liquide ou un gaz est en contact avec une source plus chaude : il se produit alors un mouvement d'ensemble des molécules du fluide transportant la chaleur vers les zones plus froides.

## La température

La température est une grandeur physique qui mesure le degré de chaleur d'un corps ou d'un milieu. Lorsque deux corps sont placés dans une enceinte adiabatique, le corps le plus chaud cède de la chaleur au corps le plus froid, jusqu'à ce que les deux corps aient la même température. Les termes *température* et *chaleur* désignent deux notions distinctes : la température est une propriété thermodynamique du corps et mesure l'agitation microscopique de la matière ; alors que la chaleur est une forme d'énergie qui peut être échangée entre deux corps.

La notion de température du point de vue quantitatif a son origine dans la sensation qu'un corps est chaud ou froid. Quand nous avons l'impression tactile qu'un corps se chauffe, nous dirons que sa température l'élève. Donc la température n'est pas une grandeur mesurable, on dit qu'elle est une grandeur repérable. Pour l'évaluer, on fait appel au phénomène physique qui accompagne la variation de la température dont nous pouvons conventionnellement numérotter la température d'un corps en observant le changement de l'une de ses propriétés spécifiques : la longueur, le volume, la résistivité électrique, l'interdépendance entre la tension  $V$  et le courant  $I$  d'une jonction PN...

Nous disposons donc un phénomène physique  $T$  et sa relation avec la température  $t$  tel que :

$$T = f(t)$$

Le corps lié à la grandeur  $T$  s'appelle *thermomètre*.

### I.5.2 Echelles de température

L'échelle de température utilisée le plus souvent en météorologie est l'échelle Celsius ( $^{\circ}\text{C}$ ).

Le **Tableau 6** donne les échelles de température utilisées aujourd'hui.

	Kelvin(K)	Celsius ( $^{\circ}\text{C}$ )	Rankin ( $^{\circ}\text{R}$ )	Fahrenheit ( $^{\circ}\text{F}$ )
Zéro absolu	0	-273,15	0	-459,67
Equilibre eau/glace à 101325 Pa	273,15	0	491,67	32
Ebullition de l'eau à 101325 Pa	373,15	100	671,67	212

*Tableau 6 : Echelle de température*

Les équations permettant de convertir les degrés Kelvin et les degrés Fahrenheit en degrés Celsius sont les suivantes :

$$t = T - 273.15$$

$$t = (F - 32)5/9$$

avec  $t$  : température en °C

$T$  : température en °K

$F$  : température en °F

Pour le chiffrage des messages d'observation devant être échangés sur le plan international, l'OMM ou Organisation Mondiale de la Météorologie a adopté officiellement l'échelle Celsius.

### II.5.3 Mesure de température en météorologie

Les météorologistes s'intéressent à la température de l'air en surface entre 1,50 et 2m au-dessus du sol, la température à la surface du sol (10cm au-dessus), la température dans le sol (5, 10, 20, 50, 100cm au-dessous de la surface).

Pour les besoins agricoles, la température de l'air peut être mesurée à des niveaux différents : depuis le niveau du sol jusqu'à une hauteur de 10m au-dessus de la limite supérieure de la végétation prédominante.

Pour donner une lecture représentative de la température de l'air, les thermomètres doivent être protégés des échanges radiatifs (rayonnement du soleil, du ciel, de la terre et de tous les objets environnants) mais en même temps, elle reste convenablement ventilée. Deux méthodes de protection sont généralement employées : abri météorologique à persienne et écran de métal poli.

On devrait mesurer la température à l'heure fixe et y joindre les valeurs extrêmes  $T_{\min}$  et  $T_{\max}$  atteinte pendant la journée. On utilise soit de thermomètre à lecture direct soit de thermomètre enregistreur. Dans ce projet, on a combiné ces deux genres : lecture directe avec trois afficheurs sept segments, enregistrement via l'ordinateur d'acquisition.

### II.5.4 Le capteur électronique du thermomètre [21]

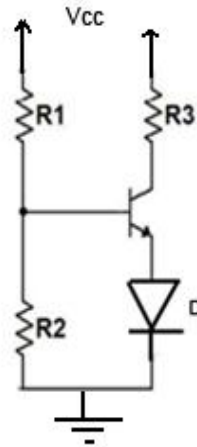
Il est possible d'utiliser des composants électroniques comme capteurs de températures. Ce ne sont pas principalement les caractéristiques fondamentales de ces composants qui nous intéressent, mais plutôt certaines de leurs particularités.

Dans une jonction  $pn$  soumise à une tension  $V$ , le courant  $I$  est lié à la tension par la relation :

$$I = I_s \left[ \exp\left(\frac{V}{\eta V_T}\right) - 1 \right] \quad \text{avec} \quad V_T = \frac{kT}{q} \quad \text{et} \quad V = \eta V_T \cdot \ln \frac{I}{I_s}$$

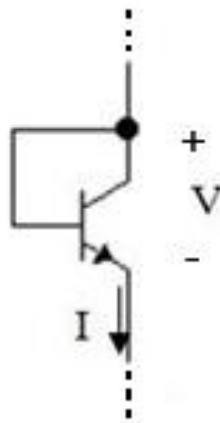
C'est la Loi de Schottky ou  $q$  est la charge de l'électron ( $1,6 \cdot 10^{-19}C$ ),  $k$  constante de Boltzmann ( $1,381 \cdot 10^{-23}J/K$ ) ;  $T$  la température absolue en  $K$  ;  $I_S$  Courant de saturation qui dépend aussi de  $T$ ,  $\eta$  coefficient de matériaux.

Pour une diode, selon la loi de Schottky, si  $I_D$  constante,  $V_D$  est seulement fonction de la température. Le montage de la **Figure 11** permet d'obtenir un courant de diode constant, relativement stable par rapport à la température. (Point de fonctionnement du BJT)



*Figure 11 : Montage utilisant la Diode comme capteur de température*

Dans la **Figure 12** ci-dessous, on peut aussi utiliser un BJT dont la base et le collecteur sont connectés. On a donc une diode.



*Figure 12 : Montage utilisant un BJT comme capteur de température*

On peut réunir ces deux dispositifs pour obtenir des capteurs intégrés comme les séries LM35 du « National Semi conductor » (**Figure 13**).





## II.6 La carte d'acquisition [17]

La carte d'acquisition est basée sur un PIC 18F4550 du Microchip comme nous l'avons déjà dit plus haut. Nous verrons plus tard un peu plus en détail le fonctionnement de ses différentes parties.

### II.6.1 Généralité sur les microcontrôleurs PIC

Un PIC est un microcontrôleur, c'est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de faciliter l'interfaçage avec le monde extérieur sans nécessiter l'ajout de composants externes.

Les PICs sont des composants RISC (Reduced Instructions Construction Set), ou encore composant à jeu d'instructions réduit. L'avantage est que plus on réduit le nombre d'instructions, plus facile et plus rapide en est le décodage, et plus vite le composant fonctionne.

#### Différentes familles des PIC

Il existe plusieurs familles de PICs. La famille Base-Line, qui utilise des mots d'instructions de 12 bits, la famille Mid-Range, qui utilise des mots de 14 bits (et dont font partie le 16F84 et 16F876), la famille High-End et la famille Enhanced (dont font partie le 18F2550 et 18F4550) qui utilisent des mots de 16 bits. Par la suite, d'autres familles sont apparues et les choses ne devraient faire qu'évoluer.

Nous nous limiterons dans ce sous chapitre à la famille Enhanced et particulièrement au PIC 18F4550, sachant que si on a tout assimilé, on pourra facilement passer à une autre famille, et même à un autre microcontrôleur pour l'automatisation de nos stations météorologiques toutes entières.

#### Identification d'un Pic

Pour identifier un PIC, on utilise simplement son numéro :

Les 2 premiers chiffres indiquent la catégorie du PIC, 18 indique un PIC Enhanced. Vient ensuite parfois une lettre L, celle-ci indique que le PIC peut fonctionner avec une plage de tension beaucoup plus tolérante. Vient ensuite une ou deux lettres pour indiquer le type de mémoire programme :

- C indique que la mémoire programme est une EPROM ou plus rarement une EEPROM
- CR pour indiquer une mémoire de type ROM.
- F pour indiquer une mémoire de type FLASH (cas du 18F4550).

On trouve ensuite un nombre qui constitue la référence du PIC. On trouve ensuite un tiret suivi des codes indiquant la fréquence d'horloge maximale que le PIC peut recevoir.

Par exemple, un 16F84-04 est un PIC Mid-Range donc la mémoire programme est de type FLASH de référence 84 et capable d'accepter une fréquence d'horloge de 4MHz. Notre 18F4550 peut supporter une fréquence d'horloge plus de 20 MHz.

Notons que les PICs sont des composants STATIQUES, c'est à dire que la fréquence d'horloge peut être abaissée jusque l'arrêt complet sans perte de données et sans dysfonctionnement.

## II.6.2 Le microcontrôleur PIC18F4550 [10]

### II.6.2.1 Vue d'ensemble

#### Les caractéristiques principales du 18F4550

- Une mémoire programme de type flash de 32768 octets de 16 bits, pouvant contenir 16384 instructions.
- Une mémoire de donnée RAM constituée de 2048 octets.
- Une mémoire de donnée EEPROM de 256 octets
- 35 lignes d'entrées/sorties numériques dont 13 peuvent être utilisées en entrées analogiques. Ils sont regroupés en 5 ports de communication bidirectionnelle nommés A, B, C, D et E.
- Alimentation sous 5 Volts
- Architecture interne révolutionnaire lui conférant une extraordinaire rapidité
- Il peut être cadencé par un oscillateur externe ou par l'oscillateur interne qui sont muni de prédiviseur
- Possibilité d'être programmé in-circuit, c'est à dire sans qu'il soit nécessaire de le retirer du support de l'application
- Vecteur de Reset situé à l'adresse Ox0000
- Deux vecteurs d'interruption, situé à l'adresse Ox0008 (High Priority) et Ox0018 (Low Priority)
- Il est également muni d'un module de communication USB
- Conversion Analogique Numérique sur 10 bits.
- Facilité de programmation
- Simplicité
- Prix abordable.

L'extrait du *Datasheet* de l'**ANNEXE A** montre plus de détails sur les différentes caractéristiques du microcontrôleur en question.

#### Aspect externe du PIC18F4550

La **Figure 15** ci-dessous montre l'aspect externe du PIC18F4550.

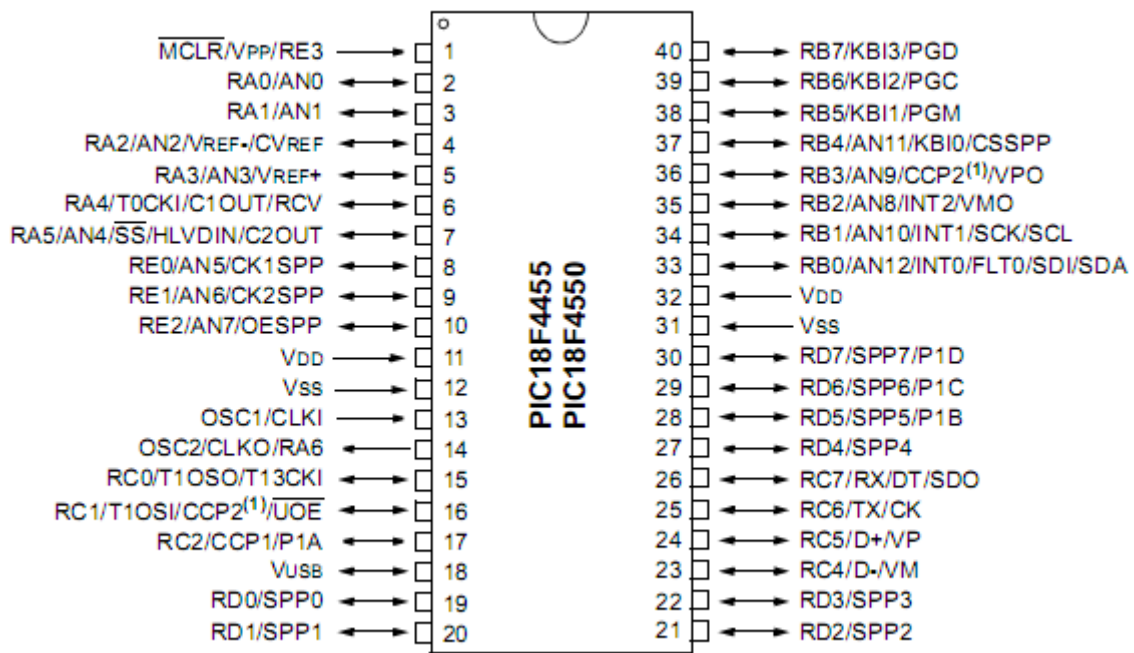


Figure 15 : Aspect externe du PIC18F4550

### Structure interne du PIC

Le microcontrôleur utilise l'architecture de « Harvard », cela veut dire qu'il dispose de bus distincts pour la mémoire de données et pour la mémoire de programme.

La **Figure 16** suivante montre l'architecture interne du PIC18F4550.



### II.6.2.2 Organisation des mémoires

Il y a trois types de mémoires dans ce microcontrôleur :

- La mémoire programme FLASH
- La mémoire de données RAM
- La mémoire de données EEPROM

#### Mémoire programme

Le mémoire programme (Program Memory) contient les instructions du programme que doit exécuter le CPU. On peut aussi l'appeler mémoire morte, et elle n'est accessible en lecture. Sa taille vaut 32768 octets. Sa programmation nécessite une procédure particulière et un matériel adéquat. Son architecture est représentée par la **Figure 17** ci-dessous.

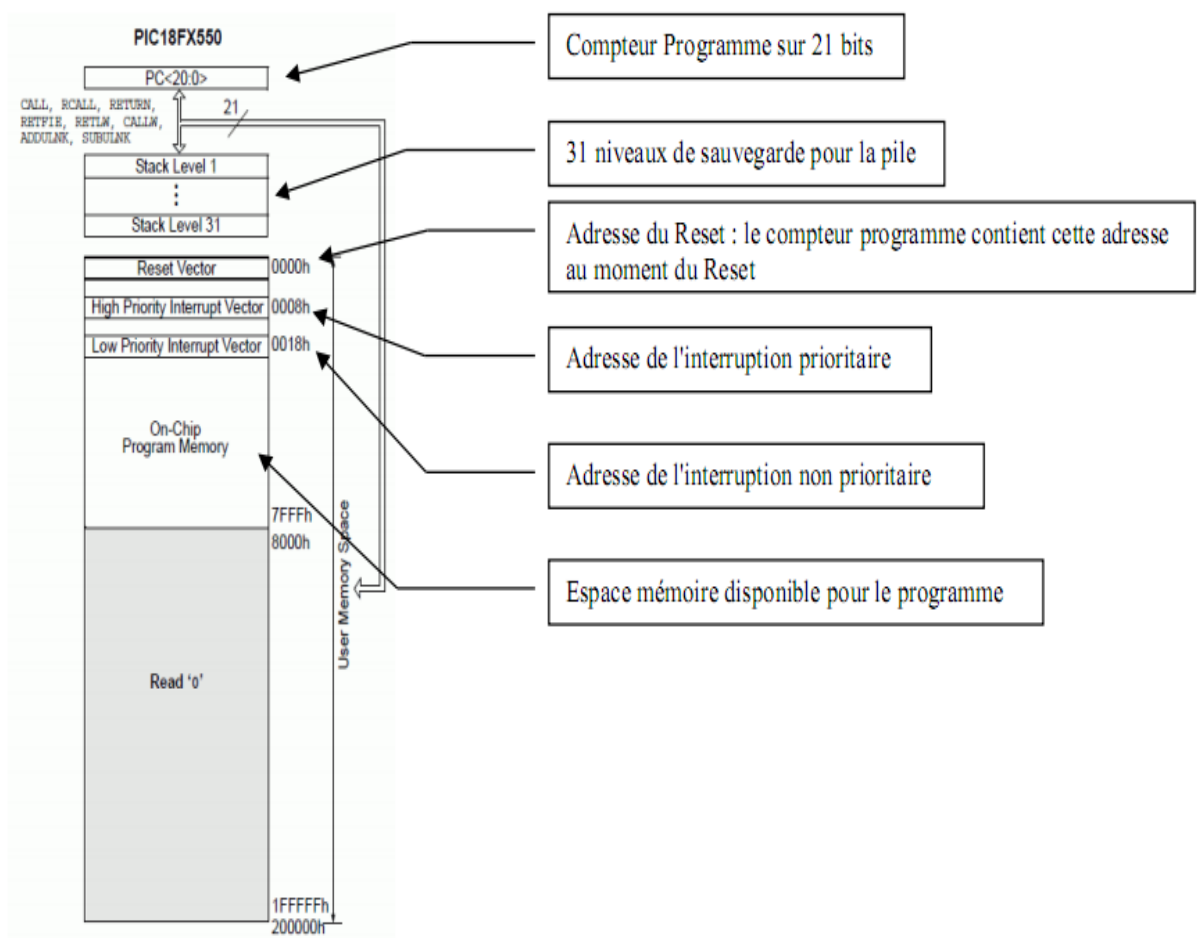


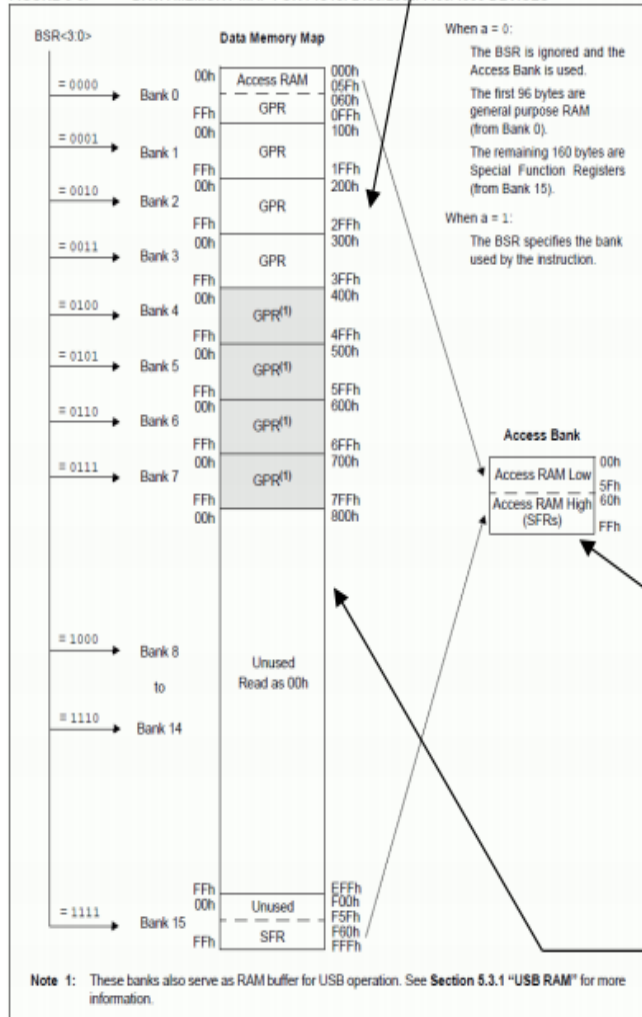
Figure 17 : Mapping du mémoire programme du 18F4550

#### Mémoire de Données RAM

Le Mémoire de données (Data Memory) permet la mémorisation temporaire des données générées par le microcontrôleur pendant les diverses phases de traitement numérique. Ce type de mémoire est accessible en écriture et en lecture.

La **RAM** (Random Access Memory) est une mémoire vive, volatile, ayant un temps de lecture et d'écriture assez court (quelques ns) et une capacité de 2048 octets dont l'organisation est représentée par la figure suivante où 'a' est un paramètre d'instruction permettant d'accéder dans l'ensemble de tous les banques.

FIGURE 5-5: DATA MEMORY MAP FOR PIC18F2455/2550/4455/4550 DEVICES



La RAM est composée de 8 banques de 256 octets chacune dont 4 sont réservées pour la communication USB. Elles sont néanmoins accessibles.

La RAM contient aussi tous les registres de configuration de tous les périphériques du microcontrôleur (PORT, TIMER, ADC...). Ces registres sont placés à partir de l'adresse 0xF60 jusqu'à l'adresse 0xFFFF.

Enfin le bit "a" contenu dans les instructions du microcontrôleur permet de compacter l'ensemble des banques en 1 seule.

Lorsque a = 0, seule une partie de la mémoire est accessible. Elle a été compactée et restreinte au bloc indiqué par la flèche.

Ce bloc contient des registres d'utilisation générale (de 0x000 à 0x05F) et les registres des périphériques (de 0xF60 à 0xFFFF).

Lorsque a = 1, l'ensemble de la mémoire RAM est accessible

Figure 18 : Mapping du mémoire de données RAM du 18F4550

### Mémoire de Données EEPROM

L'EEPROM est une mémoire morte, non volatile, avec un temps d'écriture et de lecture assez élevé (quelques ms) et une capacité de 256 octets.

#### II.6.2.3 Le reset du microcontrôleur

Il existe plusieurs sources de Reset :

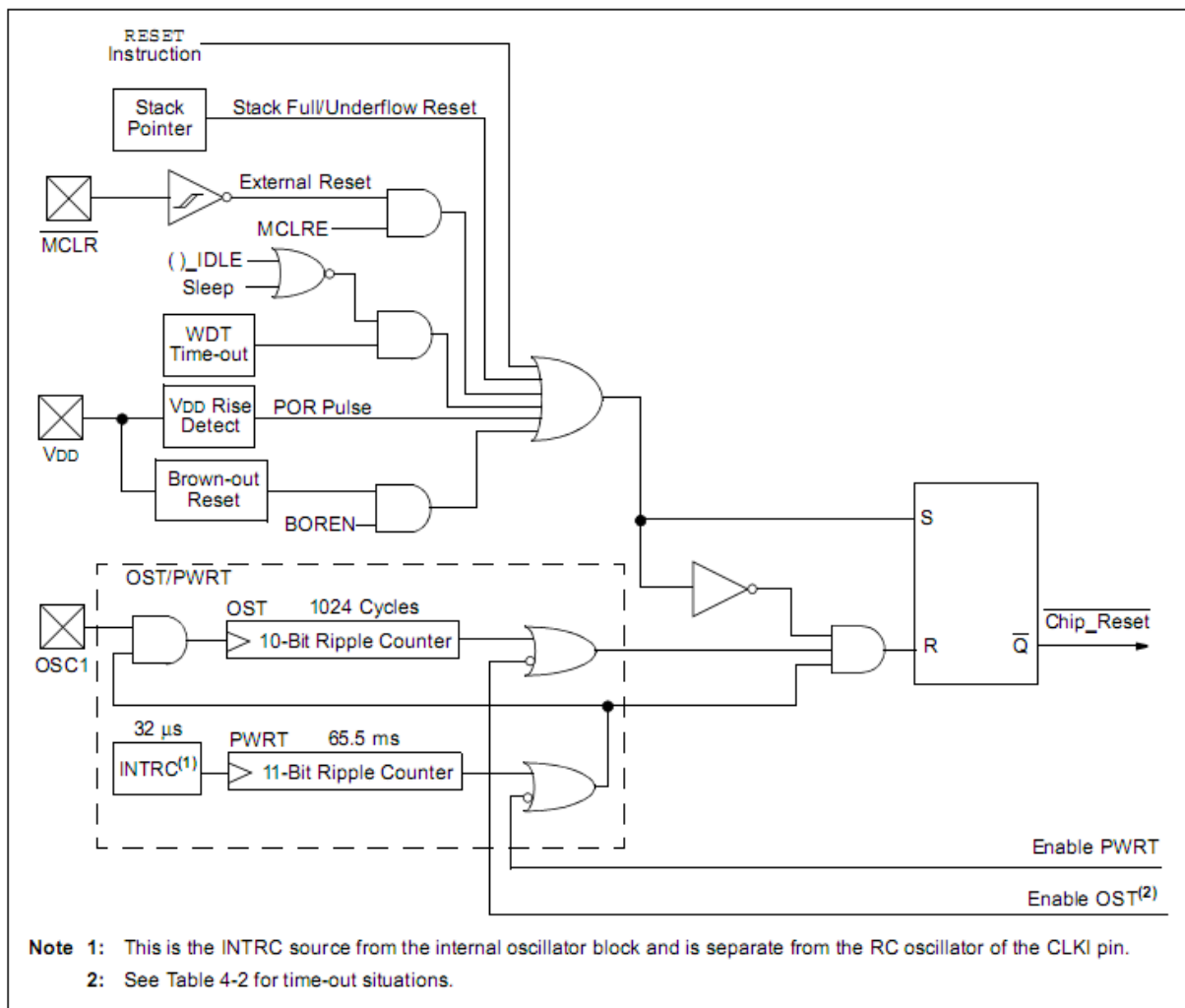
- Instruction RESET : Reset réalisé par le logiciel
- Stack pointer : Pointeur de pile (dépassement du nombre de saut logiciel)
- MCLR : Borne externe du microcontrôleur
- WDT : Chien de garde

- Power-on Reset (POR) : Mise sous tension du microcontrôleur (front montant sur VDD)
- Brown-out Reset (BOR) : Détection d'une chute de tension sur l'alimentation du microcontrôleur.

La **Figure 19** représente le schéma simplifié de circuit de reset du 18F4550.

Il existe 2 sources qui permettent de maintenir le microcontrôleur en "RESET" après la mise sous tension :

- ✚ Power-up Reset (PWRT) : Durée d'environ 65,5 ms que l'on peut ajouter
- ✚ Oscillator Start-up (OST) : Attente de la stabilisation de l'horloge (dans le cas des quartz)



*Figure 19 : Schéma bloc simplifié du circuit de reset du 18F4550*

Après la phase de Reset, le microcontrôleur commence à exécuter la première instruction présente à l'adresse 0x0000 de la mémoire Flash.



#### II.6.2.4 Les bits de configuration du microcontrôleur [8]

Le **Tableau 7** représente une liste non exhaustive des bits de configuration du PIC18F4550.

Bit	Fonction
PLLDIV	Sélection de la Prédivision de la fréquence du Quartz avant la PLL (on doit avoir 4 MHz en entrée de la PLL)
CPUDIV	Sélection de la Postdivision en sortie de la PLL (déterminera Fosc)
USBDIV	Sélection de la fréquence pour l'USB
FOSC	Sélection de l'oscillateur (voir paragraphe concernant l'horloge)
FCMEN	Permet de basculer en oscillateur interne si l'oscillateur externe est défaillant
IESO	Permet d'autoriser la commutation entre l'horloge interne et externe
PWRT	Power up timer : permet d'ajouter une durée supplémentaire au moment du Reset
BOR	Permet de faire un RESET si la tension d'alimentation passe en dessous de la valeur spécifiée par BORV
BORV	Tension d'alimentation à détecter
VREGEN	Activation du régulateur de tension du module USB
WDT	Activation du chien de garde
WDTPS	Définition de la durée du chien de garde
MCLRE	Définit RE3 soit en port d'entrée soit utiliser pour le Reset externe du microcontrôleur (MCLR)
LPT1OSC	Configuration de la puissance consommée par la Timer 1
PBADEN	Permet de configurer les bits du PORT B soit en analogique soit en numérique au moment du RESET
CCP2MX	Multiplexage du bit CCP2 soit sur la borne RB3 soit sur la borne RC1
STVREN	Activation du Reset sur dépassement de la pile (trop de saut de programme)
LVP	Utilisation d'une seule tension pour la programmation du microcontrôleur (implique certaines conditions)
ICPRT	Dédie les bits RB6, RB7 et MCLR uniquement à la programmation et au débogage
XINST	Activation du jeu d'instruction étendu du microcontrôleur
DEBUG	Dédie les bits RB6, RB7 au débogage (ICD2)

*Tableau 7 : Bits de configuration du 18F4550*

### II.6.2.5 Le circuit de génération d'horloge

Pour faire battre le cœur de notre PIC, on a besoin d'un circuit de génération d'horloge (**Figure 20**). Avec les PIC «Enhanced», plusieurs options possibles : circuit RC (résistance + condensateur), ou circuit résonateur céramique, ou oscillateur à quartz... Le mode de fonctionnement est déterminé au moment de la programmation du PIC par des « bits de configuration » qui sont en fait des emplacements de mémoire non-volatile.

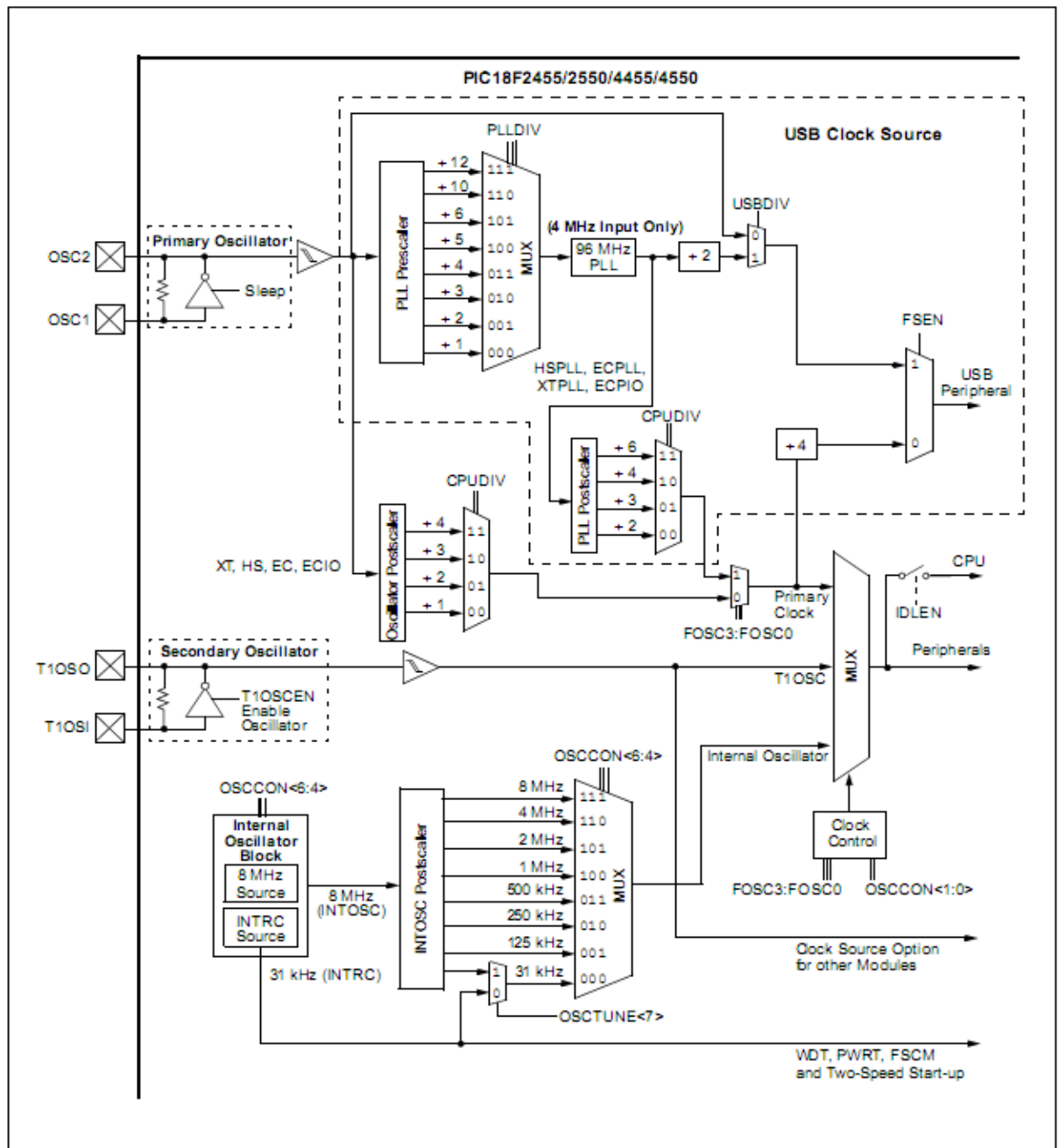


Figure 20 : Circuit de génération d'horloge du 18F4550

Pour générer un « cycle d'instruction », il faut 4 cycles d'horloge. Autrement dit, si le quartz bat à 20 MHz, les instructions du programme s'exécutent à une cadence de 5 MHz.

Il y a quatre paramètres qui nous permettent de gérer l'Horloge :

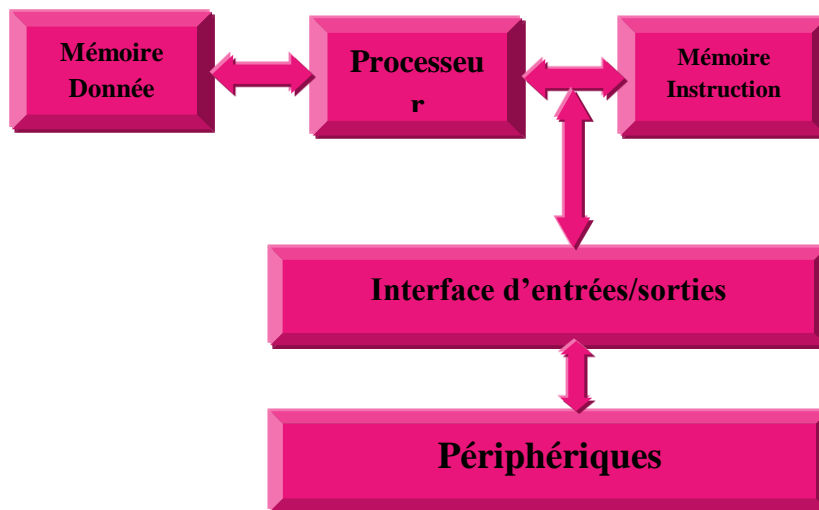
- PLLDIV
- CPUDIV
- USBDIV
- FOSC

Ces paramètres sont présents dans la mémoire Flash à partir de l'adresse 0x300000 et sont détaillés dans l'**Annexe A2.1**.

#### *II.6.2.6 Fonctionnement des entrées/sorties du Pic18f4550*

Les entrées/sorties sont l'interface entre le système embarqué et le monde extérieur.

Elles sont réparties sur cinq Ports : le port A, le port B, le port C, le port D et le port E.



*Figure 21 : Schéma simplifié des Entrées/Sorties du pic 18F4550*

La **Figure 21** montre le schéma général des Entrées/Sorties du PIC alors que le schéma interne d'une borne d'entrée / sortie est représenté par la **Figure 22**.

- ❖ Le buffer B1 est activé lorsque RD LAT est à 1, il permet de lire la sortie de la bascule L1.
- ❖ Le buffer B2 est activé lorsque la sortie de la bascule L2 est à 0 sinon la borne externe du  $\mu$ C est indépendante de la sortie de la bascule L1.
- ❖ Le buffer B3 est activé lorsque RD TRIS est à 1, il permet de lire la sortie de la bascule L2.
- ❖ Le buffer B4 est activé lorsque RD PORT est à 1, il permet de lire la sortie de la bascule L3.
- ❖ Les "Latch" ou bascules L1 et L2 recopient sur leur sortie le contenu de la donnée présente sur le Bus de donnée sur des fronts descendants des signaux de contrôles WR LAT, WR PORT et WR TRIS.

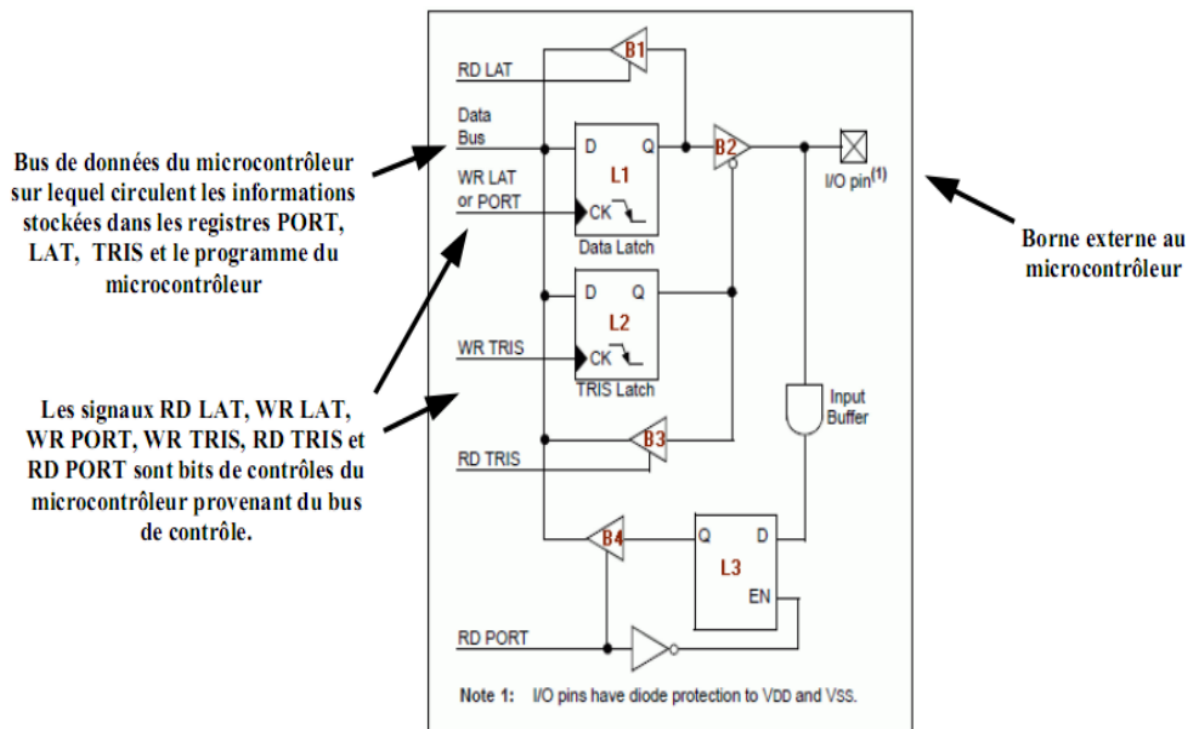


Figure 22 : schéma interne d'une borne d'entrée / sortie

Il existe 16 registres qui nous permettent d'utiliser les ports d'E/S :

- ADCON1 : configure certaines bornes en Numériques ou Analogiques
- TRISA, TRISB, TRISC, TRISD, TRISE : configure la borne du  $\mu$ C en entrée (1) ou en sortie (0)
- LATA, LATB, LATC, LATD, LATE : écrire sur la sortie de la bascule L1 ou lire la sortie de la bascule L1
- PORTA, PORTB, PORTC, PORTD, PORTE : écrire sur la sortie de la bascule L1 ou lire la borne I/O pin.

Les registres TRIS permettent de donner une direction aux bornes du microcontrôleur soit en entrée soit en sortie.

Les registres LAT permettent d'accéder à la bascule L1 ci-dessus en lecture et en écriture.

Les registres PORT permettent d'écrire sur la sortie de la bascule L1 et de lire la borne externe du microcontrôleur.

Lors de l'écriture dans le registre LAT ou PORT, la donnée se retrouvera sur la borne externe du microcontrôleur si le TRIS associé est à 0 sinon la donnée sera stockée sur la sortie de la bascule L1.

La lecture du niveau logique de la borne externe du microcontrôleur se fait par la lecture du registre PORT. Cette opération active le signal de contrôle RD PORT qui permet de copier la sortie de la bascule L3 sur le bus de données du microcontrôleur.

#### II.6.2.7 Les Timers ou Compteur

Le Timer est un compteur qui est incrémenté par des fronts montants ou descendants (par exemple horloge). Il peut être préchargé logiciellement par une valeur et peut posséder un prédiviseur (Prescaler) et/ou un Post diviseur (Postcaler).

##### Le Timer0

Le Timer0 est un compteur sur 8 ou 16 bits composé de 2 registres de 8 bits : TMR0H et TMR0L accessible en lecture et en écriture.

En mode 8 bits, seul le registre TMR0L est utilisé. En mode 16 bits, les 2 registres sont utilisés.

Son entrée est définie soit par un signal externe au microcontrôleur branchée sur la borne T0CKI, soit par l'horloge interne du microcontrôleur ( $F_{osc} / 4$ ). Le choix du signal d'entrée est défini par le bit T0CS du registre T0CON.

Un pré-diviseur programmable peut être sélectionné par le bit PSA.

A chaque front d'horloge du signal FH, le compteur  $TMR0 = TMR0H:TMR0L$  est incrémenté.

En mode 8 bits, la capacité du compteur est  $255 = 2^8 - 1$  et en mode 16 bits sa capacité est  $65535 = 2^{16} - 1$ .

Lorsqu'il y a dépassement de sa capacité, le bit TMR0IF du registre INTCON est mis à 1, et le compteur repart de valeur 0. Le bit TMR0IF doit être remis à 0 logiciellement.

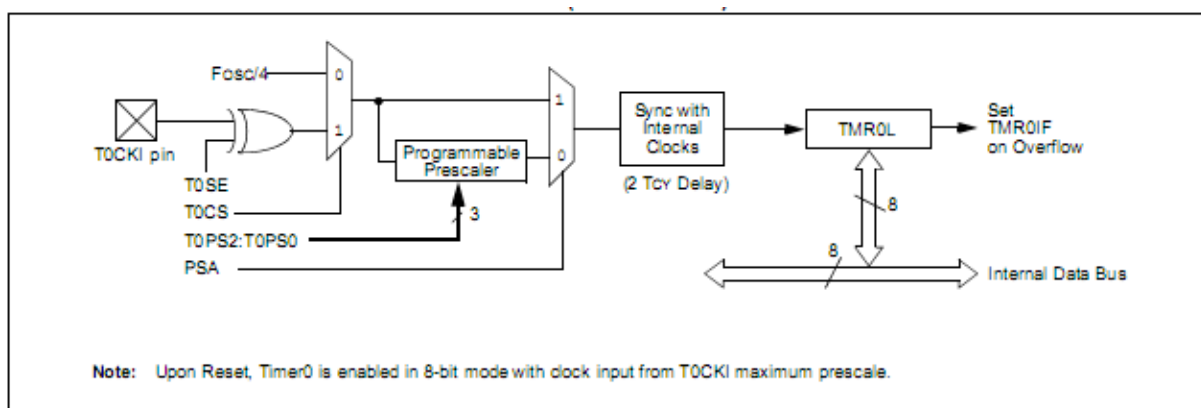


Figure 23 : Schéma bloc du Timer0

### Le Timer1

Le Timer 1 fonctionne sur le même principe que le Timer 0 mais le pré-compteur ne peut prendre que quatre valeurs : 1, 2, 4 ou 8.

La **Figure 24** suivante récapitule son fonctionnement :

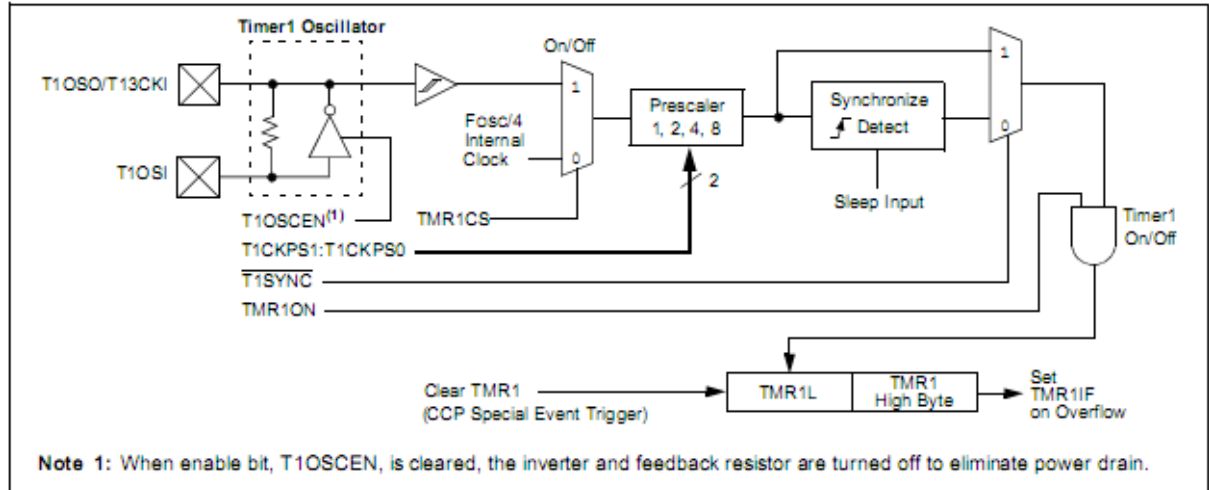


Figure 24 : Schéma bloc du Timer1

### Le Timer2

Il fonctionne avec un compteur de 8 bits TMR2, un pré-diviseur programmable, un post diviseur programmable et un registre de comparaison de 8 bits. TMR2 est incrémenté par l'horloge du microcontrôleur ( $F_{osc} / 4$ ) et comparé au registre **PR2**. En cas d'égalité le post diviseur est incrémenté et le registre TMR2 est remis à 0. Lorsque le post diviseur a atteint la valeur programmée, il génère la mise à 1 du bit de dépassement TMR2IF. (Voir **Figure 25**)

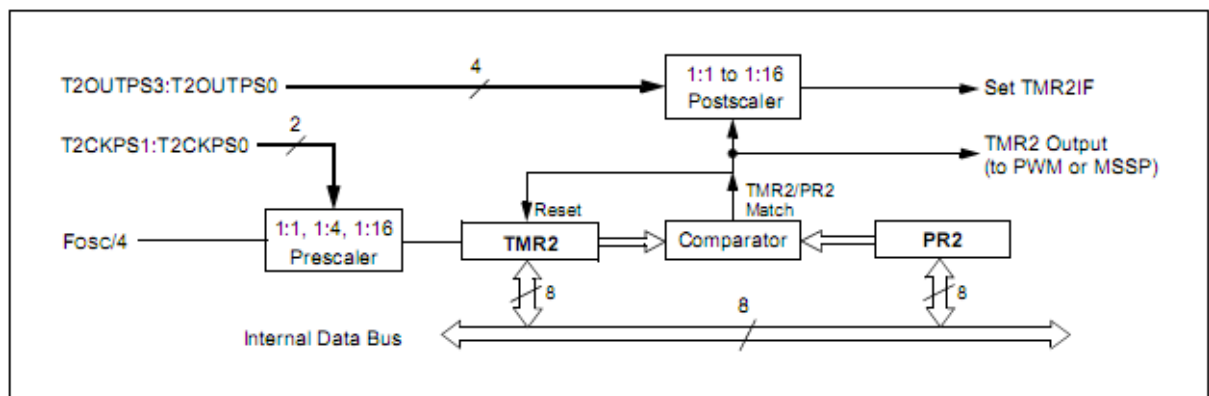


Figure 25 : Schéma bloc du Timer2

#### II.6.2.8 Le Watch Dog Timer

Le chien de garde (Watch Dog) est un système anti-plantage du microcontrôleur. Il assure que l'exécution d'une même instruction ne soit pas prolongée. Au rythme de la fréquence d'horloge, un compteur préchargeable se décrémente régulièrement. Avant qu'il

atteigne «0», et si aucun préchargement n'est effectué, un Reset est généré, relançant ainsi le microcontrôleur.

On n'a pas besoins de ce module dans ce projet.

#### *II.6.2.9 Les interruptions*

##### Généralité sur les interruptions

Une interruption est un événement extérieur qui va détourner le programme de son fonctionnement normal, pour exécuter un programme dit d'interruption permettant de gérer l'événement. Une fois le programme d'interruption fini, le système retourne dans le programme principal à l'instruction suivante.

Quand l'interruption se produit :

- Le microcontrôleur finit l'instruction en cours (l'instruction en assembleur, pas en C)
- Sauvegarde l'adresse suivante du programme principal
- Sauvegarde le contexte
- Exécute le programme lié au périphérique demandant l'interruption.
- Restitue le contexte
- Retourne au programme principal à l'adresse suivante.

Tous les circuits périphériques (Timer0, Timer1, Timer2, USB, PORT d'E/S,...) contenus dans le PIC sont susceptibles de générer une interruption prioritaire ou non prioritaire selon sa configuration.

##### Les interruptions sur le PIC18F4550

Le pic 18F4550 dispose plus de 20 sources d'interruptions.

Les sources d'interruptions ont toutes un bit permettant de les valider individuellement et un flag (bit) permettant de rechercher d'où provient l'interruption.

Ces bits sont accessibles dans les registres RCON, INTCON, INTCON2, INTCON3, PIR1, PIR2, PIE1, PIE2, IPR1, IPR2. (Page 97 de la datasheet du PIC).

Par exemple, pour le Timer0, le bit permettant de valider l'interruption est TMR0IE et le flag est TMR0IF. Ces deux bits sont dans le registre INTCON.

D'une manière générale, il est très facile de réparer les bits permettant de valider une interruption, car ils se finissent tous par les lettres **IE** qui signifient « Interrupt Enable » (Autorisation d'interruption). De même pour les drapeaux (flag) associés, ils se terminent tous par **IF** qui signifie « Interrupt Flag » (drapeau d'interruption).

Toutes ces interruptions sont des interruptions masquables. En effet le bit **GIE** du registre INTCON permet de valider ou d'invalider toutes les interruptions. Au reset, ce bit est placé à 0, dévalidant ainsi toutes les interruptions.

En outre, le PIC18F4550 possède en réalité ce qu'on appelle deux « niveaux » d'interruption, dont l'une est plus prioritaire que l'autre. Il y a donc un vecteur d'interruption dit de « haute priorité » (high-priority interrupt level) et un vecteur d'interruption dit de « basse priorité » (low-priority interrupt level) :

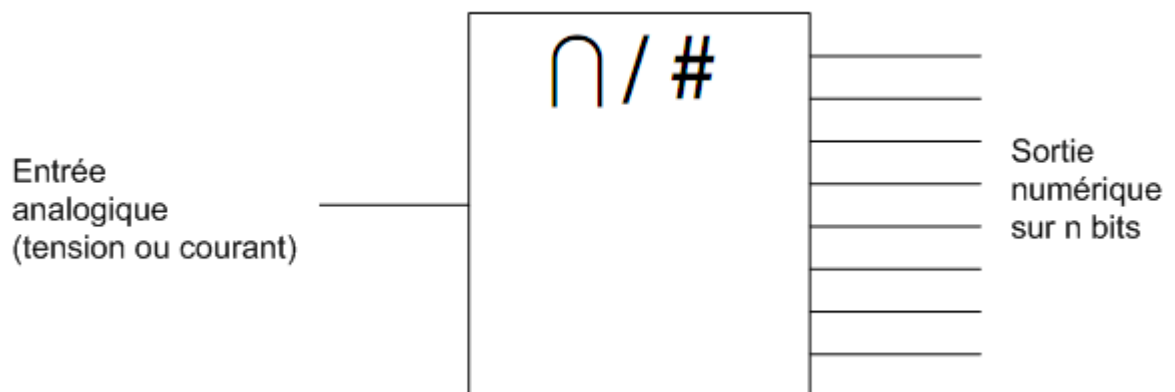
- L'interruption Prioritaire à l'adresse 0x0008
- L'interruption Non Prioritaire à l'adresse 0x0018

Le module CAN et USB de notre PIC mérite une étude détaillée que nous ne pouvons pas survoler sur quelques phrases simples. Donc nous avons intérêt à réserver un paragraphe entier pour chaque module.

## II.6.3 Le Convertisseur Analogique Numérique ou ADC [18]

### II.6.3.1. Généralité sur l'ADC (*Analogic Digital Converter ADC en anglais*)

On appelle Convertisseur Analogique Numérique (C.A.N.) tout dispositif électronique qui transforme une grandeur analogique d'entrée  $V_{en}$  en un nombre binaire de sortie  $N$  proportionnel à cette grandeur  $V_{en}$ . En anglais, le Convertisseur Analogique Numérique est appelé Analogic Digital Converter (A.D.C.).



*Figure 26 : Symbole de la fonction Conversion Analogique/Numérique*

Le signe '∩' indique que la grandeur est de type analogique (il s'agit ici de l'entrée). Le signe '#' indique que la grandeur est de type numérique (il s'agit ici de la sortie).

La fonction conversion analogique-numérique consiste donc à transformer une grandeur électrique en une grandeur numérique exprimée sur  $n$  bits comme l'indique la **Figure 26**. Cette grandeur de sortie représente, dans le système de codage qui lui est affecté, un nombre proportionnel à la grandeur analogique d'entrée. Pour le PIC18F4550, on a  $n=10$ .

D'une manière générale, convertir une grandeur analogique en une grandeur numérique nécessite trois opérations :



### Echantillonnage/Blocage

La première étape est l'échantillonnage du signal à convertir qui se fait en générale périodiquement. La fonction réalisant cette opération est l'échantillonneur. Technologiquement, un simple interrupteur commandé à fermeture par train d'impulsions (ou signal d'échantillonnage) convient. Mais, il faut pouvoir maintenir la tension pendant un temps suffisant pour permettre la quantification de cette valeur. D'où la nécessité d'un bloqueur en aval de l'échantillonneur. La réunion des deux éléments est appelée : « Echantillonneur-Bloqueur ». Le fonctionnement de l'Echantillonneur-Bloqueur ou SHS (Sample and Hold System) est donné par la **Figure 27**.

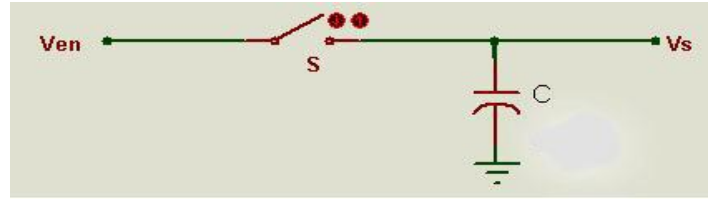


Figure 27 : Principe d'un échantillonneur bloqueur

Où S est un circuit de commande

La tension aux bornes du condensateur C suit les variations instantanées de  $V_{en}$  pendant l'intervalle de temps  $T_e$  durant lesquels un circuit de commande ferme S. A la fin de  $T_e$ , S est ouvert et le condensateur C retient la dernière valeur de  $V_{en}$  (mémorisation)

### La quantification

Opération qui consiste à associer une valeur analogique à la plus petite variation mesurable entre deux valeurs codées distinctes en sortie. Cette valeur est appelée « pas de quantification ».

$$q = \frac{\Delta V_{en_{max}}}{2^n} = \frac{V_{ref+} - V_{ref-}}{2^n}$$

Où  $q$  est le pas de quantification (en tension, ici), appelé aussi « Résolution »,  $\Delta V_{en_{max}}$  (5V-0V=5V pour notre cas) est l'écart entre la valeur min et la valeur max de  $V_e$  à numériser (en V), et  $n$  : nombre de bits en sortie du convertisseur ( $n=10$  pour le PIC18F4550).

D'où

$$q = \frac{5}{2^{10}} = 4,89mV$$

### Le codage

Opération qui assigne une valeur numérique à chacun de ces niveaux, c'est-à-dire il consiste à associer à la valeur  $V$ , issue de l'échantillonneur-bloqueur, une série de chiffres  $\{a_n\}$ ,  $n \in \mathbb{N}$  qui permettent d'approcher au mieux  $V$  par la relation :

$$V \approx q \sum_{n=0}^{N-1} a_n x^n = \frac{\Delta V}{x^N - 1} \sum_{n=0}^{N-1} a_n x^n$$

avec  $q$  : le pas de quantification  
 $x$  : la base de représentation (en numérique, on utilise exclusivement la représentation binaire, soit  $x=2$ )

$a_n$  : Le code de rang  $n$ ,  $N$  : le nombre de chiffres sur lequel est codée la valeur

$\Delta V$  : la plage de conversion, avec  $\Delta V = q(x^n - 1)$

Les codages les plus couramment utilisés sont :

- Le binaire naturel, pour les nombres non signés,
- Le complément à 2 pour les nombres signés,
- Le code binaire signé.

En un mot, le CAN utilise trois signaux différents comme le montre la **Figure 28** :

- Le signal à convertir
- Le signal d'échantillonnage
- L'horloge du convertisseur

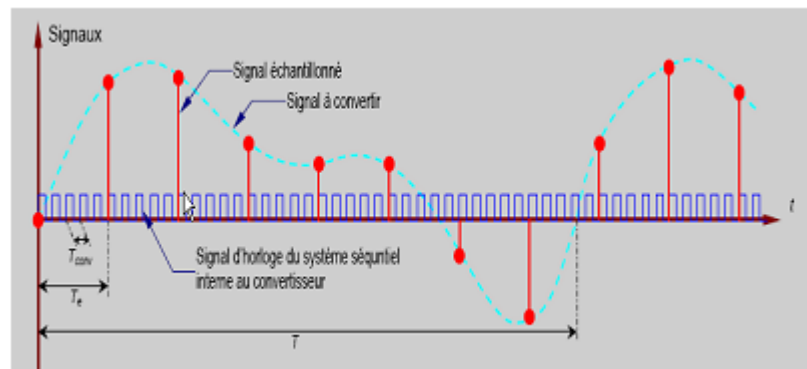


Figure 28 : Les différents signaux d'un CAN

### II.6.3.2 Le CAN du microcontrôleur

Pour le PIC il faut 12 coups d'horloge pour réaliser une conversion et le schéma bloc du CAN est représenté par la **Figure 29**.

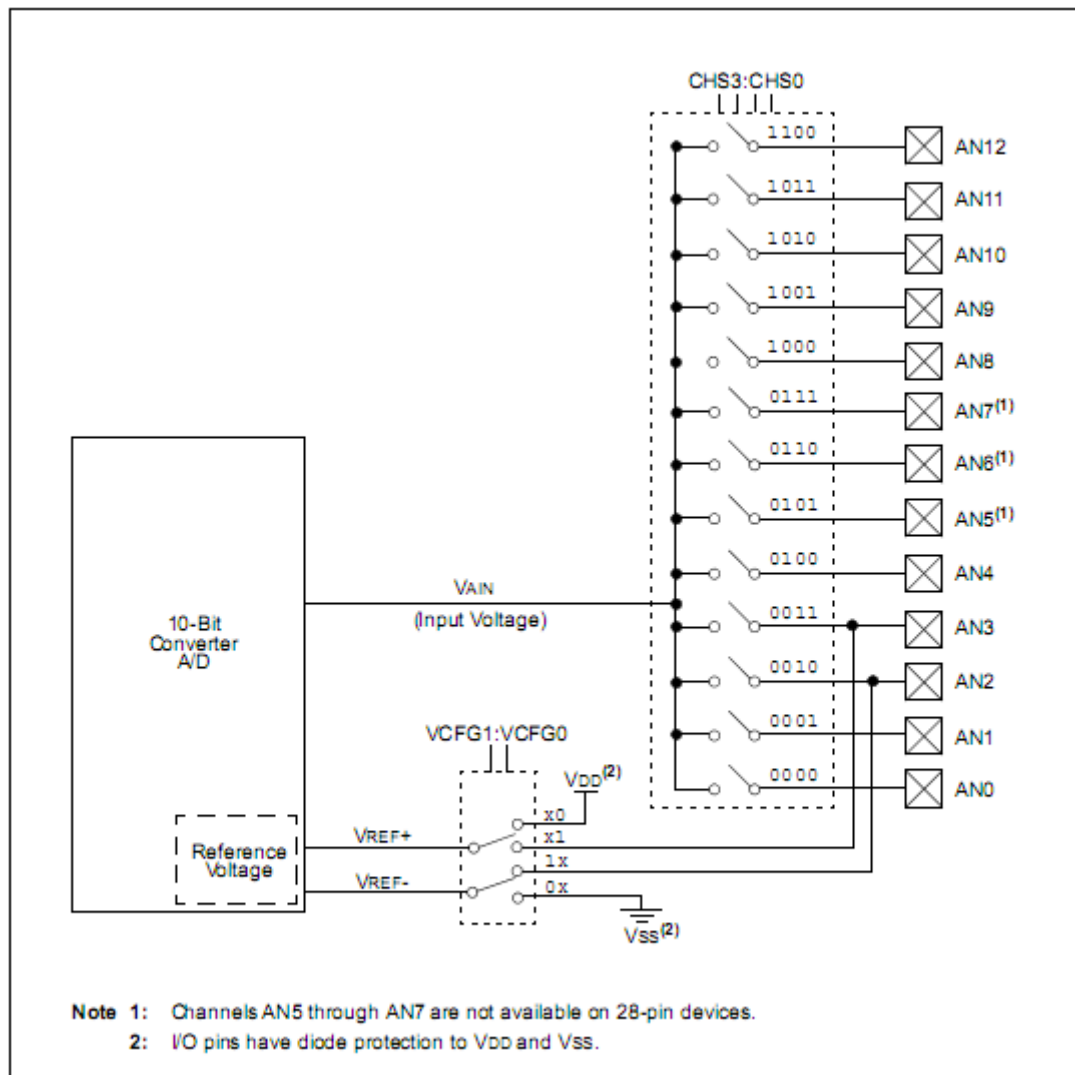


Figure 29 : Schéma bloc du CAN intégré dans le PIC18F4550

Pour pouvoir traiter le signal venant du capteur thermométrique LM35, nous allons utiliser ce Convertisseur Analogique/Numérique par l'intermédiaire de cinq principaux registres de notre microprocesseur :

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

Il y a six étapes à mettre en œuvre :

- Initialisation des paramètres du convertisseur.
- Sélection de la voie à convertir
- Attente de la stabilisation de l'échantillonneur/Bloqueur (chargement du condensateur)
- Lancement d'une conversion
- Attente de la fin de conversion
- Lecture de donnée

#### II.6.4 Le port USB et sa fonctionnalité [2]

Le microcontrôleur en question possède différent type de port d'interface : série, parallèle et USB.... Cependant, on va donner un sens au standard USB.

Le bus USB mérite une très longue explication. Mais nous trouvons que ce n'est pas la peine de réinventer la roue. Microsoft, Compaq, Hewlett-Packard, Intel, Lucent, NEC, Philips ont établi le protocole USB 2.0 en 662 pages que nous pouvons télécharger sur le site de Microsoft et dévorer à notre guise.

Nous avons donc intérêt à ne pas retenir que ce qui nous semble vital pour notre projet.

##### II.6.4.1 Généralité sur l'USB

La dénomination USB qui est « Universal Serial Bus » a dérivée vers le nom « Useless Serial Bus ».

Le bus USB est né de l'alliance en 1994 de sept partenaires industriels (Compaq, DEC, IBM, Intel, Microsoft, NEC et Northern Telecom) ;

Le bus USB a été conçu à l'origine pour faciliter les transferts de données en particulier définir une connectique « universelle » et « Plug & Play », utilisable aussi bien pour une souris que pour un modem ou un moniteur. Conçu également pour répondre au besoin d'intégration entre le monde du PC et celui du téléphone et enfin pour répondre au besoin d'extensions multiples en dehors du PC, en effet on peut brancher jusqu'à 127 périphériques en théorie... cela change de la liaison série habituelle.

En septembre 1998 paraît la version officielle (version 1.1). Cette spécification définit 2 vitesses de fonctionnement : 1,5 Mbps (Low Speed) et 12 Mbps (Full Speed).

Puis en avril 2000 est sortie la spécification 2.0. Cette nouvelle norme supporte toutes les caractéristiques de l'USB 1.1 : 1,5 Mbps (Low Speed), 12 Mbps (Full Speed), et rajoute une vitesse de 480 Mbps (High Speed) et optimise l'utilisation de la bande passante.

Le principal but du bus USB est la facilité d'utilisation qui se traduit suivant de multiples critères.

Tout d'abord on peut dire que le protocole USB est une norme très souple ; une interface unique suffit pour commander plusieurs types de périphériques, il suffit juste de posséder le bon driver.

Ensuite, un point fort de l'USB est sa configuration automatique, on l'appelle aussi le « plug & play ». Cela signifie que si l'utilisateur connecte un périphérique USB, Windows détecte automatiquement ce périphérique et charge le driver approprié s'il est disponible dans les fichiers de Windows. Si ce n'est pas le cas, Windows demande d'installer le disque (CDRom) contenant ce driver afin qu'il le copie dans son répertoire de drivers (généralement c:/windows/system32/drivers) et ceci se fait une seule fois.

Lors du prochain branchement, le périphérique USB recharge automatiquement son driver, cette étape est alors transparente pour l'utilisateur. Il faut noter aussi qu'il n'est pas nécessaire, avec le protocole USB, de lancer un fichier d'installation ou de redémarrer (rebooter) le PC avant d'utiliser le périphérique.

Il n'y a pas non plus dans le protocole USB à faire le choix de l'adresse du port comme par exemple pour une liaison série, une adresse dynamique est allouée à chaque fois que l'on branche un autre périphérique, la norme USB est capable de fournir 127 adresses dynamiques différentes (codée sur 7 bits) c'est pour cela qu'on entend toujours parler de brancher jusqu'à 127 périphériques (en théorie) sur le bus USB. Le PC alloue donc une série d'adresses de port et une commande d'interruption (IRQ) pour les interfaces USB. Contrairement aux autres périphériques qui ne sont pas USB, pour chaque ajout d'un périphérique il faut lui associer une adresse de port, souvent un IRQ ainsi qu'une carte d'extension si le PC n'en possède plus ou pas assez.

Avec les périphériques USB, tous ces problèmes sont inexistants.

Un atout majeur du bus USB est qu'il est très facile à connecter, on n'a pas besoin d'ouvrir le PC à chaque fois que l'on veut rajouter un périphérique, les anciens PC possèdent au minimum deux ports USB (de nos jours il y en a davantage). On peut aussi connecter un HUB sur un port déjà existant pour pouvoir connecter encore plus de périphériques USB. Le HUB joue en quelque sorte le rôle d'une multiprise. C'est un multiplieur de port USB. Actuellement certains moniteurs, claviers ... possèdent des HUBs intégrés. On peut, évidemment, reconnecter d'autres HUBs aux HUBs cité précédemment.

Un autre avantage du bus USB est la connectique. En effet les câbles possèdent deux extrémités bien différentes pour ne pas se tromper lors du branchement, de plus les câbles USB sont très compacts comparés aux câbles pour les liaisons parallèles ou séries.

On dit également que le bus USB est « Hot pluggable », c'est à dire que l'on peut connecter et déconnecter un périphérique USB tout en ayant le PC allumé. L'Operating System installé sur le PC (Windows ...) le reconnaît ainsi immédiatement.

L'USB possède une caractéristique très intéressante : c'est le mode veille lorsque l'on n'utilise pas le périphérique. On l'appelle aussi « Power conservation ». En effet le bus USB se met en suspend après 3 ms ou il n'est plus utilisé. Pendant ce mode, le composant ne consomme que 500µA

Enfin, le dernier point fort pour l'USB est que cette norme permet d'alimenter le périphérique directement avec le PC. Il n'est pas nécessaire d'alimenter le périphérique avec une alimentation extérieure, le PC suffit mais il y a quand même quelques conditions à respecter pour que cette « auto alimentation » soit opérationnelle.

#### *II.6.4.2 Notions importantes pour commencer avec l'USB [4]*

##### Le câble USB

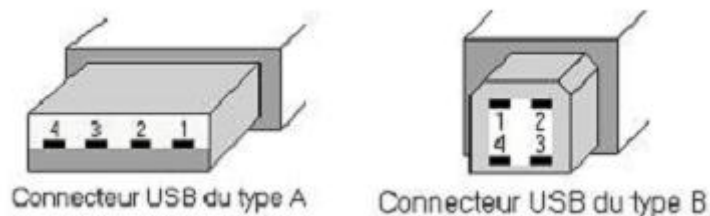
Le câblage USB est relativement simple ; il a la même structure quelle que soit la vitesse de transmission. Le câble transporte deux paires de fils :

La paire de signal destinée au transfert de données D+ et D- et une seconde paire qui peut être utilisée pour la télé alimentation GND et Vcc. La première paire est non blindée pour les périphériques lents tels que les claviers, souris fonctionnant à 1.5Mbits/s tandis que caméras, micro et autres ont recours à une paire de fils torsadée blindée pour atteindre les 12Mbits/s.

La longueur maximale autorisée par la norme est de 3m pour un câble non blindé donc généralement pour un périphérique Low USB (= 1.5Mb/s) et de 5m pour un câble blindé dans le cas d'un périphérique Full USB (=12Mb/s).

Le câble USB est composé de deux fiches bien différentes :

- En amont d'une fiche appelé connecteur USB de type A, branché au host (PC).
- L'extrémité aval par contre peut se retrouver en deux versions : connecteur USB du type B et un mini connecteur type B (appelé souvent optionnel). Ce dernier est réservé aux dispositifs de très faible dimension (ou de grande intégration) tels les appareils photo numériques.



*Figure 30 : Brochage des connecteurs USB de type A et B*

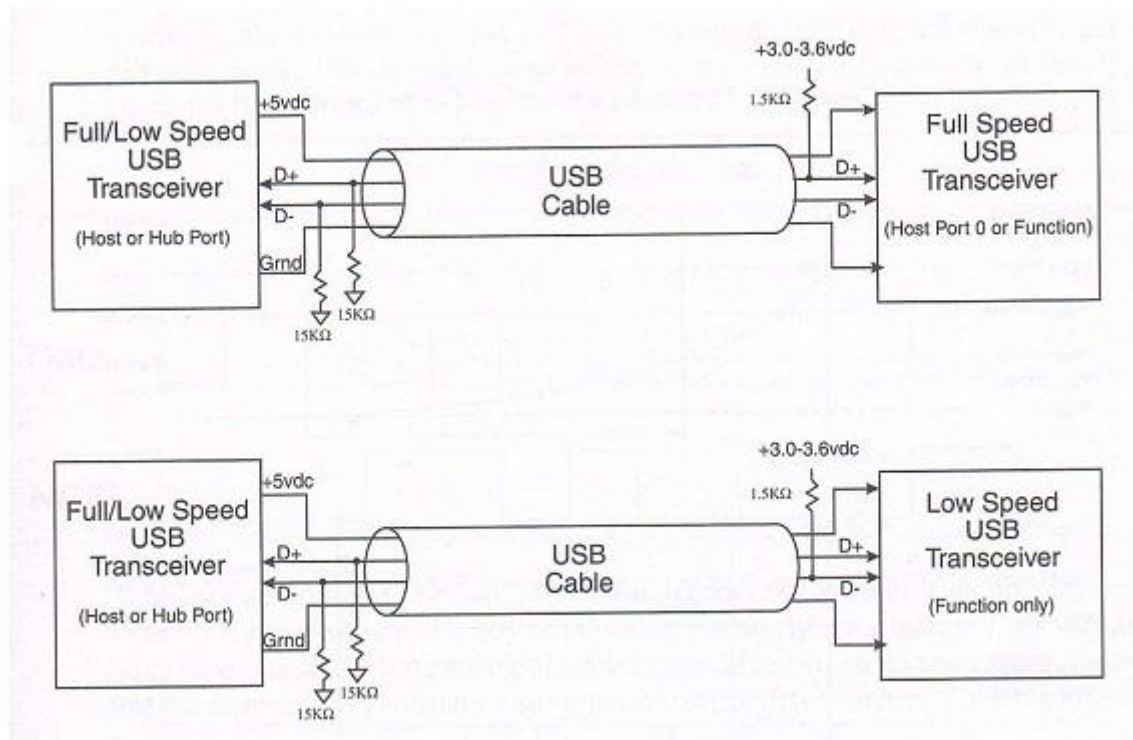
Et voici l'aspect des connecteurs en trois dimensions :



*Figure 31 : Aspect des connecteurs USB de type A et B*

### Choix « Low » ou « Full » USB

Pour faire le choix entre la version USB Low Speed et Full Speed, il suffit de placer une résistance de tirage sur l'interface d'entrée (Host ou Hub). Cette résistance de tirage (Pull-Up de  $1.5k\Omega$ ) est placée soit sur D- dans le cas du Low Speed ou sur D+ dans le cas du Full Speed. Elle est indispensable pour permettre au host de détecter la connexion ou la déconnexion. Par contre côté sortie du host il y a une résistance sur chacune des lignes D+ et D- (Pull Down de  $15k\Omega$ )



*Figure 32 : Choix Low ou Full USB – Résistance de Pull Up [2]*

On voit sur le schéma ci-dessus que s'il n'y a pas de résistance de Pull up, les deux câbles de données sont tirés à la masse par les deux résistances de Pull down. Les résistances Pull down étant de  $15k\Omega$  et les résistances de Pull up de  $1.5k\Omega$  (en fonction de la vitesse choisie) il y aura toujours une ligne de donnée qui sera à 90% de Vcc. Si le host détecte qu'une ligne de donnée avoisine les 90% de Vcc il en conclura qu'un composant USB est connecté ainsi que des informations sur la vitesse du composant, Low ou Full Speed. L'énumération ou l'identification du composant USB peut donc commencer.

### Temps de connexion et de déconnexion

Les connecteurs USB (de type A ou B), ont les deux pattes d'alimentation un peu plus longues que les pattes de données. Ceci est volontaire, d'une part pour que le périphérique USB soit alimenté avant que transitent les données. Il est normal qu'avant de transmettre des données, que le périphérique soit reconnu et que les bons drivers ont été chargés. L'USB permet de faire cette étape automatiquement. D'autre part cette différence de longueur entre

les pins d'alimentations et de données est aussi pour protéger les composants USB, car un composant doit d'abord être alimenté avant de recevoir des données

En principe, lorsqu'on branche un composant USB une différence de potentiel se crée et c'est cette différence de potentiel qui démarre le processus d'énumération. Sachant que le soft du host scanne continuellement les ports et les HUBs USB.

Les composants USB ont deux états, l'un pour les composants Low-Speed et l'autre pour les composants Full-Speed. Ces deux états sont opposés l'un par rapport à l'autre.

Ces états sont appelés état de référence K et J. Essayons à présent d'expliquer succinctement le fonctionnement. Comme on le sait déjà, les données D+ et D- sont des données différentielles, donc si un composant USB est branché, un câble de donnée (D- par exemple pour le Low-Speed) est proche de Vcc et l'autre proche de GND. On appelle cet état « l'état J ».

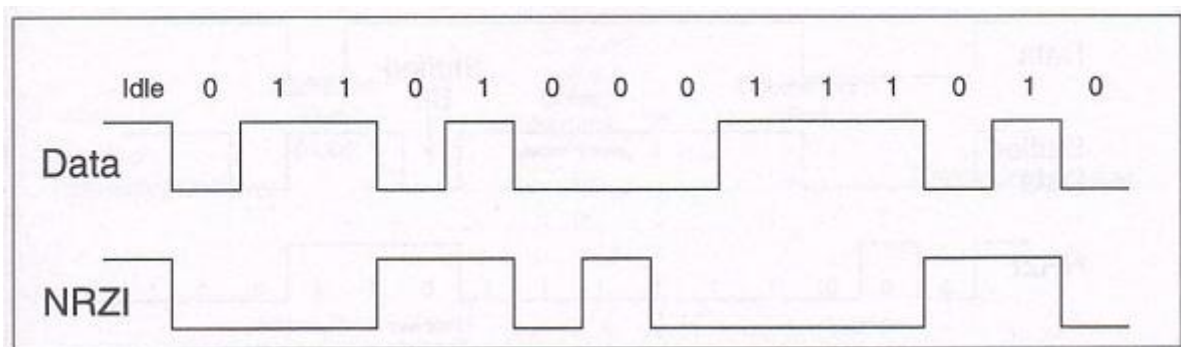
Si on débranche le composant, les deux lignes de données changent d'état et on appelle cet état l'état K.

En général, un composant est déconnecté si les deux lignes de données (D+ et D-) chutent sous la valeur de 0.8Vdc sur une durée supérieure à 2.5µs. Et de même pour la connexion : si une ligne de données (D+ ou D- selon la vitesse) dépasse 2.0Vdc sur une durée supérieure à 2.5µs, le host considère que le composant est attaché.

### Le codage NRZI

Pour transmettre les données, l'USB utilise le codage NRZI (Non-Retour à Zéro Inversé).

Le principe de ce codage est le suivant, un «1» logique est représenté par un non changement d'état en NRZI et un «0» logique est représenté par un changement d'état. Le codage va encore plus loin et utilise le Bit Stuffing ; c'est le fait de mettre un «0» après 6 «1» logique consécutif pour forcer une transition dans le code NRZI (pour éviter les pertes de données).



*Figure 33 : Principe du Codage NRZI*



La **Figure 33** montre un exemple de ce codage NRZI. Ce type de codage est uniquement utilisé pour le transport à travers le cordon USB. Il doit être décodé lors de la réception pour pouvoir retraiter les données.

### Alimentation USB

Pour simplifier l'explication de l'alimentation des périphériques USB, la norme a prévu deux niveaux d'alimentations, le premier niveau consomme une unité d'énergie, l'autre consomme cinq unités. Une unité vaut 100mA. C'est à dire qu'il existe des composants qui consomment 100mA et d'autres 500mA. Généralement les composants Low USB consomment une unité d'énergie et les composants High USB consomment jusqu'à cinq unités. Par défaut tous les composants consomment une unité et c'est par soft qu'on lui demande de consommer plus si l'application le nécessite, dans le cas d'un composant High USB bien sûr. Toutes ces informations sont contenues dans les descripteurs, le composant ne pourra jamais consommer plus que ce qui est prescrit dans son descripteur. On peut donc en déduire de cela qu'un périphérique High power doit posséder un dispositif d'alimentation séquentiel (100mA, puis 500mA).

Aucune fonction (Device ou hub) ne peut consommer plus de 100mA sur le bus avant d'être énumérée. Après énumération un Device peut consommer jusqu'à 500mA pour un device « High power device » ou rester à 100mA pour un Low Power Device.

L'USB est donc assez flexible et peut supporter plusieurs types d'alimentation.

- ❖ **Low-power bus-powered functions.** Les périphériques utilisés par un bus Low-Power puisent toutes leurs puissances de VBUS et ne peuvent consommer qu'une unité d'énergie. Les périphériques Low power alimentés par un Low Power Bus sont aussi conçus pour travailler avec une tension de VBUS s'échelonnant entre 4.4 V et 5.25V. Mais une tension de 4.4V suffit pour l'énumération. Beaucoup d'appareils fonctionnant à 3.3V doivent être munis d'un régulateur.
- ❖ **High-power bus-powered functions.** Les périphériques alimentés par un bus High-Power puisent toutes leurs puissances de VBUS et ne peuvent puiser qu'une unité d'énergie avant d'être configuré. Après la configuration, ils peuvent consommer jusqu'à 500mA. La seule condition est que ce soit défini dans les descripteurs. Les périphériques Low et High Power alimentés par un High Power Bus doivent eux aussi être détectés avec une tension s'échelonnant entre 4.75V et 5.25V.
- ❖ **Self-powered functions.** Ce type de périphérique à une alimentation mixte, c'est à dire qu'il peut absorber une unité d'énergie sur le Bus USB et le reste est tiré d'une alimentation extérieure. Il faut prévoir alors dans ce cas un peu de réserve car le bus ne délivrera pas plus qu'une unité. Ces types de périphériques « mixtes » sont les plus faciles à concevoir puisque la détection et l'énumération du périphérique peut se faire sans alimentation externe, puisqu'une unité suffit.

### Différents types de transfert

- ❖ **Transfert en mode Contrôle.** Ce mode de transfert est compatible avec le Low et Full Speed USB. Il est utilisé pour les opérations d'initialisations et de configurations. Il est éventuellement utilisable pour les transferts standards. Le mode contrôle est aussi utilisé pour tenter d'obtenir un débit Low Speed acceptable, ou pour utiliser le driver de classe HID standard.
- ❖ **Transfert en mode Interrupt.** Ce mode de transfert est également compatible avec le Low et Full Speed USB. Il est destiné à des échanges limités et périodiques, il garantit la fréquence de scrutation ainsi que la reprise sur les erreurs. Il est utilisé pour des transferts à l'initiative du périphérique (asynchrones) et pour des transferts périodiques ou permanents comme les claviers.
- ❖ **Transfert en mode Isochrone.** Ce mode de transfert est uniquement compatible avec le Full USB. La bande passante est garantie (début, latence), par contre dans ce mode il n'y a pas de reprise sur erreur. Il est utilisé pour des transferts nécessitant un flux régulier de données comme par exemple les caméras ou les téléphones ... La bande passante réclamée et non utilisée est perdue.
- ❖ **Transfert en mode Bulk.** Ce mode de transfert est uniquement compatible avec le Full USB. Ce mode est réservé pour les gros transferts de données (ex : imprimantes...) Le débit est variable et dépend de la disponibilité. Ce mode assure la reprise sur les erreurs. Les échanges isochrones sont les plus privilégiés dans le sens où le host leur réserve une bande passante garantie. Celui-ci peut refuser l'accès au bus à un périphérique s'il juge que les ressources qu'il requiert ne sont pas disponibles.

### L'énumération

L'énumération est la manière de déterminer l'appareil qui vient juste d'être branché au bus et les paramètres dont il a besoin, comme la consommation électrique, le nombre et le type de terminaison, la classe du produit, etc.... L'hôte attribuera donc à l'appareil une adresse et validera une configuration lui permettant de transférer des données sur le bus.

Une énumération sous Windows ordinaire implique les étapes suivantes :

- ❖ L'hôte ou Hub détecte la connexion d'un nouvel appareil via les résistances de rappel de l'appareil reliées sur les 2 fils de données. L'hôte attend au moins 100ms, le temps que la prise soit insérée complètement et que l'alimentation de l'appareil soit stabilisée.
- ❖ L'hôte émet un " reset " mettant l'appareil dans l'état par défaut. L'appareil peut maintenant répondre à l'adresse zéro par défaut.
- ❖ L'hôte (sous MS Windows) demande les 64 premiers octets du descripteur d'appareil.
- ❖ Après avoir reçu les 8 premiers octets du descripteur d'appareil, l'hôte émet immédiatement un autre reset sur le bus.
- ❖ L'hôte émet maintenant une commande SetAdress, mettant l'appareil dans l'état adressable.
- ❖ L'hôte demande la totalité des 18 octets du descripteur d'appareil.
- ❖ Puis il demande les 9 octets du descripteur de configuration pour déterminer la taille totale.

- ❖ L'hôte demande les 255 octets du descripteur de configuration.
- ❖ L'hôte demande l'un des descripteurs de chaînes s'ils étaient indiqués.

Tous les transferts d'informations durant cette phase d'énumération se font suivant le type Control. Il va de soi que tout composant USB doit pouvoir être capable de supporter ce type de transfert.

#### *II.6.4.3 Les descripteurs [5]*

On peut définir les descripteurs comme étant des blocs d'informations pré formatés. Tous les appareils USB ont une hiérarchie de descripteurs qui détaillent pour le compte de l'hôte des informations l'instruisant sur la nature de l'appareil, qui l'a réalisé, quelle version USB il supporte, de combien de manières il peut être configuré, le nombre de terminaisons (endpoints) et leurs types etc...

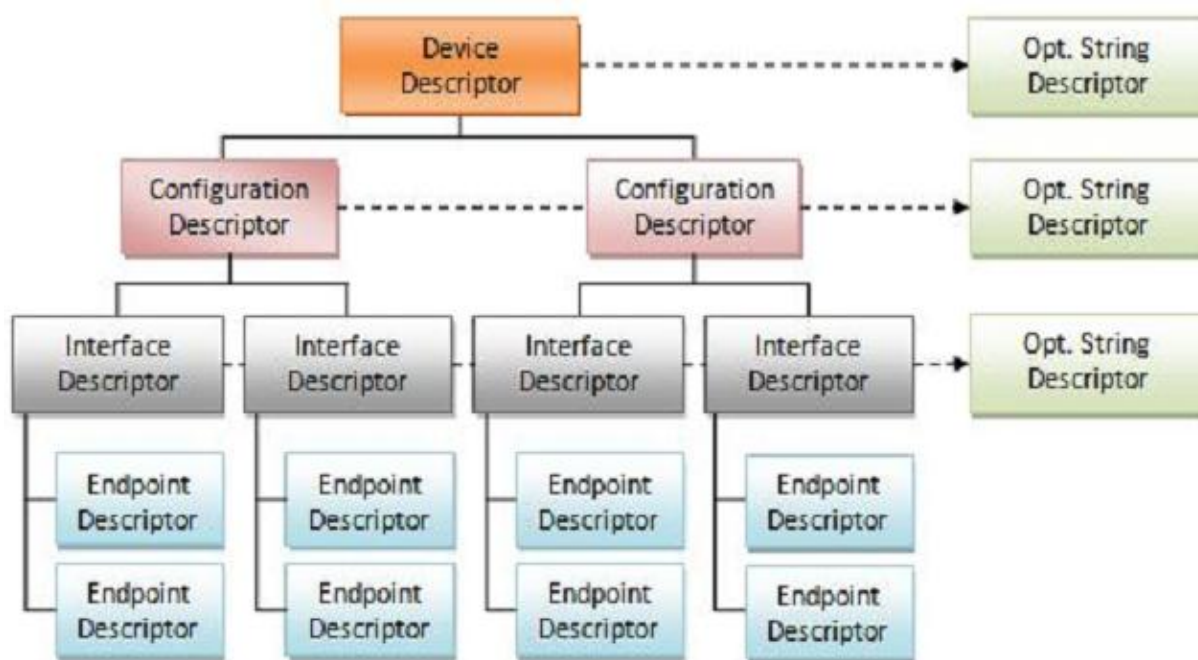
Généralement, dans la plupart des périphériques, toutes ces informations sont stockées dans la ROM des composants, et lors de l'énumération, le périphérique envoie simplement ce fichier pour se faire connaître.

Il existe plusieurs types de descripteurs, dont 4 de ces types sont standards (Device, Configuration, Interface, Endpoint). Chaque descripteur contient une valeur qui identifie le type de descripteur. Le **Tableau 8** montre les divers types de descripteurs avec leurs rôles et leurs valeurs.

Type	Valeur (hexa)	Descriptor
<b>Standard</b>	01	Device
	02	Configuration
	03	String
	04	Interface
	05	Endpoint
	06	Device_qualifier
	07	Other_speed_configuration
	08	Interface_power
<b>Class</b>	21	HID
	29	Hub
<b>Spécifique aux classes HID</b>	22	Report
	23	Physical

*Tableau 8 : Identification des descripteurs*

Ces descripteurs ont une hiérarchie bien définie (**Figure 34**).



*Figure 34 : Diagramme hiérarchique des descripteurs*

Expliquons à présent plus en détails quelques descripteurs standards et leurs rôles.

### Device Descriptor

Ce type de descripteur donne les informations générales. C'est le premier descripteur que vient lire le host. Il existe une très grande diversité de composants USB, il y a des propriétés communes à chaque dispositif d'USB comme par exemple le numéro de spécification USB qui est présent dans toutes les configurations, de même pour les numéros d'identifiant de produit et de vendeur (Product ID et Vendor ID). Un Ordinateur peut, uniquement avec ces informations contenues dans le « Device Descriptors », reconnaître un composant USB.

Il donne également des renseignements sur le canal ou pipe de communication par défaut qui est utilisé pour la configuration.

Voici un exemple des différents descripteurs faisant partie de la catégorie « Device Descriptor ».

Dans le champ description il y a la signification de chaque descripteur (**Tableau 9**).

Offset	Nom	Taille	Description
0	Blength	1	Taille du descripteur, toujours égale à 18 octets (0x12) pour un device descriptor.
1	BdescriptorType	1	Type du descripteur, toujours égale à 0x01 pour un device descriptor
2	BcdUSB	2	C'est la version de la spécification USB (codée en BCD) exemple : Version 1.0 -> 0100h Version 1.1 -> 0110h Version 2.0 -> 0200h
4	BDeviceClass	1	Si le composant appartient à une classe, ce champ renseigne sur cette classe. S'il n'appartient à aucune classe le champ doit être mis à 0. L'adresse 0xFF est réservée et signifie que la classe est spécifique au vendeur
5	BdeviceSubClass	1	Si un composant appartient déjà à une classe, ce champ permettra de restreindre encore une fois en définissant une sous classe. Si bDeviceClass = 0, BdeviceSubClass doit forcément être égal à 0 aussi.
6	BdeviceProtocol	1	Ce champ renseigne sur le protocole choisi par la classe ou la sous classe. C'est dans ce champ qu'un Hub USB 2.0 dira s'il supporte à la fois le Full et le High Speed.
7	BmaxPacketSize0	1	Taille maximum des paquets de l'Endpoint 0. Low Speed : 8 Full Speed : 8 – 16 – 32 – 64 High Speed : 64
8	IdVendor	2	Ce champ renseigne sur le Vendor ID (par exemple, VID = 0x04D8 pour Microchip)
10	IdProduct	2	Ce champ renseigne sur le Product ID (par exemple, PID = 0x0042 pour le PIC18F4550).
12	BcdDevice	2	(Device release number) Codé en BCD, ce champ, optionnel, renseigne sur le driver à charger par exemple
14	IManufacturer	1	C'est un index qui pointe sur un String qui donne le nom du Constructeur. Ce champ est optionnel. 0 si inutilisé.
15	Iproduct	1	C'est un index qui pointe sur un String qui donne des informations sur le produit. Ce champ est optionnel. 0 si inutilisé.
16	ISerialNumber	1	C'est un index qui pointe sur un String qui donne des informations sur le numéro de série du composant. Ce champ est optionnel. 0 si inutilisé. Ce champ peut être intéressant si l'on utilise plusieurs périphériques identiques possédant les même PID et PID.
17	BnumConfiguration	1	Nombre de configuration que supporte le composant.

*Tableau 9 : Tableau récapitulatif des différents champs des Device Descriptor*

Voici un extrait de programme écrit en C18 pour le Device Descriptor

```
/* Device Descriptor */  
ROM USB_DEVICE_DESCRIPTOR device_dsc=  
{  
    0x12,      // Taille du descripteur  
    0x01,      // Type du descripteur, toujours égale à 0x01  
    0x0200,    // version de la spécification USB 2.0 (en DCB)  
    0x00,      // il n'appartient à aucune classe  
    0x00,      // Subclass code, doit forcément être égal à 0  
    0x00,      // Protocol code, doit forcément être égal à 0  
    0x08,      // Taille maximum des paquets de l'Endpoint 0  
    0x04D8,    // Vendor ID, Microchip  
    0x0042,    // Product ID, PIC18F4550  
    0x0001,    // Device release number, champ optionnel  
    0x01,      // index qui pointe sur un String du constructeur  
    0x02,      // index qui pointe sur un String du produit  
    0x00,      // index qui pointe sur un String du composant  
    0x01       // Nombre de configurations possible  
};
```

### Configuration descriptor

Un descripteur de configuration renseigne sur les différents états dans lequel peut se trouver le composant USB.

Ces descripteurs de configuration définissent par exemple l'origine de l'alimentation, elle peut soit provenir d'une alimentation extérieure, soit directement du bus USB.

Il se peut qu'il y ait deux configurations différentes selon le type d'alimentation. En effet, si le système USB est directement alimenté par le bus USB il se peut en raison des conditions de puissance d'énergie, que le dispositif pourrait invalider quelques paramètres.

La numérotation des configurations commence à 1, la configuration 0 est une configuration réservée. Si le périphérique est dans cette configuration-là, on dit qu'il n'est pas configuré « unconfigured » et ne peut pas communiquer avec la host tant qu'il n'est pas sorti de cet état.

Voici un exemple des différents descripteurs faisant partie de la catégorie « Configuration Descriptor » (*Tableau 10*).

Offset	Nom	taille	Description
0	BLength	1	Taille du descripteur, toujours égale à 9 pour un configuration descriptor.
1	BDescriptorType	1	Type du descripteur, toujours 02h pour un configuration descriptor.
2	WTotalLength	2	Longueur total en bytes de toutes les données dans cette configuration. Cette longueur inclus les descripteurs d'interfaces, HID, reports et les EndPoint
4	BNumInterfaces	1	Nombre d'interfaces supportées dans cette configuration.
5	BconfigurationValue	1	Identifiant de Set_Configuration et de Get_Configuration
6	Iconfiguration	1	C'est un index qui pointe sur un String qui décrit cette configuration. Ce champ est optionnel. 0 si inutilisé.
7	BmAttributes	1	Précise si le bus alimente ou non le périphérique. Self Power ou Bus Power. Précise également si le périphérique supporte le remote wakeup.
8	MaxPower	1	Consommation du composant USB (en mA/2)

*Tableau 10 : Tableau récapitulatif des différents champs des Configuration Descriptor*

#### Interface descriptor

Une interface peut être considérée comme un ensemble d' «Endpoint ». Un Endpoint ou Terminaison est en quelque sorte l'extrémité d'un « pipe ». Un « pipe » est une sorte de tuyau par lequel transitent les données via le host. Un **descripteur d'interface** communique une information unique à tous ses Endpoints.

Si une configuration est choisie, toutes ses interfaces sont actives et par conséquent, aucun Endpoint ne peut être relié à des interfaces différentes sous la même configuration. Les interfaces sous différentes configurations peuvent partager des Endpoints.

Voici un exemple des différents descripteurs faisant partie de la catégorie « Interface descriptor » (*Tableau 11*).



offset	Nom	Taille	Description
0	Blength	1	Taille du descripteur, toujours égale à 9 pour un interface descripteur.
1	BdescriptorType	1	Type du descripteur, toujours 04h pour un interface descripteur.
2	BinterfaceNumber	1	Dans un device, une configuration possède plusieurs interfaces qui peuvent être actives en même temps. Ce champ est spécifié pour pouvoir reconnaître les interfaces de la même configuration.
3	BAlternateSetting	1	Cette valeur sélectionne par contre une interface bien précise.
4	BnumEndpoints	1	Nombre d'EndPoint supporté par l'interface. L'EndPoint 0 n'est pas compter. Si c'est le seul supporté BnumEndpoints = 0
5	BinterfaceClass	1	Si le composant appartient à une classe, ce champ renseigne sur cette classe. L'adresse FFh est réservée et signifie que la classe est spécifique au vendeur La class HID a le code 03h. L'adresse 00h est réservée.
6	BInterfaceSubClass	1	Si un composant appartient déjà à une classe, ce champ permettra de restreindre encore une fois en définissant une sous classe. Si BInterfaceClass = 0, InterfaceSubClass doit forcément être égal à 0 aussi. Si BinterfaceClass à une adresse entre 01h et FEh, bInterfaceSubClass dois être un code définit par la spécification USB. Une valeur FFh signifie que c'est une classe spécifique à un vendeur bien précis
7	BinterfaceProtocol	1	Ce champs renseigne sur le protocole choisi par la classe ou la sous classe.
8	Interface	1	C'est un index qui pointe sur un String qui décrit cette interface.

*Tableau 11 : Tableau récapitulatif des différents champs des Interfaces Descriptor*

### Endpoint descripteur

Un descripteur d'Endpoint indique la direction du transfert (IN ou OUT), ses types de transfert (ISOCHRONOUS, BULK, INTERRUPTION ou CONTROL), ainsi que d'autres informations qui sont regroupé dans le tableau suivant.

En fait, l'ordinateur « le host » communique uniquement avec ces Endpoints. Tous les transferts de paquet de données transitant sur le bus proviennent d'un Endpoint ou sont envoyés à un Endpoint. Généralement les Endpoints correspondent aux Entrées-Sorties ou au registre du dispositif USB.



Le nombre maximum d'Endpoint est différent selon que l'on utilise de l'USB Low Speed ou de l'USB High speed. Un dispositif USB High Speed peut supporter jusqu'à 15 Endpoints tandis qu'un dispositif USB Low Speed ne peut que supporter 3 Endpoints. Par contre, il est possible que deux Endpoints partagent le même numéro. Un Endpoint transitant les données dans un sens (IN par exemple) et l'autre pour les données transitant dans l'autre sens (OUT). Dans ce cas, il est nécessaire de définir deux descripteurs différents.

L'Endpoint 0 est un Endpoint particulier, c'est le seul qui est bidirectionnel et présent dans tous les dispositifs. Il est utilisé par le host pour contrôler le système. Il n'est cependant pas nécessaire de lui affecter un descripteur particulier. L'EndPoint 0 fonctionne toujours en mode Control.

Voici un exemple des différents descripteurs faisant partie de la catégorie « Endpoint descriptor» (**Tableau 12**).

Offset	Nom	Taille	Description
0	BLength	1	Taille du descripteur, toujours égale à 7 pour un Endpoint descriptor.
1	BdescriptorType	1	Type du descripteur, toujours 05h pour un EndPoint descriptor.
2	BendpointAddress	1	Renseigne sur le numéro de l'EndPoint et la direction
3	BmAttributes	1	Renseigne sur le type de transport utilisés : 00 = Contrôle    01 = Isochrone 10 = Bulk        11 = Interrupt
4	WMaxPacketSize	2	Spécifie la taille maximale des paquets <ul style="list-style-type: none"> <li>▪ Bits 0 à 9 : 0 à 1023</li> <li>▪ Bits 10 et 11 : indique le nombre de transactions supplémentaire par microframe (uniquement pour le High Speed)</li> <li>▪ Bits 12 à 15 : réservé.</li> </ul>
6	Binterval	1	Indique le temps maximal (en ms) entre le polling des EndPoints.

*Tableau 12 : Tableau récapitulatif des différents champs des Endpoint Descriptor*

### HID descriptor

La catégorie HID (Human Interface Device ) est un autre descripteur la plus utilisé à part les 4 descripteurs standards. En fait, elle regroupe tous les appareils qu'utilisent directement les personnes c'est à dire les souris, les claviers, les Gamepads, les écrans, etc.

D'après la spécification HID 1.1, les descripteurs relatifs aux HID sont envoyés après les descripteurs d'interface et avant les descripteurs d'Endpoints.

Voici un exemple des différents descripteurs faisant partie de la catégorie « HID descriptor» (**Tableau 13**).

Offset	Nom	Taille	Description
0	BLength	1	La taille varie selon le nombre de descripteur
1	BdescriptorType	1	Type du descripteur, toujours 21h pour un HID descripteur.
2	BcdHID	2	Valeur codée en DCB renseignant le type HID
3	BcountryCode	1	Optionnel, permet de localisé l'interface.
4	BnumDescriptor	1	Nombre de Classe de descripteur.
5	BdescriptorType1	1	Type de descripteur de la première class de descripteur
6	WDescriptorLength1	2	Longueur de la première classe de descripteur.

*Tableau 13: Tableau récapitulatif des différents champs des HID Descriptors*

#### *II.6.4.4 Le bus USB*

##### Principe du bus USB

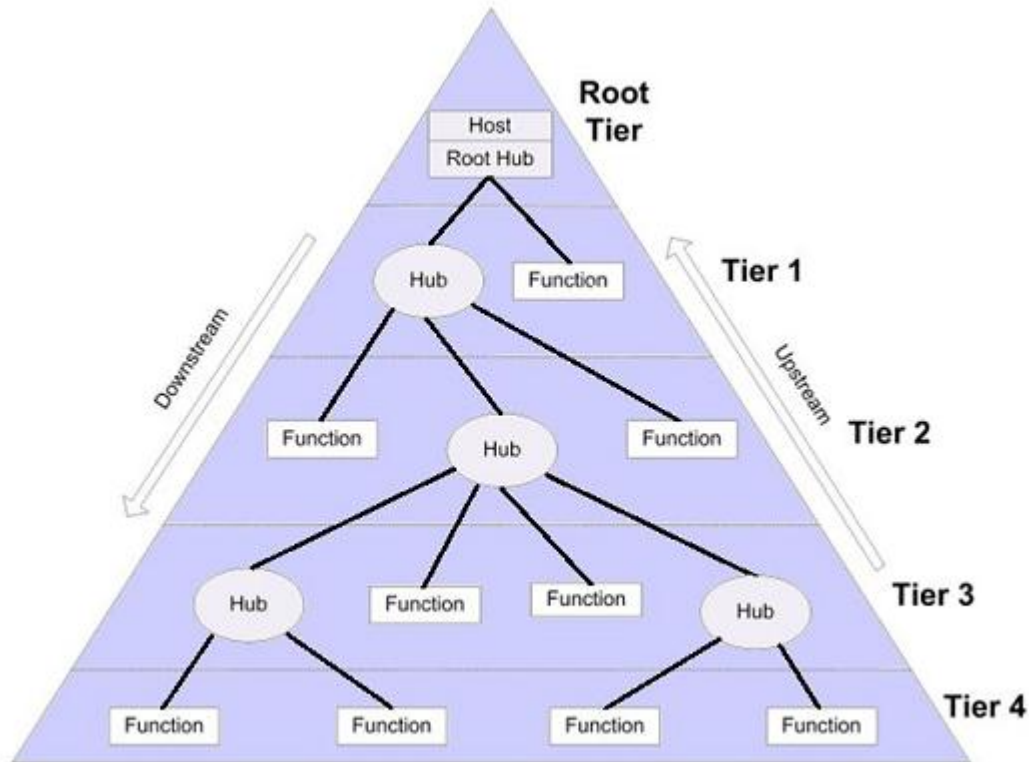
Le bus USB est un bus fonctionnant sur la hiérarchie, commandé par un host unique. Le host utilise un protocole Maître/Esclave pour communiquer avec les périphériques USB. Cela signifie que c'est le host qui décide du transfert des données et que les différents périphériques ne peuvent pas établir de connexion entre eux tant que le maître n'a pas donné l'autorisation. On peut dire que cela peut être un inconvénient par rapport aux autres protocoles mais il ne faut pas oublier que l'USB a été conçu avec des compromis de coût et de performance. Le fait que le bus USB fonctionne avec le protocole Maître/Esclave résout implicitement ces problèmes comme par exemples les problèmes de collision ou d'arbitrage de périphériques.

##### Topologie du Bus USB

La topologie est une topologie « étoile série » ou encore « tiered star » qui tolère jusqu'à 5 niveaux (tiers) de concentrateurs. La spécification limite le nombre de périphérique à 127.

Notons qu'à l'initialisation d'un "device" (ou d'une fonction selon la terminologie utilisée par le groupement USB) celui-ci reçoit une adresse. Les transferts Full Speed déclenchés par le "host" parcourent tous les tronçons, mais seule la fonction adressée va évidemment répondre. Ainsi s'il s'agit d'un ordre d'impression il sera reçu par tous les périphériques, mais seule l'imprimante le comprendra.

La **Figure 35** illustre cette topologie



*Figure 35 : Topologie du bus USB*

Les ports dirigés vers les appareils sont appelés ports descendants (Downstream) et ceux dirigés vers l'hôte ports montants (Upstream). Chaque hub contient un contrôleur surveillant les différents ports et qui rend des comptes à l'hôte. Ce dernier interroge les contrôleurs de hubs afin de connaître les connexions et les déconnexions des périphériques. Lorsqu'un appareil est connecté, l'hôte l'identifie et lui assigne une adresse unique : **c'est la phase d'énumération**. Pour la connexion d'un hub, il y a énumération de tous les appareils en aval. L'hôte attribue également la bande passante en fonction des types de transferts requis par les appareils.

### Protocole USB

Le protocole USB est, comme tous les autres protocoles, un protocole à encapsulation. Mais avant de s'intéresser à la trame proprement dite, voyons ce qu'il en est du protocole.

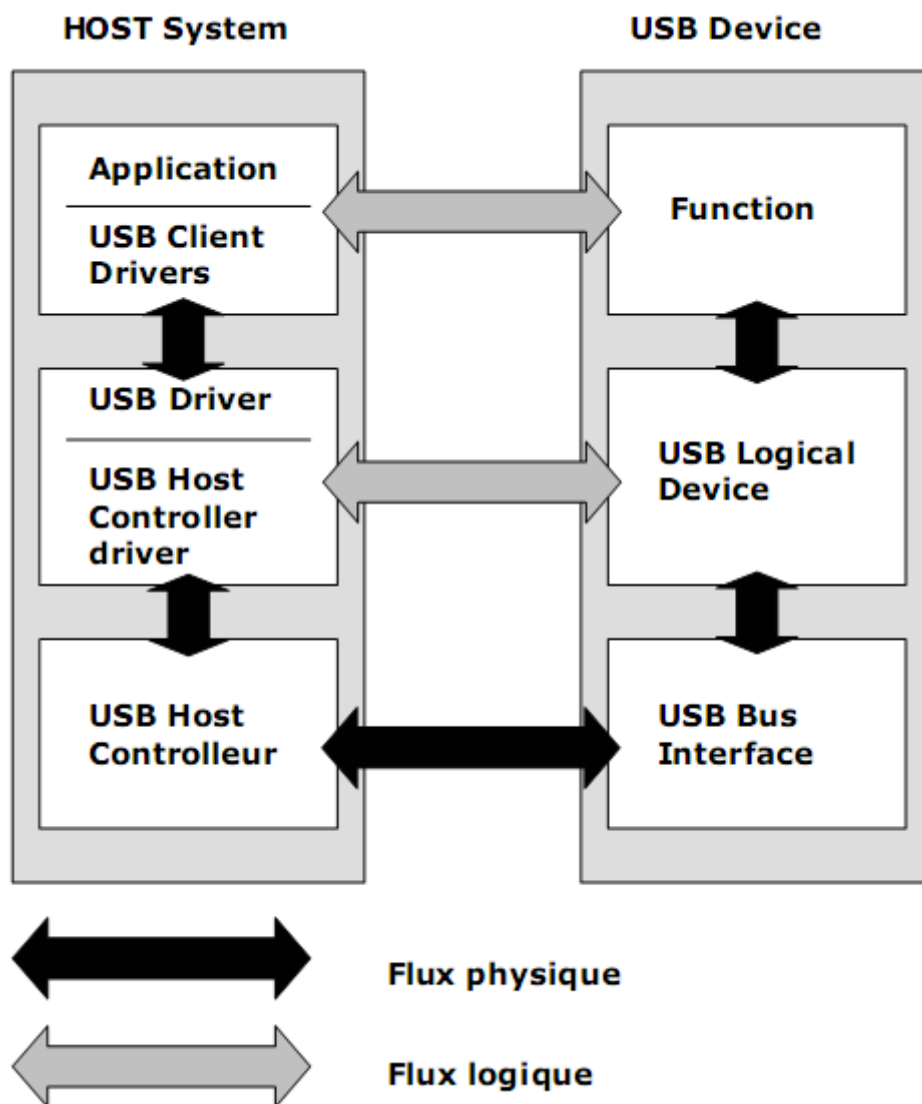
On peut représenter les échanges via USB en trois couches bien distinctes chez le host et chez les devices. Le **Tableau 14** ci-après décrit ces différentes couches avec leurs fonctions.

Le client driver communique les demandes de transfert des applications via des IRP (I/O Packet). Puis, l'USB driver traduit chaque transfert en une suite de transactions. Ensuite l'USB Host Controller driver regroupe les transactions en trames et finalement l'USB Host Controller traduit les transactions en paquets et enchaîne les trames.

<i>Couche</i>	<i>Host</i>	<i>Device</i>	<i>Fonctions</i>
<b>3</b>	Client	Function	Permet d'établir une relation fonctionnelle unique avec le dispositif.
<b>2</b>	USB System (USB Driver & USB Controller Driver)	USB Logical Device	Réalise l'étoile entre le maître et les différents dispositifs, et définit les transactions.
<b>1</b>	USB Host Controller	USB Bus Interface	Assure la connexion physique vers le bus arborescent USB.

*Tableau 14 : Différentes couches de l'échange via USB*

Le synoptique de la **Figure 36** ci-dessous traduit de façon imagée les liens entre les différents éléments.



*Figure 36 : Protocole USB*

## Type de paquet USB

Contrairement à la liaison série RS232 et des interfaces séries similaires où le format des données envoyées n'est pas défini, l'USB lui est composé de plusieurs couches de protocoles bien définis. La plupart des circuits intégrés USB s'occuperont de la couche inférieure, la rendant ainsi presque invisible au regard du concepteur final. Mais il est tout de même intéressant d'en connaître les grandes lignes.

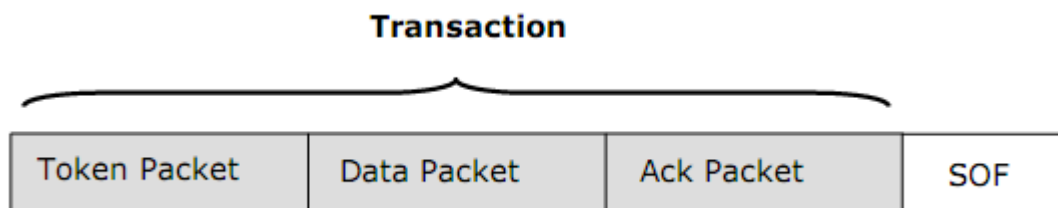
L'USB a quatre types différents de paquet :

- ❖ **Token ou Jeton (En-tête)**
- ❖ **Data (Optionnel)**
- ❖ **Acknowledge ou Etat (Handshake)**
- ❖ **SOF (Start of frame)**

Notons que l'USB est un Bus géré par l'hôte. L'hôte initie toutes les transactions. Le premier paquet, aussi appelé **Jeton** est produit par l'hôte pour décrire ce qui va suivre et si la transaction de données sera en lecture ou écriture et ce que sera l'adresse de l'appareil et la terminaison désignée. Le paquet suivant est généralement un **paquet de données** transportant la " charge utile " et est suivi par un **paquet " poignée de mains " (handShaking)**, signalant si les données ou le jeton ont été reçus correctement ou si la terminaison est bloquée, ou n'est pas disponible pour accepter de données. Et les **paquets début de trame (SOF)** indiquent le commencement d'une nouvelle trame.

L'entité de transfert USB est appelée transaction, elle est généralement constituée de paquets juxtaposés, la transition étant un paquet SOF qui indique le début d'une autre transaction. On dit alors qu'un transfert est composé d'une succession de transactions.

Voici ci-dessous le détail d'une transaction :



*Figure 37 : Transaction USB*

Les données sur le BUS USB sont transmises avec le bit LSB en premier.

Les paquets USB se composent des champs suivants :

- ❖ **Sync** : Tous les paquets doivent commencer avec un champ **Sync**. Le champ **Sync** fait de 8 bits de long pour la basse et pleine vitesse ou 32 bits pour la haute vitesse est utilisé pour synchroniser l'horloge du récepteur avec celle de l'émetteur / récepteur. Les 2 derniers bits indiquent l'endroit où le champ PID commence.

- ❖ **PID** : ici, PID signifie Packet IDentifier. Ce champ est utilisé pour identifier le type de paquet qui est envoyé.
- ❖ **ADDR** : Le champ adresse détermine à quel appareil le paquet est destiné. Sa longueur de 7 bits, lui permet de supporter 127 appareils. L'adresse 0 n'est pas valide, tant qu'un appareil qui n'a pas encore d'adresse attribuée, doit répondre aux paquets envoyés d'adresse 0.
- ❖ **ENDP** : Le champ de terminaison est composé de 4 bits, autorisant 16 terminaisons possibles. Les appareils basse vitesse, toutefois peuvent seulement avoir 2 terminaisons additionnelles au-dessus du canal de communication par défaut (4 terminaisons maximales).
- ❖ **CRC** : Les Contrôles à Redondance Cyclique sont exécutés sur les données à l'intérieur du paquet de charge utile. Tous les paquets jetons ont un CRC de 5 bits tandis que les paquets de données ont un CRC de 16 bits.
- ❖ **EOP** : Fin de Paquet. Signalé par une sortie unique zéro (SE0) pendant une durée approximative de 2 bits.

## II.7 Programmation du PIC 18F4550 [8]

### II.7.1 Les étapes de la programmation de pic

Les étapes nécessaires permettant d'avoir un programme s'exécuter sur un PIC sont :

- Ecrire un programme en langage C dans un fichier texte et le sauvegarder avec l'extension **.c**
- Compiler ce programme avec le plugin C18 fourni par Microchip. Le résultat est un fichier avec l'extension **.hex** contenant une suite d'instruction compréhensible par le pic.
- Transplanter le fichier **.hex** dans la mémoire programme du PIC (mémoire flash) à l'aide de programmeur de pic adéquat.
- Mettre le PIC dans son montage final, mettre sous tension.

### II.7.2 Programmation en langage C

Il y a de sacrés avantages : connaître la programmation en C permet de créer des programmes sur de multiples plateformes. Autrement dit, les connaissances utilisées sont en grande partie réutilisables. Le C est un langage de « haut niveau », comparé à l'assembleur, qui permet d'écrire des programmes nettement plus intelligibles et donc plus faciles à relire et corriger ou modifier. Le compilateur contrôle la cohérence du code au moment de la compilation et signale bon nombre d'erreurs, qui seront autant de bogues en moins à corriger. Le compilateur prend en charge la gestion d'un certain nombre de mécanismes fastidieux : par exemple, pas besoin de spécifier la page mémoire dans laquelle on veut écrire, le compilateur s'en charge ! De plus, certains compilateurs permettent tout de même d'accéder à des ressources de bas niveau, voir même d'insérer dans le code des portions d'assembleur. A vrai dire, on peut certainement créer avec un bon compilateur C un code plus propre et plus robuste, en nettement moins de temps.

Il y a sur le marché plusieurs compilateurs C. Le microcontrôleur utilisé ici étant un PIC Microchip, la solution la plus évidente consiste à voir ce que propose Microchip.

Notre choix s'est porté sur le compilateur C18 de la société Microchip, pour les raisons suivantes : Il existe une version gratuite avec quelques limitations mineures (taille du code généré...) permettant de se faire la main et de créer des applications personnelles, les versions payantes (à des tarifs très raisonnables) permettent de créer des applications commerciales et d'avoir accès au code source des bibliothèques. C18 est un plugin de l'environnement de développement intégré MPLAB facilitant le développement d'applications. Et surtout, C18 reste assez proche du matériel : on peut très facilement accéder à tous les bits d'un registre quelconque. On peut faire l'essentiel de ce qu'il est possible de faire en assembleur, avec les contraintes en moins. Ce qui sous-entend néanmoins de s'intéresser d'assez près à la structure du PIC et des différents sous-ensembles.

#### *II.7.2.1 L'environnement MPLAB*

Un système de développement comporte en premier lieu un assembleur et parfois un ou des compilateurs adaptés au langage évolué que l'on souhaite utiliser pour programmer.

L'assembleur traduit les instructions écrites en utilisant les mnémoniques du langage machine en code binaire exécutable par le microcontrôleur.

Pour développer des applications en assembleur ou en C, le choix, d'un tel logiciel, s'est porté sur l'outil MPLAB mis à disposition gratuitement sur Internet par la société Microchip. Ce logiciel fonctionne sur PC, sous le système d'exploitation Windows de Microsoft.

Cette logicielle comprend tout ce qu'il faut pour assembler les programmes : éditeur de texte, assembleur, simulateur, éditeur de registres, etc. Il supporte aussi plusieurs émulateurs et compilateurs.

Il offre beaucoup de flexibilité aux développeurs, notamment grâce aux nombreuses fenêtres pouvant être ouvertes à tout moment lors d'une mise au point, permettant de voir le contenu d'un quelconque emplacement mémoire et d'un quelconque registre.

MPLAB possède plusieurs points d'entrée : en ce sens qu'on peut effectuer la même tâche en empruntant des chemins différents, sans obligatoirement obéir à un mode d'emploi préétabli. Il suffit, la plupart des fois, d'adapter et de croiser les grilles écran ouvertes sur le bureau.

Les étapes nécessaires permettant d'avoir un programme s'exécuter sur un PIC sont :

- Ecrire un programme en langage assembleur ou C dans un fichier texte et le sauvegarder avec l'extension **.asm** ou **.c**.
- Compiler ce programme avec l'assembleur MPASM ou C18 fourni par *Microchip*. Le résultat est un fichier exécutable avec l'extension **.hex** contenant une suite d'instruction compréhensible par le pic.



- Transplanter le fichier .hex dans la mémoire programme du PIC (mémoire flash) à l'aide d'un programmeur adéquat. On peut utiliser les programmeurs de *Microchip* ou tout autre programmeur réalisé par soit même.
- Mettre le PIC dans son montage final, mettre sous tension et admirer le travail.

Pour implanter le binaire exécutable, nous avons besoin d'un programmeur. Ce matériel est en fait un système qui va transférer de la machine hôte, le code objet du programme dans la mémoire du microcontrôleur. MPLAB est un outil de développement spécifique aux microcontrôleurs PIC, conçu et fourni gratuitement par *Microchip*. Il permet d'écrire, mettre au point et optimiser les programmes, sous Windows.

En effet, en plus d'un éditeur et d'un assembleur, il inclut un ensemble d'outils permettant non seulement de fabriquer le code objet d'une application, mais aussi de simuler le programme, c'est à dire le voir dérouler à l'écran.

Il offre beaucoup de flexibilité aux développeurs, notamment grâce aux nombreuses fenêtres pouvant être ouvertes à tout moment lors d'une mise au point, permettant de voir le contenu d'un quelconque emplacement mémoire et d'un quelconque registre.

#### II.7.2.2 Le programmeur PIC Pickit3

Les microcontrôleurs PIC fabriqués par la société *Microchip*, comme par exemple le 18F4550, sont bien connus des électroniciens. Leurs qualités techniques en font des composants de choix pour la réalisation de notre système d'instrumentations. Bien souvent, leur programmation s'effectue en retirant le composant du montage et en l'insérant sur le support d'un programmeur relié par ailleurs à un ordinateur. Nous allons utiliser le **Pickit3** de Microchip comme programmeur.

### II.8 Dimensionnement de l'alimentation

Afin de pouvoir alimenter tous les circuits de la carte d'acquisition ainsi que les capteurs qui seront branchés dessus, il nous a fallu réaliser une alimentation 5V pouvant tourner 24h/24 et 7j/7. Pour ce faire, nous avons d'abord fait le bilan des consommations maximales que notre alimentation aura à fournir. Nous avons donc à alimenter :

• Un Pic 18F4550	10mA
• Un LCD	8mA
• 2 LEDs	2*11=22mA
• Girouette	34mA
• Anémomètre	12mA
• Thermomètre	17mA
• Amplificateur d'instrumentation	82mA
• <b>Total</b>	<b>185mA</b>

Notre alimentation aura à fournir un maximum de 200mA, ce qui n'est pas excessif. Pour la réaliser, on optera donc pour une alimentation linéaire qui est certes moins performante qu'une alimentation à découpage, mais bien plus facile à mettre en œuvre, très fiable, et qui dans le cas de la consommation qu'elle devra fournir est tout indiquée.



## II.9 Programmation host [11]

### II.9.1 Operating System (OS)

On parle essentiellement de l'OS Windows NT avec l'USB, ces deux mots sont en fait très liés car le créateur de l'USB est en fait en grande partie Microsoft. Mais il faut aussi savoir que l'USB existe également sur les iMac et autres plates-formes.

Windows NT « New Technology » est un système d'exploitation développé par Microsoft indépendamment du système historique MS-DOS (DOS : Disk Operating System, 16 bits). Les OS tels que Windows 2000, XP, Vista, Windows 7 et 8 sont basés sur Windows NT.

L'OS Windows NT détient les caractéristiques suivantes :

- ✚ **Orienté objet** : les processus, les threads, et même les fichiers sont tous des objets ;
- ✚ **Multitâche préemptif** : il est capable d'exécuter plusieurs applications et plusieurs processus sur une même machine;
- ✚ **Multiutilisateur** : il peut être utilisé par beaucoup d'utilisateurs en même temps
- ✚ **Multiprocesseur** : il autorise l'utilisation de plusieurs (2 à 32) processeurs en parallèle

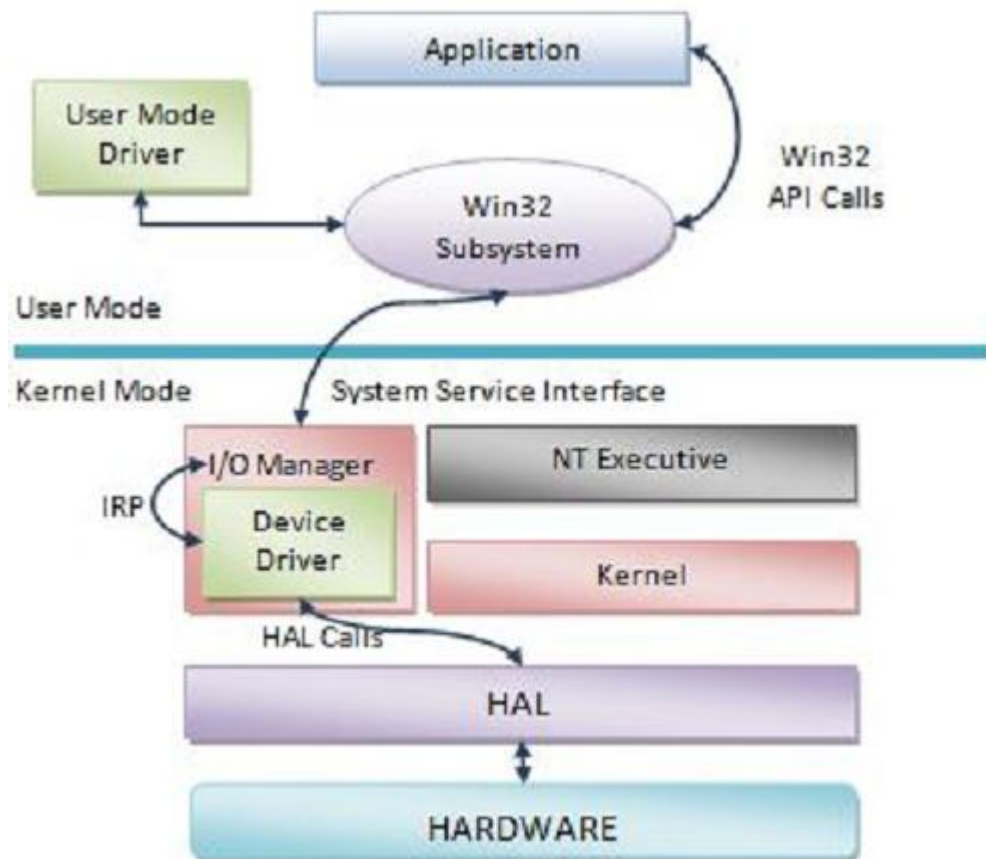
Le système NT travaille dans deux modes de fonctionnement :

- **Kernel mode** (mode noyau) : ou mode système, qui nous permet l'exécution de toutes les instructions, même les instructions privilégiées. Et on peut accéder à toutes les ressources de la machine sans restriction;
- **User mode** (mode utilisateur) : ou mode application, qui pour des raisons de sécurité, n'autorise pas l'accès et l'utilisation des instructions privilégiées directement à toutes les ressources de la machine. Il est imposé de passer par des appels système qui contrôleront et effectueront les opérations demandées, en particulier tous les accès au matériel.

Windows NT est formé par une superposition de couches (figure 3.1), et met à disposition des fonctions pour la communication entre les couches, qui sont :

- ❖ **Matériel**
- ❖ **Couche d'abstraction matérielle** (HAL : Hardware Abstraction Layer) : fournissant des fonctions pour accéder au matériel (temporisateurs, bus, DMA, contrôleurs d'interruption,...)
- ❖ **Noyau** (kernel), **le gestionnaire d'Entrée/Sortie** (I/O Manager) et **les pilotes de périphériques** (device drivers). Le noyau gère les interruptions et les exceptions, l'ordonnancement des threads, les synchronisations entre les processus et la gestion du temps (time keeping )
- ❖ **NT Exécutive** : qui gère la mémoire virtuelle, les processus, les communications interprocessus, la sécurité, les objets et les entrées-sorties
- ❖ **Couches tournant en User mode** : qui sont des sous-systèmes regroupant diverses interfaces : Win32, OS/2, POSIX et les applications utilisateur.

Schématiquement, Windows NT se présente comme le montre la **Figure 38**



*Figure 38 : Architecture de Windows NT*

Windows 98 est le premier operating system capable de gérer correctement l'USB, les successeurs Windows 2000, Windows Millenium et Windows XP... l'intègrent correctement également et sont plus riches en drivers pré-installés.

Donc il faut respecter trois conditions pour prétendre être un operating system capable de gérer l'USB. D'une part il faut qu'il puisse gérer le branchement et le débranchement de périphérique. D'autre part il faut qu'il puisse communiquer avec tous les nouveaux périphériques qui viennent d'être branchés pour trouver le moyen le plus adéquat pour pouvoir transférer des données. Et finalement, il doit pouvoir produire un mécanisme permettant aux drivers de communiquer avec le Host et le périphérique USB, que l'on appelle généralement l'énumération.

A un niveau plus élevé, on peut aussi dire qu'un OS gérant l'USB doit contenir des drivers pour différents périphériques, qui font le lien entre l'operating system et le composant USB. Si l'operating system ne possède pas le driver par défaut du périphérique à installer, c'est le fabricant du périphérique qui doit le fournir. Cependant, il est également possible de faire « soi-même » un driver via un outil de Microsoft (DDK : Device Developer's Kit).

Les drivers de composants USB utilisent le nouveau Win32 Driver Model (WDM) qui est une architecture de drivers fonctionnant sous les différentes versions de Windows à partir de Windows 98. Le but de l'utilisation de ce WDM est de fournir un driver unique.

Actuellement il existe encore des drivers différents pour les différentes versions de Windows. Il est même possible d'écrire des drivers USB sous DOS.

### *II.9.2 Les fichiers .INF [12]*

L'abréviation INF provient du mot « Information » et comme son nom l'indique, renseigne sur les **informations de configuration** de périphérique utile pour le **Plug & Play**.

Les points INF ne sont pas seulement utilisés pour les composants USB, les périphériques PCI l'utilisent également. Un fichier INF est un fichier texte qu'on peut l'éditer avec n'importe quel éditeur de texte standard.

En règle générale, les fichiers INF utiles à l'installation sont déjà présents dans les fichiers INF par défaut dans Windows. Les Fichiers INF par défaut sont des fichiers INF génériques c'est à dire qu'ils sont capable de débiter l'installation de plusieurs types de périphérique.

Un fichier INF est organisé en plusieurs sections. Chaque section (il en existe une vingtaine) possède une fonction particulière. Ces fonctions vont de la simple manipulation de fichiers jusqu'à la modification d'entrées dans la base de registres en passant par les fichiers .INI.

#### Règles à respecter et recommandation pour concevoir un fichier INF

Les fichiers INF doivent respecter les règles suivantes pour ne pas générer d'erreur lors de la création de fichier INF :

- ❖ Les sections commencent avec un nom de section entouré de crochets.
- ❖ Chaque fichier INF doit contenir la section [Version] qui identifie la compatibilité du fichier avec Windows 95 ou Windows NT.
- ❖ L'utilisation de variable est possible en utilisant la syntaxe %nom\_de\_la\_variable%.
- ❖ Les variables sont définies dans la section [Strings]. Pour utiliser le caractère % dans une chaîne, il faut utiliser la syntaxe suivante : %%
- ❖ Les lignes commençant par un point-virgule sont des commentaires.
- ❖ Un anti-slash (\) en fin de ligne indique que la suite de la ligne continue à la ligne suivante sauf s'il est entouré de guillemet ("'\")
- ❖ Les lignes vides sont ignorées.
- ❖ Le fichier INF ne peut pas dépasser 64 KO.
- ❖ Un seul fichier INF est exécuté à la fois !
- ❖ Penser à utiliser les guillemets " ", voire des doubles guillemets, surtout pour les textes avec espaces.
- ❖ Le chemin d'accès du fichier INF, ne doit pas être au format Nom Longs de Windows 95...
- ❖ Utiliser des majuscules pour les noms de fichiers courts (8+3).

- ❖ Faire bon usage des majuscules (par exemple, CurrentVersion au lieu de currentversion DisableRegistryTools au lieu de disableregistrytools, ...)
- ❖ Ne mettre aucun espace avant et après le signe '='
- ❖ Ne mettre un espace avant la virgule et mettre un espace après la virgule
- ❖ Séparer les différentes sections par une ligne vierge
- ❖ Le texte est entouré de % (%exemple\_de\_texte%).

### Applications des fichiers INF

Les fichiers INF ont diverses applications, ils servent pour :

- L'installation d'un driver (Détection d'un nouveau périphérique) ou d'un module Windows.
- L'Installation automatique d'un programme (notamment pour installer un programme à l'insu d'un utilisateur).
- La Modification d'un INI ou d'une entrée dans la base de registre lors du script de connexion.

### Rôle du fichier INF

Lors du branchement d'un périphérique USB, les descripteurs transmettent le couple PID/VID au PC. A ce moment-là, le PC scanne dans tous ces répertoires s'il trouve un fichier « d'information » (INF), qui comporte le couple PID/VID qu'il a reçu par les descripteurs. S'il ne trouve pas les fichiers concernés, il ouvrira une fenêtre Windows classique disant qu'il n'a pas réussi à trouver le driver et qu'il faut le chercher manuellement. Dans ce cas, soit on a ce fichier INF sur CD-ROM ou sur disquette et on lui indique le chemin qu'il copiera, ainsi que d'autres fichiers placés dans ses répertoires de drivers. Soit on ne possède pas ce fichier et le périphérique ne s'installera pas.

Le fichier INF est le point de départ de toute installation, s'il n'est pas là, il ne peut pas indiquer au système d'exploitation quel driver charger pour faire fonctionner l'application.

Ensuite il faut aussi noter qu'il existe des fichiers INF, déjà présent dans le système d'exploitation, qui n'ont pas de couple PID/VID bien précis, mais des plages de valeurs pour les PID et les VID. Cela permet en fait d'utiliser un seul fichier INF pour différents périphériques de même type (donc utilisant le même driver) mais de constructeurs différents.

Une fois que le fichier INF a été trouvé, il exécute toutes les commandes qui lui sont dictées. Les fichiers d'extension « INF » ont tous une commande qui demande de charger le driver en question. Ce driver a pour extension l'extension SYS. C'est ce fichier qui fait le lien entre le hardware c'est à dire le composant physique et le software. C'est le fichier INF qui reconnaît le composant et qui ordonne à l'OS d'installer son driver.

### *II.9.3 Les drivers .SYS [12]*

#### Définitions d'un driver .SYS

Un driver a le plus souvent l'extension .SYS. Ce driver doit être fourni avec le périphérique à installer... et c'est le plus souvent le cas.

Comme les .INF il existe aussi des .SYS par défaut. Par contre, contrairement au fichier INF, il n'est pas si évident de concevoir un WDM (Windows Driver Model). Il nécessite des compétences en programmation C et des connaissances sur la communication entre Windows, le hardware et les applications.

Par contre, il est beaucoup plus facile d'écrire un driver USB qu'un driver ISA. Dans la nature, un fichier SYS est un fichier compilé et il est impossible de le modifier pour l'adapter à un autre type de périphérique.

#### Création d'un driver avec le DDK [13]

Il existe une solution pour créer des points SYS, c'est en utilisant le DDK (Device Developer's Kit), un produit de Microsoft Developer's Network (MSDN). Il faut par contre, pour pouvoir utiliser le DDK, posséder Microsoft Visual C++ qui est capable de compiler des drivers WDM (Windows Driver Model).

Le DDK est en fait une sorte de base de données de code et de documentation sur différents types de périphériques. En ce qui concerne l'USB par exemple, le DDK inclut des fichiers, qui en les compilant, génère un fichier bulkusb.sys. Ce driver est un driver qui est capable de piloter un composant USB avec le type de transfert BULK. Un autre par exemple, isousb.sys qui permet de piloter de l'USB de type ISOCHRONE. Le DDK est téléchargeable sur le site de Microsoft.

## **II.10 Simulation du système**

La simulation est la reproduction artificielle et aussi réaliste que possible (d'un processus complexe) à des fins scientifiques, ludiques ou de formation. C'est aussi une représentation de la réelle assise sur un modèle théorique sous-jacent (ou caché), et elle dépend du modèle. La simulation est applicable sur divers modèles théoriques comme : le physique, la chimie, la biologie, et aussi sur les systèmes humaines (économie et sciences sociales).

Il existe diverses raisons qui nous poussent à simuler un système (électronique, électrique,...). Ces raisons sont :

- **Pour comprendre** : le désir de comprendre le fonctionnement d'un système (montage électrique) ;
- **Pour prévoir et prévenir** : anticipation de l'expérimentation, afin de la faciliter (choix des composants), et de la sécuriser ;
- **Pour comparer** : si la simulation est faite après une expérimentation, pour valider un modèle par exemple ;
- **Pour des raisons économiques** : notamment dans l'enseignement, pour certains montages gourmands en composants, alimentations, charges (onduleur,...). La simulation est alors nécessaire, et permet une économie surtout avec les logiciels gratuits.

La simulation de la carte d'interface USB a besoin d'un logiciel capable de simuler l'interface USB. Dans cet ouvrage, on a choisi d'utiliser le Proteus de Labcenter. Ce dernier

dispose deux applications dont : ARES pour le traçage des circuits imprimés et ISIS pour la simulation des composantes électroniques, et c'est ce dernier qui nous intéresse.

ISIS fournit l'environnement de développement pour Proteus VSM, qui est notre simulateur interactif évolutionnaire au niveau du système. Il permet de concevoir et de simuler les circuits par l'utilisation de ses environnements graphiques (symboles représentatifs des composants), et aussi de visualiser les graphes des signaux obtenus en simulation.

ISIS est aussi capable de simuler convenablement le fonctionnement des microcontrôleurs populaires (PIC, ATMEL-AVR, MOTOROLA, 8051, ...).

L'*Annexe G* représente quelques captures d'écran lors de la *Simulation*.

## **PARTIE III**

### **REALISATION, ETALONNAGE ET INTERPRETATION DES RESULTATS**

### III.1 Réalisation du système

#### III.1.1 L'anémomètre

A part la partie logicielle, le système anémométrique est composé de deux sous-sections :

- La partie électrique
- La partie mécanique

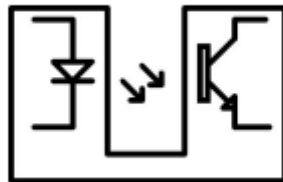
Chaque sous-section est décrite une à une ci-dessous.

##### *La partie électronique de l'anémomètre*

L'anémomètre utilise un capteur infrarouge EE-SX1115 (**Figure 39**) et doit fournir des niveaux logiques en sortie. Ce capteur infrarouge fourche est composé d'une partie émettrice (une LED infrarouge) et d'une partie réceptrice (un phototransistor), le tout moulé dans un petit boîtier compact.

La partie émettrice se comporte de la même façon qu'une DEL standard à la différence près que la lumière est invisible à l'œil.

La partie réceptrice quant à elle se comporte comme un interrupteur commandé par la réception ou non de lumière infrarouge. Quand elle reçoit de la lumière infrarouge, le capteur est à l'état ON sinon il est l'état OFF.

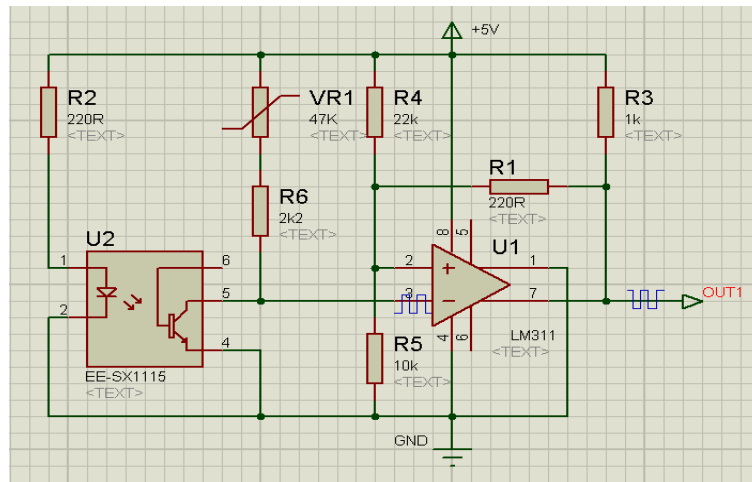


*Figure 39 : Capteur infrarouge EX-SX1115 de l'anémomètre*

Une rotation du rotor d'anémomètre donnera donc 4 cycles de variation de courant de sortie du phototransistor.

Avant que le signal du phototransistor puisse être traité il doit d'abord être converti en forme numérique. Ceci est fait par un comparateur LM311 comme l'indique la **Figure 40**.



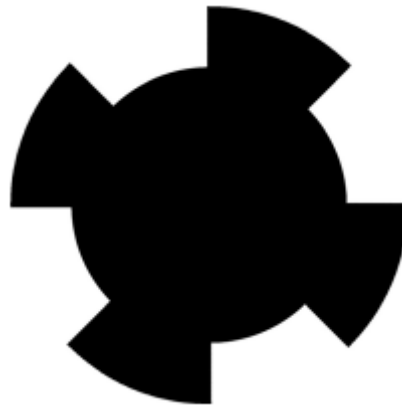


*Figure 40 : Tête de l'anémomètre sous Proteus*

Maintenant, les impulsions numériques peuvent être transmises au port d'entrée RB0 du microcontrôleur 18F4550.

#### *Réalisation de la partie mécanique*

La première étape dans la conception mécanique de l'anémomètre fut la conception de la roue de codage optique. Pour faire cela, on a utilisé de l'époxy de circuit imprimé que l'on a insolé suivant les dessins de codage (**Figure 41**).



*Figure 41 : Roue de codage de l'anémomètre*

Il fallut ensuite trouver le moyen de faire un axe dont les frottements seraient quasiment nuls et ne possédant pratiquement pas d'inertie. On a opté pour l'utilisation d'un système roulant basé sur deux roulements à billes. Il nous fut alors facile de fixer la roue de codage sur cet axe. Le prolongement supérieur de l'axe supporte la roue à godets de l'anémomètre. Cette dernière est construite avec 4 demi-balles de ping-pong.

Le typon et la liste des composants se trouvent dans l'**Annexe C**.

### III.1.2 La girouette

#### La partie électronique de la girouette

Du point de vu électronique, les capteurs infrarouges sont eux même branchés sur la base d'un transistor de type 2N2222 qui fonctionne alors en commutation. (**Figure 42**)

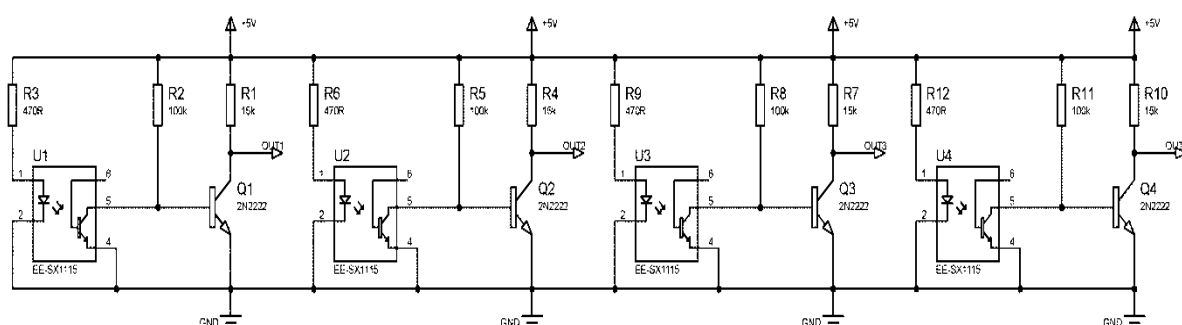


Figure 42 : Schéma de câblage du capteur de la girouette

#### Réalisation de la partie mécanique

La partie mécanique utilise un disque de Gray

L'alternance bien définie d'arcs opaques et d'arcs transparents permet d'identifier numériquement la position angulaire de la girouette. (**Figure 43**)

Ce codage de 4 bits permet d'avoir 16 directions de vent sur 0 à 360°, soit un pas de 22.5°.

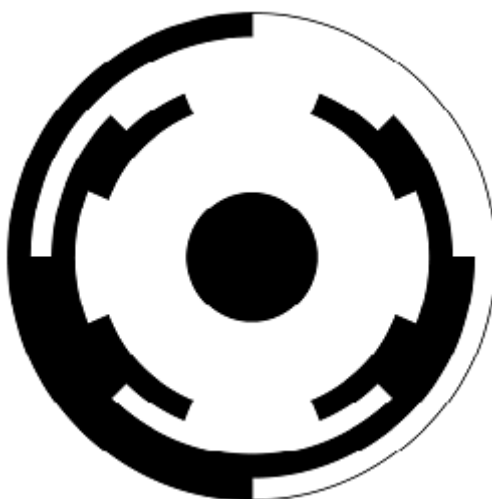


Figure 43 : Schéma du disque de codage de la girouette

Le typon et la liste des composants se trouvent dans l'**Annexe D**.

### III.1.3 Le thermomètre [21]

La tête du thermomètre est basée sur l'utilisation de LM35.

La valeur de température est obtenue en lisant directement la tension analogique en sortie des capteurs de température LM35. Ces derniers délivrent une tension qui est proportionnelle à la température, à raison de 10 mV par °C.

Le convertisseur A/N intégré au PIC18f4550 travaille en effet sur 10 bits (1024 pas de quantification) et pour une pleine échelle de mesure de 5 V cela donne environ 5 mV de résolution. Ce qui est une valeur élevée en regard de nos 10 mV par °C. Alors, nous devons amplifier la sortie du capteur LM35 pour avoir un gain en tension égale à 5 fois. Ainsi, la tension appliquée à l'entrée du CAN du PIC n'est plus de 10 mV par °C mais de 50 mV par °C. Ceci est fait par le fameux amplificateur d'instrumentation de précision TS951IN (voir la **Figure 44**).

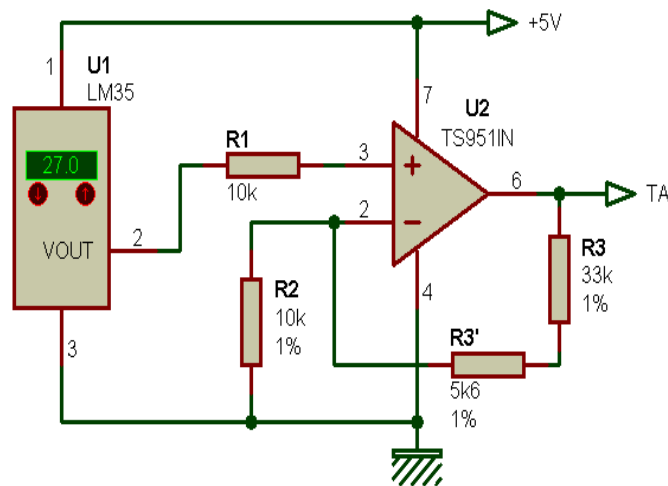


Figure 44 : Tête du thermomètre

Le typon et la liste des composants se trouvent dans l'**Annexe E**.

### III.1.3 Réalisation de la carte d'acquisition [24]

Afin de répondre au cahier des charges, la carte d'acquisition devra :

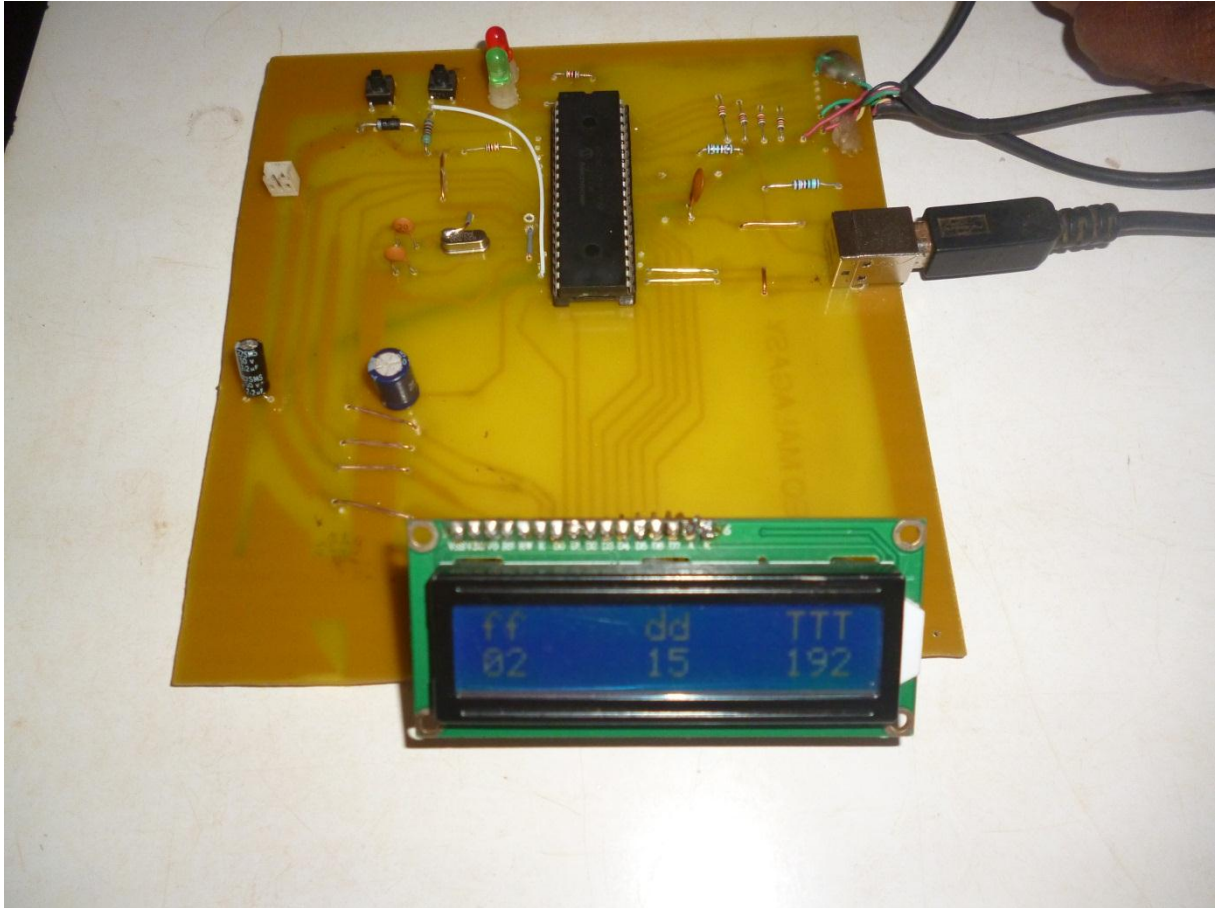
- Faire une acquisition des données numériques et analogiques.
- Réaliser des transferts vers le PC via le port USB 2.0.

Pour pouvoir réaliser des acquisitions numériques et analogiques, la carte est construite autour d'un PIC 18F4550 cadencé à 20 MHz grâce à un résonateur.

Elle peut discuter avec le PC via le port USB en utilisant deux ports du microcontrôleur (RC4 pour D- et RC5 pour D+).

Un afficheur LCD 16 lignes et deux colonnes permet à l'observateur de visualiser les paramètres de vitesse **ff**, de direction **dd** et de température **TTT**.





*Figure 46 : Photos de la carte finie*

### III.1.3 Réalisation de l'alimentation

Comme nous avons déjà parlé dans la partie « Dimensionnement de l'alimentation », nous utilisons une alimentation linéaire à base du régulateur 7805.

Le schéma de notre alimentation est donné par la **Figure 46**.

#### Choix du régulateur

Il s'agit d'un régulateur de tension fixe 5V. Nous avons opté pour le modèle LM78M05CT car il permet de délivrer jusqu'à 1A, ce qui est très largement suffisant, et par ce que nous en disposions déjà.

#### Choix du transformateur

Le régulateur délivrant une tension continue et fixe de 5V, il était nécessaire d'avoir un transformateur pouvant délivrer au moins 3V de plus que la tension régulée par le régulateur afin de garantir son bon fonctionnement. Nous avons opté pour le transformateur 220V $\approx$ 2 $\times$ 9V 5VA dont les deux enroulements secondaires sont branchés en point milieu à la masse. Ce transformateur peut délivrer jusqu'à  $5/5=1A$  soit 1000mA ce qui est plus du quadruple que ce dont nous avons réellement besoin.

### Choix de la capacité de lissage

La réserve d'énergie permettant d'avoir une tension à peu près constante à l'entrée du régulateur, sa valeur doit être assez élevée, d'où l'utilisation d'un condensateur électrolytique de valeur 470 $\mu$ F.

### Choix de la capacité de filtrage

Une capacité de filtrage en sortie du régulateur est utile pour, à la fois servir de réserve d'énergie immédiate pour l'ensemble du système, lorsqu'un composant en a besoin, et à la fois éliminer les parasites qui n'ont pas été régulés par le régulateur. Pour cette raison il faut utiliser un condensateur dont la valeur n'est ni trop faible, ni trop forte. Nous avons pris un condensateur de 10 $\mu$ F qui nous paraît bien répondre aux deux caractéristiques.

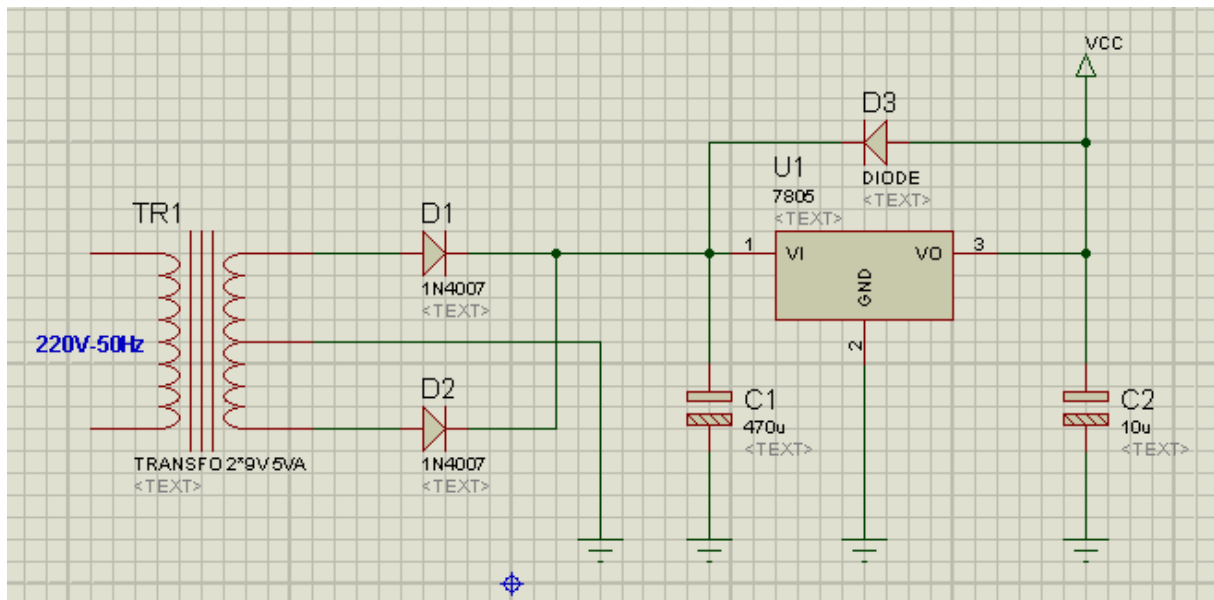


Figure 47 : Schéma de l'alimentation linéaire 5V à point milieu

Le typon et la liste des composants se trouvent dans l'**Annexe F**.

## III.2 Etalonnage et Interprétation des résultats

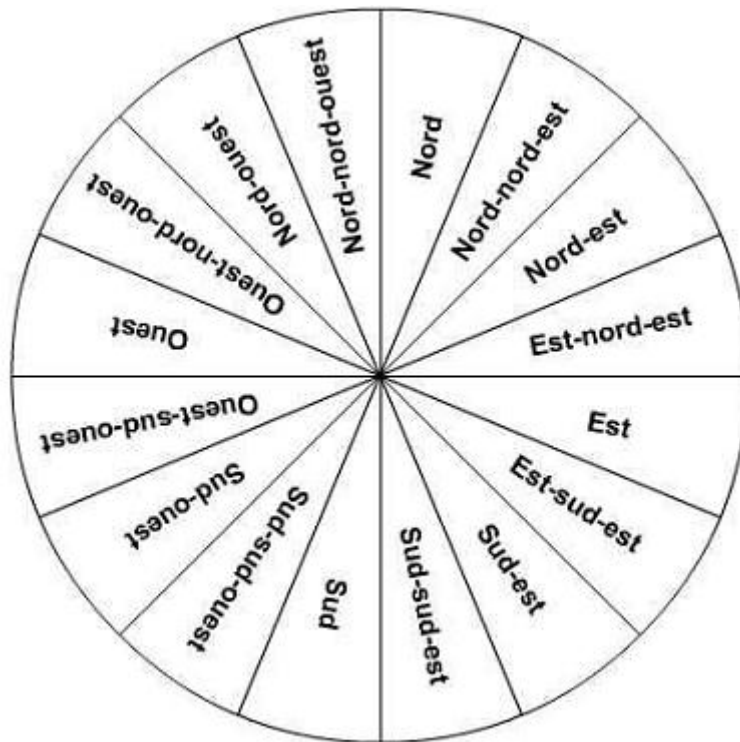
### III.2.1 L'étalonnage

Une fois la girouette, l'anémomètre, le thermomètre et la carte d'acquisition réalisés, il a fallu procéder d'abord à un test, puis à un étalonnage. Ce test en situation réelle était nécessaire afin de vérifier si le système fonctionne correctement et l'étalonnage a permis de faire correspondre les valeurs envoyées par les capteurs à des valeurs exploitables.

Pour le premier test, le montage fut rudimentaire : fixation à l'aide de ruban adhésif et protection des capteurs optiques contre les perturbations dues au soleil grâce à du carton.

### *Etalonnage de la girouette*

L'étalonnage de la girouette reste le plus simple car il a suffi de repérer le nord avec une boussole et de la fixer en conséquence. La girouette permet de différencier 16 positions ce qui correspond à un pas de  $22,5^\circ$  (*Figure 47*).



*Figure 48 : Représentation des 16 directions pouvant désigner la girouette*

### *Etalonnage de l'anémomètre*

Pour calculer le coefficient entre le nombre d'impulsions par seconde que renvoie l'anémomètre et la vitesse du vent en m/s, nous avons utilisé un anémomètre portable pris comme référence. La manipulation d'étalonnage consistait à placer l'anémomètre et la référence à un endroit où les conditions de vent seraient les mêmes puis à enregistrer les informations envoyées par l'anémomètre ainsi que celles de la référence pendant un temps donné. Il a suffi ensuite de calculer un coefficient à l'aide des données obtenues.

Après quelques calculs et quelques manipulations de notre anémomètre, on a conclu que le coefficient entre les valeurs en impulsions par seconde renvoyée par l'anémomètre et la vitesse du vent en m/s est de 0,543 (une impulsion correspond à 0,543 m/s).

### *Etalonnage du thermomètre*

Pour étalonner le thermomètre, on a utilisé celui qui est déjà en place dans le parc météorologique d'Ampanzianomby comme référence puis on a calculé un coefficient avec la valeur que nous renvoie la conversion analogique/numérique sur 10 bits du microcontrôleur

Le coefficient entre le résultat de la conversion analogique/numérique et la valeur réelle de la température en °C est de **0,4935**. Donc le microcontrôleur doit faire une multiplication entre la valeur renvoyée par le CAN et ce coefficient avant de rafraîchir l'affichage LCD et avant de transmettre les données vers le microordinateur.

### **III.2.2 Interpretation des résultats**

#### *III.2.2.1 La carte d'acquisition*

Rappelons que la carte d'acquisition est basée sur le microcontrôleur PIC18F4550 du Microchip. Nous avons dû utiliser ce circuit intelligent pour satisfaire les différentes tâches mentionnés dans le cahier de charges.

Durant notre test sur terrain, ce microcontrôleur nécessite une grande protection contre les différents bruits in situ. Ceci est dû au fait qu'il travaille avec une fréquence d'horloge de 48MHz.

Dans tout le cas, nous trouvons quand même qu'il est le mieux adapté pour notre projet. Il se comporte bien pour les acquisitions de données numériques et analogiques et sa mémoire de programme est plus que nous en avons besoin.

Le microcontrôleur dispose de 32Kbytes de mémoire programme et nous en avons utilisés seulement 20Kbytes et les 4 broches Entrées/Sorties restantes permettent d'utilisation futur de la carte.

#### *III.2.2.2 L'anémomètre*

L'anémomètre démarre avec une faible force de vent. Alors nous pouvons dire que la sensibilité est meilleure. Ceci s'explique par le fait qu'il n'y a aucun contact physique entre le capteur et l'axe de l'anémomètre.

Le coefficient que nous avons calculé lors de l'étalonnage permet d'avoir une précision acceptable. Nous constatons aussi que cette précision devient de plus en plus grande avec l'augmentation de la force du vent. Mais la plage de mesure reste entre 0-99m/s sans décimal.

#### *III.2.2.3 La girouette*

Contrairement à une girouette analogique qui, utilisant un potentiomètre comme capteur de direction, présente une zone inexploitable de 3 à 5°, notre girouette balaie un angle complet de 360° avec un pas de 22,5°.

Il s'oriente avec une force de vent moins de 0,5m/s. Ceci est possible grâce à la réduction de frottement au niveau de l'axe et l'absence de contact physique entre cet axe et les capteurs.

L'afficheur LCD peut écrire le chiffre 15 dans le champ 'dd' pour le Nord géographique et 00 pour le Nord-Nord-Ouest. Ces valeurs 15 à 00 sont les résultats bruts



venant du microcontrôleur du fait que dans le monde Numérique on commence le comptage à partir de 00.

### *III.2.2.3 Le thermomètre*

Nous avons ajusté la précision du thermomètre grâce à la modification du coefficient de température. Avec le coefficient **0,4935**, notre précision est de 1/10 de degrés que nous supposons presque acceptable.

La plage de mesure de notre thermomètre est de 0-65°C, mais la meilleure précision se trouve entre 5-40°C. Cette plage de précision provient du comportement propre du capteur LM35. Cette plage peut être améliorée en réalisant un algorithme sophistiqué au niveau du microcontrôleur pour des fins non-météorologiques.

Dans le champ « TTT », le LCD affiche la température où le dernier chiffre indique le décimal selon le code météorologique.

## CONCLUSION

Qu'elle soit théorique ou appliquée, la météorologie se fonde sur des séries continues de mesures effectuées à l'aide d'instruments et de systèmes de plus en plus élaborés.

Aussi ce présent mémoire traite-t-il de la « **CONCEPTION, SIMULATION ET REALISATION D'UN SYSTEME D'INSTRUMENTS METEOROLOGIQUES NUMERIQUES POUR LA MESURE DE VENT ET DE TEMPERATURE (ff, dd, TT) AVEC INTERFACE USB 2.** »<sup>0</sup> basé sur un microcontrôleur PIC18F4550 du Microchip.


Vu que la totalité des anciens ports de l'ordinateur commence à disparaître, l'utilisation du port USB devient actuellement incontournable au niveau de transmission de données. Or, ce nouveau technologie requiert plusieurs domaines de compétences et difficile à mettre en œuvre.

L'interface USB permet de gérer l'acquisition de données sur le bus, de minimiser nos dépenses sur l'interfaçage avec les anciens ports, d'avoir une vitesse de transfert plus rapide, et un dispositif moins encombrant.

Au terme de cette étude, nous avons pu acquérir des connaissances sur l'utilisation de dispositif basé sur un microcontrôleur ainsi que sa programmation, le protocole USB, la programmation d'un pilote de Windows, et le développement d'une application capable de gérer l'acquisition de données via USB.

Ainsi, dans un projet avenir, l'application de ce projet sur un dispositif utilisant la technologie sans fil et de l'automatisation de station météorologique pourrait être envisageable.

## ANNEXES



# MICROCHIP PIC18F2455/2550/4455/4550

## 28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

### Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1 Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

### Power-Managed Modes:

- Run: CPU on, Peripherals on
- Idle: CPU off, Peripherals on
- Sleep: CPU off, Peripherals off
- Idle mode Currents Down to 5.8  $\mu$ A Typical
- Sleep mode Currents Down to 0.1  $\mu$ A Typical
- Timer1 Oscillator: 1.1  $\mu$ A Typical, 32 kHz, 2V
- Watchdog Timer: 2.1  $\mu$ A Typical
- Two-Speed Oscillator Start-up

### Flexible Oscillator Structure:

- Four Crystal modes, including High-Precision PLL for USB
- Two External Clock modes, Up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator Options allow Microcontroller and USB module to Run at Different Clock Speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

### Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 5.2 ns ( $T_{CY}/16$ )
  - Compare is 16-bit, max. resolution 83.3 ns ( $T_{CY}$ )
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I<sup>2</sup>C™ Master and Slave modes
- 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

### Special Microcontroller Features:

- C Compiler Optimized Architecture with Optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory Typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory Typical
- Flash/Data EEPROM Retention: > 40 Years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) via Two Pins
- Optional Dedicated ICD/ICSP Port (44-pin, TQFP package only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I <sup>2</sup> C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

## A2. Paramètres de configuration de l'horloge et le registre OSCCON

### A2.1 Detail des 4 paramètres de configuration de l'horloge du PIC18F4550

Bit	Valeur	Fonction	Remplacer X par #pragma config en C18 CONFIG en assembleur
PLLDIV	8	Divide by 12 (48 MHz oscillator input)	X PLLDIV = 8
	7	Divide by 10 (40 MHz oscillator input)	X PLLDIV = 7
	6	Divide by 6 (24 MHz oscillator input)	X PLLDIV = 6
	5	Divide by 5 (20 MHz oscillator input)	X PLLDIV = 5
	4	Divide by 4 (16 MHz oscillator input)	X PLLDIV = 4
	3	Divide by 3 (12 MHz oscillator input)	X PLLDIV = 3
	2	Divide by 2 (8 MHz oscillator input)	X PLLDIV = 2
	1	No Prescale (4 MHz oscillator input drives PLL directly)	X PLLDIV = 1
CPUDIV	OSC4_PLL6	[Primary oscillator Src: /4] [96 Mhz PLL Src: /6]	X CPUDIV = OSC4_PLL6
	OSC3_PLL4	[Primary oscillator Src: /3] [96 Mhz PLL Src: /4]	X CPUDIV = OSC3_PLL4
	OSC2_PLL3	[Primary oscillator Src: /2] [96 Mhz PLL Src: /3]	X CPUDIV = OSC2_PLL3
	OSC1_PLL2	[Primary oscillator Src: /1] [96 Mhz PLL Src: /2]	X CPUDIV = OSC1_PLL2
USBDIV	2	USB clock source comes from the 96 MHz PLL divided by 2	X USBDIV = 2
	1	USB clock source comes from the primary oscillator block with no postscale	X USBDIV = 1
FOSC	HSPLL_HS	HS oscillator, PLL enabled (HSPLL)	X FOSC = HSPLL_HS
	HS	HS oscillator (HS)	X FOSC = HS
	INTOSC_HS	Internal oscillator, HS oscillator used by USB (INTHS)	X FOSC = INTOSC_HS
	INTOSC_XT	Internal oscillator, XT oscillator used by USB (INTXT)	X FOSC = INTOSC_XT
	INTOSC_EC	Internal oscillator, CLKO function on RA6, EC used by USB (INTCKO)	X FOSC = INTOSC_EC
	INTOSCIO_EC	Internal oscillator, port function on RA6, EC used by USB (INTIO)	X FOSC = INTOSC IO_EC
	ECPLL_EC	EC oscillator, PLL enabled, CLKO function on RA6 (ECPLL)	X FOSC = ECPLL_EC
	ECPLLIO_EC	EC oscillator, PLL enabled, port function on RA6 (ECPIO)	X FOSC = ECPLLIO_EC
	EC_EC	EC oscillator, CLKO function on RA6 (EC)	X FOSC = EC_EC
	ECIO_EC	EC oscillator, port function on RA6 (ECIO)	X FOSC = ECIO_EC
	XTPLL_XT	XT oscillator, PLL enabled (XTPLL)	X FOSC = XTPLL_XT
	XT_XT	XT oscillator (XT)	X FOSC = XT_XT

### A2.2 Détail du registre OSCCON ou Oscillator Control Register

R/W-0	R/W-1	R/W-0	R/W-0	R <sup>(1)</sup>	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
bit 7							bit 0

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

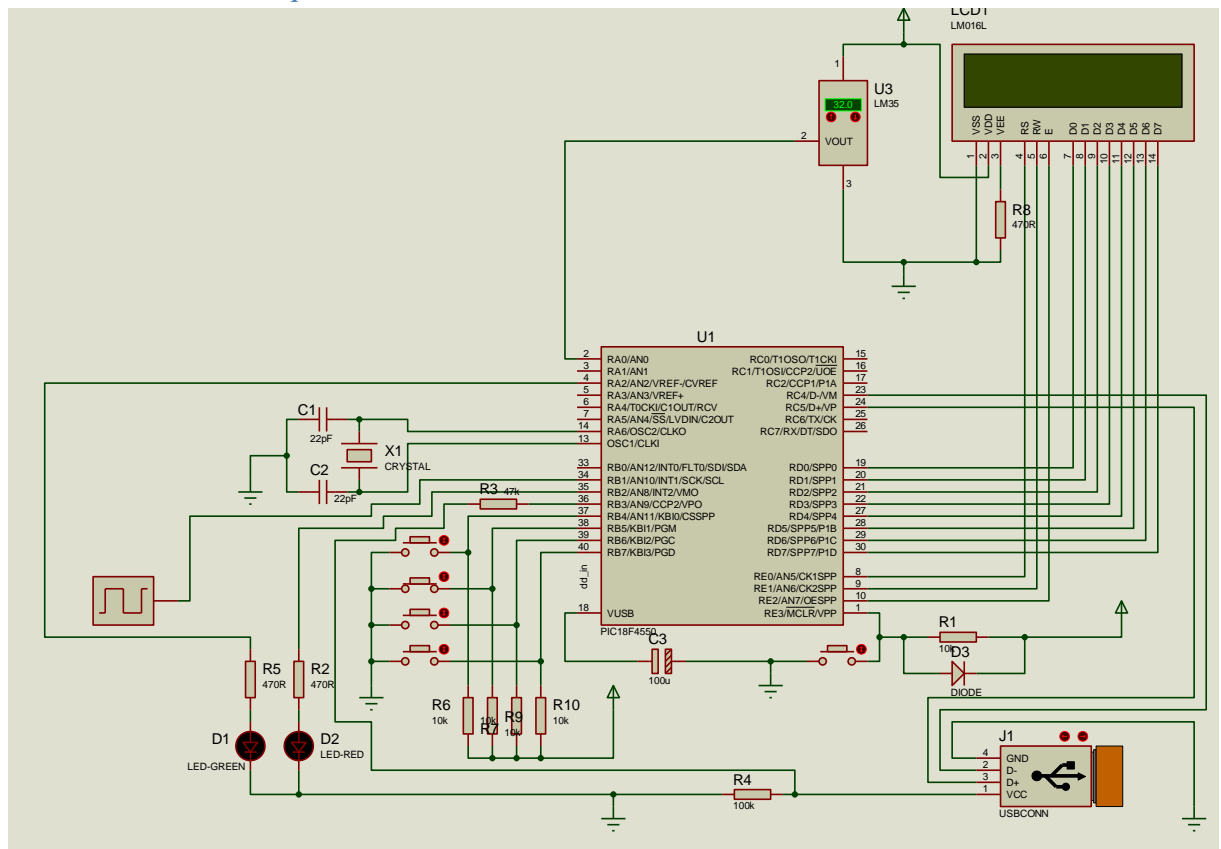
x = Bit is unknown

bit 7	<b>IDLEN:</b> Idle Enable bit 1 = Device enters Idle mode on <i>SLEEP</i> instruction 0 = Device enters Sleep mode on <i>SLEEP</i> instruction
bit 6-4	<b>IRCF2:IRCF0:</b> Internal Oscillator Frequency Select bits 111 = 8 MHz (INTOSC drives clock directly) 110 = 4 MHz 101 = 2 MHz 100 = 1 MHz <sup>(3)</sup> 011 = 500 kHz 010 = 250 kHz 001 = 125 kHz 000 = 31 kHz (from either INTOSC/256 or INTRC directly) <sup>(2)</sup>
bit 3	<b>OSTS:</b> Oscillator Start-up Time-out Status bit <sup>(1)</sup> 1 = Oscillator Start-up Timer time-out has expired; primary oscillator is running 0 = Oscillator Start-up Timer time-out is running; primary oscillator is not ready
bit 2	<b>IOFS:</b> INTOSC Frequency Stable bit 1 = INTOSC frequency is stable 0 = INTOSC frequency is not stable
bit 1-0	<b>SCS1:SCS0:</b> System Clock Select bits 1x = Internal oscillator 01 = Timer1 oscillator 00 = Primary oscillator

- Note 1:** Depends on the state of the IESO Configuration bit.  
**2:** Source selected by the INTSRC bit (OSCTUNE<7>), see text.  
**3:** Default output frequency of INTOSC on Reset.

## Annexe B : La carte d'acquisition

### B1. Schéma électronique de la carte



## B2. Liste des composants électroniques de la carte d'acquisition

### Semi-conducteurs :

U1 = Microcontrôleur PIC18F4550

D1 = LED vert

D2 = LED rouge

D3 = Diode de décharge

**Résistances (1/4 watt):**

$$R1, R6, R7, R9, R10 = 10k\Omega$$
$$R2, R5, R8 = 470\Omega$$
$$R3 = 47k\Omega$$
$$R4 = 100k\Omega$$

### Condensateurs :

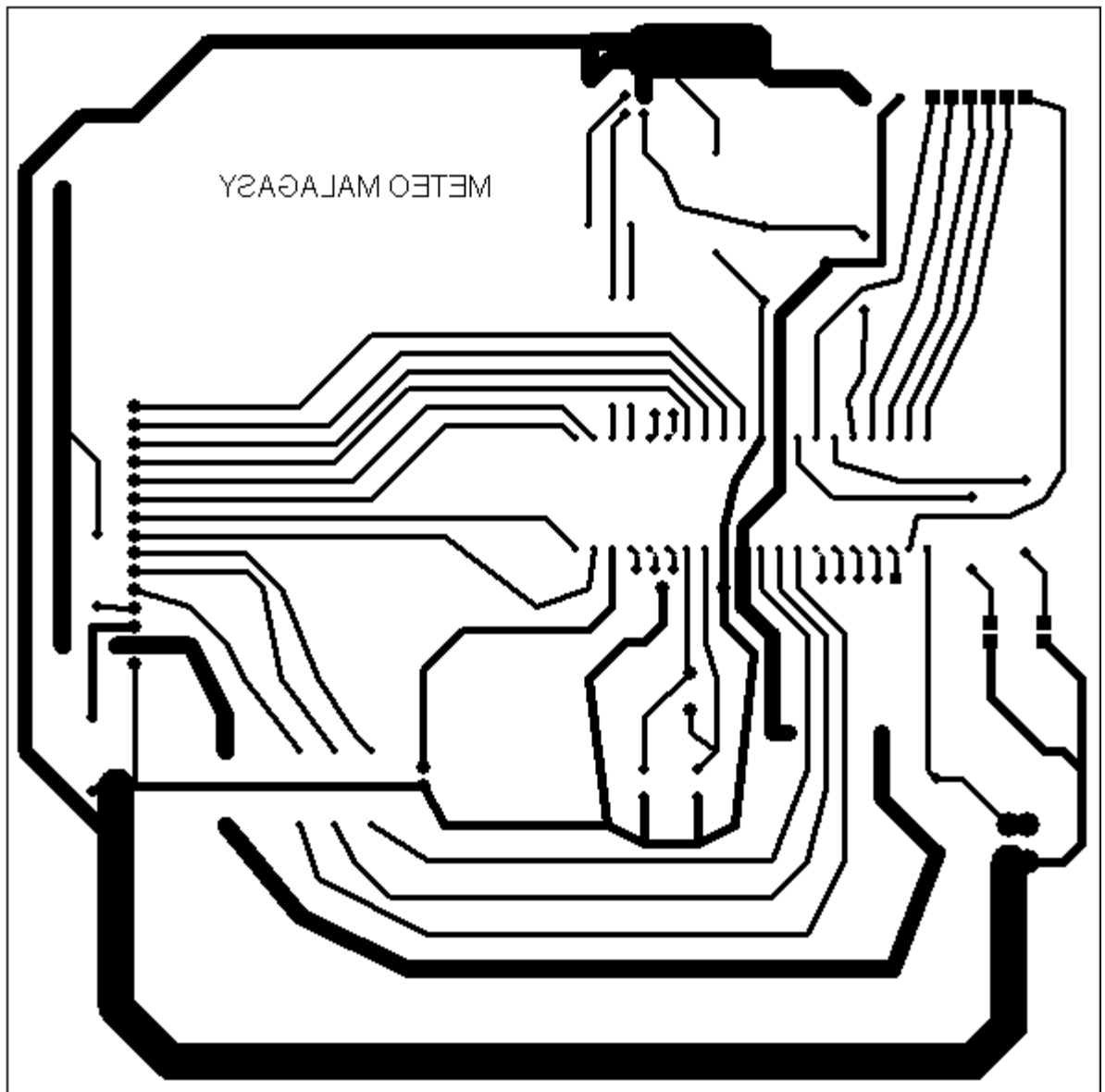
C1, C2 =22pF

$$C3 = 100\mu F$$

### Divers :

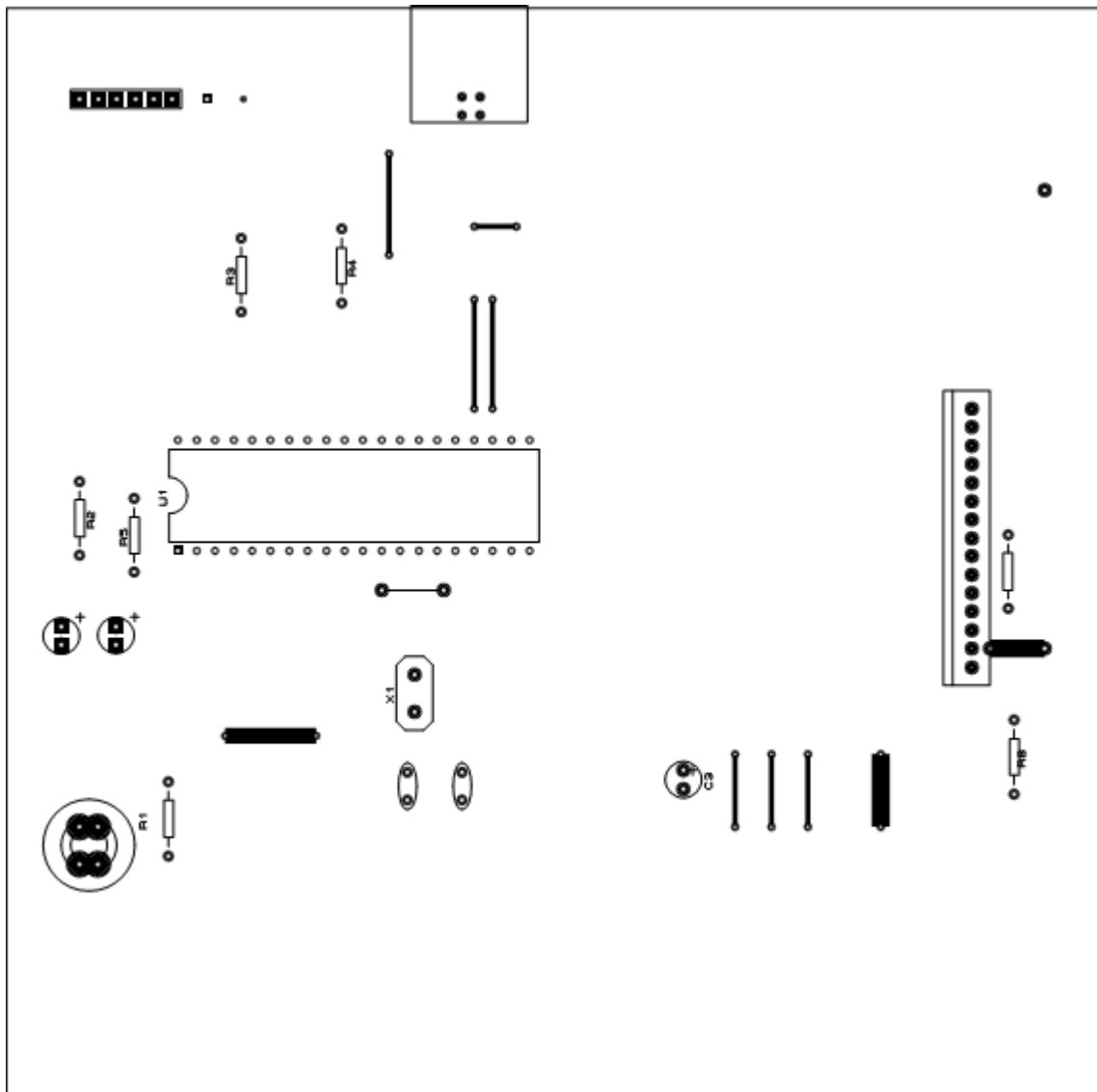
Bouton poussoirS  
Bornier à vis  
Connecteur USB femelle  
Support DIL 40  
LCD QC1602A  
Circuit imprimé simple face 150×150mm

### B3. Tracés des circuits imprimés (typons)



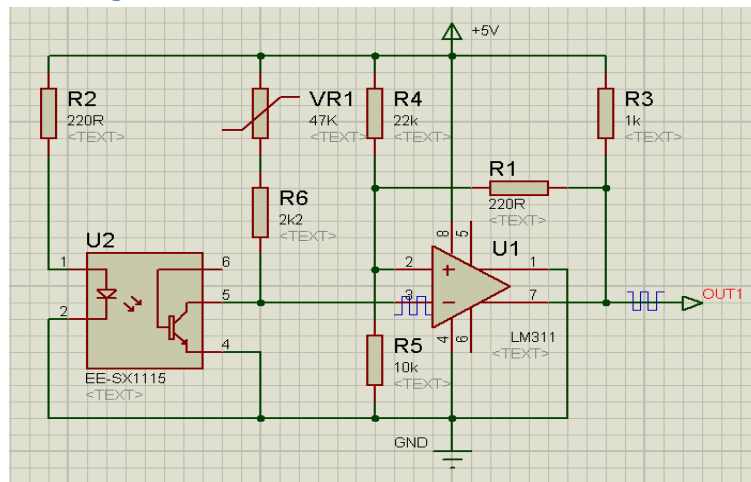


#### B4. Implantation des composants



## Annexe C : L'anémomètre

### C1. Schéma électronique



### C2. Listes des composants tête anémomètre

#### Semi-conducteurs :

U1 = Comparateur LM311

U2 = Detecteur opto fourche

#### Resistances (¼ watts) :

R1= R2= 220Ω

R3= 1k Ω

R4= 22k Ω

R5= 10k Ω

R6=2,2k Ω

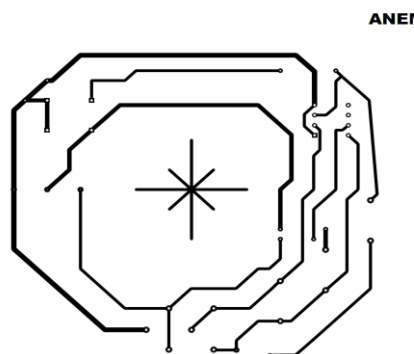
VR1=47k Ω

#### Divers:

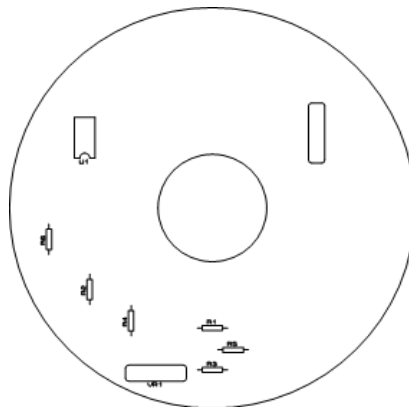
Bornier a vis

Circuit imprimé simple face en forme de disque de 50mm de diamètre

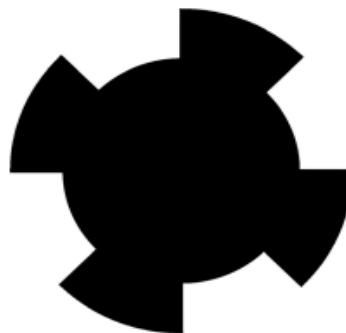
### C3. Typons tête anémomètre



#### C4. Implantation des composants

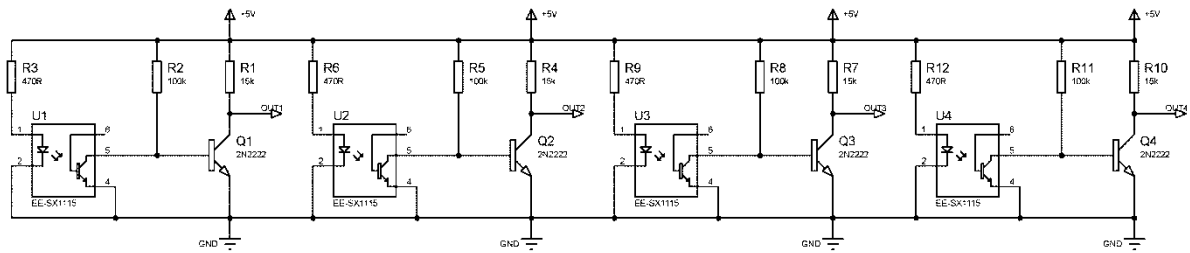


#### C5. Disque de codage



## Annexe D : La girouette

### D1. Schéma électronique



### D2. Listes des composants tête de la girouette

#### Semi-conducteurs :

U1=U2=U3=U4= Détecteur opto à réflexion TCRT5000

Q1=Q2=Q3=Q4= Transistor NPN 2N2222

#### Resistances (1/4 watts):

R1=R4=R7=R10= 15k  $\Omega$

R2=R5=R8=R11= 100k  $\Omega$

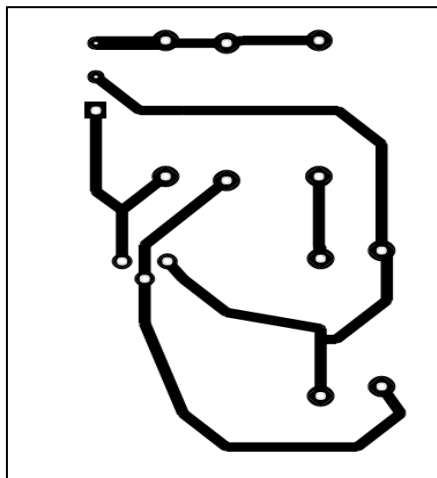
R3=R6=R9=R12=470  $\Omega$

#### Divers :

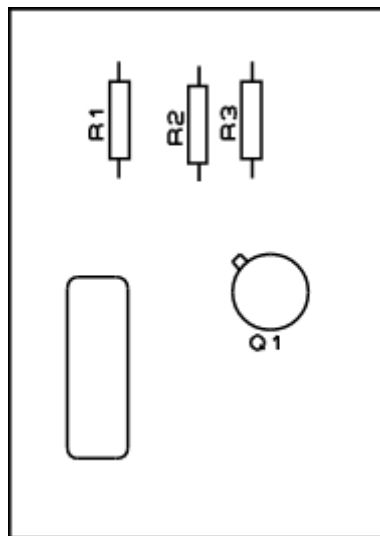
Bornier à vis

Circuit imprimé simple face (40×40mm)×4

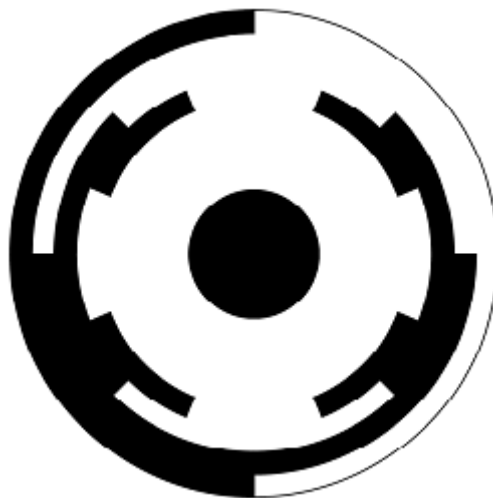
### D3. Typons de la girouette



#### D4. Implantation des composants

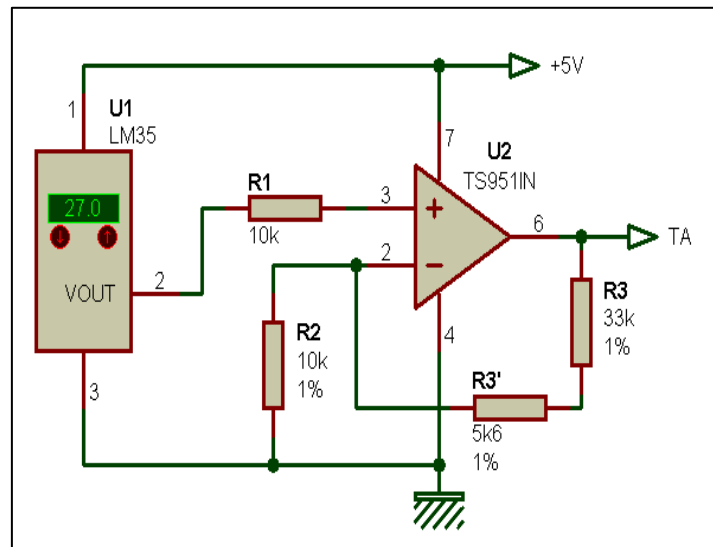


#### D5. Disque de codage



## Annexe E : Le thermomètre

### E1. Schéma électronique



### E2. Liste des composants

#### Semi-conducteurs :

U1= Capteur de température LM35

U2= Amplificateur d'instrumentation TS951N

#### Resistances (¼ watts):

R1= 10k  $\Omega$

R2=10k  $\Omega$  (1%)

R3=33k  $\Omega$  (1%)

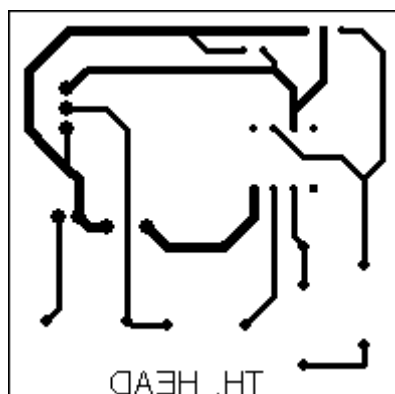
R4= 5,6k  $\Omega$  (1%)

#### Divers :

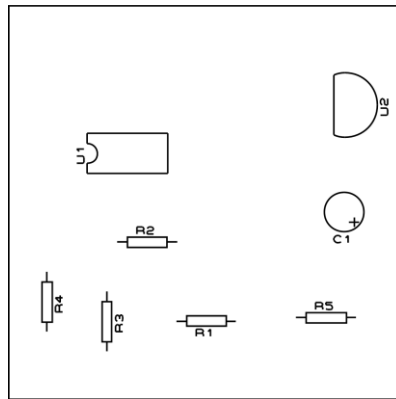
Bornier à vis

Circuit imprimé simple face 40×40mm

### E3. Typons du thermomètre

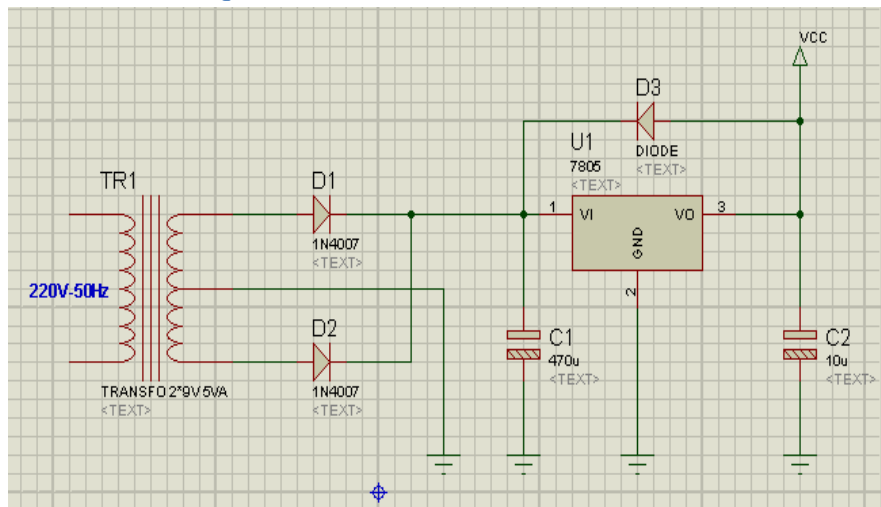


#### E4. Implantation des composants



## Annexe F : L'alimentation 5V

### F1. Schéma électronique



### F2. Liste des composants

#### Semi-conducteurs :

U1= Régulateur de tension 7805

D1=D2=D3= Diode de redressement

#### Condensateurs :

C1= 470 $\mu$ F

C2= 10  $\mu$ F

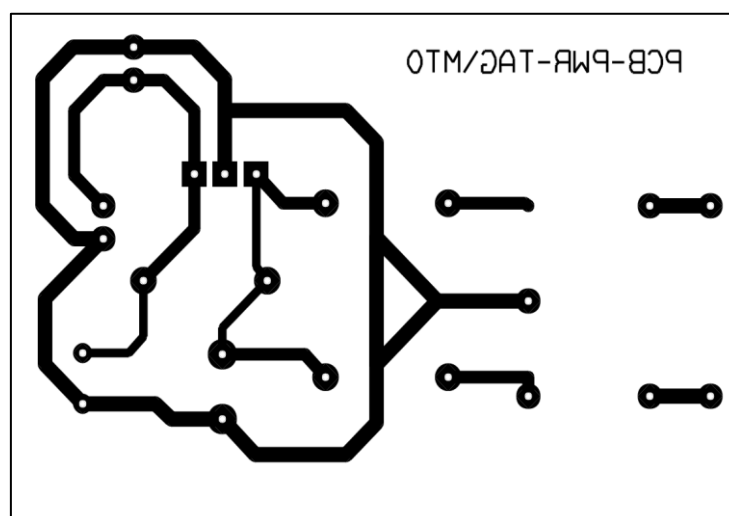
#### Divers :

Bornier à vis

Circuit imprimé simple face 100×60mm

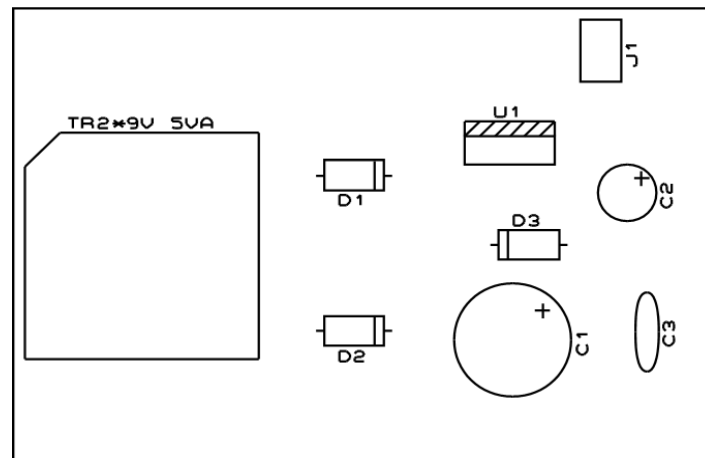
Transformateur 2×9V 5VA

### F3. Typons de l'alimentation 5V



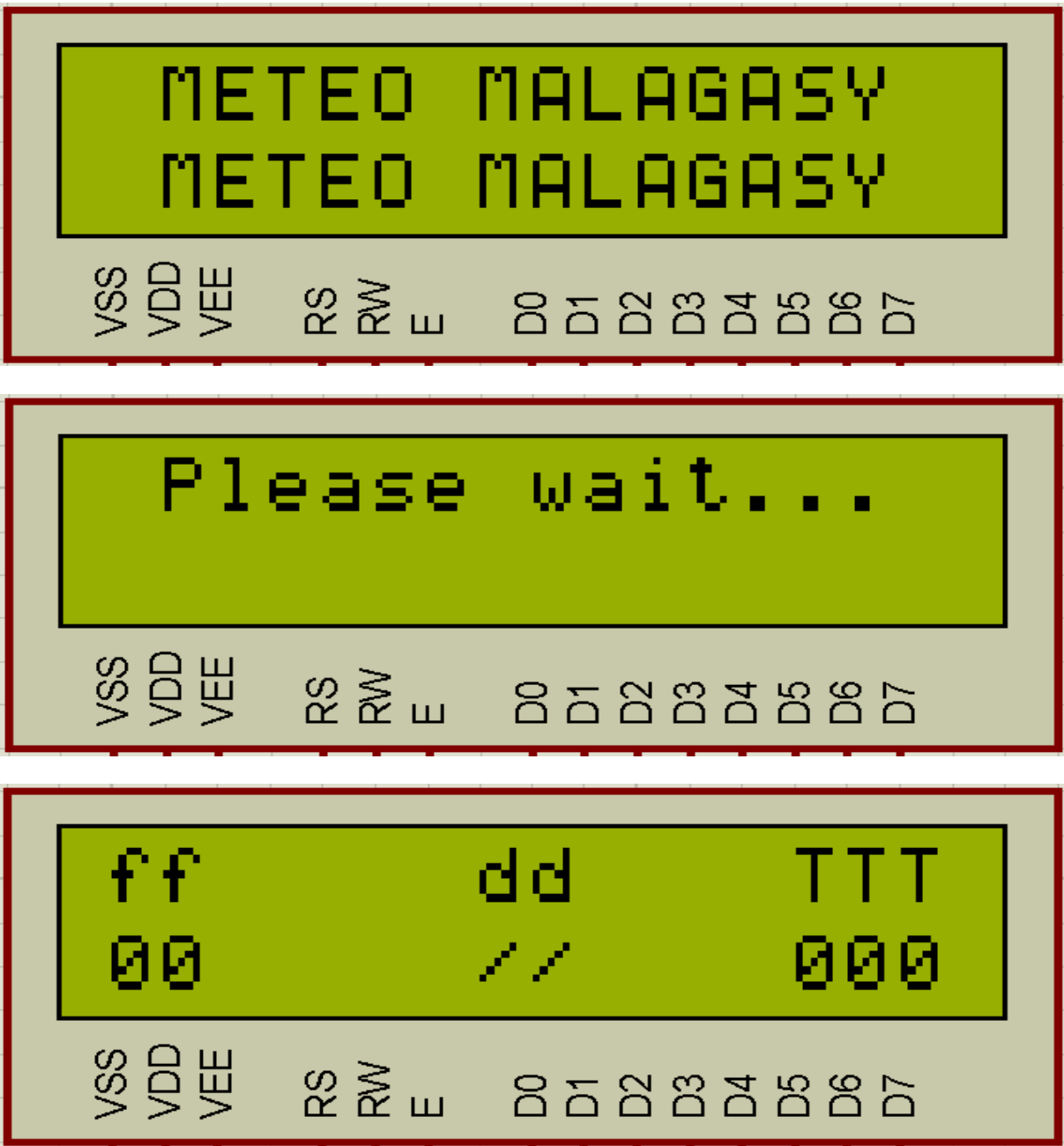


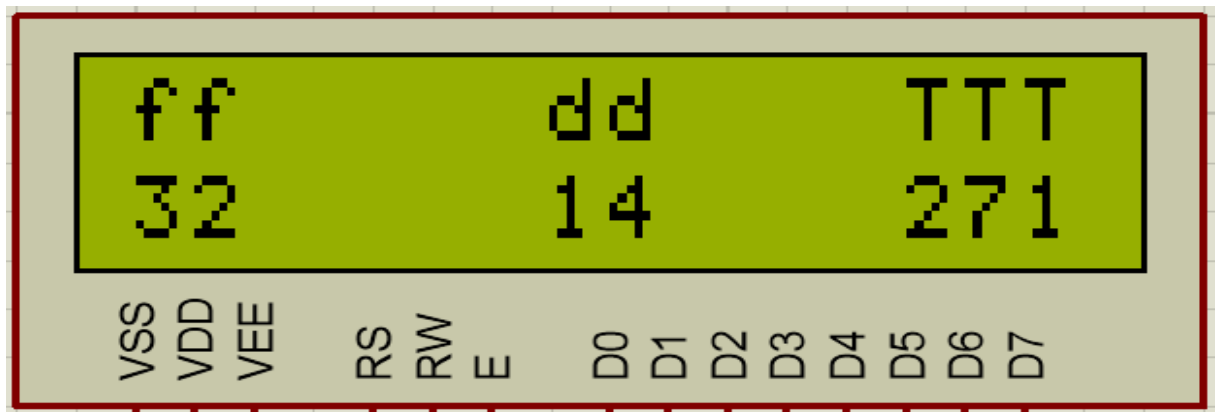
#### F4. Implantation des composants pour l'alimentation



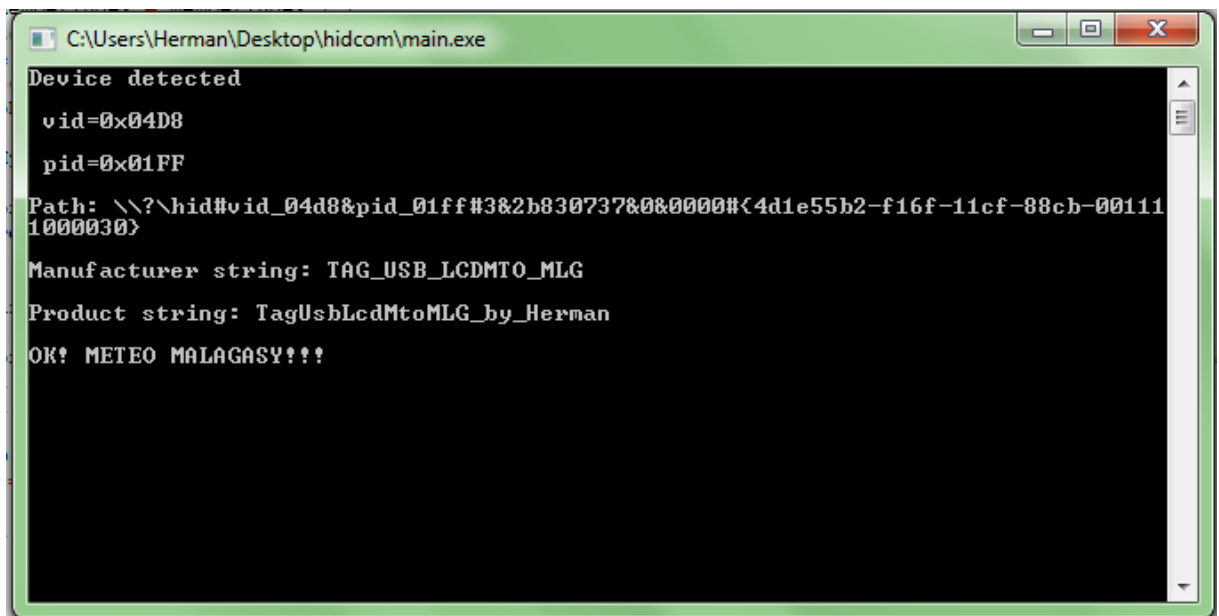
Annexe G : Quelque capture d’écran lors de la simulation

G1. Capture de l’afficheur LCD





## G2. Capture du console



## G3. Capture de la fenêtre “ Périphériques et imprimantes” sous Windows 7



## REFERENCES BIBLIOGRAPHIQUES

### A. BIBLIOGRAPHIE

- [1] : D.A Simidchiev, Recueil de note de cours sur les instruments météorologiques pour la formation du personnel météorologique des classes III et IV, 1990, 408p
- [2]: Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips, "Usb specification", 2000, 662p
- [3]: Dogan Ibrahim," Advanced PIC Microcontroller Projects in C", 2009, 539p
- [4]: Don Anderson, "USB System Architecture (USB 2.0)", 2002, 506p
- [5]: Jan Axelson, "USB COMPLETE", Troisième édition, 2005, 572p
- [6]: Labcenter, "ISIS Help", 2010
- [7]: Labcenter, "ARES Help", 2010
- [8]: Microchip Technology, "MPLAB C18 C Compiler Getting Started", 2004, 124p
- [9]: Microchip Technology," MPLAB C18 Users Guide", 2004, 128p
- [10]: Microchip Technology, "PIC18F2455/2550/4455/4550 Data Sheet", 2004, 428p
- [11]: Microsoft Corporation, "MSDN Library", 1995-2000
- [12]: Microsoft Corporation, "MSDN 2008 Library [WDK]", 2009
- [13]: Microsoft Corporation, "WDK Documentation", 2009
- [14]: Walter Oney, "Programming the Microsoft Windows Driver Model 2nd Edition", 2003, 448p
- [15]: John Hyde, Embedded USB Design By Example, 2010, Revision 2.0, 134p

### B. WEBOGRAPHIE

- [16] : <http://www.wmo.com>
- [17] : <http://www.abcelectronique.com/bigonoff>
- [18] : <http://www.beyondlogic.org>
- [19] : [http://www.codeproject.com/KB/system/WDM\\_Driver\\_development](http://www.codeproject.com/KB/system/WDM_Driver_development)
- [20] : <http://www6.conestogac.on.ca/%7Eset/courses/year3/drivers>
- [21] : <http://www.datasheetcatalog.com>

- [22] : [http://www.intel-u-press.com/usb\\_dbe](http://www.intel-u-press.com/usb_dbe)
- [23] : <http://www.Lvr.com>
- [24] : <http://membres.lycos.fr/grandzebu/electronique/usb>
- [25]: <http://www.microchip.com>
- [26] : <http://www.microchip.com/downloads/en/Appnotes>
- [27]: <http://www.microchip.com/downloads/en/devicedoc>
- [28]: <http://www.microsoft.com/whdc/ddk>
- [29]: <http://msdn.microsoft.com>
- [30]: <http://www.newsnespress.com>
- [31]: <http://www.osr.com>
- [32]: <http://pic18fusb.online.fr>
- [33]: [http://www.pulsewan.com/data101/usb\\_basics.htm](http://www.pulsewan.com/data101/usb_basics.htm)
- [34]: <http://www.usb.org>
- [35]: <http://www.usb-by-example.com>
- [36]: <http://www.usbman.com>
- [37]: <http://www.usb.org/developers>
- [38]: [http://u.s.b.free.fr/pdf/L\\_USB\\_et\\_sa\\_norme\\_v1.pdf](http://u.s.b.free.fr/pdf/L_USB_et_sa_norme_v1.pdf)
- [39]: [http://www.wikipedia.org/wiki/usb\\_cours.html](http://www.wikipedia.org/wiki/usb_cours.html)

**TITRE :**  
**« CONCEPTION, SIMULATION ET REALISATION D'UN SYSTEME  
D'INSTRUMENTS METEOROLOGIQUES NUMERIQUES POUR LA  
MESURE DE VENT ET DE TEMPERATURE (ff, dd, TT) AVEC  
INTERFACE USB 2.0 »**

**RESUME**

*La présente étude se rapporte à la conception et réalisation d'un système d'instruments météorologiques constitué d'un anémomètre, d'une girouette et d'un thermomètre.*

*Le système est basé sur le microcontrôleur PIC18F4550 et doté d'un afficheur LCD qui vise à homogénéiser les données d'observation météorologique.*

*Il permet aussi de faire le transfert de données via le port USB de l'ordinateur tout en facilitant donc leur recueil et leur traitement en une courte durée.*

**Mots clés:** Instrumentation, Météorologie, USB, WDM, pilote, microcontrôleur

**ABSTRACT**

*The present study refers to the design and realization of a system of weather instruments consisted of an anemometer, a wind vane and a thermometer.*

*The system is based on microcontroller PIC18F4550 and is equipped with a digital display panel LCD who aims at homogenizing the data of meteorological observation.*

*It also makes it possible to transmit the data via port USB of the computer while thus facilitating their collection and their treatment in a short duration.*

**Key words:** Instrumentation, Meteorology, USB, WDM, driver, microcontroller

**Nombre de pages** : 92

**Nombre de figures** : 48

**Nombre de tableaux** : 14

**Rubrique** : Météorologie

**Auteur:** ANDRIAMAOLISON Herman

*Lot ITH 72 Amboatavo Itaosy, 101 Antananarivo Madagascar*

[maoolly@yahoo.fr](mailto:maoolly@yahoo.fr)

+261 33 12 757 98

**Directeur de mémoire** : Monsieur RAKOTOARIMANANA Nirison

*Chef de Département de la Navigation Aérienne à l'Aviation Civile  
de Madagascar (ACM).*