



**CLES-FACIL**

Club Lyonnais d'Expérimentations Spatiales

Fusée Astronautique Club de l'INSA de Lyon

CLES-FACIL

BDE-INSA

20 avenue Albert Einstein

69621 Villeurbanne CEDEX

tél : 04-78-94-31-73

Fax : 04-72-44-37-53 (préciser "CLES-FACIL")

E-mail : [clesfacil@genserver.insa-lyon.fr](mailto:clesfacil@genserver.insa-lyon.fr)

<http://www.insa-lyon.fr/Associations/ClesFacil/>

# Pauline

Fusée Expérimentale 1997



**Emmanuel Jolly**

**Bruno Tardy**

**Yann Gouy**

**Cyrille Wagner**

**Benoît Pothier**

**Sébastien Deschaux**

**Dominique Walter**

**Jean-Luc Dumont**

**Mickaël Lefèvre**

**Thomas Ledoux**

# Sommaire

<b>1. PRÉSENTATION DU CLUB</b>	<b>4</b>
1.1 LES CLUBS AÉROSPATIAUX	4
1.2 HISTORIQUE DU CLES-FACIL	5
1.3 FONCTIONNEMENT DU CLUB	5
<b>2. LE PROJET PAULINE.</b>	<b>7</b>
2.1 PRÉSENTATION DU PROJET.	7
2.2 L'ÉQUIPE.	7
2.3 LES OBJECTIFS.	7
<b>3. LA CONCEPTION DE PAULINE.</b>	<b>9</b>
3.1 CHOIX DES CAPTEURS.	9
3.2 SYNOPTIQUE DE LA FUSÉE	11
3.3 CHOIX LOGICIEL AU SOL.	11
3.4 L'ÉLECTRONIQUE	14
3.4.1 LE SÉQUENCEUR.	14
3.4.2 L'ALIMENTATION.	15
3.4.3 LES INTERFACES AVEC LES CAPTEURS.	19
3.4.4 LA CARTE PROCESSEUR.	19
3.5 L'INFORMATIQUE EMBARQUÉE.	22
3.5.1 CAHIER DES CHARGES.	22
<b>4. LA RÉALISATION DE PAULINE.</b>	<b>24</b>
<b>LA MÉCANIQUE</b>	<b>24</b>
4.1.1 LE CORPS DE LA FUSÉE.	25
4.1.2 LE CÔNE.	25
4.1.3 LES BAGUES INTERMÉDIAIRES.	25
4.1.4 LE SYSTÈME DE RÉCUPÉRATION.	27
4.1.5 LES AILERONS.	27
4.2 L'ÉLECTRONIQUE	28
4.2.1 LE SÉQUENCEUR.	28
4.2.2 L'ALIMENTATION.	29
4.2.3 LES INTERFACES AVEC LES CAPTEURS.	30
4.2.4 LA CARTE PROCESSEUR.	31
4.3 L'INFORMATIQUE EMBARQUÉE.	32
4.3.1 DÉCOUPAGE LOGIQUE.	32
4.3.2 DÉCOUPAGE TEMPOREL.	32
4.3.3 ÉTUDE DÉTAILLÉE DE LA PROGRAMMATION	32
4.4 L'INFORMATIQUE AU SOL.	36
<b>5. LA CAMPAGNE DE LANCEMENT.</b>	<b>38</b>

<b>5.1</b>	<b>ETAT DU PROJET LORS DE L'ARRIVÉE À BOURGES.</b>	<b>38</b>
<b>5.2</b>	<b>LES QUALIFICATIONS.</b>	<b>38</b>
<b>5.3</b>	<b>LA CHRONOLOGIE FINALE</b>	<b>39</b>
<b>5.4</b>	<b>LE LANCEMENT.</b>	<b>40</b>
<b>5.5</b>	<b>RÉCUPÉRATION DE LA FUSÉE.</b>	<b>40</b>
<b>6. LES RÉSULTATS OBTENUS.</b>		<b>42</b>
<b>6.1</b>	<b>LES RÉSULTATS IMMÉDIATS : LA 3D.</b>	<b>42</b>
<b>6.2</b>	<b>ANALYSE POST-LANCEMENT.</b>	<b>44</b>
6.2.1	LES REPÈRES UTILISÉS	44
	ROTATION ATOURS DES AXES TRANSVERSAUX (X ET Z)	45
6.2.3	ROTATION ATOUR DE L'AXE LONGITUDINAL DE LA FUSÉE.	46
6.2.4	VITESSE DE LA FUSÉE	48
6.2.5	ACCÉLÉRATION	49
6.2.6	DISTANCE PARCOURUE PAR LA FUSÉE	50
6.2.7	RELATION ENTRE LA VITESSE LINÉAIRE ET LA VITESSE DE ROTATION.	51
	INTÉGRATION DES VITESSES ANGULAIRES, STABILITÉ DE LA FUSÉE	52
6.2.9	TRAJECTOIRE ET VENT	53
<b>7. BILAN DU PROJET.</b>		<b>55</b>
<b>7.1</b>	<b>L'EXPÉRIENCE.</b>	<b>55</b>
<b>7.2</b>	<b>CE QU'IL NE FAUDRA PAS REFAIRE.</b>	<b>55</b>
<b>7.3</b>	<b>CE QUE L'ON POURRA REFAIRE.</b>	<b>56</b>
<b>8. REMERCIEMENTS</b>		<b>58</b>
<b>9. ANNEXES</b>		<b>59</b>
	CHAÎNE ANALOGIQUE CAPTEUR	59
	MODULATION FSK	60
	SÉQUENCEUR	61
<b>9.4</b>	<b>CARTE PROCESSEUR</b>	<b>62</b>

## 1. Présentation du club

### 1.1 Les clubs aérospatiaux

La vocation initiale des clubs aérospatiaux est de réaliser des fusées, dont le gabarit peut aller de la micro fusée, d'une taille d'environ 20 cm, à celui d'une fusée expérimentale, d'une taille maximum de 3 m. L'objectif est double : il est d'une part de mener à bien une expérience physique (mesures de pressions, vitesses, contraintes...), et d'autre part de réaliser une fusée respectant le cahier des charges axé sur la sécurité établit par l'ANSTJ<sup>1</sup>.

Par ailleurs, depuis quelques années, la réalisation d'expériences basées sur un nouveau vecteur, le ballon stratosphérique, tend à se développer. Si les objectifs restent les mêmes, les caractéristiques et les potentialités des ballons nous ont ouvert de nouveaux horizons. Les projets Taranis et Pégase concrétisent depuis deux ans cet intérêt.

L'ANSTJ, association loi 1901, aide chaque club dans son projet en lui fournissant gratuitement un propulseur et un émetteur FM pour les fusées ou l'enveloppe et l'hélium pour les ballons. Elle organise enfin une campagne de lancement (le Festival des clubs aérospatiaux) qui a lieu chaque année à Bourges. Cette campagne est l'occasion pour tous les clubs aérospatiaux français et canadiens de se rencontrer pendant l'aboutissement de longs mois de travail : la mise à feu de la fusée ou le lâcher du ballon.

Le lancement d'une fusée expérimentale commence par la mise à feu de son propulseur, lorsque toutes les longues procédures de mise en fonctionnement ont été terminées. Les mesures décrivant l'expérience embarquée sont généralement retransmises en temps réel grâce à l'émetteur radio. Lorsque la fusée arrive à son apogée, un parachute est alors ouvert pour ralentir la descente. Le travail du club est de concevoir, réaliser, et tester l'ensemble de la fusée, que l'on peut décomposer en deux secteurs : l'électronique et la mécanique. La nécessité de mettre en forme les signaux issus des capteurs et d'effectuer des calculs en temps réels implique souvent l'utilisation de microprocesseurs. Il faut donc développer des cartes suffisamment souples pour répondre aux exigences attendues. La partie mécanique se compose essentiellement d'un cylindre creux (Duralumin ou carbone) formant le corps de la fusée, d'une coiffe dans laquelle est logé l'émetteur, et de 4 ailerons afin d'assurer la stabilité de l'engin.

Le lâcher d'un ballon stratosphérique ne nécessite aucun élément pyrotechnique mais se révèle tout aussi délicat. Il s'agit en effet de gonfler l'enveloppe du ballon avec un volume déterminé d'hélium, environ 8m<sup>3</sup>. Ce volume doit permettre une ascension rapide de la nacelle dont la masse est limitée à 2.5 kg. Après plus d'une heure, le ballon se trouve dans des zones ayant une pression si faible qu'il finit par éclater, l'enveloppe ne pouvant plus contenir l'hélium. L'altitude est alors comprise entre 20000 m et 40000m. Un petit parachute fixé sur la nacelle permet ensuite de ralentir la chute de la nacelle qui dure près d'une demi-heure. Au terme de ce périple d'une centaine de kilomètres, il s'agit ensuite de récupérer la nacelle, ce qui relève encore souvent du miracle.

---

<sup>1</sup>ANSTJ: Association Nationale Sciences & Techniques Jeunesse. Cette association a pour objectif, entre autres, de fédérer les clubs aérospatiaux afin que chaque club ait la possibilité de mener à bien son projet.

## 1.2 Historique du CLES-FACIL

Le FACIL<sup>2</sup> est l'un des premiers clubs fondés en France dans les années 60. Il reste peu de traces de cette époque à l'exception d'une superbe fusée, IPHIGENIE, dont l'objectif était de prendre des photos en vol par un dispositif de miroir très astucieux. Le club fabriqua de nombreuses fusées, mais sa production commença à ralentir. En 1980, le CLES<sup>3</sup> fut fondé, et fusionna rapidement avec le FACIL pour finalement donner le CLES-FACIL<sup>4</sup> qui subsiste actuellement. De nombreuses fusées sont sorties des ateliers du CLES-FACIL. Voici un bref récapitulatif des projets menés à terme depuis 1980 :

Date	Nom, Taille, Masse Altitude maximale	Expériences embarquées
1981	DAHU	Pas de données sur cette fusée (succès)
1985	MARIANNE L01 1.8 m 15 Kg 900 m	Prise de photo et mesure de la rotation propre de la fusée par cellule photorésistive. (succès)
1985	SPRITE 1.1 m 5 Kg 800 m	Prise de vues
1987	JUMBO 2.5 m 30 Kg 1000 m	Mesure de vitesse par Pitot. Fusée modulaire en PVC renforcé par de la fibre de verre (échec)
1988	D3D7 1.7 m 13 Kg 1000 m	Mesure des contraintes sur la structure (échec)
1989	PHOENIX 3.1 m 40 Kg 900 m	Mesure d'accélération et de vitesse. Fusée modulaire en Duralumin soudé. (succès)
1991	SUPER PHOENIX 3.2 m 38 Kg 1000 m	Mesure d'accélération 3 axes. Utilisation du carbone pour les ailerons et une partie de la structure. (succès)
1995	TARANIS (ballon) 2 Kg 20 000 m	Mesure de pression et de température, prise de photos en altitude. (succès)
1995	PSO 2.2m 15 Kg 1000 m	Asservissement en rotation de la fusée. (succès)
1996	PEGASE (ballon) 2.5 Kg 20 000 m	Essai cellules solaires. Suivi temps réel par GPS. (succès)

## 1.3 Fonctionnement du club

Le club est une association régie par la loi de 1901, et donc à but non lucratif. Il est basé sur le campus de l'INSA, qui nous prête un local. Les financements proviennent essentiellement du bureau des élèves et de quelques départements d'enseignement de l'Institut mais ils ne suffisent qu'à assurer le fonctionnement du club ou de petits projets.

---

<sup>2</sup> **FACIL** : Fusée Astronautique Club de l'INSA de Lyon

<sup>3</sup> **CLES** : Club Lyonnais d'Expérimentation Spatiales

<sup>4</sup> **CLES-FACIL**: Club Lyonnais d'Expérimentation Spatiales - Fusée Astronautique Club de l'INSA de Lyon

Pour les projets plus ambitieux, nous faisons appel aux entreprises ou aux laboratoires de l'INSA qui nous fournissent essentiellement du matériel. L'aide des sociétés SENSOREX, Lenoir Métallerie, HONEYWELL, RPCI et CLESSE nous a ainsi permis de réaliser nos derniers projets.

Nous bénéficions également des moyens dont dispose l'INSA, notamment en informatique et en mécanique.

Nous sommes de plus en excellentes relations avec le CLUB ELEK de l'INSA, avec qui nous partageons des projets. L'ensemble des relations que nous conservons permet en général de réaliser nos projets avec des moyens financiers réduits, mais d'importants moyens matériels.

Les membres du club font le plus souvent leurs études à l'INSA, mais le club est ouvert à tous. Nous nous réunissons au moins une fois par semaine afin de faire le point sur l'avancement du projet. La qualité des projets réalisés tient essentiellement à la motivation de l'équipe. La structure du club est classique : un bureau est élu chaque année par vote, comme l'indiquent les statuts du club.

## 2. Le projet Pauline.

### 2.1 Présentation du projet.

L'idée de la fusée Pauline nous est venue en visualisant les images du suivi infrarouge d'une fusée, proposées par la DGA lors du festival de l'espace à Bourges : Quel est le comportement en vol de la fusée ? Quelle est sa trajectoire ?

Pour cette fusée, nous avons donc décidé de représenter, en image de synthèse 3D le comportement de la fusée suivant ses 3 axes de rotation.

On essayera aussi de connaître sa trajectoire. Cependant cette dernière ne sera pas l'expérience principale.

La fusée emportera donc les capteurs nécessaires à la connaissance de son attitude, ainsi qu'une mesure de vitesse longitudinale.

Ces 2 mesures combinées permettront alors de connaître la trajectoire.

Les mesures seront transmises au sol par liaison HF, interprétées par un ordinateur. La fusée sera alors dessinée dans un "monde virtuel", en temps réel.

### 2.2 L'équipe.

L'équipe de Pauline se compose de 10 personnes environs de compétences et de savoir-faire divers et variés :

Nom	Formation	Fonction
Emmanuel JOLLY	5GE	Electronicien + Président du club.
Bruno TARDY	5GE	Electronicien + informatique embarquée.
Sébastien DESCHAUX	3GE	Electronicien.
Mickaël LEFEVRE	3GE	Electronicien.
Cyrille WAGNER	1 <sup>ère</sup> année	Informatique 3D.
Benoît POTIER	3GE	Informatique 3D.
Dominique WALTER	4GMC	Mécanique.
Yann GOUY	euh...	Mécanique.
Thomas LEDOUX	2 <sup>ème</sup> année	Mécanique.
Jean-Luc DUMONT	3GE	Mécanique, Electronique, Informatique.

### 2.3 Les objectifs.

Les objectifs de Pauline sont multiples.

D'un point de vue humain, c'est de pouvoir former un certain nombre de jeunes sur une nouvelle fusée. En effet, le CLES-FACIL est un club qui "vieilli" facilement. Ainsi, il n'est pas rare de voir la même équipe réaliser plusieurs projets, mais lorsque celle-ci disparaît (cas des écoles), la relève est souvent très difficile. Nous voulions profiter de cette année, pour réaliser une fusée d'une complexité relative (c'est à dire plus simple qu'une fusée asservie mais plus complète qu'une simple fusée de mesure) afin de permettre aux plus anciens d'exprimer leur créativité mais aussi aux débutants, de pouvoir suivre le projet dans sa totalité.

Un des autres objectifs du club avec Pauline, était d'apporter la compétence en trajectographie inertielle, et la mesure d'attitude au sein du club. Ce genre d'expérience est en effet en parfaite adéquation avec les projets précédemment réalisés (asservissement en rotation et trajectographie par GPS).



### 3. La conception de Pauline.

#### 3.1 Choix des capteurs.

Pour des mesures de rotation, voire même d'attitude, beaucoup de clubs aérospatiaux se tournent aujourd'hui vers l'utilisation de capteurs magnétorésistifs. Ces capteurs permettent la mesure du vecteur  $B$  (champ magnétique terrestre). Ceci s'effectuant grâce à 3 capteurs placés en trièdre.

Cependant, notre problème consiste à connaître les attitudes de la fusée par rapport à un repère fixe : celui de la Terre.

Mathématiquement, l'algèbre nous apprend que pour passer d'un repère à un autre, il faut définir une matrice de passage. On obtiendrait alors :  $Y=P.X$ , où  $P$  serait la matrice de passage du repère fusée au repère terrestre,  $X$  les vecteurs unitaires dans le repère fusée, et  $Y$  ces mêmes vecteurs dans le repère terrestre.

$$X \text{ aurait alors pour forme : } \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Connaître  $P$  permettra donc de déterminer les attitudes de la fusée. Or pour déterminer  $P$ , il faut déterminer  $Y$ .

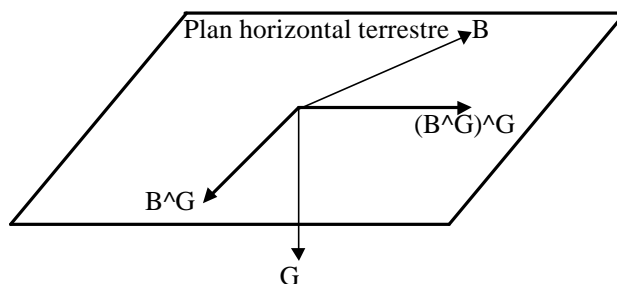
$Y$  étant une matrice de 3 vecteurs 3D, et sachant que le repère est orthonormal, il suffit de déterminer 2 vecteurs perpendiculaires pour déterminer  $Y$  et par conséquent  $P$ .

Or les capteurs magnétorésistifs n'apportant qu'un seul vecteur, il est clair que cette méthode mène à une indétermination physique des attitudes.

Une solution possible est donc d'apporter la connaissance d'un second vecteur. Le plus simple à mesurer est a priori le vecteur gravité  $G$ . Ce vecteur peut être mesuré par 3 accéléromètres (cas où la fusée est à l'arrêt).

$G$  étant orienté sur la verticale terrestre,  $B$  étant orienté vers le nord magnétique (soit à  $2.1^\circ$  du nord géographique) mais n'étant pas compris dans le plan horizontal terrestre, le produit vectoriel  $B \wedge G$  est donc dans le plan horizontal terrestre et orienté vers l'est :  $C$  est le vecteur Est par définition.

Enfin le produit  $(B \wedge G) \wedge G$ , va définir le vecteur perpendiculaire à l'est (donc orientée vers le nord), et situé dans le plan horizontal.



Cette méthode est intéressante pour déterminer les attitudes d'un véhicule à l'arrêt. En effet, dès que le véhicule est soumis aux accélérations, la mesure du vecteur G devient incohérente.

Dans le cas des centrales inertielles. Cette méthode est utilisée pour l'initialisation (phase que l'on appelle aussi "alignement") et qui doit être effectuée à l'arrêt. Ensuite ce sont des gyromètres qui vont intégrer les attitudes.

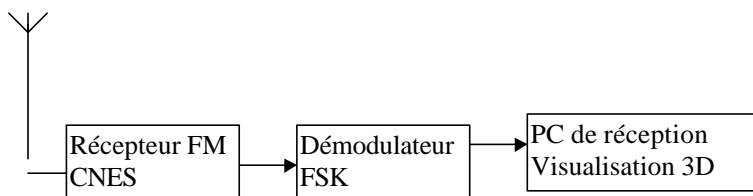
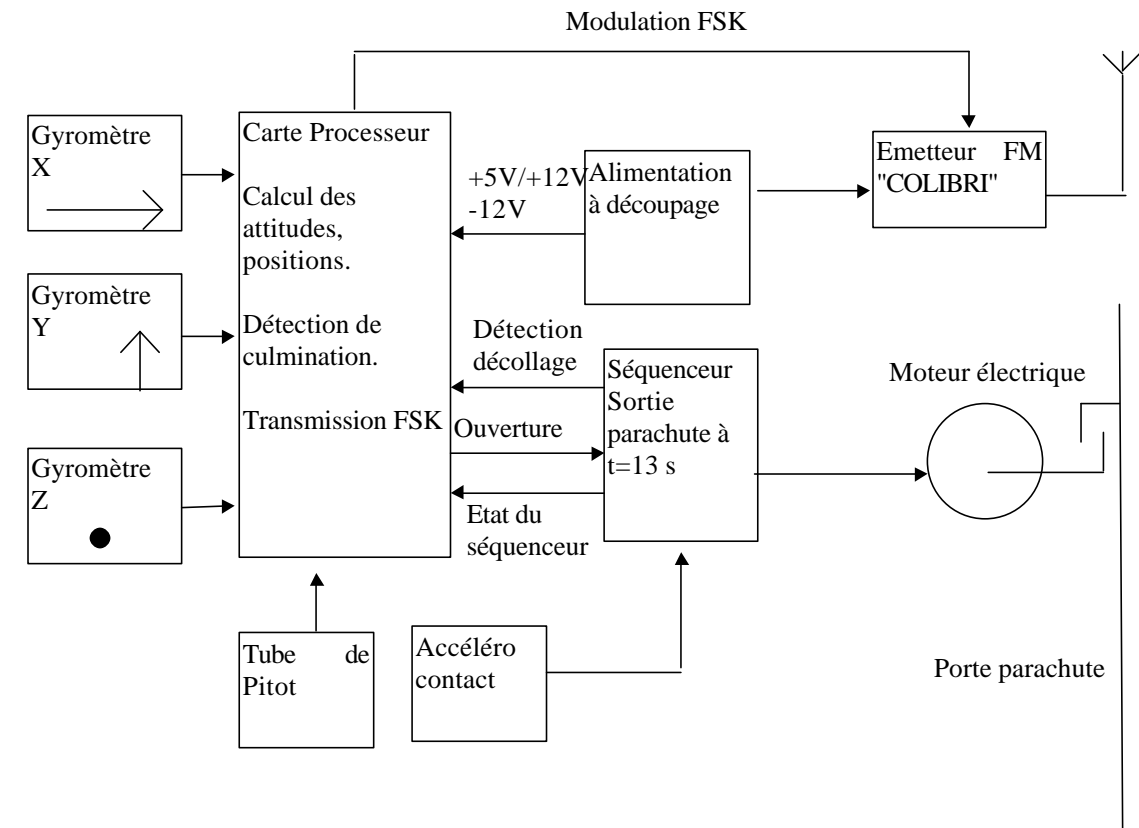
Dans notre cas, la façon la plus simple, est de mesurer les attitudes par l'utilisation de gyromètres.

En effet, dès que la fusée sera en mouvement, on ne pourra plus mesurer G (et on sait que dans une fusée, ce vecteur est soumis à des variations importantes : 10G au décollage !).

L'utilisation des accéléromètres n'étant donc pas possible, et n'ayant pas trouvé d'autre vecteur physique indépendant du mouvement de la fusée à mesurer, l'utilisation des capteurs magnétorésistifs s'en trouve compromis.

La solution la plus simple semble donc d'utiliser des gyromètres. On va intégrer les angles de rotations au fur et à mesure du vol de la fusée. Le problème de l'initialisation est résolu par le fait que l'on connaît à priori l'attitude de la fusée à l'instant précédent le décollage. Il suffit pour cela de donner les consignes nécessaires à l'équipe de mise en rampe.

### 3.2 Synoptique de la fusée



### 3.3 Choix logiciel au sol.

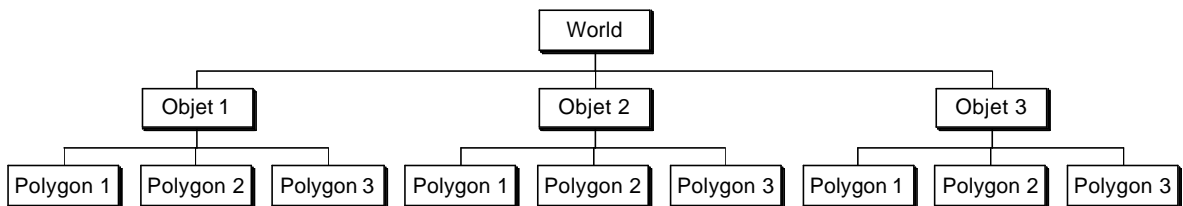
Afin de garantir une bonne lisibilité du programme et de permettre une réutilisation potentielle, le programme a été écrit en Programmation Orientée Objet sous Borland C++. Le programme se décompose donc en différents modules qui permettent de mieux localiser le code.

- **La programmation orientée objet :**

Module	Rôle
Video.cpp	Initialisation vidéo, changement de page vidéo, svga...
Bitmap.cpp	Classe de gestion des images bitmaps de la console
Console.cpp	Mise à jour de la console, des indicateurs, des graphes
Input.cpp	Contrôle les entrées claviers, la prise série et le séquenceur
Trames.cpp	S'occupe du stockage des données des trames
View.cpp	Classe de gestion du point de vue, des lumières, des zooms, des caméras...
World.cpp	Classe contenant les données de la scène et la liste d'objets
Objet.cpp	Classe contenant les données des objets et la liste des polygones
Polygon.cpp	Classe contenant les données des polygones
Pauline.cpp	Programme principal et boucle

**Tableau 1 : Classes utilisées dans Pauline**

Le programme fait une utilisation intensive des constructeurs, des destructeurs et du contrôle de l'allocation dynamique de mémoire. En ce qui concerne les données de la scène 3D, leur stockage se fait de manière hiérarchique. La classe World contient les différents objets, qui contiennent eux-mêmes l'ensemble des polygones. La fonction membre « Load » de chaque classe permet de récupérer les informations qui lui sont nécessaires à partir d'un fichier général.



**Organigramme 1 : La hiérarchie de classe**

- **Le système de stockage des informations**

Le stockage des informations de la scène 3D se fait dans le fichier .WLD. L'utilitaire Asc2Wld permet de convertir directement des fichiers élémentaires issus de 3D Studio au format .ASC en fichiers .WLD. Ceci facilite l'édition des objets 3D.

Le stockage des trames durant le vol ne se fait pas directement sur le disque mais dans la mémoire, ceci afin d'éviter des pertes de trames durant les temps d'accès au disque dur. On utilise dans ce but une liste chaînée à allocation dynamique. A la fin de l'acquisition les données sont directement sauvegardées dans le fichier précédemment sélectionné. Ils correspondent aux fichiers « Savetram.00? ».

- **La programmation de la prise série**

La programmation de la prise série a été faite à bas niveau à partir d'une documentation existante. La librairie établie se trouve dans le fichier serlib.c : L'arrivée de données sur la prise série déclenche une interruption. Les données sont alors placées dans un buffer. Cette

tâche est ensuite cachée du point de vue de l'utilisateur. Ce dernier récupère alors les données à l'adresse du buffer.

Référence : (voir dossier informatique : programmation de la prise série)

LaMothe, Ratcliff, Seminator & Tyler

1994, *Tricks of the Game Programming Gurus*, USA, SAMS Publishing

- **Les connaissances mathématiques utilisées**

La programmation de la partie 3D nécessite l'utilisation de connaissances mathématiques :

⇒ Les matrices de rotation et de translations en 3D

⇒ La projection 3D → 2D

⇒ Calcul des normales aux surfaces

⇒ Equations de plan et intersections

⇒ Calculs trigonométriques pour le placement des vues et des caméras

⇒ Algorithmes de faces cachées

⇒ Algorithmes : Z Sorting, Clipping...

De plus afin d'établir la trajectoire complète de la fusée on utilise :

⇒ L'intégration des données des gyromètres

Pour un complément d'informations concernant les algorithmes 3D, le livre suivant traite de la programmation d'un simulateur de vol.

Référence :

Christopher Lampton,

Build Your Own Flight Simulator With C++, USA, Coriolis

## 3.4 L'électronique

### **3.4.1 Le séquenceur.**

On connaît l'importance du séquenceur dans une fusée : Il devra déclencher le système de récupération. De son bon fonctionnement dépendra donc la bonne récupération de la fusée.

Avec Pauline, c'est le microprocesseur, qui grâce à la mesure de la vitesse, va décider du meilleur moment pour l'ouverture du parachute (Lorsque la vitesse est minimale, à culmination).

Le séquenceur n'est donc dans ce cas qu'une sécurité : Il devra, en cas de défaillance du cerveau de Pauline, interdire toute ouverture prématurée, et imposer l'ouverture au bout d'un temps  $t$  si le microprocesseur n'a pas ordonné cette dernière. Ce temps  $t$  est estimé avant le lancement comme légèrement postérieur à la culmination.

Ainsi, notre séquenceur devra interdire toute décision du processeur si elle est antérieure à l'instant  $t=8s$ , et imposer une ouverture si aucune décision n'a été donnée au bout de  $t=12.5s$ .

Le séquenceur sera donc composé de deux minuteriers : la première d'interdiction, la seconde d'obligation. Ces deux minuteriers ont été conçus avec des schémas très anciens (ils ont pratiquement 10 ans), mais qui ont fait leurs preuves sur de très nombreux projets au club.

Le principe en est simple :

- Le déclenchement de la minuterie, est détecté par un accélérocontact. Ce dernier va se fermer sur une bascule RS.
- La bascule autorise un 555 à osciller, et les 393 à compter.
- Deux 393 sont cascades permettant de faire une division suffisante de la fréquence du 555.
- Le bit de sortie du 393 servant de sortie va donc osciller avec une période de 25s ou 16s suivant les minuteriers.

Ces deux minuteriers utilisent donc des composants très simples mais robuste : NE555, 74HCT393,...

Le réglage des périodes (8s ou 12.5s) s'effectue grâce au potentiomètre du pont RC du 555.

L'initialisation à zéro (la mise en veille) est effectuée grâce à un pont RC placé sur la bascule RS de détection de décollage.

La sortie de la minuterie 1, servant à interdire toute décision antérieure à  $t=8s$ , est donc connectée à une logique à base de portes.

Cette logique gère les informations provenant de la minuterie 1, de la minuterie 2, de l'entrée décision processeur.

Sa sortie attaque un optocoupleur, déclenchant un transistor MOS pour la mise en route du moteur électrique.

Ce moteur libère alors le parachute. Toutes les informations entrantes et sortantes en direction du microprocesseur sont optocouplées. Ce qui permet de se dégager de toute défaillance électrique de ce dernier.

On a de plus, pris soin de placer des capacités de découplage sur les alimentations des circuits intégrés.

### 3.4.2 L'alimentation.

Cette année, au niveau des alimentations, le but était double :

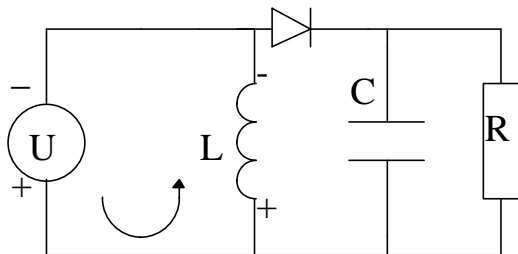
- Faire des alimentations efficaces au niveau du rendement : donc plus de régulateurs linéaires, dissipant le surplus d'énergie sous forme thermique (pouvant être considéré comme du gaspillage).
- Faire des alimentations légères : donc embarquer le moins de piles ou d'accus possible et optimiser le rendement de ceux-ci.

La solution à ces impératifs a donc été le remplacement des alimentations à régulateurs, faciles à mettre en œuvre, mais lourdes (nombreux accus pour obtenir une autonomie suffisante), et surtout au rendement trop faible, par des alimentations à découpages, peut-être plus onéreuses et plus difficiles à mettre en œuvre mais beaucoup plus économes en énergie et nécessitant de ce fait moins d'accumulateurs d'où un poids réduit.

#### 3.4.2.1 Principe de l'alimentation à découpage :

On peut le découper en deux étapes :

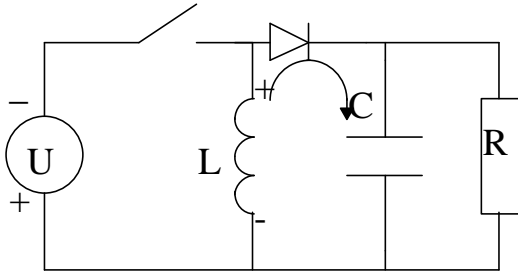
##### Etape 1 : interrupteur fermé



Il y a accumulation d'énergie dans la self qui est soumise à la tension du générateur.

Le courant croît linéairement dans la self qui stocke de l'énergie ( $E = \frac{1}{2}LI^2$ )

## Etape 2 : interrupteur ouvert



La self devient génératrice, la tension s'inverse à ses bornes, l'énergie est acheminée par la diode au circuit récepteur.

Le courant traversant le self est décroissant mais toujours dans le même sens.

Dans ces schémas simplifiés, l'interrupteur simule en fait le transistor de découpage.

### 3.4.2.2 Mise en œuvre

Nous avons trouvé les schémas des alimentations dans les notes d'applications de linear technology.

La première alimentation réalisée en tant que prototype a été réalisée autour d'un LT1172 selon le schéma suivant :

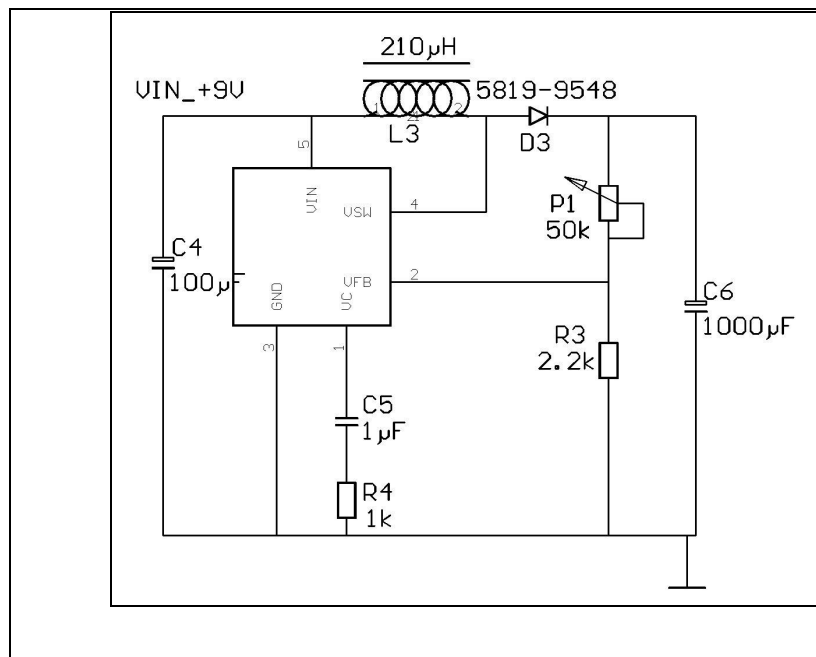


Schéma 1 : alimentation +12V selon data-sheets

Nous avons remarqué certains défauts rapidement :

- Sans la capacité C4 beaucoup de parasites remontent sur l'alimentation  $\Rightarrow$  nécessité de rajouter ce filtre passe bas qui améliore nettement les choses.



- On observe également beaucoup de parasites sur la sortie  $\Rightarrow$  il sera nécessaire de rajouter un filtre passe bas constitué d'une self (valeur  $100\mu\text{H}$ ) et d'une capa ( $100\mu\text{F}$ ) (cf. schémas finals)
- Le réseau RC monté sur  $V_c$  représente en fait le temps de réponse du montage pour se stabiliser. On peut modifier R et C : mais si R est grand et C faible (temps de réponse faible), dans ce cas : attention aux oscillations ! Ainsi on trouve que le réseau RC monté sur  $V_c$  doit être ramené à  $10\text{k}\Omega$  et  $22\text{nF}$ .

Nous avons procédé à des essais sur le prototype avec un  $4,5\text{V}$  apporté par une alimentation réglable en tension et en courant : le problème est qu'il faut autoriser les surintensités pour que l'alimentation démarre car il y a deux régulateurs dans le LT1171 : un premier en fonction de la tension d'entrée (Si  $V_{\text{out}} < V_{\text{out}} \text{ consigne}$ ) et un deuxième en fonction de la tension de sortie (si  $V_{\text{out}} \approx V_{\text{out}} \text{ consigne}$ ). En bref : ne pas limiter le courant d'entrée à l'alimentation à découpage car ça peut l'empêcher de démarrer. Si le montage ne démarre pas, essayer de remplacer l'alimentation de laboratoire par une pile de tension adéquate. Une alimentation sous dimensionnée peut empêcher un montage fonctionnel de démarrer.

Le cas du  $-12\text{ Volt}$  :

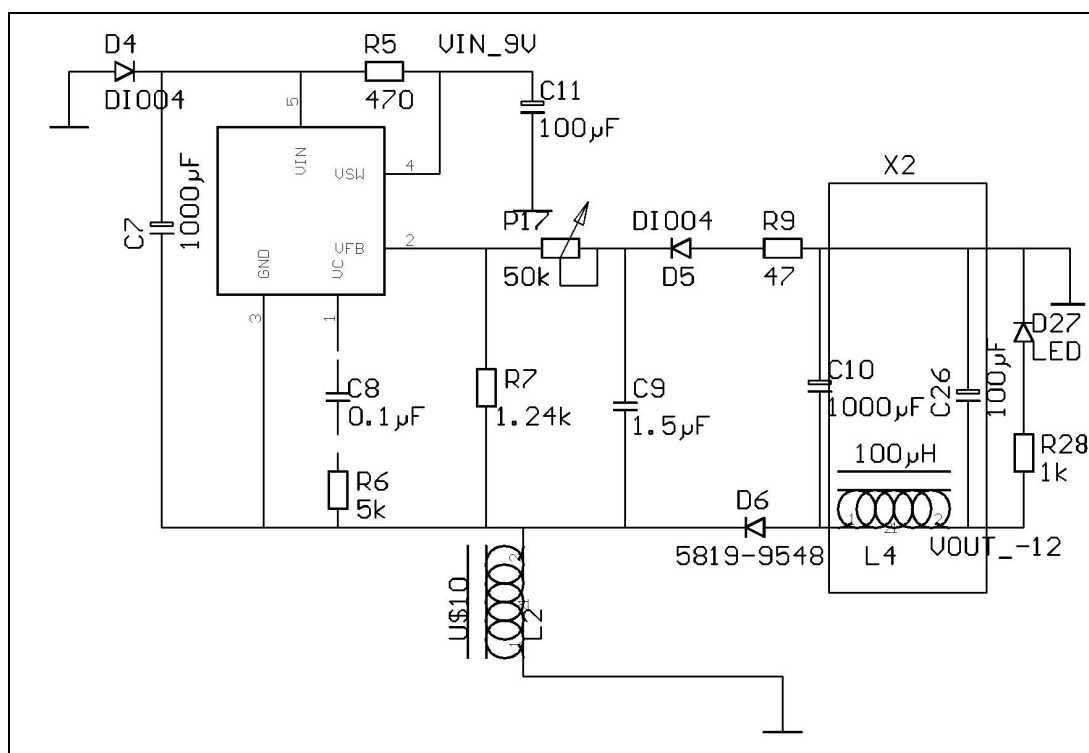


Schéma 2 : alimentation  $-12\text{V}$  selon data-sheets

Ce montage a fonctionné sans trop de problèmes. Toutefois son fonctionnement est déroutant : les broches  $V_{\text{in}}$  et  $\text{GND}$  sont flottantes par rapport à la masse :  $\text{GND}$  varie de  $V_{\text{out}}$  à  $V_{\text{in}}$  en passant par 0 et la tension  $V_{\text{in}}-\text{GND}$  est constante et égale à  $-V_{\text{out}}$ .

En bref pour le mettre au point : mesurer FB par rapport à  $\text{GND}$  et optimiser la capa  $1,5\mu\text{F}$  de façon à avoir FB le plus stable possible. Ensuite, mettre le filtre au point, cette fois par rapport à la masse.

On prendra notamment garde à bien utiliser en D5 une diode signal pour des performances optimales.

### ***3.4.2.3 Conclusion sur les alimentations***

La réalisation de ces alimentations à découpage n'a pas posé de problèmes majeurs.

Cependant, nous avons du faire face durant la campagne à un problème inattendu : l'autonomie de la fusée était beaucoup trop faible  $\Rightarrow$  voir les erreurs à ne pas refaire.

### **3.4.3 Les interfaces avec les capteurs.**

Pauline possède deux types de capteurs :

D'une part les gyromètres, d'autre part le capteur de pression différentielle permettant de faire la mesure de vitesse.

C'est ce dernier qui a demandé le plus d'attention.

En effet, la dynamique de sortie des gyromètres est  $-5V ; 5V$  alors que celle du Pitot est de  $0-20mV$

De plus ce capteur étant proche du Pitot, il se trouve donc dans le cône. Or on y trouve aussi l'émetteur FM.

Et il est bien connu que  $20mV$  et  $300mW$  HF ne font jamais bon ménage.

Le signal extrait du capteur de pression devait donc être aussitôt traité électroniquement. Comme nous l'avons dit plus haut, pour des questions d'intégrations, nous devons placer le capteur et son électronique de traitement face à l'émetteur. Cette électronique devra donc être particulièrement soignée.

La sortie du capteur étant du type différentielle, on a un étage d'entrée à AOP effectuant l'adaptation.

Il est à noter que ce type d'entrée introduit un gain de 2.

Ensuite, on retrouve un AOP monté en ampli non-inverseur. Il dispose de plus d'un potentiomètre permettant de régler ce gain.

Enfin, on retrouve un ampli permettant d'ajouter un offset et donc d'ajuster le minimum de vitesse.

L'offset ajouté provient d'une référence de tension, lui assurant ainsi stabilité et tenue en température.

De plus, on a ajouté un condensateur en parallèle à la contre réaction de ce sommateur. Ceci enlève toutes les perturbations HF pouvant s'ajouter au signal pendant son traitement.

Enfin, pour garantir un minimum de bruit sur la sortie du signal, on a découplé les alimentations par un double filtre LC, ainsi que les alimentations des CIs grâce à des condensateurs de  $100nF$ .

Pour éviter toutes perturbations d'origine externe, le circuit imprimé sera du type double face : une face signal, et une face ne comportant qu'un plan de masse unique.

En ce qui concerne les gyromètres, nous avons dû réaliser de simples amplificateurs atténuateurs pour diminuer la dynamique de sortie des capteurs.

### **3.4.4 La carte processeur.**

C'est le cœur du système électronique. Le processeur va rassembler les informations pour en déduire les angles de rotations de la fusée, ainsi que sa trajectoire dans le repère associé au lieu de lancement.

Ce processeur sera d'autre part chargé de la transmission au sol des informations (téléométrie), de la sauvegarde interne sur EEPROM, et de la sortie du parachute.

La carte devait de plus répondre à des contraintes de tailles.

On s'est donc orienté rapidement vers un 80C552 comme choix de processeur : Ce microcontrôleur a comme avantage de posséder un CAN 8 voies 10 bits en interne, un bus I2C, plusieurs bits d'entrées et de sorties, ainsi qu'un chien de garde.

Les caractéristiques de cette carte sont donc :

- Processeur 80C552.
- 32Ko de RAM externe.
- 32Ko de zone programme externe.
- 4 entrées analogiques avec entrées différentielles, et compensée en offset.
- 4 E/S logiques.
- EEPROM 2Ko au format I2C
- générateur FSK 4800 bauds

Pour détailler l'électronique de cette carte, on va la diviser en sous-ensemble :

#### ***3.4.4.1 La partie numérique.***

Pour son fonctionnement, nous devons démultiplexer les adresses des données du bus du processeur. Pour cela nous utilisons un latch 74574 dont l'horloge de verrouillage est synchronisée par le signal ALE.

Attention: Il ne faut pas oublier les résistances de pull-up de AD0 à AD7.

Une fois le bus de données et d'adresses démultiplexés, on va donc pouvoir y placer la mémoire RAM et l'UVPROM.

L'UVPROM sera validé par le signal PSEN (zone programme), et la RAM par le signal RD ou WR (zone de données).

L'EEPROM I2C est placée sur le bus I2C du processeur, donc directement sur les signaux SCL et SDA.

Attention: ne pas oublier les résistances de pull-up à 10 K sur ces deux signaux. Les ports étant du type à drains ouverts.

Sur l'EEPROM, on a configuré l'adresse à 0 en plaçant  $E0=E1=E2=0$ .

Enfin on a autorisé l'écriture en imposant  $WR=0$ .

Le protocole I2C sera expliqué succinctement dans la partie informatique embarquée.

Enfin le Reset est obtenu grâce à un condensateur externe de 1 $\mu$ F (La résistance du pont RC étant en interne).

#### ***3.4.4.2 La partie analogique.***

Cette partie a été conçue avec le souci de la précision.

Il est en effet très difficile d'obtenir une mesure sur 8 bits vrais. Nous avons donc voulu mettre toutes les chances de notre côté en immunisant l'étage précédant le CAN au bruit et à l'erreur d'offset.

Le bruit généralement constaté dans les fusées, est l'émission FM rayonnée (ou bien conduit par les alimentations) par la partie HF.

Ce bruit est notamment absorbé lors de la transmission par câble, du capteur vers l'étage d'acquisition.

Pour augmenter le rapport signal/bruit du câble de transmission, on a tout simplement placé une résistance d'entrée faible. On aura donc une attaque à faible impédance d'entrée (meilleur qu'une attaque à forte impédance d'entrée).

D'autre part, l'étage entrée est du type différentiel : cela permettra de diminuer le bruit transmis dans le câble. Par contre, ce dernier devra être du type différentiel blindé.

L'étage suivant permet de faire un filtre d'ordre 2, suffisant pour éviter le repliement de spectre lors de la numérisation du signal.

Enfin, le dernier étage ajoute un offset référencé au Convertisseur Analogique/Numérique.

En effet, le signal reçu des gyromètres se situe entre  $-2.5V$  et  $+2.5V$  : l'absence de rotation étant donc représentée par une sortie nulle.

L'entrée du CAN est quant à elle comprise entre 0 et 5V. Il apparaît donc qu'il va falloir ajouter un offset de 2.5V.

Or cet offset doit être de grande qualité : Une erreur de + de 19mV entraînerait alors un offset numérique d'un bit.

Ce bit étant intégré 30 fois par seconde pendant 13s, on comprend alors que les erreurs engendrées peuvent être catastrophiques.

L'idée pour remédier à ce problème, est de se servir de la référence servant au CAN. Ce dernier en ayant besoin pour son 5V.

On va donc utiliser un pont de grande précision (résistances à 0.1%) pour obtenir le 2.5V à partir du 5V de référence.

Ainsi si le 5V baisse de 0.1V le 2.5V baissera de 0.05V. L'offset ajouté sera donc toujours en milieu de la gamme de conversion, et aucune erreur numérique supplémentaire ne sera introduite.

Pour garantir une grande stabilité à la tension de référence de 5V, nous avons opté pour un montage autour d'un AD589, « Boosté » par un transistor en sortie.

Enfin la tension générée par le pont de précision, est injectée dans un AOP de précision, servant de suiveur, avant d'être envoyée à toutes les chaînes analogiques.

Pour garantir des faibles dérives et des gains précis, toutes les résistances de la chaîne sont à 1%.

#### **3.4.4.3 La télémétrie.**

La télémétrie est le moyen de codage des informations sortant du microcontrôleur et devant être transmises au sol.

Pour être au standard de l'ANSTJ, nous avons opté pour un générateur FSK<sup>5</sup> à 4800 Bauds.

Pour simplifier le montage, ce dernier a été conçu autour d'un composant ayant fait ses preuves et remplissant parfaitement cette tâche : le XR2206.

Quelques composants sont nécessaires en externes pour son bon fonctionnement. Nous avons de plus rajouté un AOP permettant une amplitude suffisante à l'attaque de l'émetteur.

#### ***3.4.4.4 Les ports d'entrées - sorties.***

Le montage de ces ports d'entrées / sorties est relativement simple : En général des simples résistances de Pull-up suffisent.

#### ***3.4.4.5 Les alimentations.***

Pour les alimentations, nous avons filtré les alimentations analogiques en plaçant un double filtre LC sur ces alimentations.

D'autre part, pour le 5V, nous avons placé une capacité de forte capacité à l'entrée de la carte, ainsi que des capacités de plus faible valeur à proximité des composants numériques.

### **3.5 L'informatique embarquée.**

#### **3.5.1 Cahier des charges.**

Le logiciel embarqué a pour rôle d'interroger tous les capteurs, de traiter les données de façon à les rendre exploitables, de les envoyer au sol, et enfin de décider des actions à entreprendre.

##### ***3.5.1.1 Contraintes spécifiques.***

###### **3.5.1.1.1 Le débit de données.**

Le débit des données est la première contrainte. En effet, la visualisation au sol en temps réel nous impose un débit minimal de 25 images par seconde pour obtenir une qualité visuelle satisfaisante. L'autre facteur est la longueur de la trame de donnée envoyée par télémétrie. Le débit étant fixé à 4800 bauds, soit 480 octets (de 10 bits) par seconde, atteindre 25 images par secondes nous donne des trames de 19 octets au maximum. Comme on peut s'en tirer à 17 octets, on a fixé la trame à 17 octets. Partant de là, la fréquence des trames est de 28,2 trames par secondes, soit 28,2 images par seconde. Au niveau qualité visuelle c'est satisfaisant.

###### **3.5.1.1.2 L'intégration numérique**

Le calcul de l'attitude de la fusée est, schématiquement, une simple intégration numérique. C'est à dire qu'à intervalles de temps réguliers et fixes, on doit ajouter à l'attitude actuelle de la fusée, la variation d'attitude depuis l'instant de sommation précédent. D'autre part la qualité de la liaison HF est trop médiocre pour espérer avoir 100% des trames justes au sol.

---

<sup>5</sup> **FSK** : Frequency Shift Modulation ( Modulation par décalage de fréquence)

De cela on déduit que le calcul de l'attitude de la fusée doit se faire dans la fusée elle-même, sur la base d'un échantillonnage des capteurs réalisé à fréquence fixe.

#### 3.5.1.1.3 La sauvegarde des données

Comme on l'a souligné auparavant, la transmission des données n'est pas fiable, c'est pourquoi on n'aura pas au sol la totalité des trames. Du moins pas en temps réel, par contre on souhaite les avoir quand même complètement en temps différé. Pour se convaincre de l'utilité d'une telle sauvegarde, il suffit de d'imaginer les conséquences d'une liaison radio totalement défaillante.

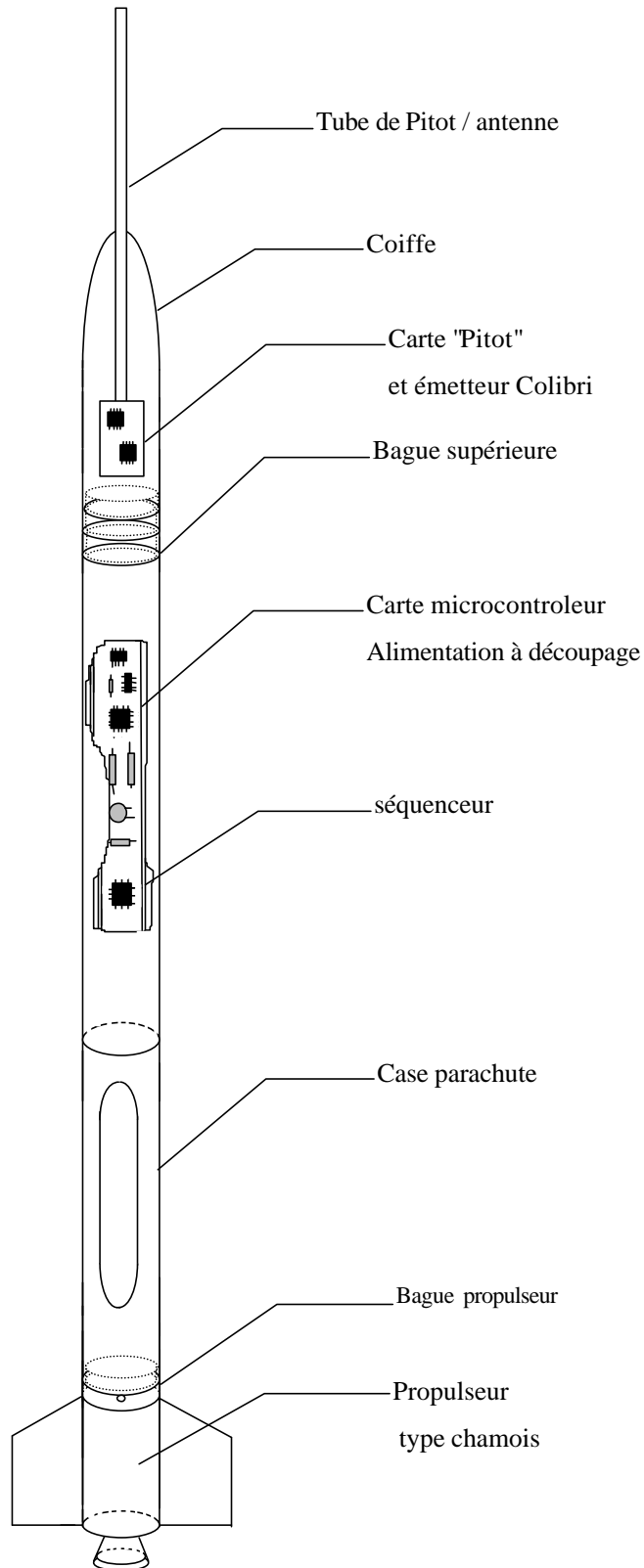
Il est donc nécessaire de sauvegarder sur un support inaltérable, en cours de vol, toutes les données indispensables au calcul de l'attitude de la fusée.

#### 3.5.1.1.4 Détection du décollage

Cela permet de mettre en route les temporisateurs de fenêtrage de l'ouverture parachute. Cela se lit sur une broche d'entrée sortie du microcontrôleur, car c'est mémorisé sur la carte de temporisation.

## 4. La réalisation de Pauline.

### 4.1 La mécanique





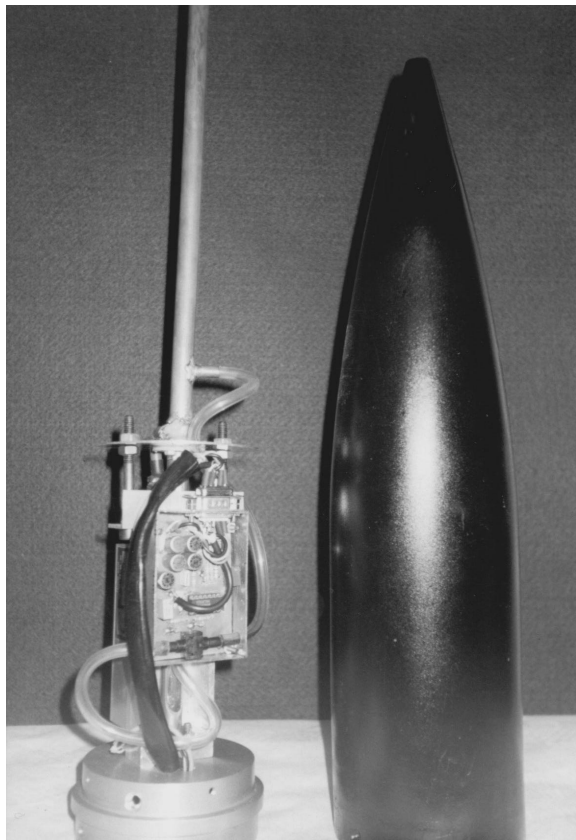
#### 4.1.1 Le corps de la fusée.

Le corps de la fusée n'a demandé aucun traitement, si ce n'est l'ouverture de la porte parachute effectué au moyen d'une scie sauteuse.

Une fois cette découpe effectuée, le tube a été poncé avant d'être anodisé (couleur noire). Le ponçage doit être parfait, l'anodisation ne masquant pas les défauts d'état de surface.

#### 4.1.2 Le cône.

Il convient d'abord de réaliser un contre-moule de la forme du futur cône, en bois. De la qualité de ce modèle dépend fortement celle de la pièce finale.



En prenant l'empreinte du moule en bois dans du plâtre frais, on obtient le moule final, qui a donc la même forme, mais creuse. L'état de surface de la pièce mâle est très important afin de pouvoir espérer la retirer du plâtre. Il faut également éviter les formes "exotiques" et les tronçons cylindriques. Toutes les parties du cône doivent donc former un angle aigu d'au moins quelques degrés, même les derniers centimètres qui lient le cône au tube.

Le temps de séchage du plâtre varie en fonction de son type ("Plâtre Paris", "plâtre de moulage", "dur", "très dur"...).

L'état de surface du futur cône dépend fortement de celui de l'intérieur du moule en plâtre que l'on aura poncé, poncé et re-poncé puis peint pour le rendre lisse et imperméable. Après de longues et laborieuses heures, le moule en plâtre est enfin prêt. On couvre alors

la surface intérieure du moule de bandes de fibre de verre préalablement imprégnées de résine (la célèbre résine-qui-va-bien). Il est nécessaire de presser la couche de fibre de verre afin d'empêcher la formation de bulles disgracieuses. On utilise pour cela une baudruche que l'on gonfle à quelques bars à l'intérieur du futur cône. Cette étape est critique et difficile à réussir.

Après durcissement de la résine, on peut démouler le cône ou, plus simplement, casser le moule en plâtre.

Il ne reste alors plus qu'à poncer et peindre cette œuvre d'art...

#### 4.1.3 Les bagues intermédiaires.

Les bagues intermédiaires ont été réalisées au club. Elles n'ont pas posé de problèmes particuliers si ce n'est que notre tour commence à voir ses outils vieillir.

Les premières bagues faites, furent la bague du cône, et son complément sur la fusée. Ceci à permis d'utiliser rapidement la première bague pour l'insérer (par collage) dans le cône. Celui-ci se voyant donc renforcé.

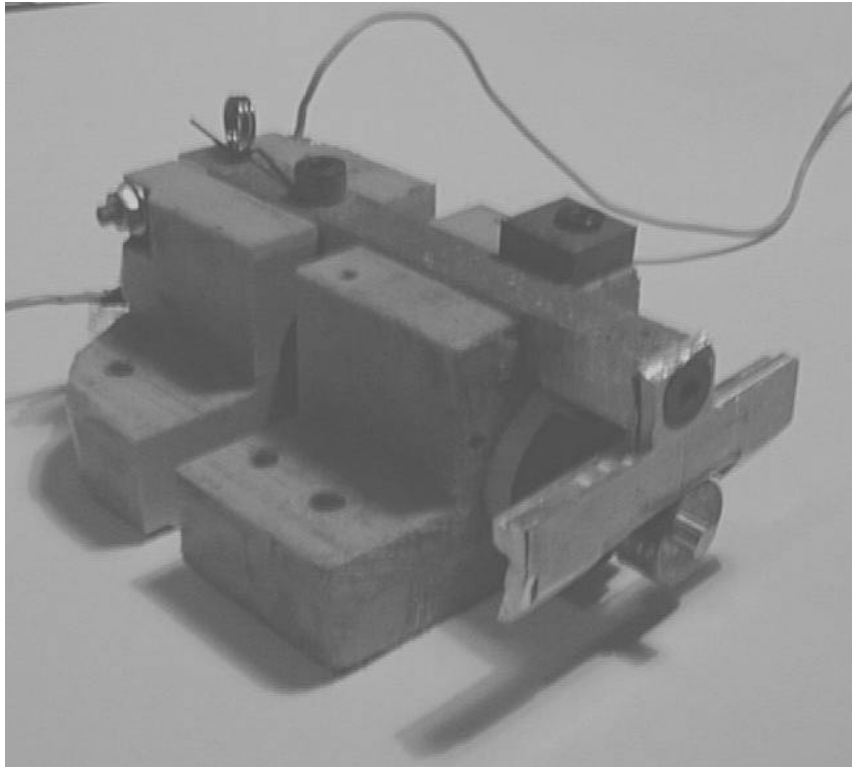
Enfin, toutes les bagues ont été poncées et anodisées couleur laiton (mais nous dirons plutôt couleur or !)

#### **4.1.4 Le système de récupération.**

Nous sommes partis du système de récupération de PSO (qui provenait lui-même de Super PHOENIX, qui provenait lui-même de PHOENIX, qui provenait...).

Nous avons du refaire certaines pièces de manière à faire entrer ce système dans les 100 mm de diamètre de Pauline.

Enfin, l'ensemble moteur +réducteur a été changé...



#### **4.1.5 Les ailerons.**

Une fois dessinés sous DMT, ils ont été découpés par plasma dans les ateliers de l'INSA.

Ensuite, la partie se fixant sur le corps de la fusée a été pliée de manière à suivre la forme du tube.

Des rivets ont été posés sur cette partie après alignement des ailerons.

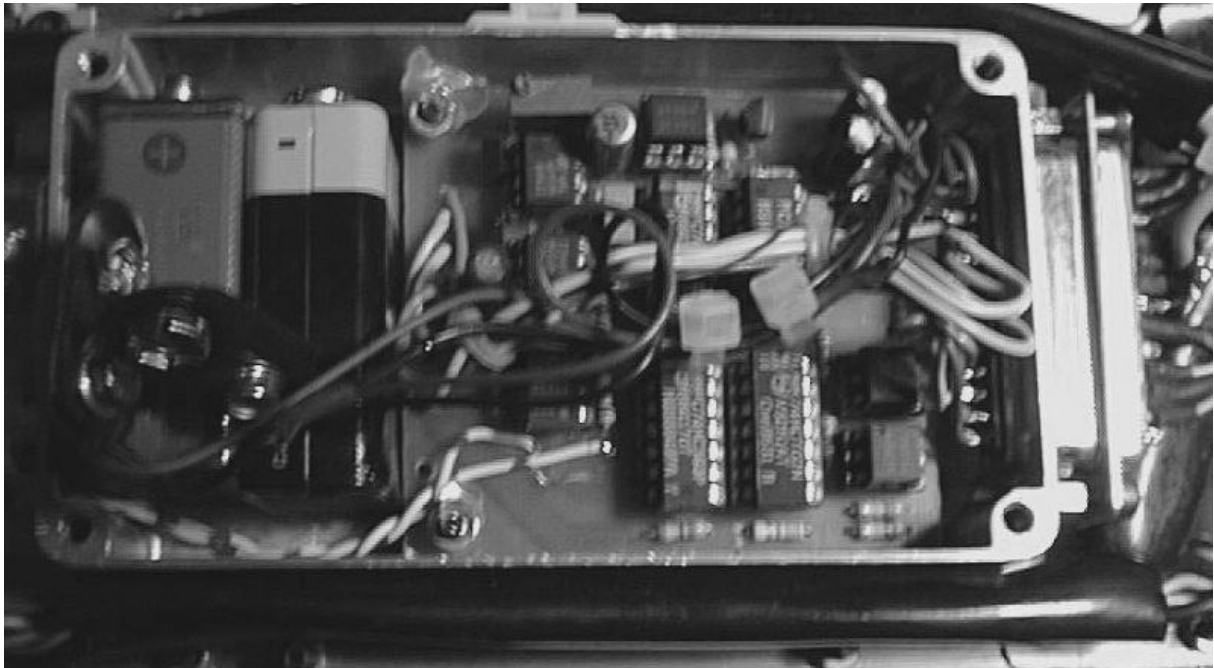
## 4.2 L'électronique

### **4.2.1 Le séquenceur.**

La réalisation du séquenceur n'a pas posé de problème majeur.

Il a fallu prêter attention à l'intégration de la carte dans le boîtier minuterie, de telle manière que l'on puisse y placer les deux piles 9V (alimentations séquenceurs et minuterie) qui sont indépendantes de l'alimentation du reste du montage.

D'autre part, accélérocontact prenant trop de place dans le boîtier, on a préféré le mettre à l'extérieur du boîtier.



#### 4.2.2 L'alimentation.

L'alimentation à découpage est par définition source de parasites. On a donc pris soin de réaliser un circuit en double face dont la face composant était réservée à un unique plan de masse.

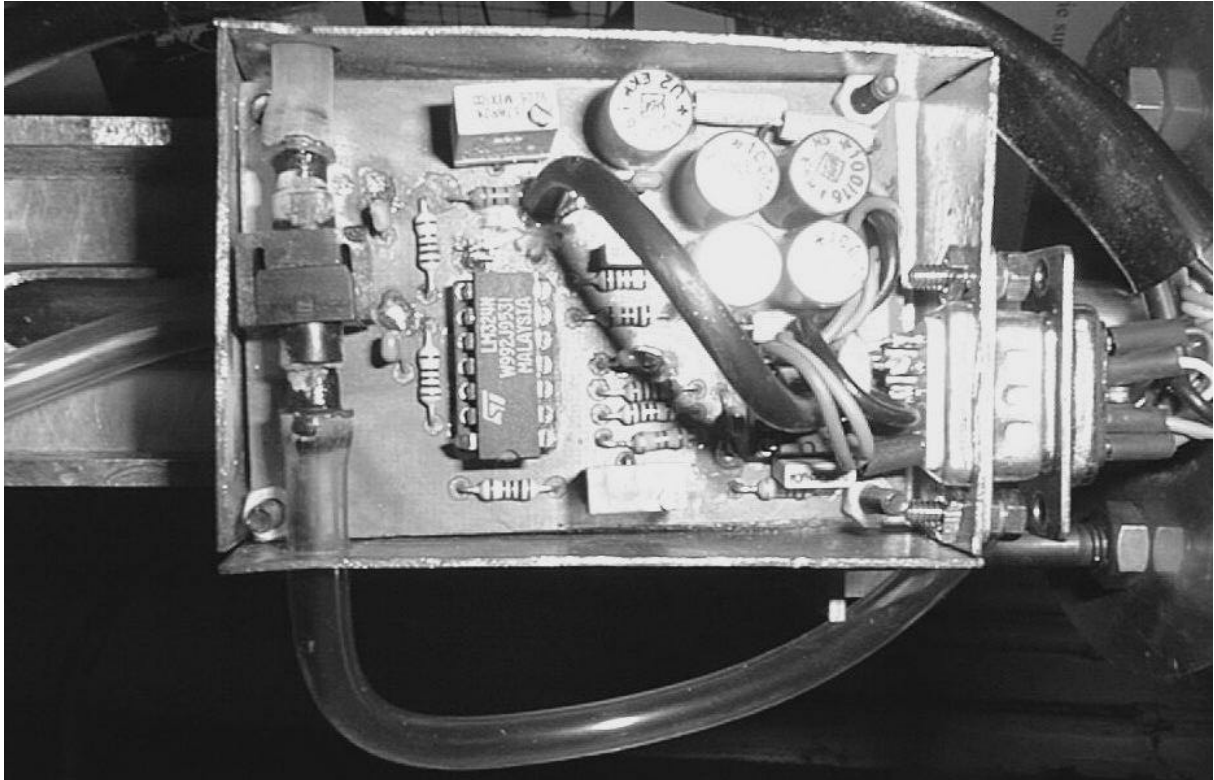
Par contre, nous avons fait attention que certains composants, qui disposent d'un radiateur dont le potentiel n'est pas relié à la masse, ne touchent pas ce plan de masse.



Enfin, le circuit électronique a été placé dans un boîtier métallique pour limiter le rayonnement émis.

### 4.2.3 Les interfaces avec les capteurs.

Pour le tube de Pitot, on a aussi réalisé le circuit en double face, la couche supérieure étant réservée au plan de masse, afin de s'affranchir au mieux des émissions de l'émetteur radio

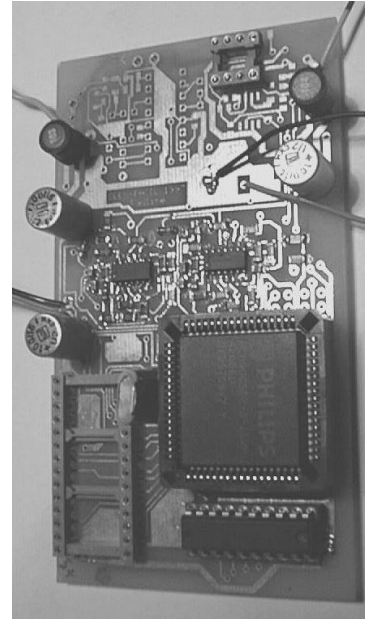
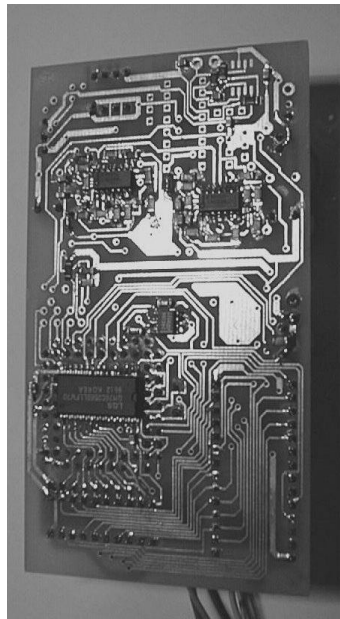


Le capteur est soudé directement sur le circuit, et le tout est placé dans un boîtier en aluminium traité EMI/RFI, pour diminuer au maximum les influences néfastes de l'émetteur COLIBRI.

#### 4.2.4 La carte processeur.

Pour optimiser la taille de la carte, il a été choisi d'utiliser au maximum les composants CMS. Cette technique permet de diminuer la taille des empreintes physiques, mais aussi de pouvoir placer des composants des deux côtés de la carte tout en permettant de fixer la carte dans un boîtier grâce à la hauteur limitée des composants.

Faces inférieure et supérieure de la carte processeur en cours de montage :

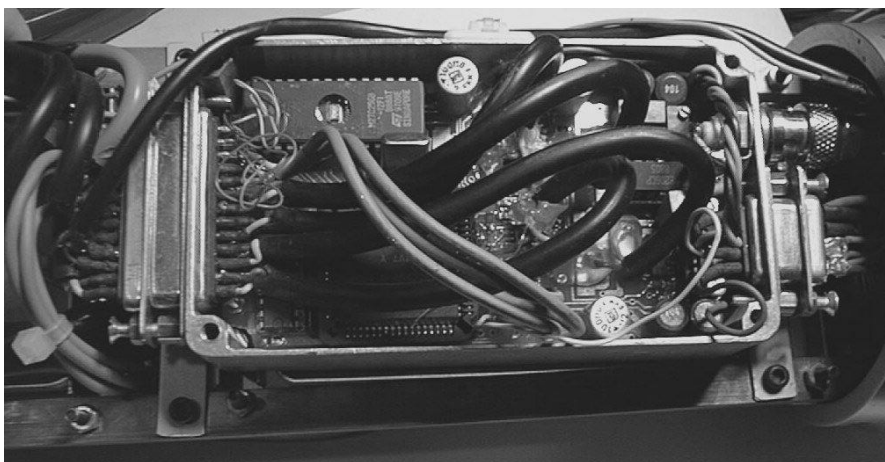


Au final on a obtenu une carte avec toutes les fonctionnalités décrite au chapitre 3.4.4, et dont les dimensions sont extrêmement faibles.

La réalisation de cette carte était déjà plus compliquée puisqu'elle nécessitait un circuit double face à trous métallisés.

Le circuit a donc été routé au club, mais tiré par la société RPCI. Les composants CMS difficiles d'accès ont d'abord été soudés, puis se fut le tour des traversants les plus encombrants, et enfin les CMS classiques furent posés en dernier.

Pour cette carte, il a fallu veiller à ne pas réaliser de pont de soudure, notamment lors de la pose de la RAM CMS.



La même carte après intégration dans un boîtier et connexion des entrées/sorties.

### **4.3 L'informatique embarquée.**

#### **4.3.1 Découpage logique.**

Etant donné le cahier des charges, le découpage du soft est tout fait :

1. Interrogation des capteurs
2. Calcul des nouvelles valeurs
3. Construction de la trame de données
4. Expédition de la trame
5. Sauvegarde des données
6. Ouverture de la porte parachute

Il ne manque plus qu'une étape d'initialisation et une boucle pour envoyer les trames en boucle.

#### **4.3.2 Découpage temporel.**

Comme on l'a vu au chapitre 3.5.1.1.2, il est indispensable d'avoir une base de temps régulière pour échantillonner et faire les calculs. Cette base de temps nous est fournie par la transmission série des données au sol. En effet, le temps mis pour envoyer une trame au sol est connu à priori, puisque à la fois le débit et la longueur de la trame sont fixes.

Le programme réalise donc la boucle suivante :

1. Initialisation de l'envoi d'une trame au sol en tâche de fond par port série
2. sauvegarde des données en tâche de fond par bus I2C
3. échantillonnage des capteurs (la date est connue avec précision)
4. intégration numérique
5. exploitation des données de la vitesse
6. attente de la fin de l'émission de la trame
7. retour en 1

On voit aisément que cette organisation permet échantillonner à date connue à la condition expresse que le temps d'exécution d'une boucle soit inférieur à la durée de transmission d'une trame, soit 28,2Hz, c'est à dire 35ms.

### **4.3.3 Etude détaillée de la programmation**

#### **4.3.3.1 Détection du décollage**

Cette étape, est une clé du programme.

En effet, ce point consiste simplement en la lecture d'un port d'entrée sortie.

Par contre, lorsque l'on vient de décoller, on remet à la position prédéfinie la fusée. En effet, en un quart d'heure en rampe, l'intégration numérique a largement eu le temps de faire



dérivée la fusée. Donc, au décollage, on la remet dans une position connue lors de la programmation, et respectée lors de la mise en rampe.

On commence de même à compter le nombre de trames émises depuis le décollage pour savoir si la propulsion est finie et si l'on peut envisager l'ouverture parachute. En effet, on verra au chapitre 4.3.3.6 que l'on décide l'ouverture parachute lors d'une phase d'accélération. Or de ce point de vue rien ne distingue la propulsion de la chute.

### ***4.3.3.2 Initialisation de l'envoi d'une trame au sol en tâche de fond***

#### ***4.3.3.2.1 Transmission en tâche de fond***

La liaison série du 80552 est dotée d'une interruption de fin d'émission d'un caractère. Il suffit donc de programmer l'interruption de telle manière qu'elle aille lire les données à transmettre dans un buffer linéaire de (en l'occurrence) 17 octets, en prenant à chaque occurrence de l'interruption l'octet suivant.

La liaison série émet alors un octet et se manifeste à la fin de son émission par une interruption.

Il ne reste plus à l'interruption qu'à changer une variable flag lorsqu'elle a émis tout le buffer pour signaler au programme principal qu'il faut de nouveau remplir le buffer d'émission, et qu'il peut recommencer une boucle de son programme principal.

#### ***4.3.3.2.2 Initialisation de la transmission***

Elle consiste à remplir le buffer d'émission, à initialiser le pointeur de position dans la trame, et à faire émettre un premier caractère. C'est tout, car à la fin de l'émission de ce caractère, une interruption interviendra, et l'on rentre dans le chapitre 4.3.3.2.1

### ***4.3.3.3 Sauvegarde des données en tâche de fond***

Comme on l'a vu au chapitre 3.5.1.1.3 il faut sauvegarder les données indispensables. Pour cela on dispose d'une EEPROM dans la fusée. Le premier point important est que cette EEPROM est de petite taille, on ne sauvegardera donc que ce qui est strictement nécessaire : les trois vitesses de rotation et la mesure du Pitot. C'est tout, et c'est suffisant pour recréer la totalité des autres données. L'autre point, est que l'EEPROM est programmable uniquement par un bus trifilaire à la norme I2C. On a de la chance, le 80552 gère le bus I2C de façon hardware, c'est à dire que l'on peut exécuter toutes les communications uniquement par interruption.

Le programme se résume donc à utiliser la bibliothèque I2C.C et I2C.H qui a été programmée pour l'occasion et permet de faire un transfert d'un buffer de longueur programmable vers un endroit programmable de l'EEPROM.

Le programme doit donc initialiser un certain nombre de variable (que faire et avec quoi) puis demander le transfert. Le bus étant rapide, et les données peu nombreuses, on est certain que ce transfert sera terminé avant que l'on en demande un autre au cours de la boucle suivante.

Un mot sur le bus I2C. Le bus I2C est un bus trifilaire : donnée, horloge et masse, qui permet une liaison multipoints entre plusieurs composants. Un composant au moins est un maître, c'est à dire qu'il peut interroger les autres composants. Les autres sont des esclaves,

c'est à dire qu'ils embarquent une logique non programmable, qui leur permet de comprendre des requêtes faites sur le bus et d'y répondre. Ainsi une EEPROM saura quand on s'adresse à elle, ce que l'on attend d'elle (lecture ou écriture) à quelle adresse et sur combien d'octets. Du côté 80552, l'I2C étant codé en hardware tout se déroule sous forme d'interruptions. On peut donc faire des transferts de donnée en tâche de fond. L'I2C présente beaucoup d'avantage sauf que sa programmation est fastidieuse sans être réellement difficile à mettre au point.

#### ***4.3.3.4 Echantillonnage des capteurs***

Les capteurs sont branchés sur le convertisseur Analogique/Numérique intégré du 80552. On dispose des capteurs suivants :

1. rotation axe X
2. rotation axe Y
3. rotation axe Z
4. tension des piles
5. tube Pitot

Les trois premières valeurs sont indispensables pour passer à l'étape suivante qui est le calcul de l'attitude. On effectue la conversion de la manière suivante :

1. Demande de la conversion 1
2. Boucle d'attente, en attendant la fin de la conversion 1
3. stockage du résultat
4. Demande de la conversion 2
5. Boucle d'attente, en attendant la fin de la conversion 1
6. stockage du résultat
7. Demande de la conversion 3
8. Boucle d'attente, en attendant la fin de la conversion 1
9. stockage du résultat

Ensuite, les deux autres valeurs ne seront utiles qu'après le calcul de l'attitude qui est très long (quelques millisecondes) par rapport au temps de conversion (quelques microsecondes), on les effectue donc en utilisant les capacités d'interruption du 80552.

1. Programmation des variables adéquates
2. Appel de la conversion du Pitot
3. Au bout d'un certain temps, l'interruption de fin de conversion apparaît. Elle stocke le résultat et demande la conversion de l'alimentation
4. Au bout d'un certain temps, l'interruption de fin de conversion apparaît. Elle stocke le résultat et c'est tout.

#### ***4.3.3.5 Calcul de l'attitude***

Le calcul de l'attitude se fait en plusieurs étapes.

1. Calcul des nouveaux angles selon les trois axes de la fusée.
2. passage au repère terrestre
3. Calcul de la position de la fusée dans le repère terrestre

#### 4.3.3.5.1 Calcul des nouveaux angles

C'est une bête intégration numérique : On prend l'angle précédent, et on y ajoute le produit de la vitesse de rotation par la période d'intégration. La seule complication c'est les mises à l'échelle, qui imposent de travailler non pas en degrés mais en une unité fictive telle que  $360^\circ$  soient codés par 64k valeurs entières.

#### 4.3.3.5.2 Passage au repère terrestre

La formule de changement de repère implique le calcul de 6 sinus et cosinus. Si l'on utilise les bibliothèques mathématiques du compilateur C, on obtient un temps de calcul de 100ms (à la louche) au lieu de 20 imparties. On ne peut donc utiliser cette bibliothèque. L'astuce est d'utiliser une table de sinus précalculée et logée dans la ROM non utilisée par le code. Pour calculer un sinus d'un float, on appelle la fonction `sinus(float angle)` qui va chercher dans un tableau de 2000 angles  $(0..pi/2)*4$   $(0..2pi)*4$  (float)=16000 octets la valeur float du sinus correspondant. Le gain de temps obtenu est très important, pour une perte en précision faible, et surtout négligeable vis à vis de l'intégration numérique. D'autant plus que cette erreur n'est pas accumulée car non intégrée.

#### 4.3.3.5.3 Calcul de la position de la fusée

Il suffit de passer la mesure du tube de Pitot dans un tableau de 256 valeurs, pour avoir un octet codant la vitesse.

N.B. La vitesse est codée sur un octet seulement, pour deux raisons : le Pitot est très imprécis surtout que l'on peut difficilement l'étalonner sur toute sa plage de mesure, et deuxièmement, ce n'est qu'une expérience accessoire dans Pauline.

#### 4.3.3.6 *Exploitation des données de la vitesse*

La vitesse nous permet de détecter la fin de propulsion, et la culmination.

Ces mesures sont basées sur la dérivée de la vitesse. Lorsque la vitesse décroît une première fois après le décollage, c'est la fin de propulsion. Lorsque la vitesse croît, c'est la fin de la culmination.

La détection de ces points se fait donc par simple comparaison avec la vitesse précédente. On a simplement rajouté un test pour vérifier que cette tendance se confirmait sur plusieurs valeurs successives, de façon à éviter les mesures défectueuses (c'est un peu l'idée des antirebonds). Bien sûr cela retarde la détection de l'événement, mais ce n'est absolument pas critique car le retard est de l'ordre de la centaine de millisecondes.

Lorsque la culmination est détectée, on demande l'ouverture de la porte parachute de façon permanente.

#### 4.3.3.7 *Attente de la fin de l'émission de la trame*

Là, c'est une simple boucle qui attend que le flag mis par interruption de transmission série passe à l'état fin de transmission/ début d'une nouvelle boucle. On suppose en effet que

les temps de calcul sont inférieurs au temps d'émission d'une trame. On n'a donc pas besoin d'interruption.

#### 4.4 L'informatique au sol.

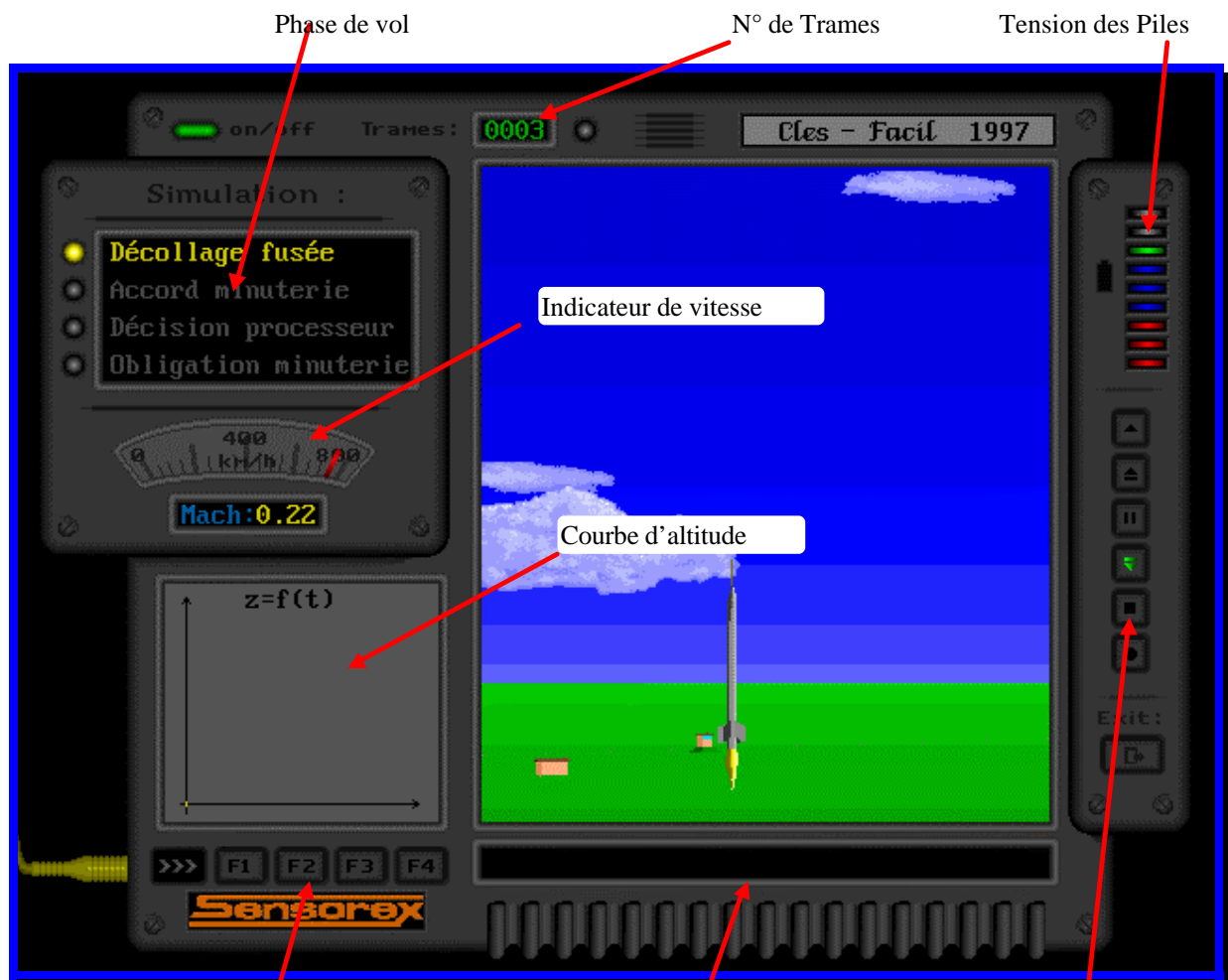


Figure 1 : Présentation du logiciel

Touches de fonction

Zone de messages

Touches de contrôle

- **Connections :** Avant la mise en route du logiciel, vérifier que la liaison série est connectée convenablement.
- **Mise en route :** Démarrez le logiciel en tapant PAULINE [ options ].
  - com1 ou -com2 : Port utilisé
  - 1200, -4800 ou -9600 : Vitesse en baud

-x2, ..., -x10 : Permet de n'afficher qu'une image sur deux  
-pinup, -nopinup : Permet de désactiver la pin-up Pauline

- **Mode démonstration**

Après le démarrage du logiciel, appuyez simplement sur une touche pour passer la page d'accueil. La démonstration peut être enclenchée à tout moment par la touche **F3**.

- **Didacticiel 1 : Visualisation d'une séquence**

Après le démarrage du logiciel, le logo du CLES-FACIL s'affiche. Appuyez sur une touche quelconque. Appuyez ensuite sur la touche **F4** afin d'accéder aux fichiers. Sélectionnez par exemple « CLES-FACIL » puis validez. L'animation s'enclenche alors automatiquement ou par la touche **F10**. Les touches **F6** à **F10** permettent de naviguer dans l'animation comme on le ferait avec un magnétoscope.

- **Didacticiel 2 : Mode acquisition**

Après le démarrage du logiciel, le logo du CLES-FACIL s'affiche. Appuyez sur une touche quelconque. Appuyez ensuite sur la touche **F5** pour enclencher le mode d'acquisition. Sélectionnez alors un emplacement qui sera supprimé et modifiez le nom si vous le désirez puis validez. Le bouton d'enregistrement devient rouge et le message « Connexion non établie » ou « connexion OK » s'affiche selon que des données parviennent ou non. Le logiciel est prêt à recevoir les données.

## **5. La campagne de lancement.**

### **5.1 Etat du projet lors de l'arrivée à Bourges.**

Lors de notre arrivée au festival, Pauline n'était pas entièrement terminée.

On distinguait 3 problèmes majeurs :

- L'accélérocontact, servant à détecter le décollage, n'était pas fixé correctement au séquenceur.
- Au niveau informatique embarquée : l'algorithme de changement de repère capteur-terre n'était pas tout à fait opérationnel.
- L'autonomie des piles était insuffisante.

### **5.2 Les qualifications.**

Les problèmes évoqués ont été résolus plus ou moins difficilement. Le moins évident a été de rajouter des piles pour garantir une autonomie suffisante.

Une fois ces modifications réalisées, les qualifications ont débuté.

Au niveau mécanique, la grosse surprise fut dans un premier temps l'instabilité de la fusée. En effet, lors de la visite d'avancement en fin juin, nous avons diminué la taille des ailerons pour assurer la stabilité de la fusée.

Au festival, les mesures donnaient tendance à agrandir les ailerons.

En fait, notre fusée était en limite de stabilité, et les mesures (dimensions de la fusée) prises lors du festival se sont avérées fausses. En refaisant les mesures telles qu'elles avaient été effectuées en juin, nous avons alors retrouvé la stabilité de la fusée.

Les tests séquenceur d'un point de vue mécanique et électronique se sont très bien déroulés.

Le test de la partie capteur, pré-HF, émission se sont bien déroulés.

Un nouveau problème est apparu lors du test Vol-Simulé. La télémétrie FSK ne fonctionnait plus du tout. En fait, à force de faire des tests et donc de démonter et de remonter l'intégration de la fusée, un des fils conduisant le -12V c'était dénudé, et en touchant la carcasse métallique, court-circuitait l'alimentation.

Après avoir résolu ce problème, Pauline a donc été validée ( à 4H00 le Samedi matin...).

### 5.3 La chronologie finale

Nous sommes arrivés sur l'aire de lancement vers 10H00 environ.



Une fois tous les tests et montages finaux effectués en tente club, nous nous sommes dirigés vers l'aire de lancement (accompagné de TF1-LCI).

Le dernier test de télémétrie a montré le traditionnel problème d'émission HF en rampe. D'autre part, la rampe se trouvant dans une sorte de «cuvette », l'émission se trouvait très affaiblie.

Nous avons cependant décidé d'opérer au lancement, en espérant retrouver la liaison HF une fois la fusée décollée.

#### **5.4 Le lancement.**

Après l'évacuation de la quasi totalité du club et des journalistes, Bruno Tardy et Dominique Walter sont restés avec les pyrotechniciens pour procéder à la mise en route de la fusée.

Une fois les interrupteurs de télémétrie basculés, la ligne de mise à feu raccordé, Bruno a alors appuyé « bien fort et bien longtemps » sur le bouton, et Pauline s'est alors élevée dans les airs...

Nos espérances ont été exaucées. Dès la sortie de rampe, nous avons retrouvé l'émission HF.

Après quelques secondes seulement, Pauline n'est devenue plus qu'un point dans le ciel nuageux de Bourges.

Heureusement, au bout de 15 secondes, nous l'avons vue redescendre sous son parachute.

Nous avons alors pu nous précipiter sur l'écran de contrôle de notre récepteur de télémétrie.



#### **5.5 Récupération de la fusée.**

La fusée a été retrouvée dans le gabarit de l'ETBS. Les observateurs avaient repéré très précisément Pauline dans le domaine, et nous avons alors pu converger rapidement vers la fusée.





«Par hasard», et l'auteur sait de quoi il parle, nous avons retrouvé la porte qui s'était arrachée lors de la sortie du parachute. Cette porte a été récupérée à environ 100 m de Pauline.

La fusée a été récupérée en bon état. Seuls l'extrémité du cône et un aileron ont été légèrement endommagés lors du choc du tube de Pitot sur le sol.

Enfin, les renforts verticaux de la peau de la fusée placés à proximité de la porte latérale, ont été éjectés à l'ouverture et n'ont pas été retrouvés.

L'intérieur de la fusée (intégration, électronique), n'a en aucune sorte souffert.

## 6. Les résultats obtenus.

### 6.1 Les résultats immédiats : la 3D.

Dans sa phase finale, le logiciel a fonctionné de façon très stable grâce aux nombreux tests effectués. Il a répondu à toutes les exigences formulées. Toutefois lors de la campagne de tir, la réception HF était assez faible, ce qui a causé la perte de certaines trames. En particulier, il a été impossible de réceptionner un signal lorsque la fusée se trouvait dans la rampe à cause de la structure métallique. Il a donc été impossible de vérifier la bonne transmission des informations dans la période précédant le lancer. Une fois la fusée sortie de la rampe la transmission fut opérationnelle et le vol fut visualisé en direct.

Le vol visualisé ne fut pas d'une grande fluidité à cause des trames perdues, mais ces circonstances avaient été prévues. Pour cette raison, nous avons décidé d'effectuer les calculs d'intégration et de trajectoires dans la fusée. L'avantage est que la perte d'une trame n'altère pas la trajectoire puisque la prochaine trame contiendra de nouveau la position exacte. Si ce choix n'avait pas été fait la trajectoire obtenue aurait été tout simplement erronée.

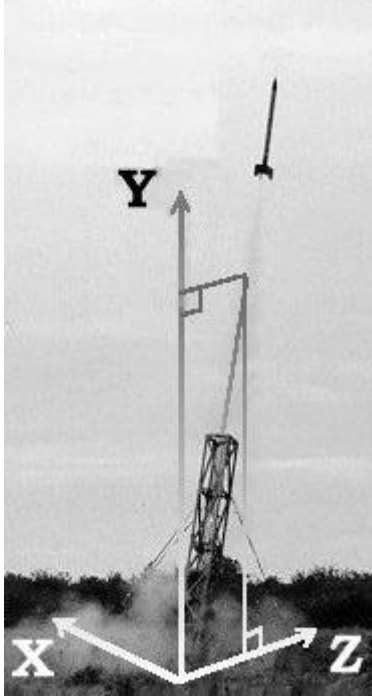
Le système de stockage, de visualisation, de démonstration et le système de vues ont très bien fonctionné et l'animation fut retransmise sur l'ensemble des écrans de l'aire de lancement. Le but consistant à présenter en temps réel les résultats au public a donc été atteint.

Enfin, bien que la transmission HF ne fut pas optimale, nous avons pris soin de sauvegarder l'ensemble des données dans la fusée sur EEprom, ce qui a permis de récupérer des données non altérées et de reconstituer le vol réel.



## 6.2 Analyse post-lancement.

### 6.2.1 Les repères utilisés

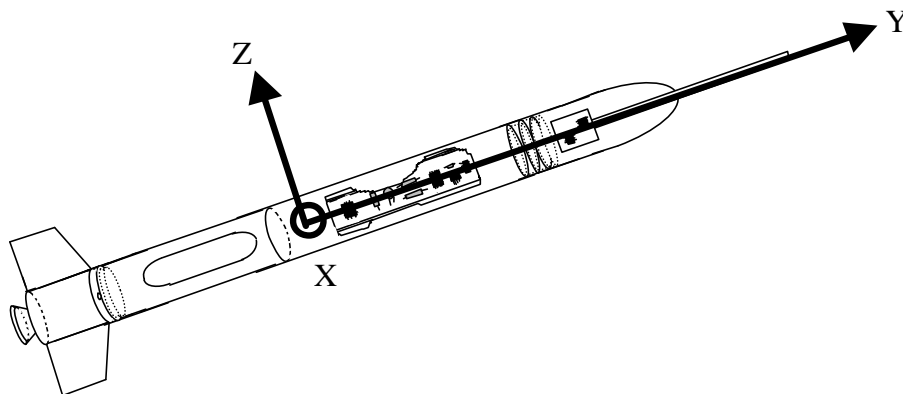


L'attitude et la position de la fusée sont données dans le repère suivant :

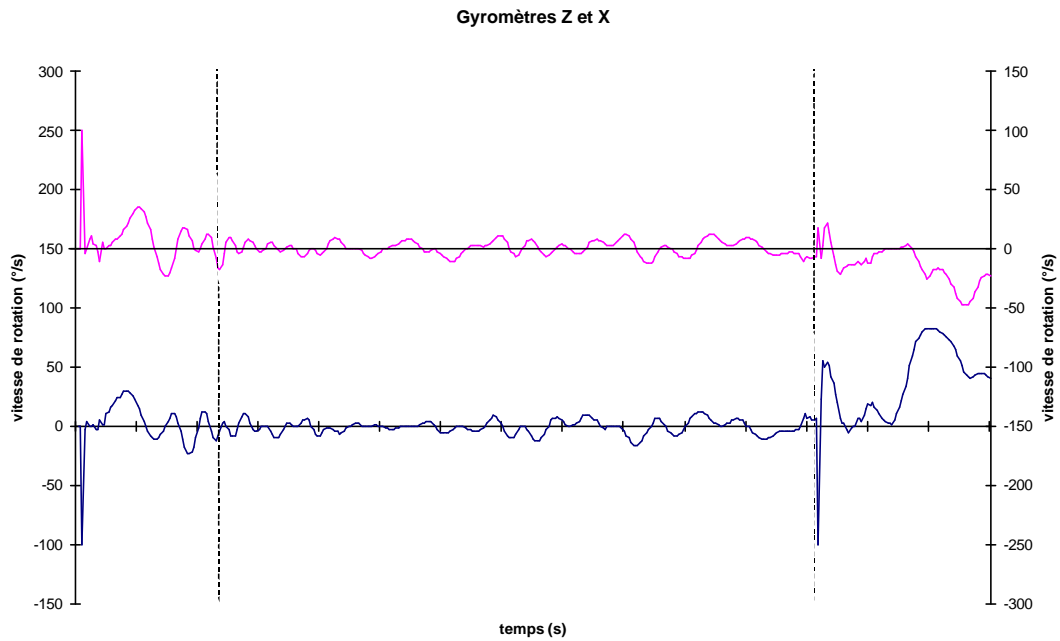
- L'origine est la base de la rampe
- Le vecteur Y est vertical, dirigé vers le haut
- Le vecteur Z est le projeté de l'axe de la rampe sur le plan horizontal dirigé vers la zone de lancement
- Le vecteur X est le résultat du produit vectoriel entre Y et Z

Les gyromètres sont placés selon les 3 axes du repère associé à la fusée en mouvement.

L'origine est le centre d'inertie de Pauline, le vecteur Y est porté par son axe longitudinal, dirigé vers la coiffe. X est perpendiculaire à la porte parachute. Le produit vectoriel de X et Y donne Z.



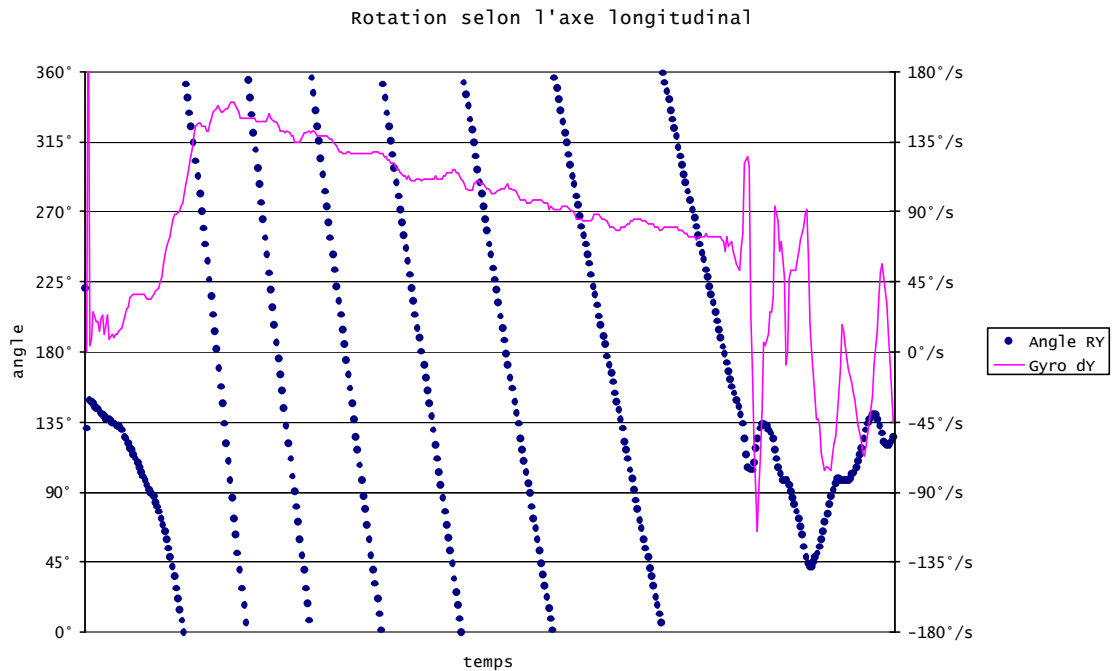
## 6.2.2 Rotation autour des axes transversaux (X et Z)



Les mesures de vitesse angulaire selon les axes transversaux de la fusée évoluent dans une plage de  $\pm 30^\circ/\text{s}$  durant la phase propulsée et  $\pm 20^\circ/\text{s}$  durant le vol balistique. Deux pics apparaissent lors de la mise à feu du propulseur et de l'ouverture du parachute. Hormis ces deux instant, le choix d'une plage de mesure de  $\pm 90^\circ/\text{s}$  est justifiée.

### 6.2.3 Rotation autour de l'axe longitudinal de la fusée.

Ce point est important dans la perspective de la réalisation d'une fusée asservie en rotation. Avant toute chose, voici les mesures que l'on a effectuées.



Il est important de se souvenir que cette courbe présente trois parties: l'accélération, la montée par la force d'inertie, et l'ouverture parachute. Il n'y a pas la descente sous parachute, qui a été volontairement omise, car peu significative.

De même on notera que le signe de la vitesse de rotation a été modifié pour les besoins du graphique. On ne perd toutefois aucune donnée significative.

#### 1. La première phase: l'accélération

Cette partie dure trois secondes approximativement, et contient la sortie de rampe. La vitesse initiale de rotation selon l'axe longitudinal est donc nulle. Cette vitesse s'accroît ensuite rapidement pour atteindre 160°/s ce qui fait presque un demi tour par seconde. On ignore dans cette interprétation le pic initial qui est probablement dû au choc de l'explosion initiale du propulseur.

#### 2. La deuxième phase: la montée

La majorité du vol se fait par force inertielle. Durant cette phase, on voit que la vitesse de rotation décroît régulièrement. Durant le même temps on sait que la vitesse linéaire décroît aussi. On peut donc affirmer, sans surprise, que la vitesse de rotation est fonction de la vitesse linéaire.

### 3. La troisième phase: l'ouverture parachute

La position de la fusée se fixe alors sur une position angulaire de  $120^\circ$ . Toutefois il y a de fortes variations autour de cette valeur. Ce mouvement est probablement la combinaison de deux facteurs: un "enroulement" de la fusée autour de la sangle du parachute, à la manière d'un yoyo, et d'autre part un mouvement pendulaire au bout du parachute.

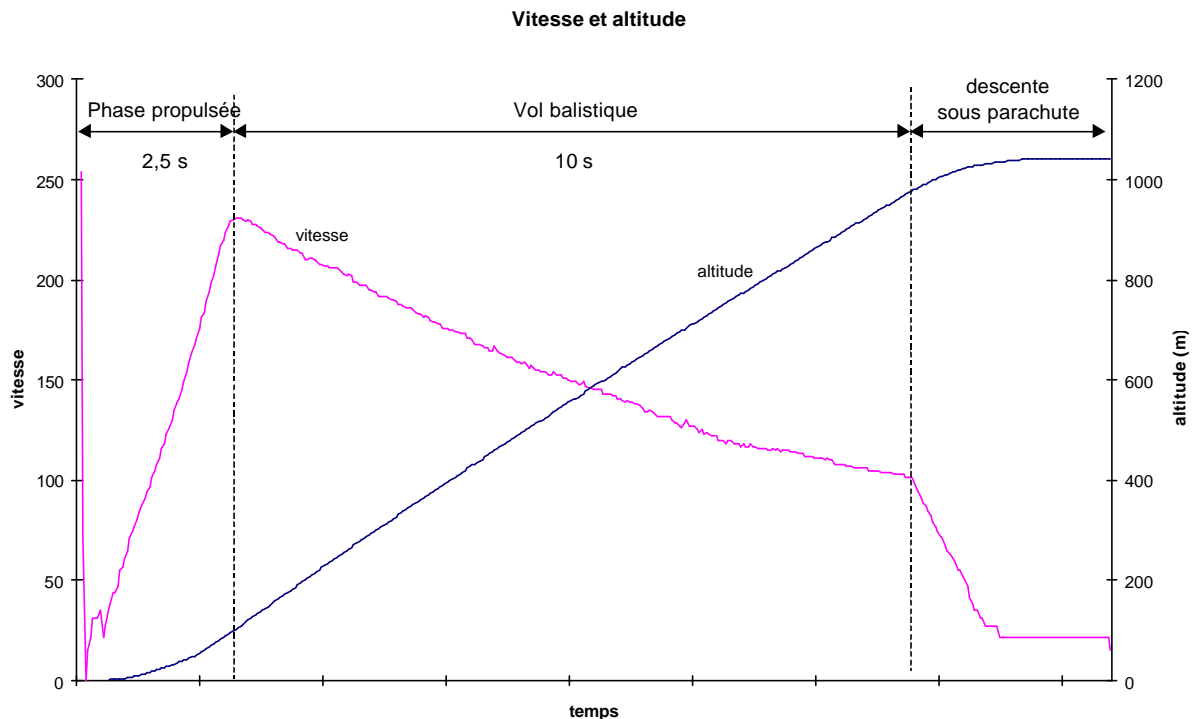
On notera d'autre part que la vitesse de rotation est assez peu bruitée. Au vu de la position angulaire, on ne peut pas objectivement affirmer que la vitesse de rotation n'est pas uniforme. Ce point permet de filmer pendant le vol, sans avoir un bougé trop important.

Ces données confirment le choix judicieux qui a été fait de choisir un gyromètre capable de  $\pm 500^\circ/\text{s}$ , puisque c'est le mieux adapté au niveau de la plage d'utilisation.

On n'omettra pas de se rappeler, si l'on réutilise ces données pour une autre fusée que Pauline, que si l'allure de la courbe a toutes les chances d'être la même les valeurs seront différentes, puisque la géométrie de la fusée entre en compte, et que chaque fusée est unique.



## 6.2.4 Vitesse de la fusée



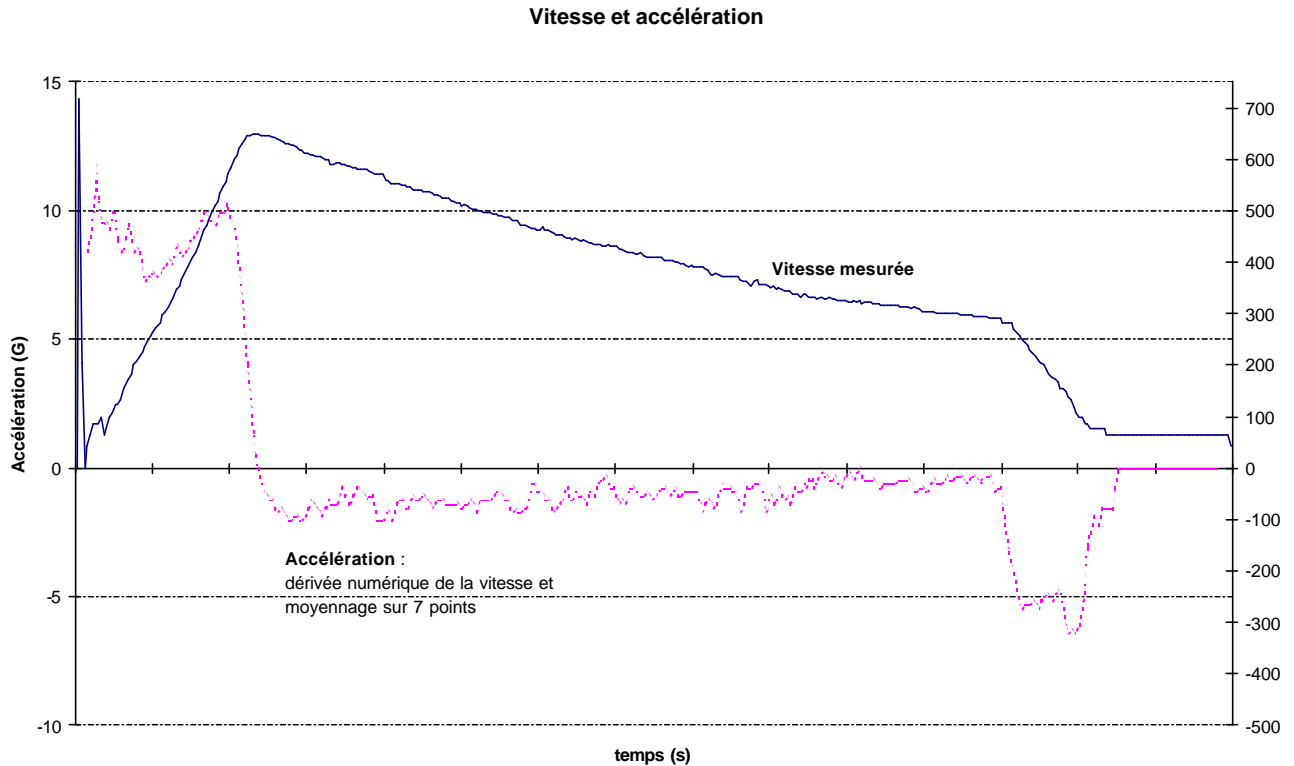
Durant la première phase, la poussée relativement constante du propulseur chamois fait croître la vitesse de façon linéaire. Après épuisement du combustible, la fusée est soumise au frottements de l'air qui la freine. Lorsque le microcontrôleur détecte un minimum de vitesse, l'ouverture du parachute est déclenchée, ce qui a pour conséquence (voulue) de réduire rapidement la vitesse de la fusée.

On remarque un pic de vitesse au début de la phase propulsée qui est vraisemblablement dû à la nature piézoélectrique du capteur de pression différentielle et au violent pic de poussée du propulseur lors de l'ignition.

La présence de rafales de vent et la faible vitesse en sortie de rampe, ont imposé le choix d'un angle de lancement faible ( $65^\circ$ ). Par conséquent, la vitesse de la fusée était supérieure à 100 km/h lors de l'ouverture du parachute. (surdimensionnez vos parachutes !)



## 6.2.5 Accélération



De la mesure de vitesse, on déduit l'accélération par dérivation. Cette opération amplifie les fréquences élevées, on applique alors un filtre passe-bas sous la forme d'un moyennage des valeurs calculées.

Malgré la présence de mesures de vitesse erronées, durant la phase propulsée, l'accélération suit relativement bien la courbe de poussée du propulseur chamois (cf. "les propulseurs des clubs scientifiques spatiaux" Ed. ANSTJ). L'accélération est comprise entre 7 et 10 G. Les résultats du logiciel Trajec s'étendaient de 6 à 8 G. On peut attribuer cette différence à un paramétrage approximatif de la simulation - la valeur du  $C_x$ , par exemple, était estimée "à vue de nez" -. De même, il est probable que deux propulseurs différents ne fournissent pas exactement la même poussée au cours du temps et on peut également mettre en doute la précision des mesures de vitesse et d'accélération qui en découle.

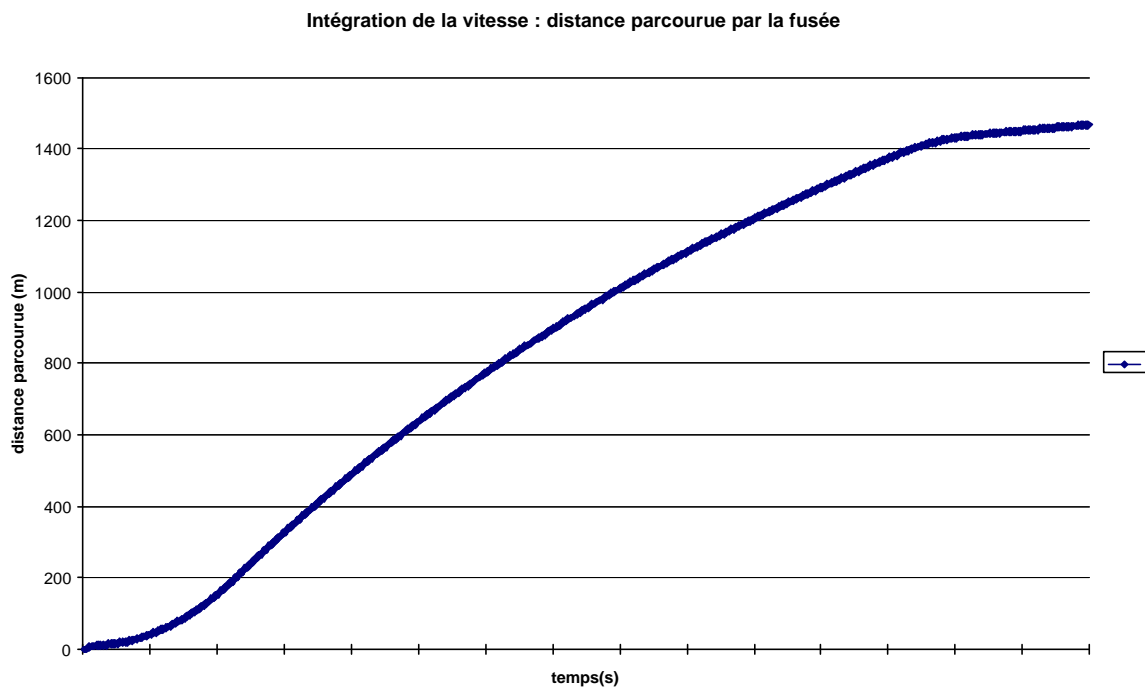
Lors du vol balistique, l'accélération (le freinage), due aux frottements de l'air et à la gravité terrestre est voisine de 1 à 2 G, décroît avec la vitesse.

L'ouverture du parachute cause une décélération brutale de 5 à 6.5 G pendant une seconde.

### 6.2.6 Distance parcourue par la fusée

En intégrant la vitesse au cours du temps, on obtient l'abscisse curviligne de la fusée, ce qui correspond à la distance parcourue par la fusée depuis la mise à feu.

On doit préciser que l'éventuelle présence d'un offset sur la mesure de vitesse génère une erreur sur ce graphe.



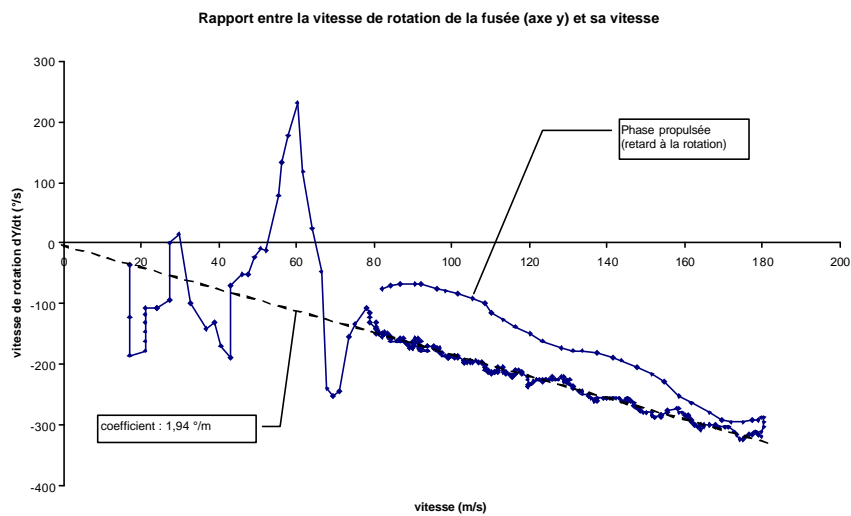
Pauline a donc réalisé un vol d'environ 1.4 km.

### 6.2.7 Relation entre la vitesse linéaire et la vitesse de rotation.

Voici un graphique comparant la mesure de la vitesse linéaire longitudinale de la fusée mesurée par le tube de Pitot, et la vitesse de rotation selon ce même axe longitudinal.

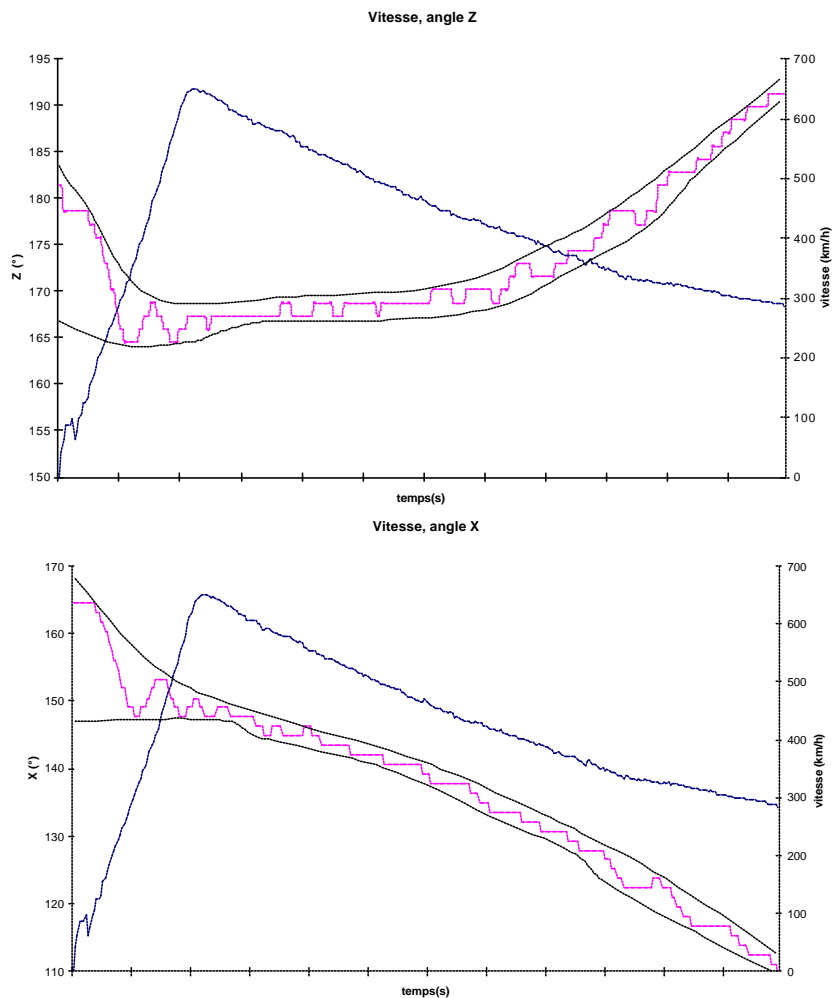
**Erreur! Liaison incorrecte.**

Avouez que la ressemblance est frappante! L'écart du début traduit l'inertie de la fusée. Par contre, à la décélération l'inertie de la fusée n'est plus apparente, et là, la vitesse de rotation colle à la vitesse longitudinale. La vitesse de rotation est proportionnelle à la vitesse longitudinale : la fusée pénètre dans l'air comme une vis. La linéarité n'est pas parfaite car la fusée oppose son inertie au changement de vitesse de rotation.



On peut calculer le coefficient de proportionnalité entre l'angle et la vitesse :  $1.94 \text{ }^{\circ}/\text{m}$  et le "pas" de la fusée : 190

## 6.2.8 Intégration des vitesses angulaires, stabilité de la fusée

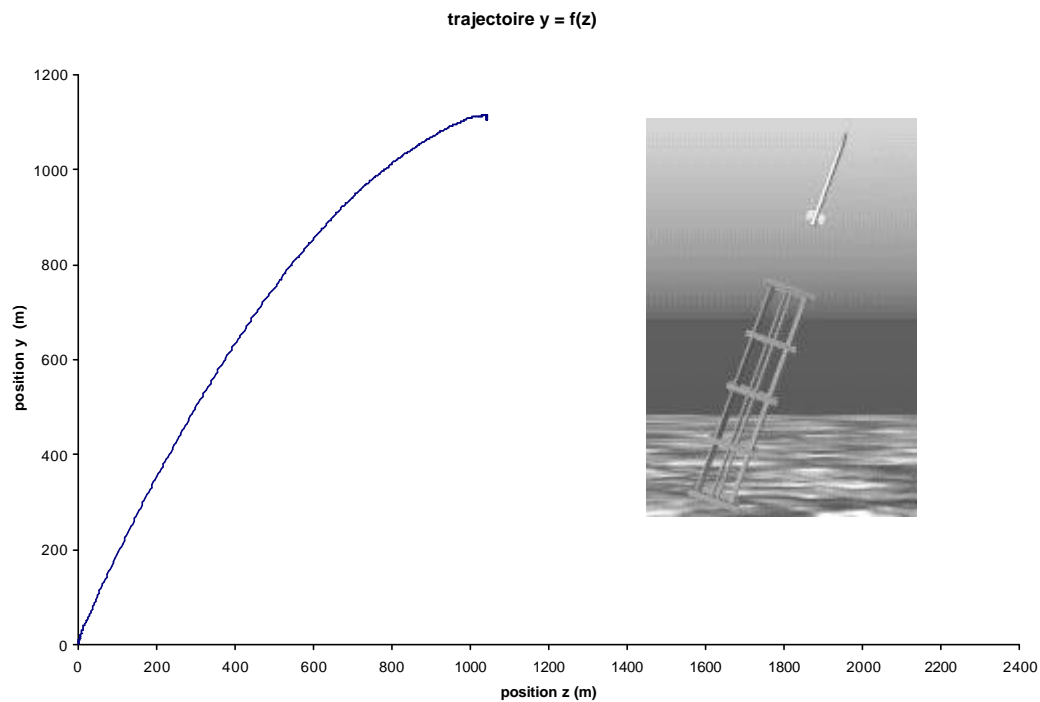


Ces graphes montrent l'angle parcouru par la fusée au cours du temps sur chacun des deux axes transversaux de la fusée. On obtient ces résultats en intégrant la vitesse angulaire. La courbe de vitesse permet de situer les différentes phases de vol.

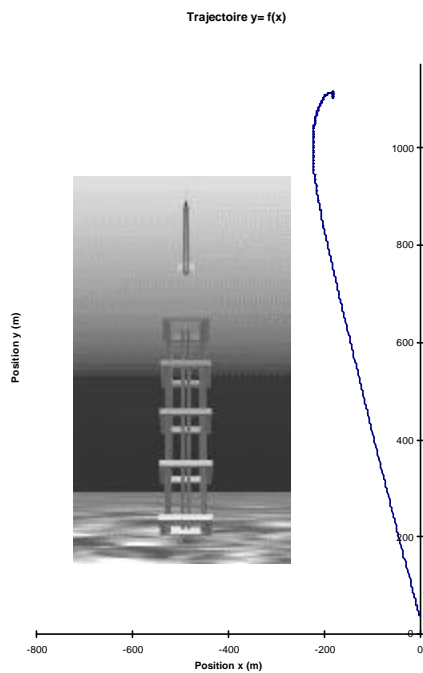
On remarque que la fusée est peu stable au début de la phase propulsée, c'est à dire lorsque la vitesse est faible. Ce fait est imputable soit au sous-dimensionnement de la taille des ailerons (Pauline était en limite de stabilité sous le logiciel Trajec), soit aux rafales de vent à basse altitude qui ont beaucoup perturbé les vols en ce vendredi après-midi.

Durant le vol balistique, les oscillations de la fusée ont une amplitude plus réduite.

## 6.2.9 Trajectoire et vent

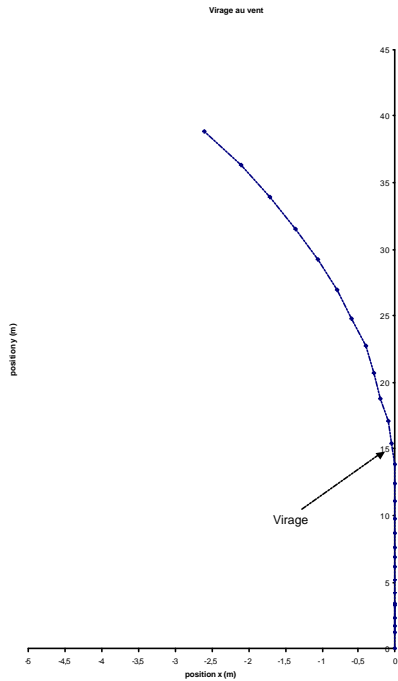


Ce graphe est la projection de la trajectoire dans le plan YZ. On retrouve la courbe "classique" de vol d'un engin propulsé.

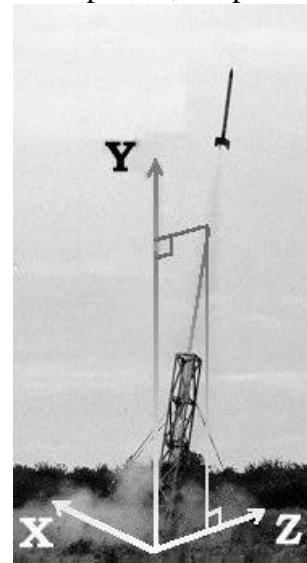


La courbe ci-contre montre la trajectoire que le public à pu voir, c'est à dire derrière la rampe, dans l'axe de lancement.

La fusée devrait théoriquement seulement se déplacer dans le plan YZ. Ici, on note une déviation importante due aux conditions météorologiques. (cf graphe suivant)

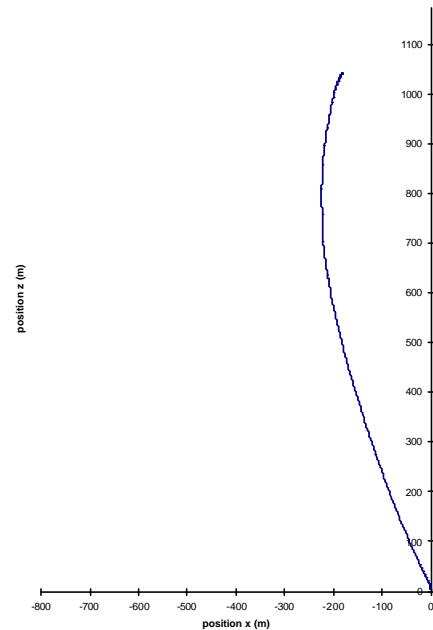


Le graphe ci-contre est un "zoom" du graphe précédent au niveau du début de trajectoire. Si la trajectoire semble débiter parfaitement verticalement, elle s'infléchit subitement lorsque la fusée atteint une altitude de 15 m. Cet effet était parfaitement visible des spectateurs et a affecté tous les vols du jours. Le vent particulièrement important contraignait les organisateurs à déclencher les lancements entre deux rafales. Ainsi si le départ était parfait, la première rafale contraignait la fusée à orienter son vol dans la direction de l'origine du vent. Ce virage est bien visible dans la démo du logiciel de suivi 3D.



Trajectoire Z = f(X)

Projection de la trajectoire sur le plan du sol :  
 On retrouve la déviation due au vent.  
 RAS, non ?



## 7. Bilan du projet.

### 7.1 L'expérience.

L'expérience peut être qualifiée de succès. En effet le vol a été nominal, les mesures effectuées ont été fiables et précises. De ce point de vue tout est parfait.

Puisque le but d'un tel document est de faire une critique constructive de l'expérience, relevons les points « défaillants »

1. La transmission radio était très perturbée. « Comme d'habitude ! » diraient les anciens. Mais il faudrait voir comment font les autres pour avoir une bonne transmission.
2. La temporisation de la minuterie parachute était calculée au plus juste. En effet ce n'est pas le microcontrôleur mais bien la minuterie qui a ordonné l'ouverture parachute quelques centaines de millisecondes avant la culmination réelle. Il faudrait donc être un « peu » plus large sur le calcul de la fenêtre de temporisation.
3. Les mesures Pitot étaient très satisfaisantes eut égard à la proximité de l'émetteur. Le concept utilisé pour le blindage et la transmission différentielle est à garder à l'esprit pour le futur.
4. L'alimentation était encore largement surdimensionnée du point de vue de la capacité. Chercher une solution conciliant faible capacité et fort courant.

### 7.2 Ce qu'il ne faudra pas refaire.

Voici une liste d'erreurs commises sur Pauline qui ne doivent pas être reprises pour les projets suivants :

**Un interrupteur général unique :** L'erreur est fondamentale, puisque nous avons donc effectué un essai télémètre avec un séquenceur actif. On aurait donc pu avoir une ouverture porte en rampe.

Deux interrupteurs sont donc nécessaires : Un pour la télémétrie et un second pour le séquenceur.

On peut même se poser la question d'un troisième interrupteur pour l'émetteur (indispensable pour les essais). Par contre il pourrait être court-circuité la veille du lancement. On n'est jamais à l'abri d'un oubli lors du basculement des interrupteurs lors de la mise en marche en rampe. Moins il y a d'interrupteur à basculer, mieux c'est : Deux semble être un bon compromis.

**Surestimer la capacité des piles :** On dispose de deux piles plates 4,5V formant un bloc de 9V, avec une capacité de 4500mAh. L'alimentation nécessite 1 A environ, et a besoin d'une autonomie de 45 minutes.

On pourrait donc faire le calcul rapide :  $4.5A.H/1A = 4.5H$ . Avec une pile plate on tire bien 1A, et même si elle ne tient pas 4 heures, elle tiendra bien au moins 2 heures. On pourrait se le dire... Mais on ne doit pas !!!

Sur de telles piles, le courant de décharge nominal est d'environ 400-500 mA. En tirant 1A l'autonomie passe à moins de 15 minutes !

Si on rajoute 2 piles de plus en parallèles, on tombe à 500 mA pour chaque pile. Dans ce cas le courant de décharge nominal est tout juste atteint. Le fonctionnement est alors normal. Le calcul  $4.5AH/0.5A$  est alors possible. On obtient des autonomies très importantes. Ceci est d'autant plus vrai si on utilise une alimentation à découpage qui va pouvoir utiliser à fond les piles.

**Utiliser une UVEPROM pour le programme embarqué :** En effet, il est extrêmement pénible d'avoir à effacer, puis griller, puis replacer sur support, une UVEPROM pour tester un bug de programme. L'opération est laborieuse, prend environ 10 minutes et n'est pas très bonne pour la durée de vie des connecteurs du support de l'UVEPROM.

Il serait plutôt préférable d'utiliser un moniteur gravé dans la zone UVEPROM d'un microcontrôleur à fenêtre.

Le programme serait alors téléchargé par une liaison série, depuis un PC vers une mémoire FLASH externe.

On pourrait alors faire rebooter le microcontrôleur sur la mémoire FLASH.

**Ne pas utiliser de connecteurs sur CI :** On a intérêt à poser directement les connecteurs sur le CI et à les disposer de façon à ce qu'ils puissent se visser directement sur le boîtier externe.

Sur Pauline, les connecteurs étaient indépendants des CIs et reliés par fils. Il y avait donc une liasse de fil qui allait du CI au boîtier. Inévitablement il y en a eu qui se sont amusés à créer des courts circuits. N'oubliez pas la loi de Murphy : « If something can go wrong, it will »

**Oublier de préparer l'après vol :** Encore une fois, avec Pauline on se retrouve à exploiter les données du vol, sans avoir de source unique permettant de savoir quelle donnée code quoi et avec quelle échelle. De même, la méthode pour extraire les données sauvegardées sur l'EEPROM I2C était connue de peu de personnes. Bref il faut penser à faire un document expliquant en détail ce que contient chaque octet de la trame et comment l'exploiter. De même pour les sauvegardes de données.

### 7.3 Ce que l'on pourra refaire.

**Utiliser une référence de tension** pour référencer le convertisseur A/N par rapport aux offsets ajoutés sur les tensions d'entrées.

Cela permet d'avoir des corrections d'offsets soft quasiment nulles.



**Utiliser l'I<sup>2</sup>C** : C'est pratique et ça diminue le nombre de connexion.

**Utiliser une Flash** : Pour stocker le programme c'est plus pratique qu'une UVROM. Par contre, il faut se procurer ou coder un programme de téléchargement du code.

**Utiliser des CMS** : Ce type de package, permet d'augmenter la densité de composants.

## 8. Remerciements

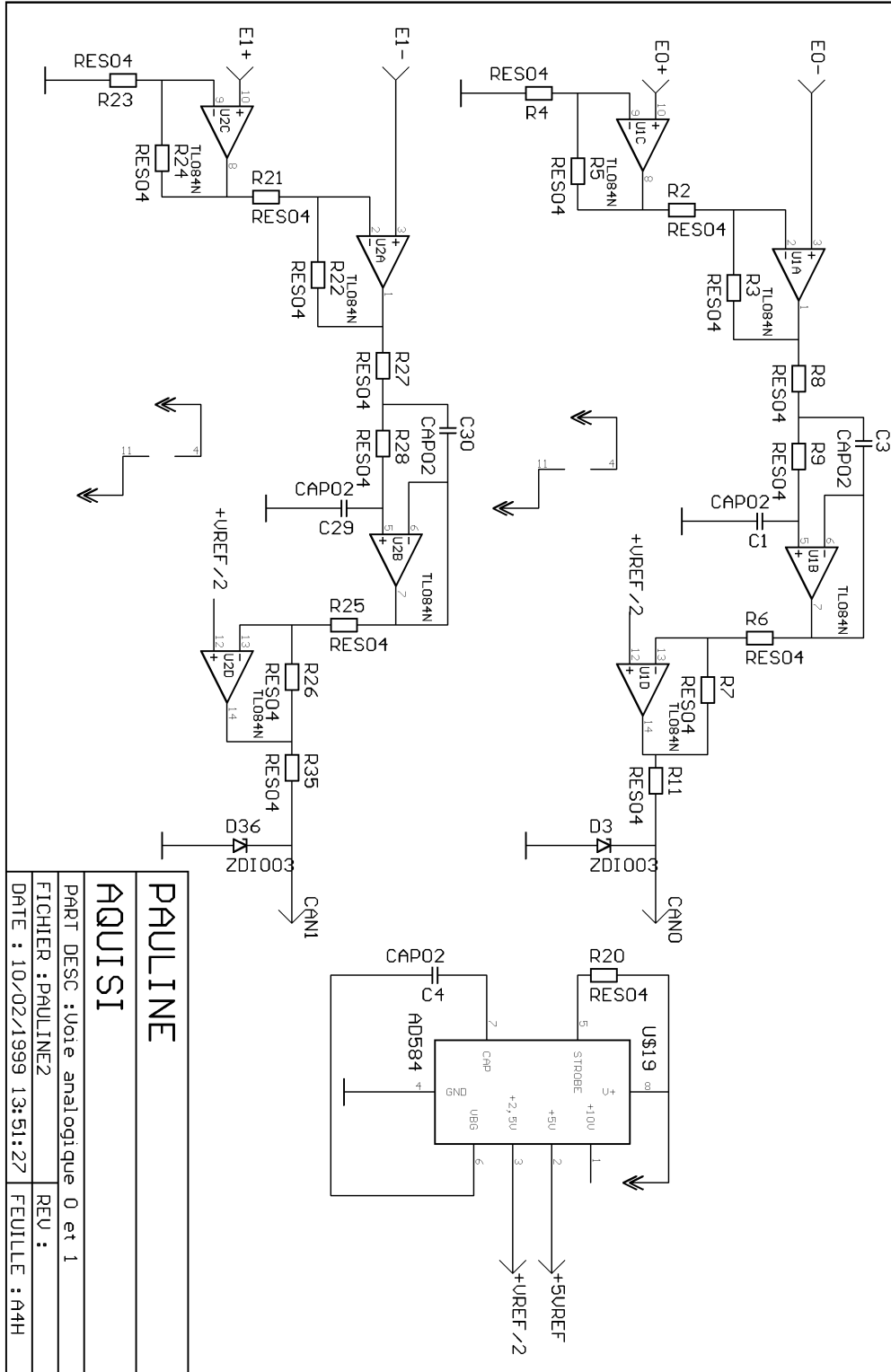
Nous tenons à remercier toutes les sociétés qui nous ont accordé du temps et du matériel pour nous permettre d'occuper nos temps libres passés ou à venir.

INSA	formation, locaux
SENSOREX	gyromètres
LENOIR MÉTALLERIE	
HONEYWELL	capteurs magnétoélectriques
RPCI	tirage de circuits imprimés
CLESSE	matériaux composites

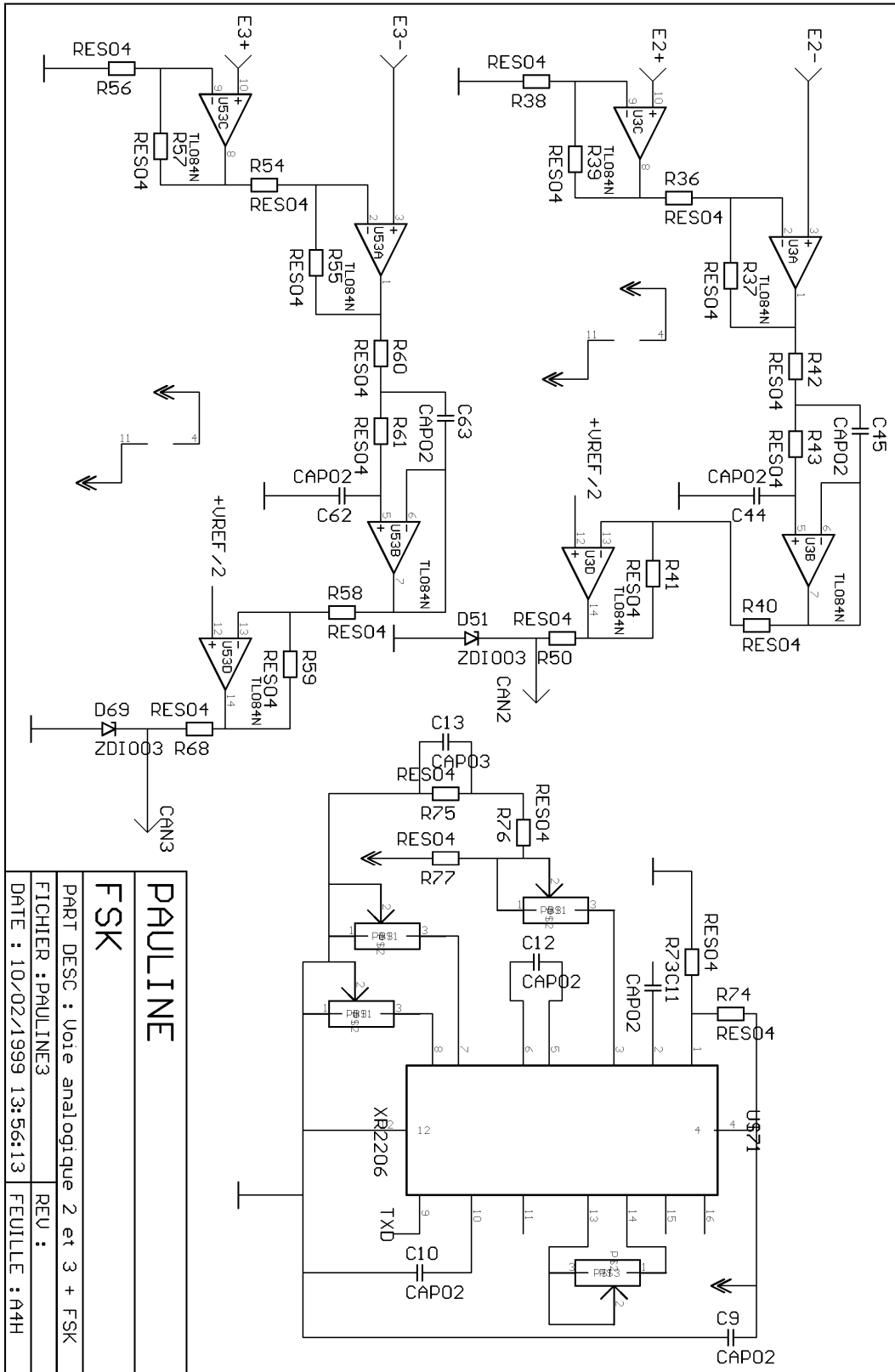


## 9. Annexes

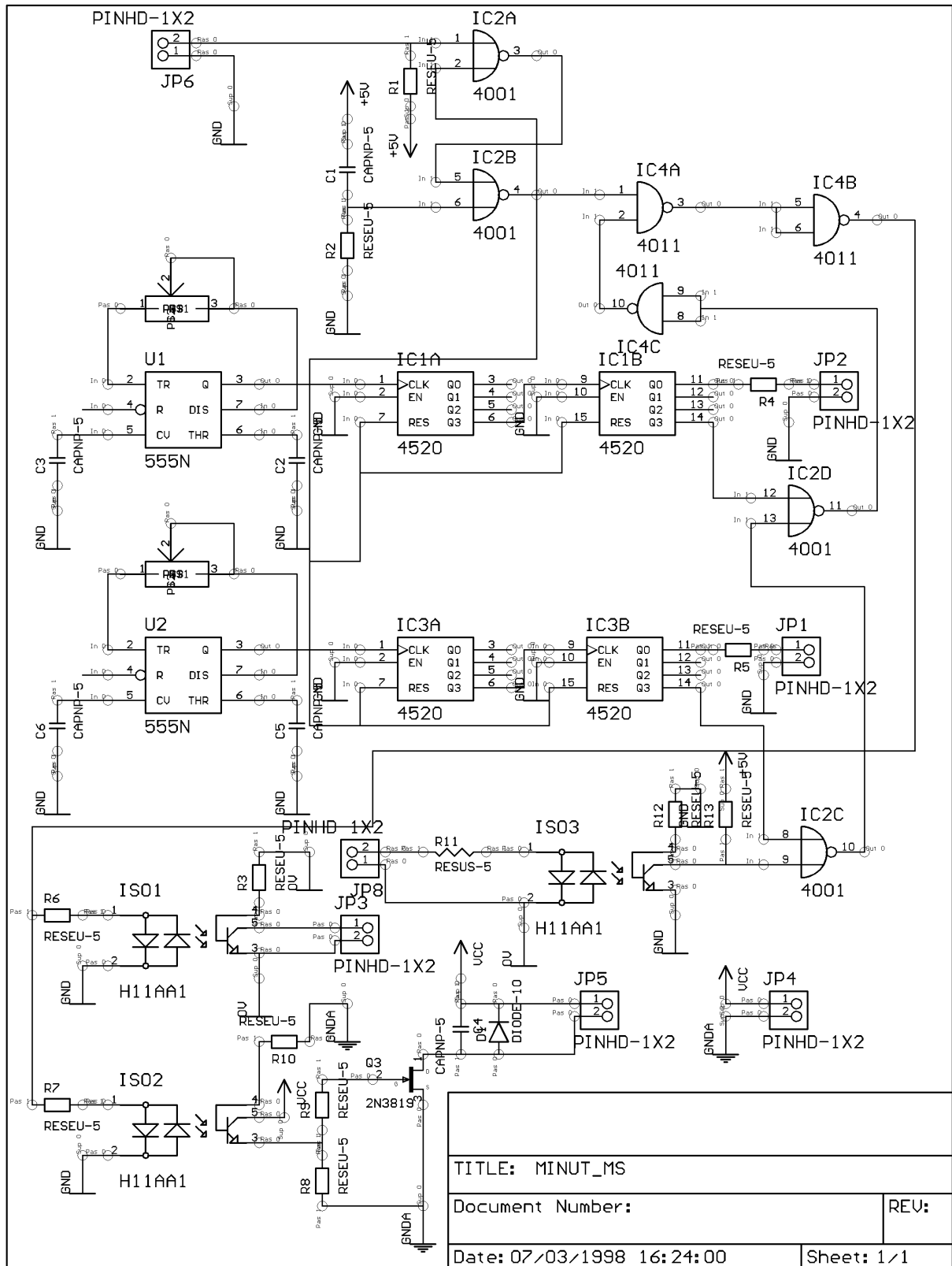
### 9.1 Chaîne analogique capteur



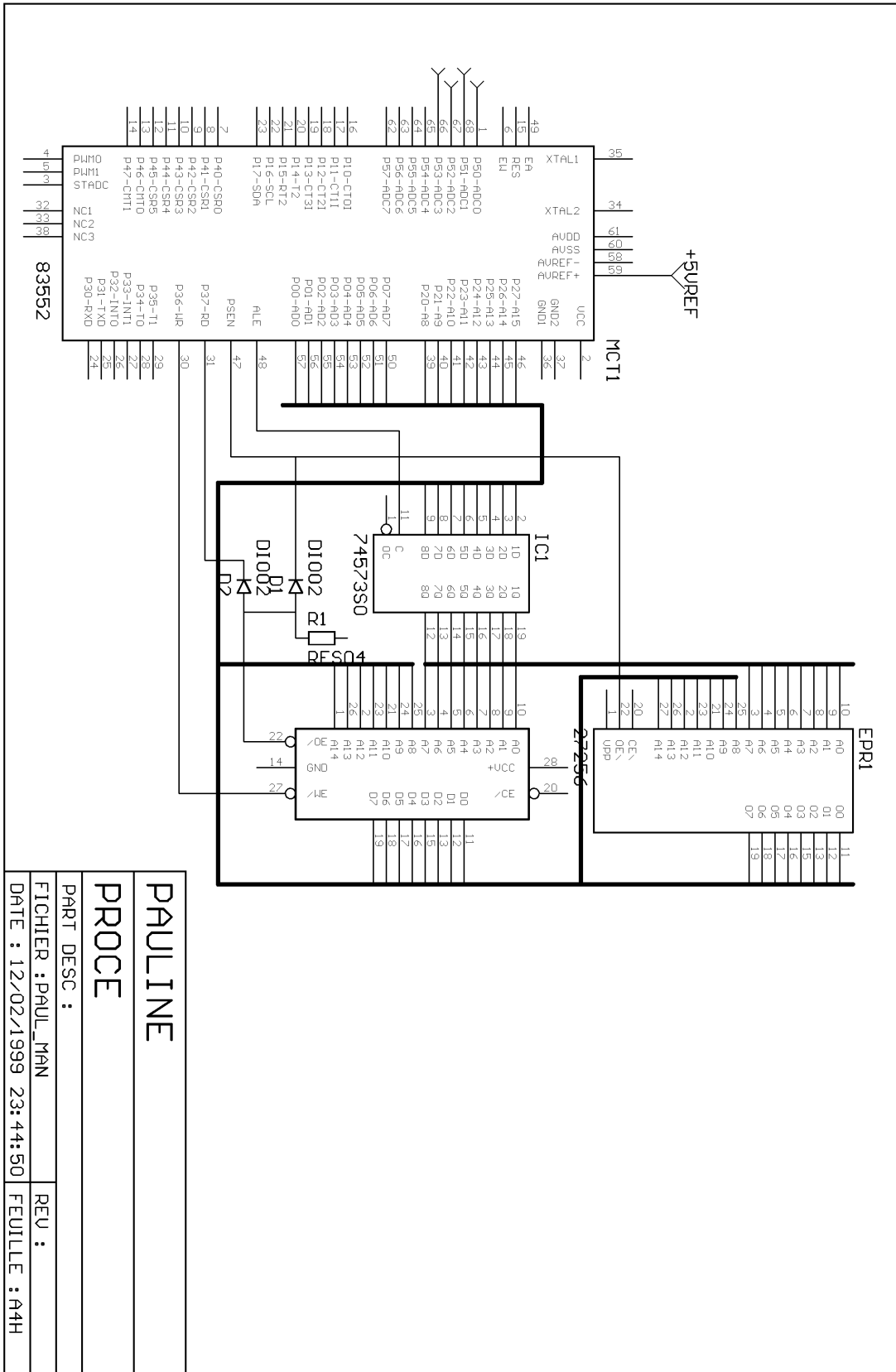
## 9.2 Modulation FSK



### 9.3 séquenceur



## 9.4 carte processeur



PAULINE  
 PROCE  
 PART DESC :  
 FICHER : PAUL\_MAN  
 DATE : 12/02/1999 23:44:50 FEUILLE : 44H