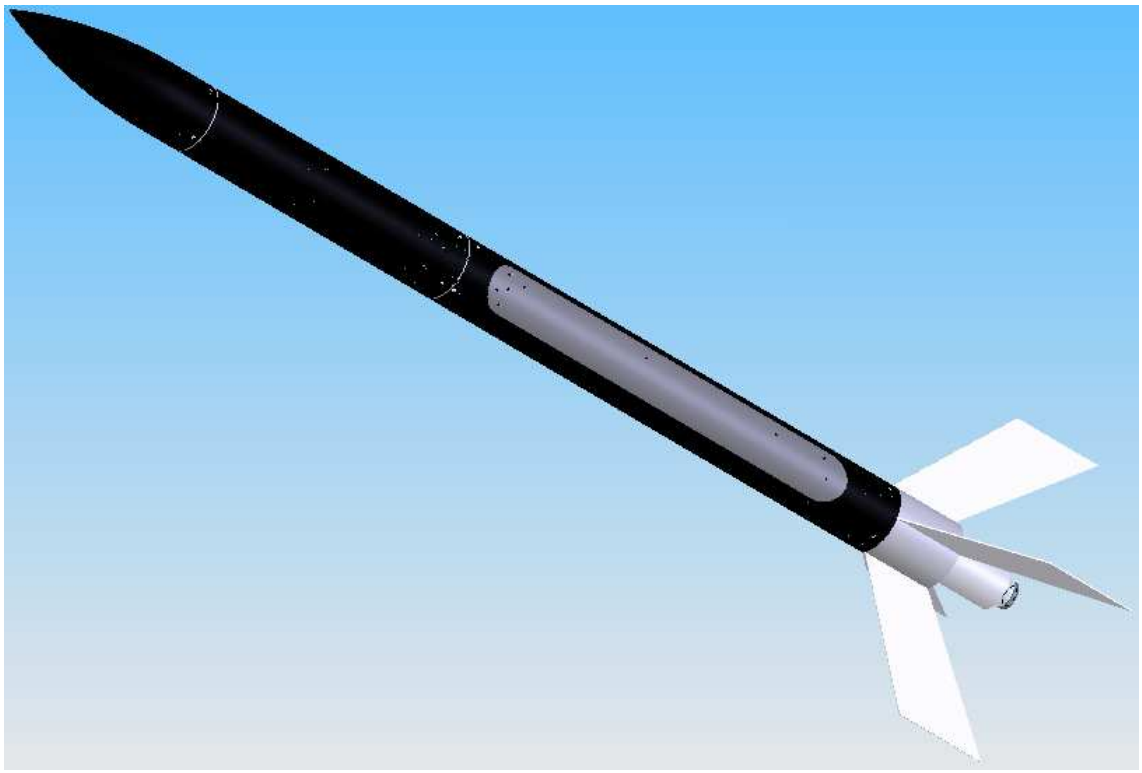
Projet de Fusée Expérimentale AMER

Année Scolaire 2005-2006



Fin de Rédaction le 15/11/06

Rédigé par :
Roux Félicien
Grandadam Rémi

Remerciements

Nous remercions vivement nos sponsors et partenaires qui nous ont cette année encore aidé à réaliser nos projets. Ceux-ci sont principalement :

- CNES : Propulseur, Déroulement de la campagne de lancement
- Planète Sciences : Organisation de la campagne de lancement, suivi.
- Union Thermique : Bruts d'Aluminium
- Zodiac : Parachute
- ESTACA : aide financière, facilités techniques et mise à disposition de machines et d'un local.
- FUSION ARGON : Soudure des Ailerons

N'oublions pas non plus Mr Mailly, le responsable de l'atelier de l'ESTACA, qui nous a aidé à faire certaines pièces et qui nous a donné accès au tour et à la fraiseuse.

Sommaire

I) Introduction 1
- 1) Présentation de l'ESTACA 1
- 2) Présentation de l'ESO 1
- 3) Présentation de l'équipe AMER 1

II) Analyse de l'Expérience 2
- 1) Mesure avec des Jauges de Contrainte 2
- 2) Pointage laser 2
- 3) Expérience Secondaire : Recherche de la fusée par ARVA 3

III) Moyens mis à disposition 3
- 1) Moyens Financiers 3
- 2) Moyens Techniques 3
- 3) Planning 4

IV) Conception et Dimensionnement Mécanique 5
- 1) Généralités : 5
- 2) Architecture de la fusée : 5
- 3) Dimensionnement 5

V) Conception de la Partie Electronique 7
- 1) Le Séquenceur 8
- 2) L'expérience 8

VI) Résultats de l'Expérience 9
- 1) Récupération grâce à l'émetteur ARVA 9
- 2) Mesure de Flèche durant le vol 9

Annexes 12

Estaca Space Odyssey

I) Introduction

Ce document regroupe toutes les données et informations qui nous ont permis de concevoir la fusée expérimentale AMER.

Le projet, qui a débuté en octobre 2005, a été lancé le 29 Juillet 2006, lors de la campagne de lancement nationale organisée par le CNES et Planète Sciences. Celle-ci c'est déroulée du 23 au 30 juillet 2006 sur le terrain militaire de La Courtine dans la Creuse (23).

1) Présentation de l'ESTACA

L'Ecole Supérieure des Techniques Aéronautiques et de Construction Automobile forme des ingénieurs en 5 ans. Elle recrute ses étudiants en grande majorité après un baccalauréat scientifique.

Les études, axées autour des mathématiques, de la mécanique et des sciences physiques, préparent aux métiers des transports grâce à quatre dominantes: Automobile, Ferroviaire, Aéronautique et Espace.

Elle délivre un diplôme reconnu par la commission des titres d'ingénieurs, dans quatre filières : Structure et Matériaux, Commande et Systèmes, Fluides et Energétique et Vibrations et Acoustique.

Chaque année, environ 160 élèves sont diplômés et sont embauchés par les grands noms de l'industrie des transports comme Renault, PSA, Airbus, Dassault Aviation ...

2) Présentation de l'ESO

L'ESO (ESTACA Space Odyssey) est une association loi 1901 dont l'objectif est de promouvoir l'activité aérospatiale au sein de l'école et du grand public. Elle y parvient en réalisant, notamment, la conception, la fabrication et le lancement de fusées ou de ballons expérimentaux, grâce à l'encadrement du CNES et de Planète Sciences. L'association a réalisé une trentaine de fusées et de ballons depuis sa fondation en 1991. Elle compte aujourd'hui une quarantaine de membres, tous étudiants à l'ESTACA, travaillant sur plusieurs projets.

Adresse postale : ESO
34 rue Victor Hugo
92300 Levallois Perret

Adresse électronique : eso@estaca.fr
Site Internet : www.eso.online.fr

Pour l'année 2005-2006, le Conseil d'Administration comprenait :

Président :	Pierre SERIN
Vice-président :	Thomas GARNIER
Trésorière :	Aude MOREL
Vice trésorier :	Yvan MADEC
Secrétaire :	Vincent GAUTHERON
Vice secrétaire :	Félicien ROUX

3) Présentation de l'équipe AMER

L'équipe de conception de la fusée AMER est composé de :

Félicien Roux	(3e année)	(chef de projet)
Rémi Grandadam	(1e année)	
Loïc Chappaz	(1e année)	
Marline Bagard	(1e année)	
Thomas Gorry	(3e année)	

Sur ces 5 étudiants de l'ESTACA, il n'y avait que des nouveaux membres de l'ESO, n'ayant jamais réalisé de Fusées auparavant. La réalisation de celle-ci remplit donc un objectif double : tout d'abord la formation de ceux-ci à la construction d'une fusée, ainsi que la réalisation d'une expérience originale, que l'on expliquera par la suite.

II) Analyse de l'Expérience

L'expérience choisit pour cette fusée est une mesure de flèche tout au long du vol de la fusée, expérience jamais réalisée à ce jour à bord d'une fusée expérimentale. En effet, celle-ci, en plus de son originalité, impliquait plus ou moins d'électronique à concevoir, selon les méthode de mesure expliquées par la suite. Cette flexibilité était nécessaire du à cet aspect de formation.

Deux méthodes de mesure ont donc été exposées :

1) Mesure avec des Jauges de Contrainte

La première méthode de mesure consiste à placer un profilé de PVC au milieu de la fusée, et d'y coller des jauges de contraintes afin de mesurer les contraintes à sa surface et donc de déterminer la flèche de a fusée.

Cependant le problème majeur de cette méthode, est que les jauges de contraintes auraient été collées à proximité de la fibre neutre de la fusée, ce qui aurait impliqué des mesures de petites déformations, augmentant notre risque d'erreur. L'idée a alors été émise de les coller sur la coque intérieure de la fusée, mais seulement s'il nous restait du temps en fin d'année pour le réaliser. En effet, nous avions alors opté pour la solution du laser expliquée plus loin. Cette méthode de mesure a donc été abandonnée faute de temps. De plus, les jauges de contraintes nécessitent une grande précision dans leur collage, ce qui aurait impliqué de le faire par des professionnels. Or, sur Galak, une Fusée Expérimentale réalisée la même année que la Fusée AMER, l'équipe avait besoin d'une jauge de contrainte pour leur expérience principale, mais l'équipe n'a jamais réussi à trouver un sponsor pour le leur coller. Du coup, nous ne pouvions pas non plus réaliser cette méthode d'acquisition.

2) Pointage laser

La deuxième méthode de mesure consiste à placer un laser au niveau de la plaque de poussée, et de filmer le déplacement du point rouge du laser projeté sur un écran.

Le problème le plus important était alors que ce système était fortement soumis aux vibrations hautes fréquences. Nous espérerions alors l'éliminer via un traitement du signal approprié, ainsi que grâce à des systèmes amortisseurs.

C'est donc cette méthode qui a été retenue.

3) Expérience Secondaire : Recherche de la fusée par ARVA

Suite au vol des fusées, afin de les retrouver au milieu de la forêt Creusoise, plusieurs systèmes ont été développés, comme entre autre un buzzer qui permet à la fusée d' « appeler » ses concepteurs lors de la phase de récupération, ou encore l'envoie des coordonnées GPS de la fusée via la télémesure. Sur notre projet, nous avons donc décidé d'expérimenter le système de recherche de victimes d'avalanche ARVA.

Pour cela, nous avons donc placé un tel émetteur dans la fusée, que l'on « entendra » grâce au récepteur prêté par un des membres du projet

III) Moyens mis à disposition

1) Moyens Financiers

Domaine	Type	Objet	Coût	Partenariat	Partenaire
Mécanique	Structure Alu	Bague Milieu/Supérieure	180,00 €	250,00 €	Union Thermique
		Bague Aileron/Poussée	100,00 €	100,00 €	Union Thermique
		Usinage	475,00 €	475,00 €	ESTACA
		Soudure Ailerons	80,00 €	80,00 €	Fusion Argon
	Structure	Fibre de Carbone 1,5m²	90,00 €		
		Fibre de Verre 1,5m²	22,50 €		
		Matériel	75,00 €		
	Propulsion	Propulseur Isard	800,00 €	800,00 €	CNES
	Système de récupération	Electro-Aimant	20,00 €	20,00 €	
		Parachute	150,00 €	150,00 €	Zodiac
		Emérillon	20,00 €		
		Cordes	10,00 €		
Electronique	Système	Minuterie	40,00 €	10,00 €	
		Carte Caméra	40,00 €	10,00 €	Microship
		ARVA	80 €		
		Laser	20 €		
		Caméra	80 €		
	Connectique	Intégration	50,00 €		
	Alimentation	Piles	50,00 €		
Campagne	Transport		100,00 €	100,00 €	ESO
	Hébergement	par personne	100,00 €	100,00 €	chaque membre
	Logistique		2 000,00 €	2 000,00 €	CNES - Planète Sciences
Total			4582,5 €	4095 €	
Reste à notre charge			487,5 €		
Budget			800,00 €		ESO

2) Moyens Techniques

L'ESO utilise le matériel de l'ESTACA, ce qui facilite la conception ainsi que la fabrication mécanique. Au niveau de l'électronique, l'école est pourvue de labo mais ne possède pas de moyen de gravure des cartes. L'ESO a donc son propre matériel, mais la mise en oeuvre pratique est moins aisée.

3) Planning

semaine	2005-2006	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Méca	Définition																																			
	Conception																																			
	Dimensionnement																																			
	Usinage																																			
	Tubes carbone																																			
	Ailerons																																			
	Intégration																																			
Elec	Définition Générale																																			
	Minuterie				C								R									Réalisation			T											
	Test Caméra																																			
	Carte Cam				C								R									Réalisation			T											
Divers	Vacances																																			
	Partiels et révisions																																			
	Départ 4°																																			

IV) Conception et Dimensionnement Mécanique

1) Généralités :

Diamètre Extérieur : 105 mm
Hauteur de la Fusée : 1898 mm avec les ailerons.
Masse : 6kg

La faible masse de la fusée ne nécessitait donc pas la capacité propulsive du moteur Chamois, nous avons donc opté pour le propulseur Isard, qui suffisait amplement.

2) Architecture de la fusée :

Le fait de placer un laser au milieu de la fusée a imposé une très grande contrainte : nous ne pouvions rien placer sur l'axe de révolution de la fusée, et même légèrement autour, afin de laisser la place au débattement du laser nécessaire à l'expérience.

Du coup, nous avons opté pour une séparation à porte via électroaimant, en plaçant le parachute sur un coté de la fusée. (un demi cylindre). De l'autre coté du coffre parachute sera alors placé les deux minuteries. Cet ensemble sera placé dans la partie basse de la fusée, séparé de la partie haute par une bague. En haut de la fusée sera placé un treillis carré afin de faire passer le faisceau lumineux en son milieu. C'est sur ce treillis que se trouvera la carte interrupteurs, les portes jack, les piles et autres cartes électroniques nécessaires à l'expérience. Sous l'ogive sera alors placée la caméra ainsi que l'ARVA ; la coiffe n'étant pas en fibre de carbone, elle ne créera donc pas une cage de Faraday. Quant au laser, il sera placé dans une toute petite bague vissée au centre de la plaque de poussée, disposant de 3 vis de réglages. Pour plus de détails, des mises en plan ont été placées en annexes.

3) Dimensionnement

Dimensionnement des treillis dans la partie haute de la fusée.

Concernant le dimensionnement du « cône de visibilité du laser », nous avons placé l'écran où se reflétera le laser à 1175 mm de la plaque de poussée ; du coup, au grand maximum, le laser peut se déporter de 2% selon le cahier des charges soit 23,5 mm. Par conséquent, l'écran devra faire 47 mm sur 47 mm. Mais afin de laisser la place pour le treillis, celui-ci fera 57 mm de coté.

Dimensionnement du parachute

La formule de la vitesse de descente d'un objet sous parachute est :

$$V_d = \sqrt{\frac{2 \times M \times g}{\rho_0 \times C_x \times S}}$$

Avec M la masse de la fusée, dans notre cas 6kg
S la surface du parachute à déterminer
$g = 9{,}81\ m/s^2$
$\rho_0 = 1{,}23\ kg/m^3$
$Cx = 1$

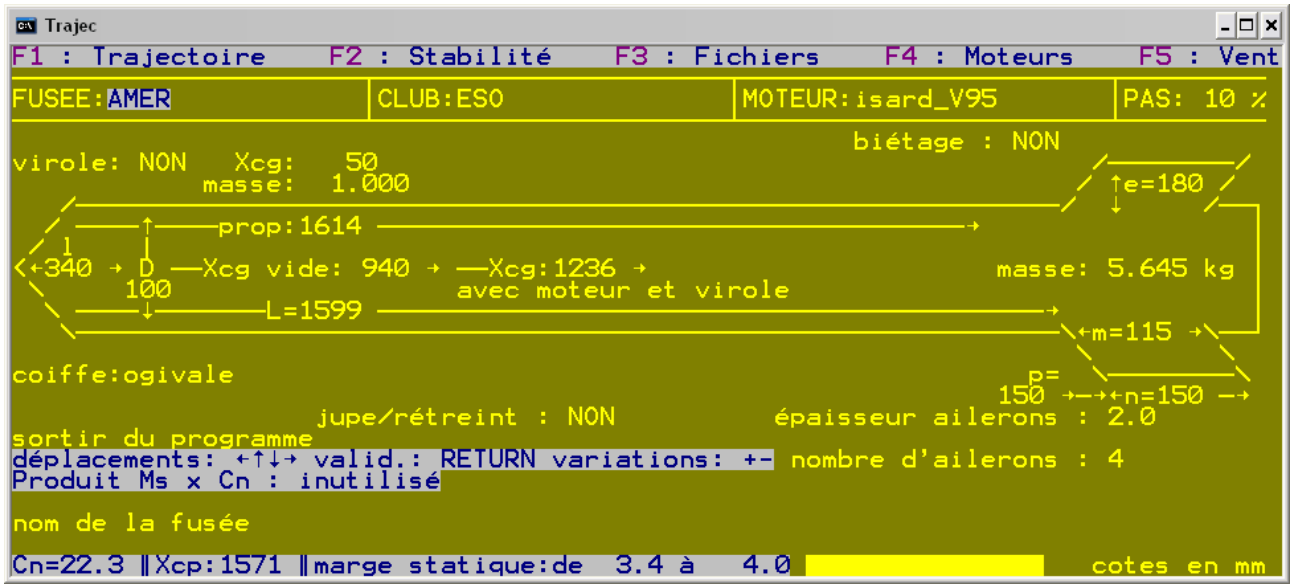
On en déduit donc que $S = 1{,}5\ m^2$

Dimensionnement de l'électroaimant

L'électroaimant, qui maintient la porte fermée, peut être soumis à plusieurs types de forces : traction et cisaillement. En traction, seules les forces du parachute et du ressort aidant la porte à s'ouvrir s'appliquent. C'est alors le ressort qui est dimensionné, en fonction de l'électroaimant, par des méthodes expérimentales. En cisaillement, c'est la torsion de la fusée au cours du vol qui peut faire que l'électroaimant s'ouvre à un moment non désiré. Pour cela, des fourchettes ont été placées en haut et en bas de la porte (avec en bas un système de pivot maintenant la porte en phase ascensionnelle, et réalisant une liaison pivot à l'apogée), afin de reprendre ces efforts de cisaillement. En effet, un électro-aimant ne supporte quasiment aucune charge de cisaillement.

Dimensionnement des ailerons

Grâce au logiciel donné par Planètes Sciences, nous dimensionnons les ailerons de la fusée afin que cette dernière soit stable. Nous rentrons alors les mensurations de la fusée dans ce logiciel :



Lorsque la stabilité est atteinte, (ce qui est le cas dans la capture d'écran, mais du à un bug du logiciel n'apparaît pas), nous pouvons alors récupérer la taille des ailerons afin de les découper.

Nota : Plusieurs combinaisons de dimensions d'ailerons conduisent à une fusée stable, ce qui permet également de jouer sur l'esthétisme.

V) Conception de la Partie Electronique

L'électronique se compose de deux grandes parties : le séquenceur et l'expérience.

Une carte "Interrupteurs" accessible permet l'alimentation des différents circuits électriques de la fusée. Cette carte est composée de 3 interrupteurs qui sont associés, respectivement de haut en bas, à la mise sous tension du laser et de l'Arva, à la carte commandant la caméra, et enfin au séquenceur.

Comme le cahier des charges l'impose, le séquenceur est électriquement séparé des autres cartes (sauf la masse qui est commune à toute la fusée). Alimenté en amont par deux piles 9V de type 6LR61 montées en série, toutes les cartes d'AMER nécessitent la présence de régulateurs de tension (LM7805) associés à des condensateurs ; cela permettant d'une part d'abaisser la tension à 5V (tension nominale des composants électroniques utilisés), et d'autre part de lisser et réguler la tension, en effaçant le bruit constitué par les hautes et basses fréquences parasites.

1) Le Séquenceur

Le séquenceur d'AMER est une minuterie numérique constituée d'une horloge (LM555) délivrant un signal carré de 13 Hz (ajustable par potentiomètre) à un compteur 8 bits (74HC40103). Après l'arrachage d'un Jack accroché à la rampe de lancement, une LED jaune de contrôle s'éteint sur la carte "Interrupteurs" ; parallèlement, le compteur s'active et s'incrémente d'une unité à chaque état haut fourni par l'horloge. A tout instant, il compare le résultat du comptage à un nombre binaire réalisé par un Switch ; après le calcul du temps T nécessaire à la fusée pour atteindre l'apogée (grâce au logiciel "Trajec"), on peut régler le Switch pour obtenir, au bout du temps T, une impulsion à la sortie du compteur. Cette impulsion représente la commande du Mosfet (transistor de puissance), permettant de laisser passer ou non un courant dans la ventouse électromagnétique. Malheureusement, cette impulsion de très courte durée ne permet pas à la ventouse électromagnétique de s'actionner correctement. Il faut donc rajouter (entre le compteur et le Mosfet) un composant électronique jouant un rôle de mémoire, pour maintenir l'état haut, en entrée du Mosfet, et par conséquent dans la ventouse). On intègre donc une bascule. Il en existe de plusieurs types, en fonction des applications souhaitées : RS, D, J-K. On a choisi une bascule J-K avec Set/Reset (74HCT74).

Pour récapituler, une fois le jack arraché, le compteur se met en route ; dès que le résultat du comptage est égal au nombre choisi via le Switch, une impulsion est envoyée à la bascule qui met et maintient sa sortie à l'état haut. Le Mosfet laisse alors passer le courant dans la ventouse qui libère la porte de la fusée. Une LED rouge de contrôle s'allume sur la carte "Interrupteurs" <jokemode>, permettant à une mouette d'éviter un gros risque </jokemode>.

Le principal inconvénient de ce type de minuterie est qu'il est nécessaire de bien dimensionner toutes les résistances et condensateurs influant sur la fréquence de sortie de l'horloge. Il s'agit d'une minuterie somme toute assez difficile à mettre en œuvre.

C'est la raison pour laquelle, parallèlement, on a décidé de réaliser une minuterie secondaire, cette fois-ci grâce à un PIC. Ce dernier exécute un programme qui, une fois le jack arraché, patiente la durée nécessaire à la fusée pour atteindre l'apogée, tout comme la minuterie numérique, puis met à l'état haut une de ses sorties pour saturer un Mosfet, relié à la ventouse de la même manière que la minuterie principale.

2) L'expérience

L'expérience étant de calculer la flèche qu'AMER subit au cours du vol, il a fallu choisir un moyen d'acquisition : ce fut une caméra filmant un laser.

La caméra, alimentée par ses propres piles, doit être commandée pré-décollage suivant une procédure simple mais bien définie.

Un jack est enlevé avant le lancement de la fusée ; l'information est envoyée directement à un PIC 12F683 qui va mettre à l'état haut une de ses sorties. Il s'agit de la commande de mise en route de la caméra. En effet, cette sortie est reliée à un transistor. L'émetteur et le collecteur du transistor sont reliés directement au bouton poussoir permettant d'allumer la caméra. Ainsi, quand le transistor devient passant, il effectue un contact au niveau du bouton poussoir. Le transistor substitue le bouton poussoir puisque ce dernier est inaccessible une fois la fusée intégrée. Comme il s'agit de la reproduction d'une impulsion due au bouton poussoir – et non d'un interrupteur – le transistor n'a besoin d'être passant que durant quelques millisecondes. C'est la raison pour laquelle le PIC remet à l'état bas la sortie en question, après un court laps de temps.

Après un délai de quelques secondes, laissant le temps à la caméra de s'allumer et de faire la mise au point, on procède à l'enregistrement. La commande est analogue : le PIC met à l'état haut une autre de ses sorties durant un court instant, ce qui sature un transistor qui devient passant et remplace l'action du bouton poussoir "Rec ●".

Quant au système de récupération (Arva) et au laser, cœur même de l'expérience, ils sont alimentés via une mini-carte servant à diminuer la tension d'arrivée de 18V, et en fournissant en sortie du 3V et du 1,5V respectivement au laser et à l'Arva.

VI) Résultats de l'Expérience

1) Récupération grâce à l'émetteur ARVA

Voici une photo de la fusée telle qu'on l'a retrouvée après le vol :

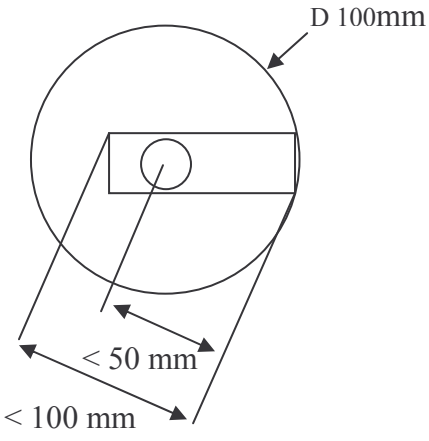
Comme on peut le voir, la hauteur maximale des buissons était alors de 30 cm ; pour le coup, nous avons eu la fusée en visu, avant de l'entendre grâce à l'émetteur ARVA, ce qui l'a rendu totalement inutile. Nous n'avons donc pas pu en déduire si ce système de récupération était d'un grand avantage par rapport à ceux qui existaient jusqu'alors. Cela dit, il est envisagé que l'année prochaine, une autre fusée en soit équipée, pour tester à nouveau ce système. On peut alors « espérer » que celle-ci atterrisse dans la forêt afin de tester une autre condition de récupération avec cet émetteur.

2) Mesure de Flèche durant le vol

La Caméra :

Cette expérience originale et jamais réalisée jusqu'alors représentait deux défis :

- Laisser la place au centre de la fusée afin que le faisceau laser ne puisse pas être perturbé.
- Faire tenir dans un diamètre de 100 mm une caméra de qualité moyenne.



Le premier point étant réglé par une intégration mécanique adéquate, le deuxième point s'est avéré nettement plus difficile à régler. En effet, il fallait que la caméra ait une diagonale inférieure au 100 mm intérieur de la coiffe, et que la distance entre le centre de l'objectif et le point le plus éloigné sur le même plan soit inférieure à 50 mm.

Une longue recherche sur Internet a alors été effectuée afin de trouver la caméra idéale, et nous nous sommes donc arrêté sur celle-ci : la *Mustsek DV 5200*

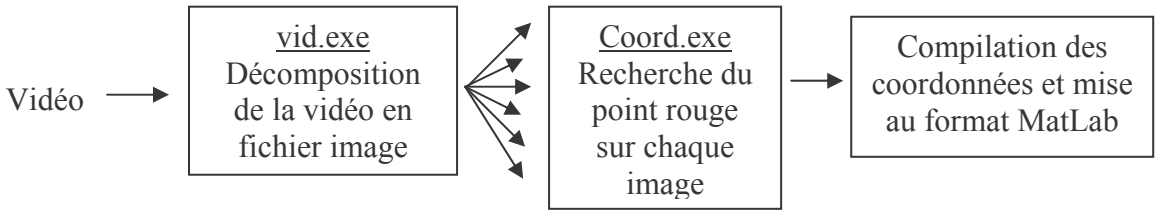
Le Traitement de l'Image

Suite au vol, nous devions donc recevoir la vidéo d'un point rouge ce baladant sur un fond noir. Fort intéressant certes, mais encore fallait-il l'interpréter. Nous avons donc envisagé de réaliser un programme informatique avec comme entrée la vidéo, et en sortie les coordonnées de chaque points rouges en fonction du temps (t, x, y) sous un format MatLab, afin de pouvoir réaliser par la suite un traitement du signal sur ce logiciel (enlever les hautes fréquences). Le code source de ce Programme est disponible en annexe.

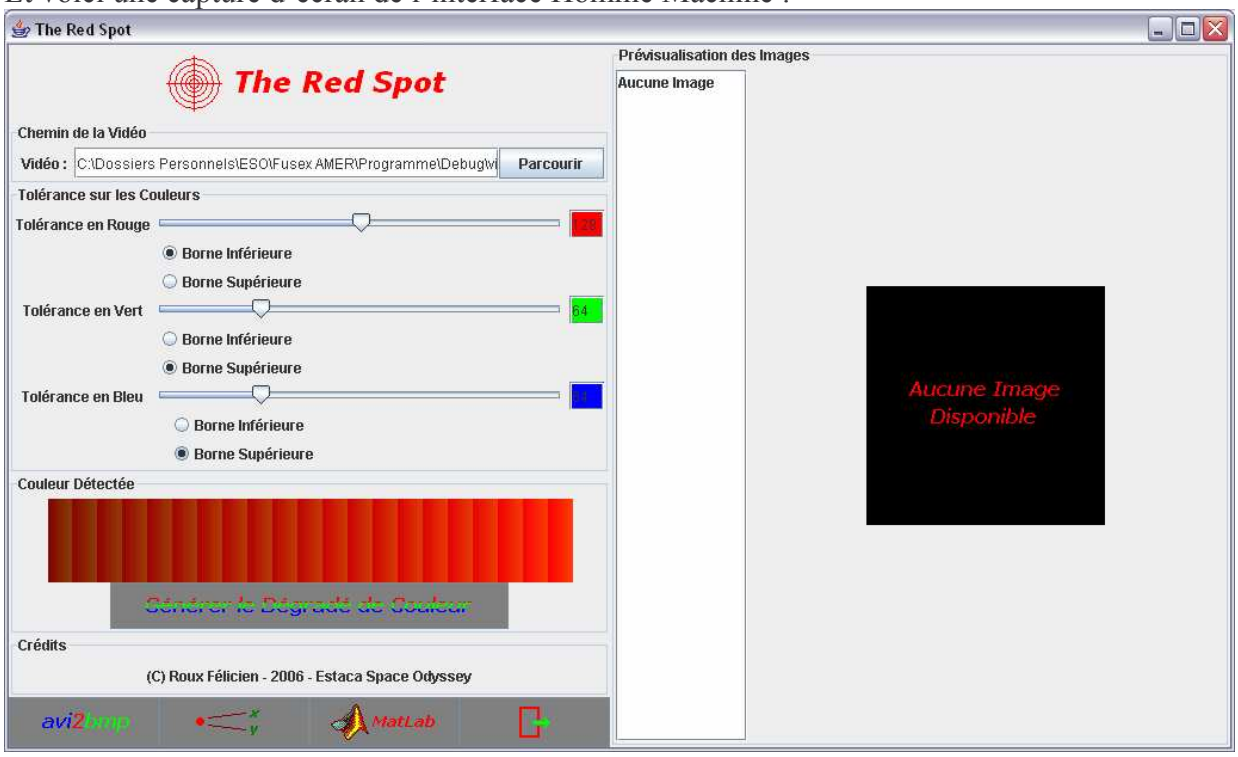
Or, deux langages de programmation sont enseignés à l'ESTACA, le Java et le C. Ce sont donc ces deux langages que nous avons utilisé pour réaliser ce dernier, en exploitant les qualités de chacun d'eux.

Du coup, toute l'interface graphique a été réalisée en Java, et tous les calculs de fond sur l'image ont été réalisés en C grâce à la libraire Open CV.

Voici le schéma de fonctionnement de ce programme :



Et voici une capture d'écran de l'interface Homme Machine :



Résultats

Malheureusement, suite au vol de la fusée, nous avons remarqué que la SD-Card était vide de tout fichier vidéo. On suppose que celle-ci a du trop s'enfoncer durant la phase d'accélération de la fusée, ce qui a donc coupé l'enregistrement. En effet, il existe un jeu permettant de retirer la carte, en effectuant un légère pression sur celle-ci. Ce n'est donc pas encore cette année qu'il y aura un résultat sur la flèche de la fusée durant le vol ; l'expérience aurait essentiellement permis de comparer les mesures réalisées lors des contrôles, par rapport à ce qui se passe réellement durant le vol.

L'étalonnage a été réalisé à La Courtine cet été, mais comme malheureusement aucune vidéo du vol n'a été récupérée, nous n'avons pas passé la vidéo d'étalonnage dans le programme dont on a parlé précédemment ; celui-ci demandant beaucoup de ressources système et n'aurait été d'aucune utilité.

Interprétations et Remarques

Une remarque est cependant à faire : nous avons pu nous apercevoir durant l'étalonnage de la fusée, que la méthode utilisée lors des qualifications est plus qu'aléatoire. En effet, lorsque nous rajoutions du poids au bout de la fusée pour simuler différentes flèches dynamiques, nous nous sommes aperçus que la flèche n'augmentait pas forcément dans le même sens. En effet, de part le fait que c'est une personne qui tient la barre de référence sur la bague de poussée de la fusée, celle-ci provoque des erreurs aléatoires selon l'endroit ou elle maintient la barre.

Aussi, lors de l'étalonnage, on s'est aperçu que la différence entre la flèche dynamique et statique est quasi nulle. En effet pour augmenter la flèche de manière significative (~2%), il a fallu rajouter des masses considérables au niveau de l'ogive, provoquant des forces qui ne sont jamais atteintes en vol (ex : force équivalente au poids même de la fusée, au niveau de l'ogive).

Annexes

Chronologie AMER 13

Code Source du Programme vid.exe réalisé en C 14

Code Source du Programme Coordlc.exe réalisé en C 15

Code Source de l'IHM réalisé avec le langage Java 25

Plans Mécaniques et Electroniques 47

Chronologie AMER

Temps :	Tâches :	Qui :
Au R3 :	Mise en place du jeu de piles neuves (9Vx4 et 1,5Vx2)	
	Vérification des systèmes + pisto coller :	Rémi
	- PIC Commande Caméra — Rilesans + carton pour bloquer	
	- Séparation et Minuterie — les piles.	Marline
	- Alimentation Laser / ARVA	Félicien
	Fixation et pliage du parachute	
	Montage de la fusée **(ATTENTION AU DB9 et STOKO)**	Félicien
	Vérification aspect extérieur (ailerons, vis, fixation propulseur, jacks)	et Rémi
H-90 min :	Montée sur l'aire de lancement	
	Outillage : Piles en spare / Récepteur ARVA	
	BTR (M3 en deux exemplaires, + M4)	
	Clefs Plate et à Pipe de 5,5	Marline
	Vis M3 (6,8,12 mm) et M4	Félicien
	Notre outils spécial en L / Rilesans (à la rigueur)	Rémi
	Ordinateur(s) Portable(s)	
H-70 min :	Arrivée en tente club	
H-60 min :	Descente sur rampe sans oublier le matériel suivant :	Marline
	- BTR (M3 en deux exemplaires, + M4)	Félicien
	- Les deux Jacks	Rémi
H-50 min :	Réglage en gisement de la rampe	Marline
	Réglage des patins (Ø 107 + 5)	Félicien
	Compatibilité rampe (mise en rampe)	Rémi
	Sortie de la fusée de la rampe	
	Mise en place du propulseur par l'artificier, retrait de la mousse.	Pyro
	Mise en rampe	
	Massages de Rémi par Mylène, et gros câlin à Félicien par Mylène	Mylène
H-32 min :	Fixation du cordon sur la rampe et mise en place des jacks et	Pyro
H-27 min :	Tout le monde regagne la tente pyro. Ne reste que Félicien.	
H-25 min :	Mise en place de la canne d'allumage	Pyro
H-20 min :	Orientation en site de la rampe	Pyro
H-15 min :	Mise sous tension des interrupteurs dans l'ordre (de bas en haut) :	Félicien
	-> Minuterie / Carte Caméra / Arva et Laser	
	Vérification des diodes (les vertes sont allumées ainsi que la jaune)	
	Arrachage du jack caméra	
	Vérification à l'écran de l'enregistrement de la cam.	
	Prière en direction de la Mecque.	
	Félicien rejoint le poste de lancement	Pyro
H-4 min :	Armement du propulseur par l'artificier	Lui
	Evacuation de la rampe	même
H-10 s :	Compte à rebours	Tous
H-0 :	3, 2, unité, Allumage Vulcain	Félicien

Code Source du Programme vid.exe réalisé en C

```
#include "C:\Dossiers Personnels\ESO\Fusex AMER\Programme\Librairie\highgui.h"
#include "C:\Dossiers Personnels\ESO\Fusex AMER\Programme\Librairie\cv.h"
#include <stdio.h>

void main(int nb, char **string){

	char cheminvideo[1000];
	char cheminrep[1000];
	char cheminimg[1000];
	char tmp[1000];

	typedef struct CvCapture CvCapture;

	FILE *Fichier=0;
	IplImage *img=NULL;
	CvCapture *video=NULL;
	int L;
	int i;

	//strcpy(cheminrep,"C:/Dossiers Personnels/ESO/Fusex AMER/Programme/Debug/");
	//strcpy(cheminvideo,"C:/Dossiers Personnels/ESO/Fusex AMER/Programme/Debug/vid.avi");
	strcpy(cheminvideo,string[1]);
	strcpy(cheminrep,string[2]);

	video = cvCaptureFromFile(cheminvideo);

	if(video==NULL){
		printf("erreur");
	}
	else
	{
		L = cvGetCaptureProperty(video,CV_CAP_PROP_FRAME_COUNT);
		for(i=0;i<L;i++)
		{
			cvSetCaptureProperty(video,CV_CAP_PROP_POS_FRAMES,i);

			img = cvQueryFrame(video);
			sprintf(cheminimg, "%sImg%d.bmp", cheminrep,i);
			cvSaveImage(cheminimg, img);
		}
		printf("Finit !\n");

		sprintf(tmp, "%snbimage.txt", cheminrep);
		Fichier = fopen(tmp,"w");

		if(Fichier==NULL)
			printf("Hic");
		else
		{
			fprintf(Fichier, "%d", L);
		}
		fclose(Fichier);
	}

	cvReleaseCapture(&video);
}
```

Code Source du Programme Coordlc.exe réalisé en C

```
#include "C:\Dossiers Personnels\ESO\Fusex AMER\Programme\Librairie\highgui.h"
#include "C:\Dossiers Personnels\ESO\Fusex AMER\Programme\Librairie\cv.h"
#include <stdio.h>

char rep[1000];
char nom[1000];

//LISTE CHAINE Rouge
struct Rouge{
	int x;
	int y;
	struct Rouge* suivant;
} rouge;

struct Rouge* CoorRouge = NULL;

InsertionEnFinRouge(int x, int y)
{
	struct Rouge* pcourant = CoorRouge;
	struct Rouge* NewCell = (struct Rouge*)malloc(sizeof(struct Rouge));
	NewCell->x=x;
	NewCell->y=y;
	NewCell->suivant=NULL;

	if(CoorRouge!=NULL)
	{
		while(pcourant->suivant!=NULL)
		{
			pcourant=pcourant->suivant;
		}
		pcourant->suivant=NewCell;
	}
	else
	{
		CoorRouge = NewCell;
	}

}

AffichageCoorRouge()
{
	struct Rouge* pcourant = CoorRouge;
	if(pcourant==NULL)
		printf("Liste Vide\n");
	else
	{
		while(pcourant!=NULL)
		{
			printf("Rouge : %d _ %d\n", pcourant->x, pcourant->y);
			pcourant=pcourant->suivant;
		}
	}
}


//LISTE CHAINE Image
```

```
struct Image{
	int x;
	int y;
	int R;
	int V;
	int B;
	struct Imahe* suivant;
}image;

struct Image* Image = NULL;

InsertionEnFinImage(int x, int y, int R, int V, int B)
{
	struct Image* pcourant = Image;
	struct Image* NewCell = (struct Image*)malloc(sizeof(struct Image));
	NewCell->x=x;
	NewCell->y=y;
	NewCell->R=R;
	NewCell->V=V;
	NewCell->B=B;
	NewCell->suivant=NULL;

	if(Image!=NULL)
	{
		while(pcourant->suivant!=NULL)
		{
			pcourant=pcourant->suivant;
		}
		pcourant->suivant=NewCell;
	}
	else
	{
		Image = NewCell;
	}

}

AffichageImage()
{
	struct  Image* pcourant = Image;
	if(pcourant==NULL)
		printf("Liste Vide\n");
	else
	{
		while(pcourant!=NULL)
		{
			printf("Coor : %d  %d // R=%d  V=%d  B=%d \n", pcourant->x, pcourant->y, pcourant->R,
pcourant->V, pcourant->B);
			pcourant=pcourant->suivant;
		}
	}
}

// Liste Chainée des coordonées des points
struct Coor{
	int nom;
	int x;
	int y;
```

```
	struct Coor* suivant;
} coor;

struct Coor* Coor = NULL;

InsertionEnFin(int nom[50],int x, int y)
{
	struct Coor* pcourant = Coor;
	struct Coor* NewCell = (struct Coor*)malloc(sizeof(struct Coor));
	NewCell->nom=nom;
	NewCell->x=x;
	NewCell->y=y;
	NewCell->suivant=NULL;

	if(Coor!=NULL)
	{
		while(pcourant->suivant!=NULL)
		{
			pcourant=pcourant->suivant;
		}
		pcourant->suivant=NewCell;
	}
	else
	{
		Coor = NewCell;
	}

}
InsertionEnTete(int nom[50],int x, int y)
{
	struct Coor* NewCell = (struct Coor*)malloc(sizeof(struct Coor));
	NewCell->suivant=Coor;
	NewCell->nom=nom;
	NewCell->x=x;
	NewCell->y=y;
	Coor=NewCell;
}

AffichageCoor()
{
	struct Coor* pcourant = Coor;
	if(pcourant==NULL)
		printf("Liste Vide\n");
	else
	{
		while(pcourant!=NULL)
		{
			printf("Coor : %d _ %d _ %d\n", pcourant->nom, pcourant->x, pcourant->y);
			pcourant=pcourant->suivant;
		}
	}
}

// Fonctions Complémentaire
CopieImageListeChaine()
{
	int i,j,l,R,V,B;
			// Problème pour l'image 5 et 15 de la Vidéo Test
```

```
	IplImage *img=NULL;
	CvSize imgSize;

	img = cvLoadImage( nom, -1);

	imgSize.width = img->width;
	imgSize.height = img->height;

	for(j=0;j<imgSize.height;j++)
	{
		for (i=0;i<imgSize.width;i++)
		{
			for(l=0;l<256;l++)
			{
				if((uchar*)(img->imageData + img->widthStep*j)[3*i]==(char)(l))
				{
					B=l;
				}
				if((uchar*)(img->imageData + img->widthStep*j)[3*i+1]==(char)(l))
				{
					V=l;
				}
				if((uchar*)(img->imageData + img->widthStep*j)[3*i+2]==(char)(l))
				{
					R=l;
				}
			}
			InsertionEnFinImage(i,j,R,V,B);
		}
	}
	cvReleaseImage(&img);
}

AjoutFichierCoordonnee(char *s, int X, int Y)
{
	if((X!=-1)&&(Y!=-1))
	{
		int nbLigne=0,i;
		char temp0[50];
		char temp1[50];
		char temp2[50];

		FILE *Fichier=0;

		char tmp[1000];
		strcpy(tmp,rep);
		sprintf(tmp,"%sCoordonnée.txt",rep);

		Fichier = fopen(tmp,"r");

		if(Fichier==NULL)
			printf("Bizare mais possible");
		else
		{
			fscanf(Fichier, "%d", &nbLigne);
			for(i=0;i<nbLigne;i++)
			{
				fscanf(Fichier, "%s", temp0);
```

```
                        fscanf(Fichier, "%s", temp1);
                        fscanf(Fichier, "%s", temp2);
                        InsertionEnFin(atoi(temp0),atoi(temp1), atoi(temp2));
                }
                fclose(Fichier);
        }


        // Partie Lecture faite, maintenant, écrivons
        Fichier = fopen(tmp,"w");
        if(Fichier==NULL)
                printf("Etrange ... et même impossible");
        else
        {
                struct Coor* pcourant = Coor;
                nbLigne++;
                fprintf(Fichier, "%d\n",nbLigne);
                while(pcourant!=NULL)
                {
                        fprintf(Fichier, "%d %d %d\n", pcourant->nom, pcourant->x, pcourant->y);
                        pcourant=pcourant->suivant;
                }
                fprintf(Fichier, "%d %d %d\n", s, X, Y);
                fclose(Fichier);
        }
    }
    else
    {
        if(CoorRouge==NULL)
        {
                char tmp[1000];
                FILE *Fichier=0;
                strcpy(tmp,rep);
                sprintf(tmp,"%sCoordonnée.txt",rep);
                Fichier = fopen(tmp,"w");
                fclose(Fichier);
        }
    }
}

RemplissageCoorRouge(int R,int G,int B, int RInf, int VInf, int BInf)
{
    if((RInf==1)&&(VInf==0)&&(BInf==0))
    {
        struct Image* pcourant = Image;
        while(pcourant!=NULL)
        {
                if((pcourant->R>=R)&&(pcourant->V<=G)&&(pcourant->B<=B))
                {
                        InsertionEnFinRouge(pcourant->x,pcourant->y);
                }
                pcourant=pcourant->suivant;
        }
    }
    if((RInf==1)&&(VInf==1)&&(BInf==0))
    {
        struct Image* pcourant = Image;
        while(pcourant!=NULL)
```

```
		{
			if((pcourant->R>=R)&&(pcourant->V>=G)&&(pcourant->B<=B))
			{
				InsertionEnFinRouge(pcourant->x,pcourant->y);
			}
			pcourant=pcourant->suivant;
		}
	}
	if((RInf==1)&&(VInf==1)&&(BInf==1))
	{
		struct Image* pcourant = Image;
		while(pcourant!=NULL)
		{
			if((pcourant->R>=R)&&(pcourant->V>=G)&&(pcourant->B>=B))
			{
				InsertionEnFinRouge(pcourant->x,pcourant->y);
			}
			pcourant=pcourant->suivant;
		}
	}
	if((RInf==0)&&(VInf==1)&&(BInf==0))
	{
		struct Image* pcourant = Image;
		while(pcourant!=NULL)
		{
			if((pcourant->R<=R)&&(pcourant->V>=G)&&(pcourant->B<=B))
			{
				InsertionEnFinRouge(pcourant->x,pcourant->y);
			}
			pcourant=pcourant->suivant;
		}
	}
	if((RInf==0)&&(VInf==0)&&(BInf==1))
	{
		struct Image* pcourant = Image;
		while(pcourant!=NULL)
		{
			if((pcourant->R<=R)&&(pcourant->V<=G)&&(pcourant->B>=B))
			{
				InsertionEnFinRouge(pcourant->x,pcourant->y);
			}
			pcourant=pcourant->suivant;
		}
	}
	if((RInf==1)&&(VInf==1)&&(BInf==0))
	{
		struct Image* pcourant = Image;
		while(pcourant!=NULL)
		{
			if((pcourant->R>=R)&&(pcourant->V>=G)&&(pcourant->B<=B))
			{
				InsertionEnFinRouge(pcourant->x,pcourant->y);
			}
			pcourant=pcourant->suivant;
		}
	}
	if((RInf==1)&&(VInf==0)&&(BInf==1))
	{
```

```
		struct Image* pcourant = Image;
		while(pcourant!=NULL)
		{
			if((pcourant->R>=R)&&(pcourant->V<=G)&&(pcourant->B>=B))
			{
				InsertionEnFinRouge(pcourant->x,pcourant->y);
			}
			pcourant=pcourant->suivant;
		}
	}
	if((RInf==0)&&(VInf==1)&&(BInf==1))
	{
		struct Image* pcourant = Image;
		while(pcourant!=NULL)
		{
			if((pcourant->R<=R)&&(pcourant->V>=G)&&(pcourant->B>=B))
			{
				InsertionEnFinRouge(pcourant->x,pcourant->y);
			}
			pcourant=pcourant->suivant;
		}
	}
}

PointMoyen(int *pX, int *pY, int *psX, int *psY)
{
	int n=0;
	struct Rouge* pcourant = CoorRouge;

	if(pcourant!=NULL)
	{
		*pX=0;
		*pY=0;
		do
		{
			*pX=*pX+pcourant->x;
			*pY=*pY+pcourant->y;
			n++;
			pcourant=pcourant->suivant;
		}while(pcourant!=NULL);

		*pX=*pX/n;
		*pY=*pY/n;
	}



	pcourant = CoorRouge;
	if(pcourant!=NULL)
	{
		*psX=0;
		*psY=0;
		do
		{
			*psX=*psX+(pcourant->x-*pX)*(pcourant->x-*pX);
			*psY=*psY+(pcourant->y-*pY)*(pcourant->y-*pY);
			pcourant=pcourant->suivant;
		}while(pcourant!=NULL);
```

```
                *psX=sqrt(*psX/n);
                *psY=sqrt(*psY/n);
        }


}


CarreAutourMoyenne(int X, int Y, int sX, int sY, int tmp, char rep[]){
        IplImage *img=NULL;
        CvSize imgSize;
        int demilarg;
        int demihauteur;
        int i,j,Ymin,Xmin,Ymax,Xmax=0;

        img = cvLoadImage( nom, -1);

        imgSize.width = img->width;
        imgSize.height = img->height;


        if(sX==0)
        {
                sX=1;
        }
        if(sY==0)
        {
                sY=1;
        }


        if((3*sX<imgSize.width)&&(3*sY<imgSize.height))
        {
                demilarg = sX*3;
                demihauteur = sY*3;
        }
        else
        {
                demilarg=imgSize.width/2;
                demihauteur=imgSize.height/2;
        }



        if(Y-demihauteur<0)
        {
                Ymin=0;
        }
        else
        {
                Ymin=Y-demihauteur;
        }
        if(Y+demihauteur+1>=imgSize.height)
        {
                Ymax=imgSize.height-1;
        }
        else
```

```
{
	Ymax=Y+demihauteur+1;
}
if(X-demilarg<0)
{
	Xmin=0;
}
else
{
	Xmin=X-demilarg;
}
if(X+demilarg+1>=imgSize.width)
{
	Xmax=imgSize.width-1;
}
else
{
	Xmax=X+demilarg+1;
}

for(i=Xmin;i<Xmax+1;i++)
{
	((uchar*)(img->imageData + img->widthStep*Ymin))[3*i]=(char) (255);
	((uchar*)(img->imageData + img->widthStep*Ymin))[3*i+1]=(char) (0);
	((uchar*)(img->imageData + img->widthStep*Ymin))[3*i+2]=(char) (0);

	((uchar*)(img->imageData + img->widthStep*(Ymin+1)))[3*i]=(char) (255);
	((uchar*)(img->imageData + img->widthStep*(Ymin+1)))[3*i+1]=(char) (0);
	((uchar*)(img->imageData + img->widthStep*(Ymin+1)))[3*i+2]=(char) (0);
}
for(i=Xmin;i<Xmax+1;i++)
{
	((uchar*)(img->imageData + img->widthStep*Ymax))[3*i]=(char) (255);
	((uchar*)(img->imageData + img->widthStep*Ymax))[3*i+1]=(char) (0);
	((uchar*)(img->imageData + img->widthStep*Ymax))[3*i+2]=(char) (0);

	((uchar*)(img->imageData + img->widthStep*(Ymax-1)))[3*i]=(char) (255);
	((uchar*)(img->imageData + img->widthStep*(Ymax-1)))[3*i+1]=(char) (0);
	((uchar*)(img->imageData + img->widthStep*(Ymax-1)))[3*i+2]=(char) (0);
}
for(j=Y-demihauteur;j<Y+demihauteur+1;j++)
{
	((uchar*)(img->imageData + img->widthStep*j))[3*Xmin]=(char) (255);
	((uchar*)(img->imageData + img->widthStep*j))[3*Xmin+1]=(char) (0);
	((uchar*)(img->imageData + img->widthStep*j))[3*Xmin+2]=(char) (0);

	((uchar*)(img->imageData + img->widthStep*j))[3*(Xmin-1)]=(char) (255);
	((uchar*)(img->imageData + img->widthStep*j))[3*(Xmin-1)+1]=(char) (0);
	((uchar*)(img->imageData + img->widthStep*j))[3*(Xmin-1)+2]=(char) (0);
}
for(j=Y-demihauteur;j<Y+demihauteur+1;j++)
{
	((uchar*)(img->imageData + img->widthStep*j))[3*Xmax]=(char) (255);
	((uchar*)(img->imageData + img->widthStep*j))[3*Xmax+1]=(char) (0);
	((uchar*)(img->imageData + img->widthStep*j))[3*Xmax+2]=(char) (0);

	((uchar*)(img->imageData + img->widthStep*j))[3*(Xmax-1)]=(char) (255);
	((uchar*)(img->imageData + img->widthStep*j))[3*(Xmax-1)+1]=(char) (0);
```

```
                ((uchar*)(img->imageData + img->widthStep*j))[3*(Xmax-1)+2]=(char) (0);
        }

        sprintf(nom, "%sImg%d.jpg", rep,tmp);

        cvSaveImage(nom, img);

        //Mis en commenaire car ça buggait mais pas déterminer d'où venait le problème
        //cvReleaseImage(&img);



}


void main(int nb, char **string)
{
        int X=-1,Y=-1,sX=-1,sY=-1,*psX,*psY,*pX, *pY,tmp=0, R=128, G=64, B=64,RInf=1,BInf=0,VInf=0;

        pX=&X;
        pY=&Y;
        psX=&sX;
        psY=&sY;



        tmp = atoi(string[1]);
        R = atoi(string[2]);
        G = atoi(string[3]);
        B = atoi(string[4]);
        strcpy(rep,string[5]);
        RInf = atoi(string[6]);
        VInf = atoi(string[7]);
        BInf = atoi(string[8]);


/*
        tmp = 5;
        R = 128;
        G = 64;
        B = 64;
        strcpy(rep,"C:/Dossiers Personnels/ESO/Fusex AMER/Programme/Debug/");
        RInf = 1;
        VInf = 0;
        BInf = 0;
*/


        sprintf(nom, "%sImg%d.bmp", rep,tmp);
        CopieImageListeChaine();
        RemplissageCoorRouge(R, G, B, RInf, VInf, BInf);
        PointMoyen(pX,pY,psX,psY);
        AjoutFichierCoordonnee(tmp, X,Y);

        //CarreAutourMoyenne(X,Y,sX,sY,tmp,rep);

}
```

Code Source de l'IHM réalisé avec le langage Java

- **Classe Fenêtre Principale :**

```
import java.io.*;
import java.util.regex.Pattern;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import java.awt.event.*;
import java.util.Vector;
import java.lang.Thread;

public class FenetrePrincipale extends JFrame implements
ActionListener,ListSelectionListener,javax.swing.event.ChangeListener{


    // Variable du Constructeur
    private String Commande;
    private String Chemin=new String("");
    private int Coor=0;
    private int nbImage;
    private int prev=0;
    private int avi2bmp=0;
    private int S=0;

    // Variable de l'extérieur
    private File _coor;
    private ImageIcon icon;
    private ImageIcon imgprev;
    private ImageIcon imagedet;

    // Différents Labels
    private JLabel _titre;
    private JLabel _tolRed;
    private JLabel _tolGreen;
    private JLabel _tolBlue;
    private JLabel _chemin;
    private JLabel _prevlabel;
    private JLabel _imglabel;
    private JLabel _credits;
    private JLabel _vide1;
    private JLabel _vide2;
    private JLabel _vide3;
    private JLabel _vide4;
    private JLabel _vide5;
    private JLabel _vide6;
    private JLabel _vide7;
    private JLabel _vide8;
    private JLabel _vide9;
    private JLabel _vide10;
    private JLabel _vide11;
    private JLabel _vide12;
    private JLabel _vide13;
    private JLabel _vide14;
    private JLabel _imgCoul;
```

```
// Liste des images
private JList _img;
private Vector _imgitems;
private JScrollPane _imgavecascenceur;

// Bouton radio
private JRadioButton _InfR;
private JRadioButton _SupR;
private ButtonGroup _Red;

private JRadioButton _InfB;
private JRadioButton _SupB;
private ButtonGroup _Blue;

private JRadioButton _InfV;
private JRadioButton _SupV;
private ButtonGroup _Green;

// Text Field
private JTextField _tfRed;
private JSlider _sRed;
private JTextField _tfGreen;
private JSlider _sGreen;
private JTextField _tfBlue;
private JSlider _sBlue;
private JTextField _tfchemin;

// Bouton d'action
private JButton _aviEnBmp;
private JButton _traitement;
private JButton _exportMatLab;
private JButton _exit;
private JButton _browse;
private JButton _genererPanel;

// Diférents Panels

private JPanel _paneTitre;
private JPanel _paneChVideo;
private JPanel _paneTolCouleur;
private JPanel _paneTolRouge1;
private JPanel _paneTolVert1;
private JPanel _paneTolBleu1;
private JPanel _paneTolRouge2;
private JPanel _paneTolVert2;
private JPanel _paneTolBleu2;
private JPanel _paneTolRouge3;
private JPanel _paneTolVert3;
private JPanel _paneTolBleu3;
private JPanel _paneCouleurDet;
private JPanel _paneGenerer;
private JPanel _paneCredit;
private JPanel _paneBoutton;
private JPanel _paneConfigPrev;

private JPanel _paneGlobal;
private JPanel _paneDroite;
private JPanel _paneHaut;
```

```
    private JPanel _paneBas;
    private JPanel _paneGauche;


    // Fenêtre MatLab
    private TransformationCoorMatLab tmp;

    public FenetrePrincipale() {
        super("The Red Spot");

        // Chargement de toutes les icones
        ImageIcon icon = new ImageIcon("Titre.gif");
        ImageIcon imgprev = new ImageIcon("AucuneImageDispo.gif");
        ImageIcon avi2bmp = new ImageIcon("avi2bmp.gif");
        ImageIcon matlab = new ImageIcon("matlab.gif");
        ImageIcon traitement = new ImageIcon("traitement.gif");
        ImageIcon sortie = new ImageIcon("sortie.gif");
        ImageIcon imagedet = new ImageIcon("imagedet.gif");
        ImageIcon generer = new ImageIcon("generer.gif");


        // Constrcution des Labels
        _titre = new JLabel(icon);
        _tolRed = new JLabel("Tolérance en Rouge",JLabel.CENTER);
        _tolRed.setPreferredSize(new Dimension(114,5));
        _tolGreen = new JLabel("Tolérance en Vert",JLabel.CENTER);
        _tolGreen.setPreferredSize(new Dimension(114,5));
        _tolBlue = new JLabel("Tolérance en Bleu",JLabel.CENTER);
        _tolBlue.setPreferredSize(new Dimension(114,5));
        _chemin = new JLabel("Vidéo :",JLabel.CENTER);
        _chemin.setPreferredSize(new Dimension(50,5));
        _prevlabel = new JLabel("Prévisualisation de l'Image sélectionnée :");
        _imglabel = new JLabel(imgprev);
        _imglabel.setPreferredSize(new Dimension(400,400));
        _credits = new JLabel("(C) Roux Félicien - 2006 - Estaca Space Odyssey");
        _vide1 = new JLabel("");
        _vide1.setPreferredSize(new Dimension(120,5));
        _vide2 = new JLabel("");
        _vide3 = new JLabel("");
        _vide3.setPreferredSize(new Dimension(120,5));
        _vide4 = new JLabel("");
        _vide5 = new JLabel("");
        _vide5.setPreferredSize(new Dimension(120,5));
        _vide6 = new JLabel("");
        _vide7 = new JLabel("");
        _vide7.setPreferredSize(new Dimension(120,5));
        _vide8 = new JLabel("");
        _vide9 = new JLabel("");
        _vide9.setPreferredSize(new Dimension(130,3));
        _vide10 = new JLabel("");
        _vide11 = new JLabel("");
        _vide11.setPreferredSize(new Dimension(130,3));
        _vide12 = new JLabel("");
        _vide13 = new JLabel("");
        _vide13.setPreferredSize(new Dimension(80,5));
        _vide14 = new JLabel("");
        _vide14.setPreferredSize(new Dimension(80,5));
        _imgCoul = new JLabel(imagedet);
```

```
// Construction des Boutons Radios
_InfR = new JRadioButton("Borne Inférieure", true);
_SupR = new JRadioButton("Borne Supérieure",false);
_Red = new ButtonGroup();
_Red.add(_InfR);
_Red.add(_SupR);

_InfV = new JRadioButton("Borne Inférieure", false);
_SupV = new JRadioButton("Borne Supérieure", true);
_Green = new ButtonGroup();
_Green.add(_InfV);
_Green.add(_SupV);

_InfB = new JRadioButton("Borne Inférieure", false);
_SupB = new JRadioButton("Borne Supérieure", true);
_Blue = new ButtonGroup();
_Blue.add(_InfB);
_Blue.add(_SupB);

// Construction de la Liste
Vector temp = new Vector();
temp.addElement("Aucune Image");
_img = new JList(temp);
_imgavecascenceur = new JScrollPane(_img);
_imgavecascenceur.setPreferredSize(new Dimension(110,250));

// Construction des Textes Fields
_tfRed = new JTextField("128");
_tfGreen = new JTextField("64");
_tfBlue = new JTextField("64");
_tfchemin = new JTextField("C:\\Dossiers Personnels\\ESO\\Fusex AMER\\Programme\\Debug\\vid.avi");
_tfchemin.setPreferredSize(new Dimension(290,21));
_tfRed.setBackground(Color.red);
_tfGreen.setBackground(Color.GREEN);
_tfBlue.setBackground(Color.BLUE);
_tfRed.setPreferredSize(new Dimension(29,5));
_tfGreen.setPreferredSize(new Dimension(29,5));
_tfBlue.setPreferredSize(new Dimension(29,5));

//Constuction des Sliders
_sRed = new JSlider(0,255,128);
_sGreen = new JSlider(0,255,64);
_sBlue = new JSlider(0,255,64);

// Construction des Boutons
_aviEnBmp = new JButton(avi2bmp);
_aviEnBmp.setBackground(Color.GRAY);
_traitement = new JButton(traitement);
_traitement.setBackground(Color.GRAY);
_exportMatLab = new JButton(matlab);
_exportMatLab.setBackground(Color.GRAY);
_exit = new JButton(sortie);
_exit.setBackground(Color.GRAY);
_browse = new JButton("Parcourir");
_genererPanel = new JButton(generer);
_genererPanel.setBackground(Color.GRAY);
```

```
// Construction des Panneaux
_paneTitre = new JPanel();
_paneChVideo = new JPanel(new BorderLayout());
_paneChVideo.setBorder(BorderFactory.createTitledBorder("Chemin de la Vidéo"));
_paneTolCouleur = new JPanel(new GridLayout(9,1));
_paneTolCouleur.setBorder(BorderFactory.createTitledBorder("Tolérance sur les Couleurs"));
_paneTolRouge1 = new JPanel(new BorderLayout());
_paneTolVert1 = new JPanel(new BorderLayout());
_paneTolBleu1 = new JPanel(new BorderLayout());
_paneTolRouge2 = new JPanel(new BorderLayout());
_paneTolVert2 = new JPanel(new BorderLayout());
_paneTolBleu2 = new JPanel(new BorderLayout());
_paneTolRouge3 = new JPanel(new BorderLayout());
_paneTolVert3 = new JPanel(new BorderLayout());
_paneTolBleu3 = new JPanel(new BorderLayout());
_paneCouleurDet = new JPanel(new BorderLayout());
_paneCouleurDet.setBorder(BorderFactory.createTitledBorder("Couleur Détectée"));
_paneGenerer = new JPanel(new BorderLayout());
_paneCredit = new JPanel();
_paneCredit.setBorder(BorderFactory.createTitledBorder("Crédits"));
_paneBoutton = new JPanel(new GridLayout(1,4));
_paneConfigPrev = new JPanel(new BorderLayout());

_paneGlobal = new JPanel(new BorderLayout());
_paneDroite = new JPanel(new BorderLayout());
_paneDroite.setBorder(BorderFactory.createTitledBorder("Prévisualisation des Images"));
_paneHaut = new JPanel(new BorderLayout());
_paneBas = new JPanel(new BorderLayout());
_paneGauche = new JPanel(new BorderLayout());

// Ajout des Items dans les Panneaux
_paneTitre.add(_titre);

_paneChVideo.add(_chemin,BorderLayout.WEST);
_paneChVideo.add(_tfchemin,BorderLayout.CENTER);
_paneChVideo.add(_browse,BorderLayout.EAST);

_paneTolCouleur.add(_paneTolRouge1);
_paneTolCouleur.add(_paneTolRouge2);
_paneTolCouleur.add(_paneTolRouge3);
_paneTolCouleur.add(_paneTolVert1);
_paneTolCouleur.add(_paneTolVert2);
_paneTolCouleur.add(_paneTolVert3);
_paneTolCouleur.add(_paneTolBleu1);
_paneTolCouleur.add(_paneTolBleu2);
_paneTolCouleur.add(_paneTolBleu3);

_paneTolRouge1.add(_tolRed,BorderLayout.WEST);
_paneTolRouge1.add(_sRed,BorderLayout.CENTER);
_paneTolRouge1.add(_tfRed,BorderLayout.EAST);
_paneTolRouge2.add(_vide1,BorderLayout.WEST);
_paneTolRouge2.add(_InfR,BorderLayout.CENTER);
_paneTolRouge2.add(_vide2,BorderLayout.EAST);
_paneTolRouge3.add(_vide3,BorderLayout.WEST);
_paneTolRouge3.add(_SupR,BorderLayout.CENTER);
_paneTolRouge3.add(_vide4,BorderLayout.EAST);
```

```
_paneTolVert1.add(_tolGreen,BorderLayout.WEST);
_paneTolVert1.add(_sGreen,BorderLayout.CENTER);
_paneTolVert1.add(_tfGreen,BorderLayout.EAST);
_paneTolVert2.add(_vide5,BorderLayout.WEST);
_paneTolVert2.add(_InfV,BorderLayout.CENTER);
_paneTolVert2.add(_vide6,BorderLayout.EAST);
_paneTolVert3.add(_vide7,BorderLayout.WEST);
_paneTolVert3.add(_SupV,BorderLayout.CENTER);
_paneTolVert3.add(_vide8,BorderLayout.EAST);

_paneTolBleu1.add(_tolBlue,BorderLayout.WEST);
_paneTolBleu1.add(_sBlue,BorderLayout.CENTER);
_paneTolBleu1.add(_tfBlue,BorderLayout.EAST);
_paneTolBleu2.add(_vide9,BorderLayout.WEST);
_paneTolBleu2.add(_InfB,BorderLayout.CENTER);
_paneTolBleu2.add(_vide10,BorderLayout.EAST);
_paneTolBleu3.add(_vide11,BorderLayout.WEST);
_paneTolBleu3.add(_SupB,BorderLayout.CENTER);
_paneTolBleu3.add(_vide12,BorderLayout.EAST);

_paneGenerer.add(_vide13,BorderLayout.WEST);
_paneGenerer.add(_genererPanel,BorderLayout.CENTER);
_paneGenerer.add(_vide14,BorderLayout.EAST);

_paneCouleurDet.add(_imgCoul,BorderLayout.NORTH);
_paneCouleurDet.add(_paneGenerer,BorderLayout.CENTER);

_paneCredit.add(_credits);

_paneBoutton.add(_aviEnBmp);
_paneBoutton.add(_traitement);
_paneBoutton.add(_exportMatLab);
_paneBoutton.add(_exit);

//_paneConfig

_paneDroite.add(_imgavecascenceur, BorderLayout.WEST);
_paneDroite.add(_imglabel,BorderLayout.CENTER);
_paneDroite.add(_paneConfigPrev,BorderLayout.SOUTH);

// Ajout des Panneaux en Global
_paneHaut.add(_paneTitre,BorderLayout.NORTH);
_paneHaut.add(_paneChVideo,BorderLayout.CENTER);
_paneHaut.add(_paneTolCouleur,BorderLayout.SOUTH);

_paneBas.add(_paneCouleurDet,BorderLayout.NORTH);
_paneBas.add(_paneCredit, BorderLayout.CENTER);
_paneBas.add(_paneBoutton,BorderLayout.SOUTH);

_paneGauche.add(_paneHaut, BorderLayout.NORTH);
_paneGauche.add(_paneBas, BorderLayout.SOUTH);

_paneGlobal.add(_paneGauche,BorderLayout.WEST);
_paneGlobal.add(_paneDroite,BorderLayout.EAST);



// Listners
```

```
        _aviEnBmp.addActionListener(this);
        _traitement.addActionListener(this);
        _exportMatLab.addActionListener(this);
        _exit.addActionListener(this);
        _genererPanel.addActionListener(this);
        _browse.addActionListener(this);

        _img.addListSelectionListener(this);
        _tfRed.addActionListener(this);
        _tfGreen.addActionListener(this);
        _tfBlue.addActionListener(this);

        _InfR.addActionListener(this);
        _InfV.addActionListener(this);
        _InfB.addActionListener(this);
        _SupR.addActionListener(this);
        _SupV.addActionListener(this);
        _SupB.addActionListener(this);

        _sRed.addChangeListener(this);
        _sGreen.addChangeListener(this);
        _sBlue.addChangeListener(this);




        this.setContentPane(_paneGlobal);
    }

    /**ATTRIBUTS**/
    public String getChemin()
    {
        return Chemin;
    }
    public void setCoor(int nv)
    {
        Coor=nv;
    }

    public void actionPerformed(ActionEvent evt)
    {
        this.remplissageCommande();
        //
        String tolRed = Integer.toString(_sRed.getValue());
        String tolGreen = Integer.toString(_sGreen.getValue());
        String tolBlue = Integer.toString(_sBlue.getValue());
        int RInf;
        int VInf;
        int BInf;

        if(_InfR.isSelected())
        {
                RInf=1;
        }
        else
        {
                RInf=0;
```

```
        }
        if(_InfB.isSelected())
        {
                BInf=1;
        }
        else
        {
                BInf=0;
        }
        if(_InfV.isSelected())
        {
                VInf=1;
        }
        else
        {
                VInf=0;
        }
        //

        if(evt.getSource()==_aviEnBmp)
        {
                File fr1 = new File(Commande);
                if(fr1.canRead()==true)
                {


                        _coor = new File(Commande);

                        try
                        {
                        String[] command = new String[] {"C:\\Dossiers Personnels\\ESO\\Fusex
AMER\\Programme\\Debug\\vid.exe",Commande,Chemin};
                        Process child = Runtime.getRuntime().exec(command);
                        avi2bmp=1;
                        }
                        catch(IOException e)
                        {
                                System.out.println("Erreur IOException :" + e.toString());
                }

                }
                else
                {
                        PasdeVideo pdv = new PasdeVideo();
                        pdv.pack();
                        pdv.setVisible(true);
           }
        }

        if(evt.getSource()==_traitement)
        {
//              Remplissage du champs commande
                this.remplissageCommande();

                try
                {
                        FileReader fr1 = new FileReader(Chemin+"nbimage.txt");
                        fr1.close();
```

```
                try
                {
                        FileReader fr = new FileReader(Chemin+"nbimage.txt");
                        BufferedReader buf = new BufferedReader(fr);
                        nbImage=Integer.parseInt(buf.readLine());
                        fr.close();
                }
                catch(IOException e)
                {
                        System.out.println("Erreur IOException :" + e.toString());
        }

// On regarde si le fichier coordonée existe et si oui, on l'efface
                _coor = new File(Chemin+"Coordonnée.txt");
                if(!(_coor==null))
                {
                        try
                        {
                                FileWriter fr = new FileWriter(_coor);
                                fr.close();
                        }
                        catch(IOException e)
                        {
                                System.out.println("Erreur IOException :" + e.toString());
                }
                }

// On alanylse chaque image
                for(int i=0;i<(nbImage);i++)
                {
                        try
                        {
                                String[] command = new String[] {"C:\\Dossiers Personnels\\ESO\\Fusex
AMER\\Programme\\Debug\\Coordlc.exe", Integer.toString(i), tolRed, tolGreen, tolBlue, Chemin,
Integer.toString(RInf),Integer.toString(VInf),Integer.toString(BInf)};
                                Process child = Runtime.getRuntime().exec(command);
                        }
                        catch(IOException e)
                        {
                                System.out.println("Erreur IOException :" + e.toString());
                }
                        try
                        {
                         //Thread.sleep(5*60000);
                                Thread.sleep(100);
                        }
                        catch (InterruptedException e)
                   {
                   }
                }

                // Ajout des Images à la fenêtre
                Vector _imgitems = new Vector();
                for(int i=0;i<nbImage;i++)
                {
                        _imgitems.add("Img"+i);
                }
```

```
                    _img.setListData(_imgitems);
                    ImageIcon imgprev = new ImageIcon("AucuneImageaaff.gif");
                    _imglabel.setIcon(imgprev);
                    prev=1;
                    avi2bmp=0;
                }
                catch(IOException e)
                {
                    PasdImage pdi = new PasdImage();
                    pdi.pack();
                    pdi.setVisible(true);
        }
        }

        if(evt.getSource()==_exportMatLab)
        {
//              Remplissage du champs commande
                this.remplissageCommande();

                tmp = new TransformationCoorMatLab(this);
                if(Coor==1)
                {
                    tmp.ordre();
                    tmp.pourMatLab();
                }
        }


        if(evt.getSource()==_exit)
        {
                File fr1 = new File(Commande);
                if(fr1.canRead()==true)
                {
                    int tmp=S-1;
                    File del2 = new File(Chemin+"coul"+tmp+".jpg");
                    del2.delete();
                    if(avi2bmp==1)
                    {
                        try
                        {
                            FileReader fr = new FileReader(Chemin+"nbimage.txt");
                            BufferedReader buf = new BufferedReader(fr);
                            nbImage=Integer.parseInt(buf.readLine());
                            fr.close();
                        }
                        catch(IOException e)
                        {
                            System.out.println("Erreur IOException :" + e.toString());
                        }
                        for(int i=0;i<nbImage;i++)
                        {
                            File del = new File(Chemin+"Img"+i+".bmp");
                            del.delete();
                        }
                    }

                    if(prev==1)
                    {
```

```
                        for(int i=0;i<nbImage;i++)
                        {
                                File del = new File(Chemin+"Img"+i+".bmp");
                                del.delete();
                                del = new File(Chemin+"Img"+i+".jpg");
                                del.delete();
                        }
                        File del = new File(Chemin+"Coordonnée.txt");
                        del.delete();
                        //del = new File(Chemin+"nbimage.txt");
                        //del.delete();

                }
        }
        this.dispose();
}


if(evt.getSource()==_tfRed)
{
        _sRed.setValue(Integer.parseInt(_tfRed.getText()));
}
if(evt.getSource()==_tfGreen)
{
        _sGreen.setValue(Integer.parseInt(_tfGreen.getText()));
}
if(evt.getSource()==_tfBlue)
{
        _sBlue.setValue(Integer.parseInt(_tfBlue.getText()));
}

if(evt.getSource()==_genererPanel)
{
        try
        {
                String[] command = new String[] {"C:\\Dossiers Personnels\\ESO\\Fusex
AMER\\Programme\\Debug\\C:\\Dossiers Personnels\\ESO\\Fusex AMER\\Programme\\Debug\\Image.exe", Chemin,
Integer.toString(RInf), tolRed, Integer.toString(VInf),tolGreen, Integer.toString(BInf),tolBlue, Integer.toString(S)};
                Process child = Runtime.getRuntime().exec(command);
        }
        catch(IOException e)
        {
                System.out.println("Erreur IOException :" + e.toString());
}

        try
        {
         Thread.sleep (2000);
        }
        catch (InterruptedException e)
   {
   }

        if(S!=0)
        {
                int tmp=S-1;
                File del = new File(Chemin+"coul"+ tmp +".jpg");
                del.delete();
```

```
                }
                _imgCoul.setIcon(new ImageIcon(Chemin+"coul"+S+".jpg"));

                S++;
        }
        if(evt.getSource()==_browse)
        {
                Parcourir _parcourir = new Parcourir();
                _parcourir.setVisible(true);
                _parcourir.pack();
                if(_parcourir.getparcourir().APPROVE_OPTION==0)
                {
                        int selection = JFileChooser.APPROVE_OPTION;
                        _tfchemin.setText((_parcourir.getparcourir()).getSelectedFile().getName().toString());
                        _parcourir.dispose();
                }
                else
                {
                        _parcourir.dispose();
                }


        }


    }

    public void valueChanged(ListSelectionEvent evt)
    {
        if(prev==1)
        {
                this.remplissageCommande();
                int I = _img.getSelectedIndex();
                ImageIcon imgprev = new ImageIcon(Chemin+"Img"+I+".jpg");
                _imglabel.setIcon(imgprev);
        }
    }

    public void stateChanged(javax.swing.event.ChangeEvent evt)
    {
        if(evt.getSource()==_sRed)
        {
                _tfRed.setText(Integer.toString(_sRed.getValue()));
        }
        if(evt.getSource()==_sGreen)
        {
                _tfGreen.setText(Integer.toString(_sGreen.getValue()));
        }
        if(evt.getSource()==_sBlue)
        {
                _tfBlue.setText(Integer.toString(_sBlue.getValue()));
        }
    }



    public void remplissageCommande()
```

```
    {
        Chemin = ("");
        Commande = _tfchemin.getText();
        Pattern p = Pattern.compile("\\\\");
        String[] dossier;
        dossier = p.split(Commande);
        for(int i=0;i<dossier.length-1;i++)
        {
                Chemin = Chemin+dossier[i]+"/";
        }
    }

    public static void main(String[] args)
    {
        FenetrePrincipale nv = new FenetrePrincipale();
        nv.pack();
        nv.setVisible(true);
    }

}
```

- **Classe Mon Filtre :**

```
import java.io.*;
import java.util.regex.Pattern;
import java.awt.*;

import javax.swing.filechooser.FileFilter;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import java.awt.event.*;


public class MonFiltre extends FileFilter{

  //Description et extension acceptée par le filtre
  private String description;
  private String extension;


  //Constructeur à partir de la description et de l'extension acceptée
  public MonFiltre(String description, String extension){

    this.description = description;
    this.extension = extension;
  }


  //Implémentation de FileFilter
  public boolean accept(File file){
    if(file.isDirectory()) {
      return true;
    }
    String nomFichier = file.getName().toLowerCase();
    return nomFichier.endsWith(extension);
  }
```

```
  public String getDescription(){
    return description;
  }
}
```

- **Classe Parcourir :**

```
import java.io.*;
import java.util.regex.Pattern;
import java.awt.*;

import javax.swing.*;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import java.awt.event.*;
import javax.swing.filechooser.FileFilter;



public class Parcourir extends JFrame{

    private JFileChooser _parcourir;
    private JPanel _pane;
    private FileFilter _avi;

    public Parcourir() {
        super("Parcourir");

        JFileChooser _essai = new JFileChooser();
        FileFilter _avi = new MonFiltre("Fichiers Vidéo avi",".avi");
        _essai.addChoosableFileFilter(_avi);

        JPanel _pane = new JPanel(new GridLayout());

        _pane.add(_essai);

        this.setContentPane(_pane);
    }

    /**ACCESSEURS**/
    public JFileChooser getparcourir()
    {
        return _parcourir;
    }
}
```

- **Classe Pas de Point Rouge :**

```
import java.awt.*;
import java.util.*;
import javax.swing.*;

import java.awt.event.*;

public class PasdePointRouge extends JFrame implements ActionListener{

/** ATTRIBUTS */
        // Le Texte
```

```
	private JLabel _premiertexte;
	private JLabel _logo;

	// Le Bouton
	private JButton _OK;

	// Le Panneau
	private JPanel _pane;
	private JPanel _paneG;
	private JPanel _paneD;
	private JPanel _paneBoutton;


/** CONSTRUCTEUR */
	public PasdePointRouge() {
		super("Erreur - Pas de Point Rouge");

		// Création du Paneeau
		_pane = new JPanel(new BorderLayout());
		_paneG = new JPanel();
		_paneD = new JPanel();
		_paneBoutton = new JPanel();

		// Création du Texte
		ImageIcon icon = new ImageIcon("Erreur.gif");
		_premiertexte = new JLabel("Erreur : Pas de Résultats à Afficher",JLabel.CENTER);
		_logo = new JLabel(icon);

		//Création du Bouton
		_OK = new JButton("OK");

		// Ajout du Texte et du Bouton dans le Panneau
		_paneD.add(_premiertexte);
		_paneG.add(_logo);
		_paneBoutton.add(_OK);

		_pane.add(_paneG,BorderLayout.WEST);
		_pane.add(_paneD,BorderLayout.EAST);
		_pane.add(_paneBoutton,BorderLayout.SOUTH);

		// Ajout du Panneau dans la Fenêtre
		this.setContentPane(_pane);

		// "Ecouter" ces boutons
		_OK.addActionListener(this);

	}

/** METHODES */
	public void actionPerformed(ActionEvent evt)
	{
		if(evt.getSource()==_OK)
		{
			this.dispose();
		}
	}
		}
```

- **Classe Pas de Traitement:**

```
import java.awt.*;
import java.util.*;
import javax.swing.*;

import java.awt.event.*;

public class PasdeTraitement extends JFrame implements ActionListener{

/** ATTRIBUTS */
    // Le Texte
    private JLabel _premiertexte;
    private JLabel _logo;

    // Le Bouton
    private JButton _OK;

    // Le Panneau
    private JPanel _pane;
    private JPanel _paneG;
    private JPanel _paneD;
    private JPanel _paneBoutton;


/** CONSTRUCTEUR */
    public PasdeTraitement() {
        super("Erreur - Pas de Traitement");

        // Création du Paneeau
        _pane = new JPanel(new BorderLayout());
        _paneG = new JPanel();
        _paneD = new JPanel();
        _paneBoutton = new JPanel();

        // Création du Texte
        ImageIcon icon = new ImageIcon("Erreur.gif");
        _premiertexte = new JLabel("Erreur : Aucun Résultat à afficher ",JLabel.CENTER);
        _logo = new JLabel(icon);

        //Création du Bouton
        _OK = new JButton("OK");

        // Ajout du Texte et du Bouton dans le Panneau
        _paneD.add(_premiertexte);
        _paneG.add(_logo);
        _paneBoutton.add(_OK);

        _pane.add(_paneG,BorderLayout.WEST);
        _pane.add(_paneD,BorderLayout.EAST);
        _pane.add(_paneBoutton,BorderLayout.SOUTH);

        // Ajout du Panneau dans la Fenêtre
        this.setContentPane(_pane);

        // "Ecouter" ces boutons
        _OK.addActionListener(this);
```

```
    }

/** METHODES */
    public void actionPerformed(ActionEvent evt)
    {
        if(evt.getSource()==_OK)
        {
                this.dispose();
        }
    }
        }
```

- **Classe Pas de Vidéo :**

```
import java.awt.*;
import java.util.*;
import javax.swing.*;

import java.awt.event.*;

public class PasdeVideo extends JFrame implements ActionListener{

/** ATTRIBUTS */
        // Le Texte
        private JLabel _premiertexte;
        private JLabel _logo;

        // Le Bouton
        private JButton _OK;

        // Le Panneau
        private JPanel _pane;
        private JPanel _paneG;
        private JPanel _paneD;
        private JPanel _paneBoutton;


/** CONSTRUCTEUR */
        public PasdeVideo() {
                super("Erreur - Pas de Vidéo");

                // Création du Paneeau
                _pane = new JPanel(new BorderLayout());
                _paneG = new JPanel();
                _paneD = new JPanel();
                _paneBoutton = new JPanel();

                // Création du Texte
                ImageIcon icon = new ImageIcon("Erreur.gif");
                _premiertexte = new JLabel("Erreur : Le chemin de la vidéo n'est pas valide",JLabel.CENTER);
                _logo = new JLabel(icon);

                //Création du Bouton
                _OK = new JButton("OK");

                // Ajout du Texte et du Bouton dans le Panneau
                _paneD.add(_premiertexte);
```

```
		_paneG.add(_logo);
		_paneBoutton.add(_OK);

		_pane.add(_paneG,BorderLayout.WEST);
		_pane.add(_paneD,BorderLayout.EAST);
		_pane.add(_paneBoutton,BorderLayout.SOUTH);

		// Ajout du Panneau dans la Fenêtre
		this.setContentPane(_pane);

		// "Ecouter" ces boutons
		_OK.addActionListener(this);

	}

/** METHODES */
	public void actionPerformed(ActionEvent evt)
	{
		if(evt.getSource()==_OK)
		{
			this.dispose();
		}
	}
		}
```

- **Classe Pas d'Image :**

```
import java.awt.*;
import java.util.*;
import javax.swing.*;

import java.awt.event.*;

public class PasdImage extends JFrame implements ActionListener{

/** ATTRIBUTS */
	// Le Texte
	private JLabel _premiertexte;
	private JLabel _logo;

	// Le Bouton
	private JButton _OK;

	// Le Panneau
	private JPanel _pane;
	private JPanel _paneG;
	private JPanel _paneD;
	private JPanel _paneBoutton;


/** CONSTRUCTEUR */
	public PasdImage() {
		super("Erreur - Pas d'image");

		// Création du Paneeau
		_pane = new JPanel(new BorderLayout());
		_paneG = new JPanel();
		_paneD = new JPanel();
```

```
		_paneBoutton = new JPanel();

		// Création du Texte
		ImageIcon icon = new ImageIcon("Erreur.gif");
		_premiertexte = new JLabel("Erreur : Aucune Image Disponible à traiter",JLabel.CENTER);
		_logo = new JLabel(icon);

		//Création du Bouton
		_OK = new JButton("OK");

		// Ajout du Texte et du Bouton dans le Panneau
		_paneD.add(_premiertexte);
		_paneG.add(_logo);
		_paneBoutton.add(_OK);

		_pane.add(_paneG,BorderLayout.WEST);
		_pane.add(_paneD,BorderLayout.EAST);
		_pane.add(_paneBoutton,BorderLayout.SOUTH);

		// Ajout du Panneau dans la Fenêtre
		this.setContentPane(_pane);

		// "Ecouter" ces boutons
		_OK.addActionListener(this);

	}

/** METHODES */
	public void actionPerformed(ActionEvent evt)
	{
		if(evt.getSource()==_OK)
		{
			this.dispose();
		}
	}
		}
```

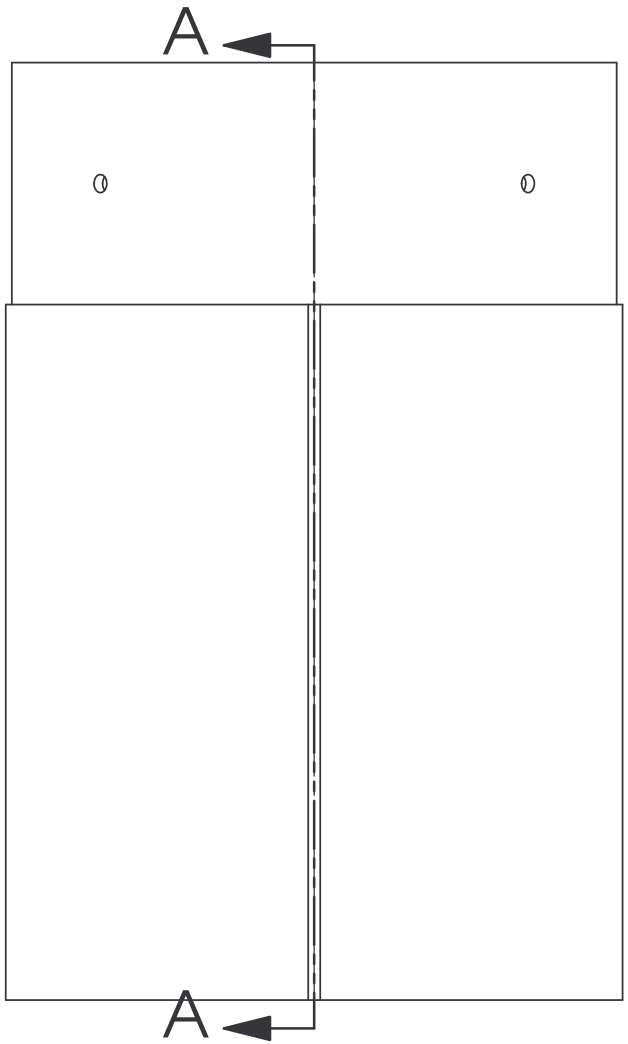
- **Classe TransformationCoorMatLab :**

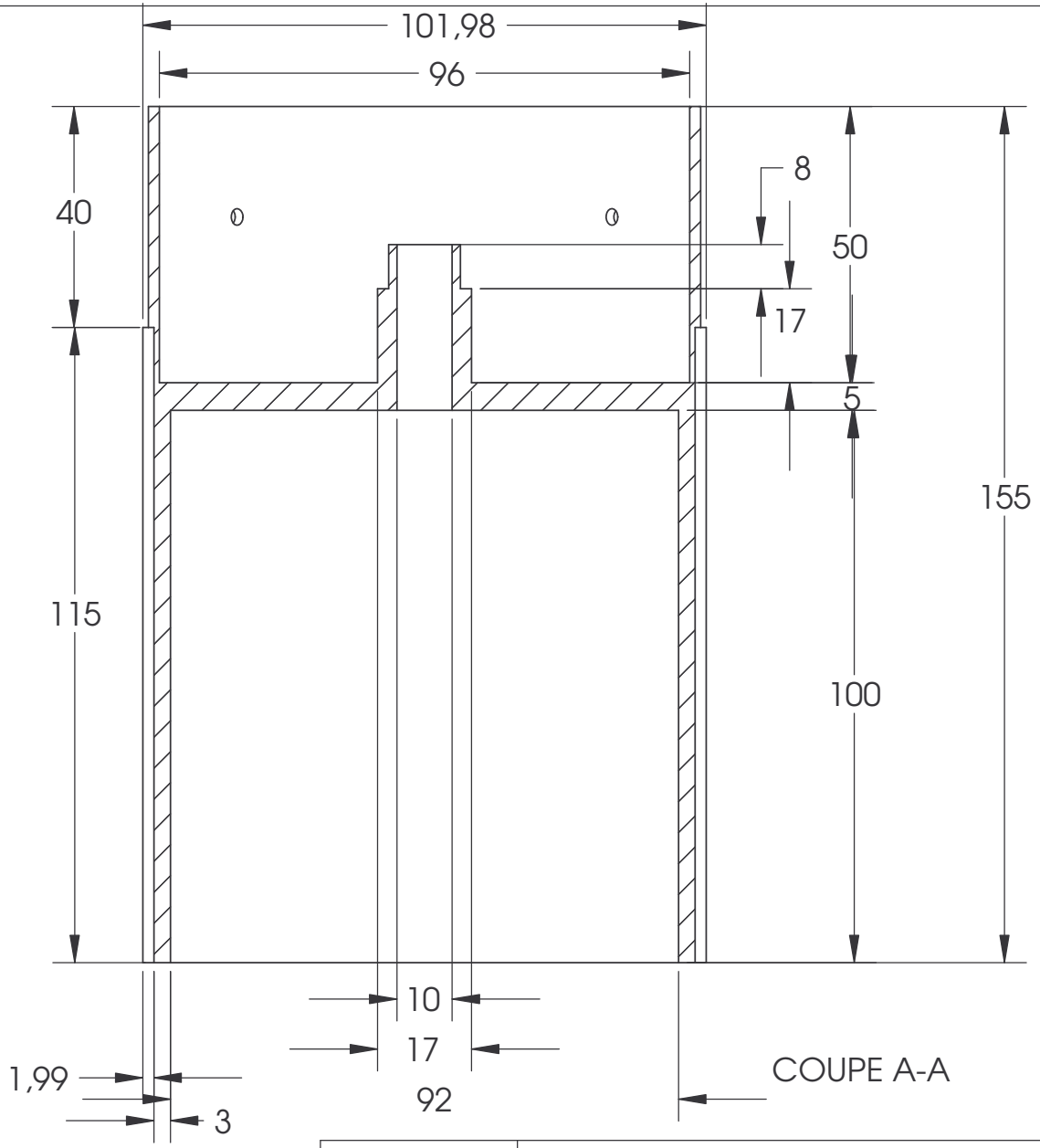
```
import java.util.regex.Pattern;
import javax.swing.JFrame;
import java.io.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import javax.swing.*;
import java.util.*;
import java.awt.event.*;


public class TransformationCoorMatLab extends JFrame implements ActionListener{

	private Vector _liste;

	private File _fichier;

	private JTextArea _result;
	private JButton _ok;
```

```
private JPanel _pane;

private FenetrePrincipale fp;

public TransformationCoorMatLab(FenetrePrincipale nv) {
	super("Résultats :");

	fp=nv;

	_result = new JTextArea("");
	_result.setPreferredSize(new Dimension(600,50));
	_ok = new JButton("OK");
	_pane = new JPanel(new BorderLayout());

	_pane.add(_result,BorderLayout.NORTH);
	_pane.add(_ok,BorderLayout.SOUTH);

	_ok.addActionListener(this);

	this.setContentPane(_pane);


	String ligne;
	Pattern p = Pattern.compile(" ");
	String[] items = new String[10];

	_fichier = new File(fp.getChemin()+"Coordonnée.txt");
	_liste = new Vector();


	try
	{
		// On ouvre un flux sur ce fichier
		FileReader fr = new FileReader(_fichier);
		BufferedReader buf = new BufferedReader(fr);

		if(buf.readLine()!=null)
		{

			while((ligne = buf.readLine()) != null)
			{
				Vector _carac = new Vector();
				items = p.split(ligne);

				_carac.addElement(items[0]);
				_carac.addElement(items[1]);
				_carac.addElement(items[2]);

				_liste.addElement( _carac);
			}
		}
		// Pour finir, on ferme le fichier
		fr.close();
		fp.setCoor(1);
	}
	catch(IOException e)
	{
	    PasdeTraitement pdpr = new PasdeTraitement();
```

```
                pdpr.pack();
                pdpr.setVisible(true);
        }


}

public void actionPerformed(ActionEvent evt)
{
        if(evt.getSource()==_ok)
        {
                this.dispose();
        }
}

public void pourMatLab(){
        if(_liste.size()==0)
        {
                PasdePointRouge nv = new PasdePointRouge();
                nv.pack();
                nv.setVisible(true);
        }
        else
        {
                _result.append("t=[");
                for(int i=0;i<_liste.size();i++)
                {
                        _result.append(((Vector)_liste.elementAt(i)).elementAt(0)+" ");
                }
                _result.append("];\n");

                _result.append("x=[");
                for(int i=0;i<_liste.size();i++)
                {
                        _result.append(((Vector)_liste.elementAt(i)).elementAt(1)+" ");
                }
                _result.append("];\n");

                _result.append("y=[");
                for(int i=0;i<_liste.size();i++)
                {
                        _result.append(((Vector)_liste.elementAt(i)).elementAt(2)+" ");
                }
                _result.append("];\n");
                this.pack();
                this.setVisible(true);
        }

}

public void ordre()
{
        String tmp0;
        String tmp1;
        String tmp2;
        String tmp3;
        for(int i=0;i<_liste.size();i++)
        {
```

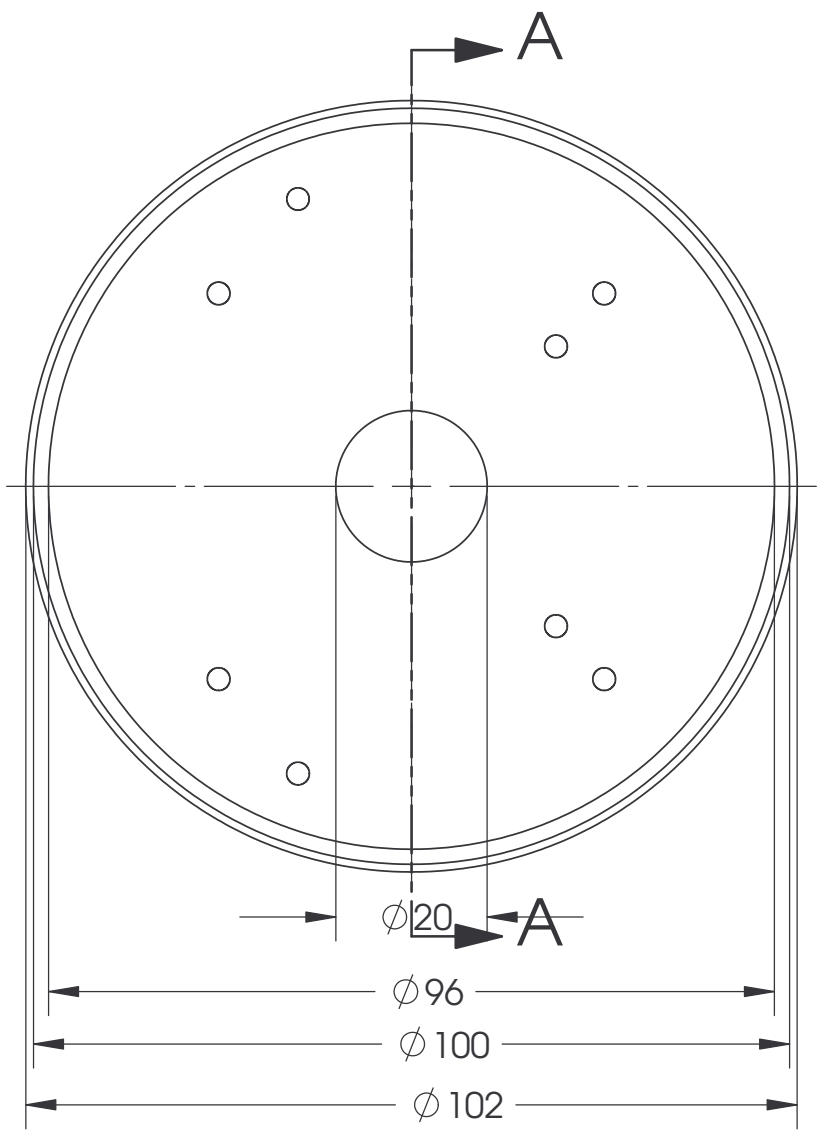
```
                        for(int j=i+1;j<_liste.size();j++)
                        {
                                int J = Integer.parseInt((String) ((Vector)(_liste.elementAt(j))).elementAt(0));
                                int I = Integer.parseInt((String) ((Vector)(_liste.elementAt(i))).elementAt(0));
                                if(I>J)
                                {
                                        tmp0=((String) ((Vector)(_liste.elementAt(j))).elementAt(0));
                                        tmp1=((String) ((Vector)(_liste.elementAt(j))).elementAt(1));
                                        tmp2=((String) ((Vector)(_liste.elementAt(j))).elementAt(2));

                                        ((Vector)(_liste.elementAt(j))).setElementAt(((String)
((Vector)(_liste.elementAt(i))).elementAt(0)),0);
                                        ((Vector)(_liste.elementAt(j))).setElementAt(((String)
((Vector)(_liste.elementAt(i))).elementAt(1)),1);
                                        ((Vector)(_liste.elementAt(j))).setElementAt(((String)
((Vector)(_liste.elementAt(i))).elementAt(2)),2);

                                        ((Vector)(_liste.elementAt(i))).setElementAt(tmp0,0);
                                        ((Vector)(_liste.elementAt(i))).setElementAt(tmp1,1);
                                        ((Vector)(_liste.elementAt(i))).setElementAt(tmp2,2);
                                }
                        }
                }



        }

}
```

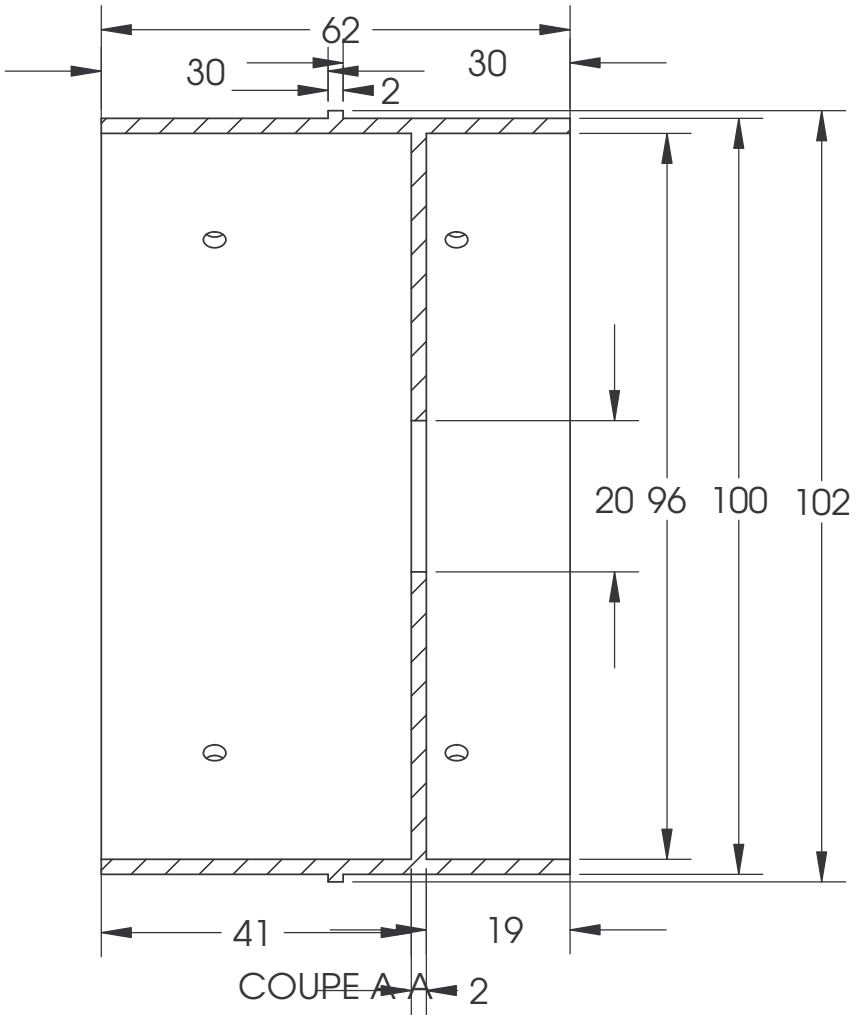




Echelle 4:5	Bague De Poussée Bague Aileron	
A4		
Le 23/06/06	Fusée AMER	ESTACA

5 4 3 2 1

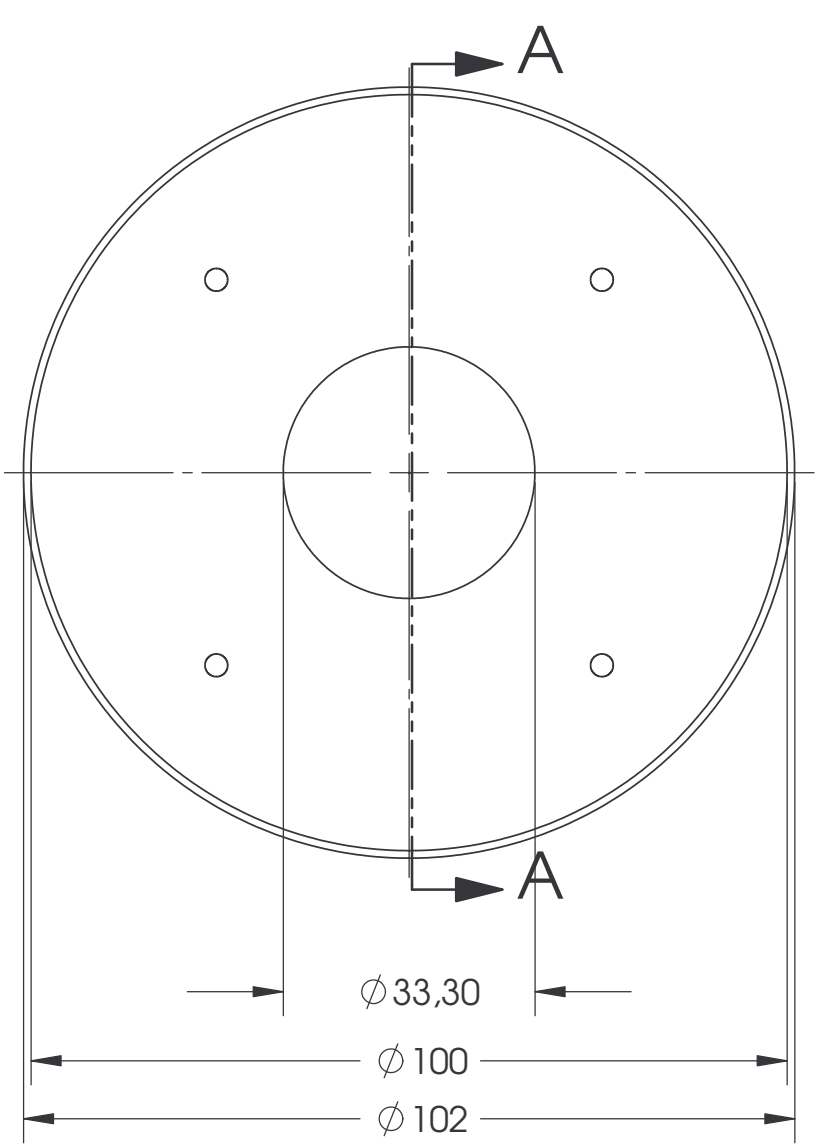


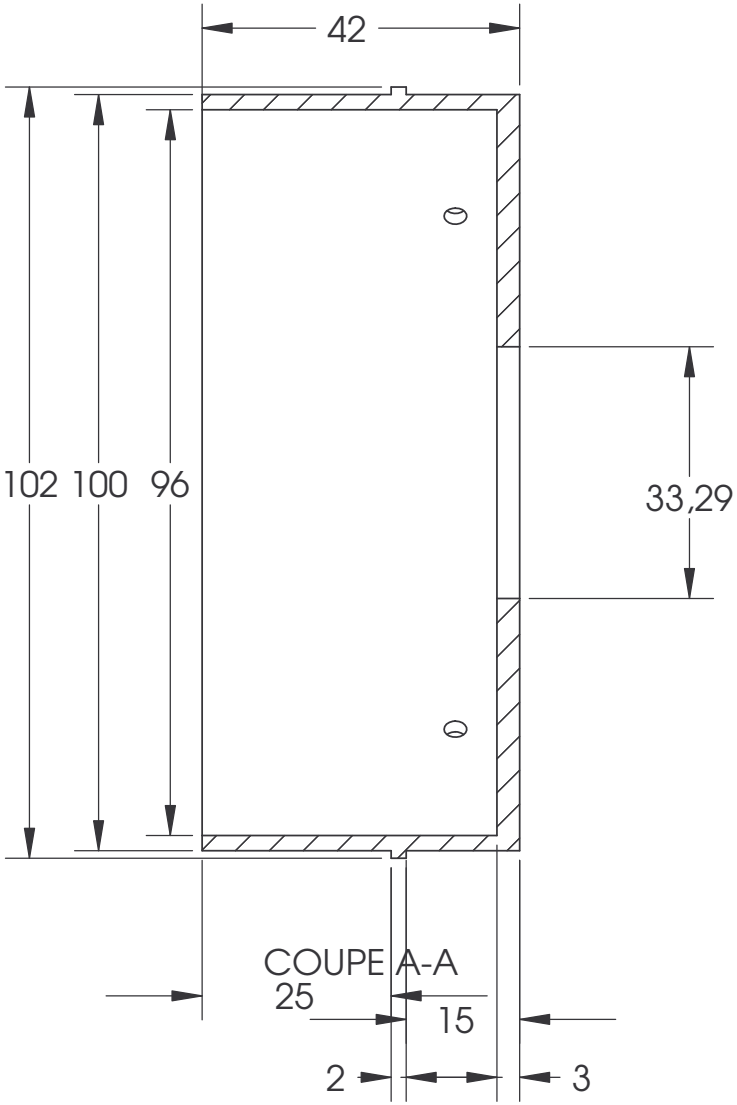


COUPE A-A

Echelle 1:1	Bague Intermédiaire	
A4		
Le 23/06/06	Fusée AMER	ESTACA

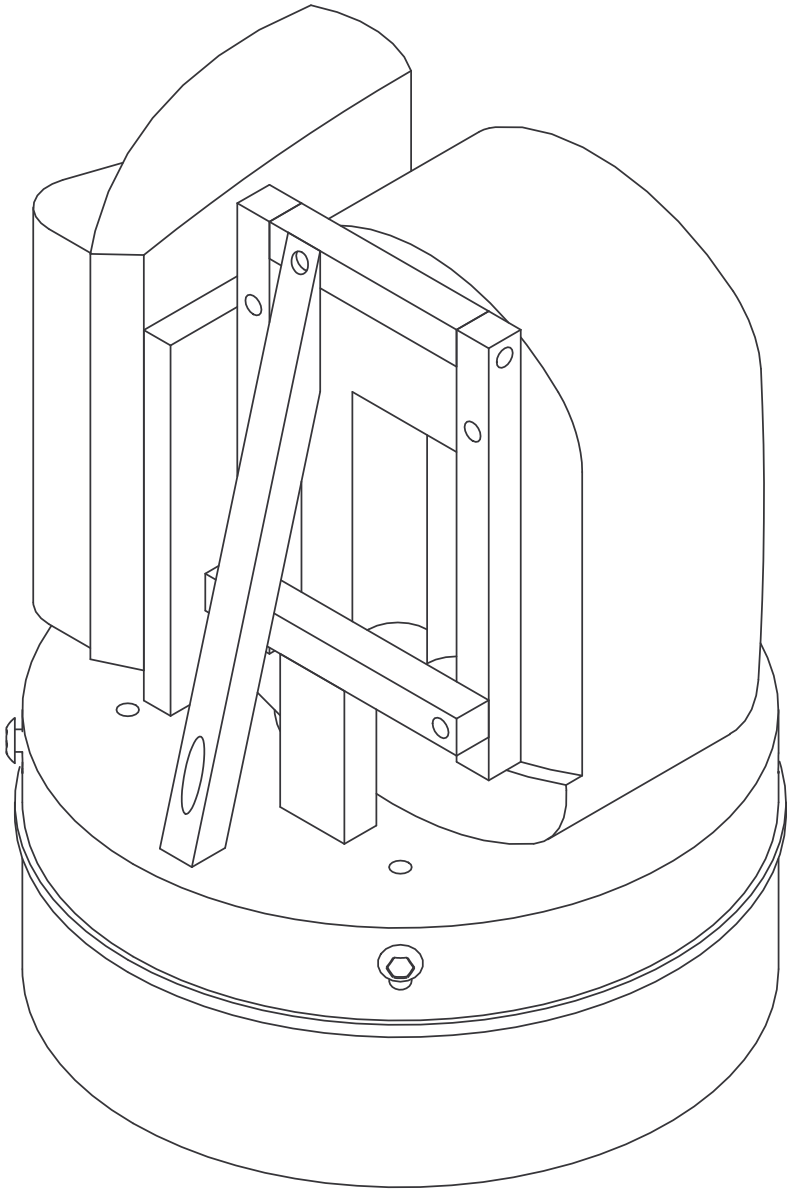
5 4 3 2 1





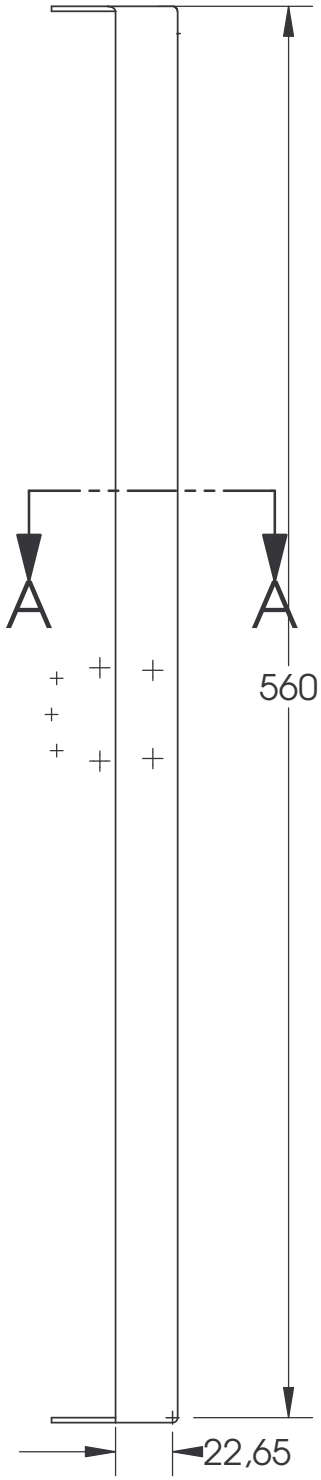
Echelle 1:1	Bague Supérieure	
A4		
Le 23/06/06	Fusée AMER	ESTACA

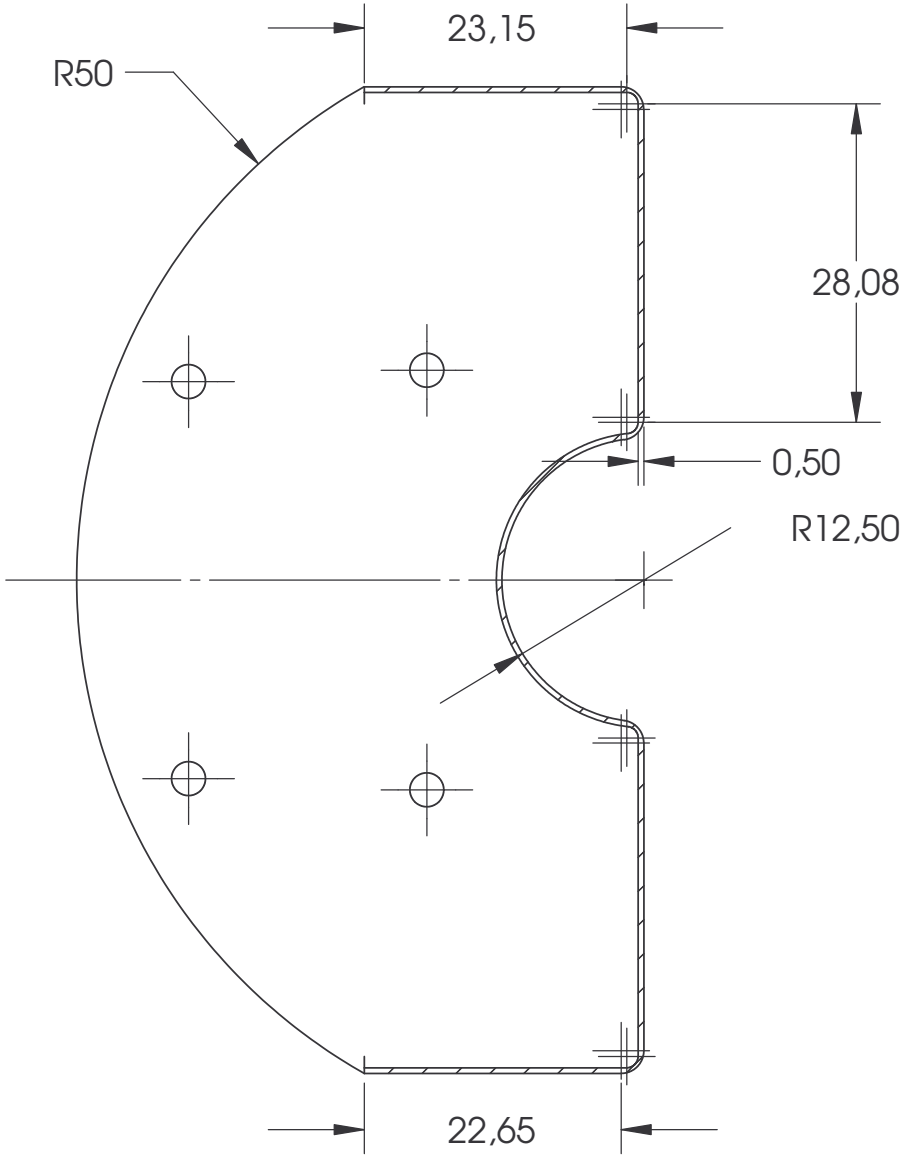
5 4 3 2 1



Echelle 1:1	Case à Equipement	
A4		
Le 23/06/06	Fusée AMER	ESTACA

5 4 3 2 1

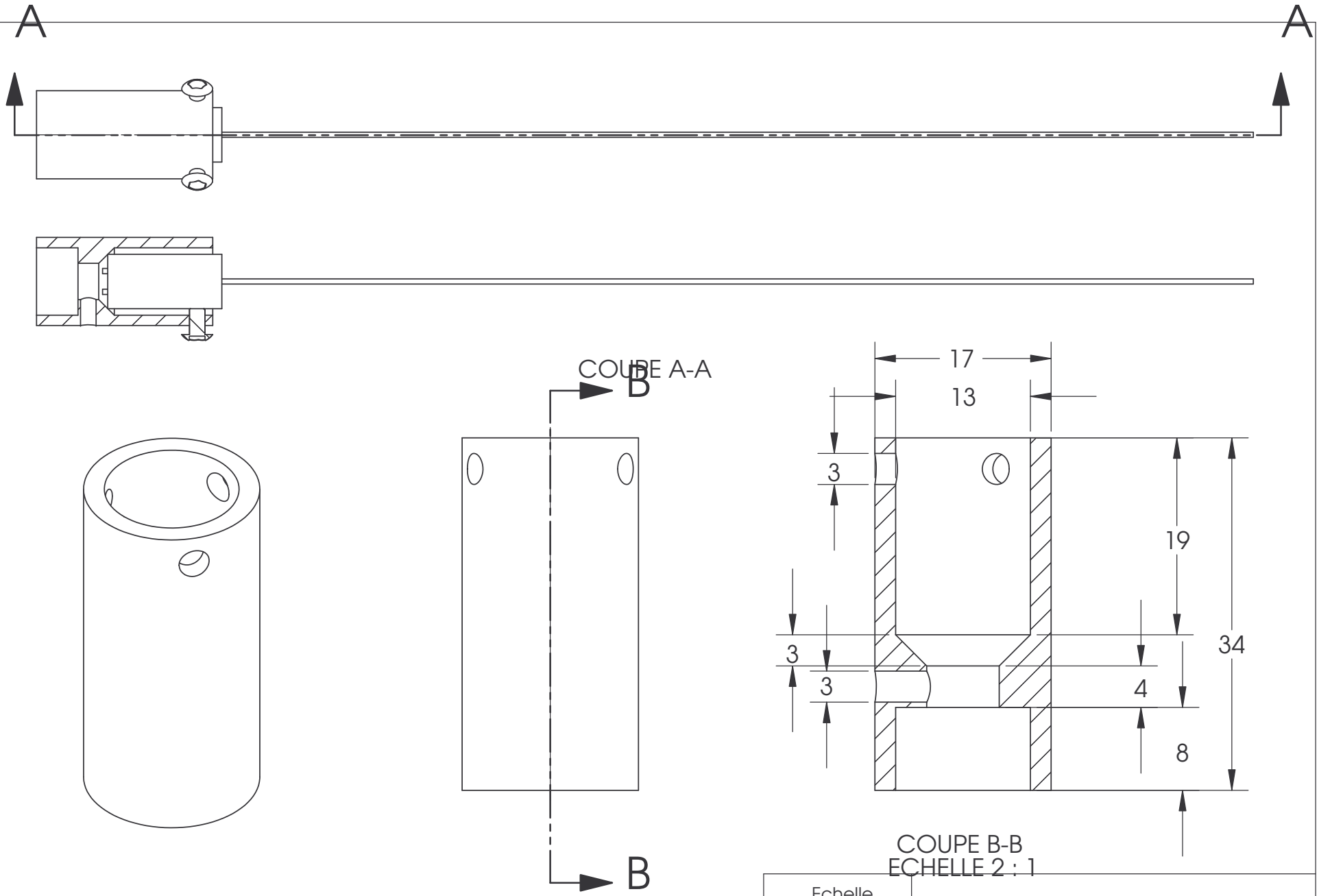




COUPE A-A
ECHELLE 3 : 2

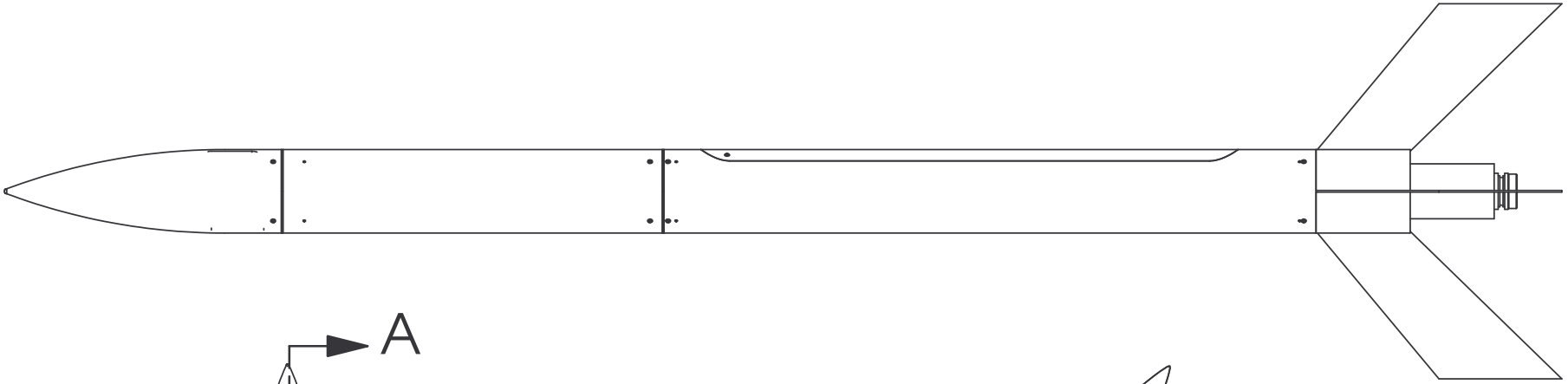
Echelle 1:3	Coffre Parachute
A4	
Le 23/06/06	Fusée AMER — ESTACA

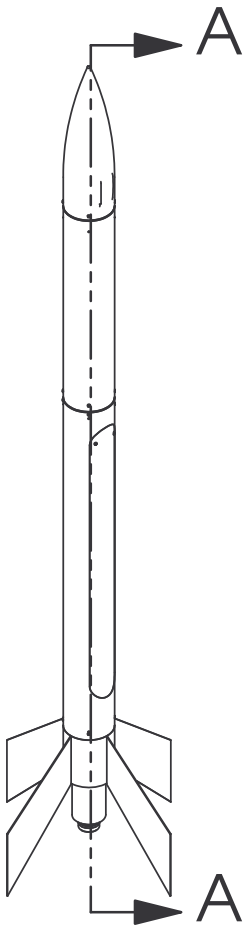
5 4 3 2 1

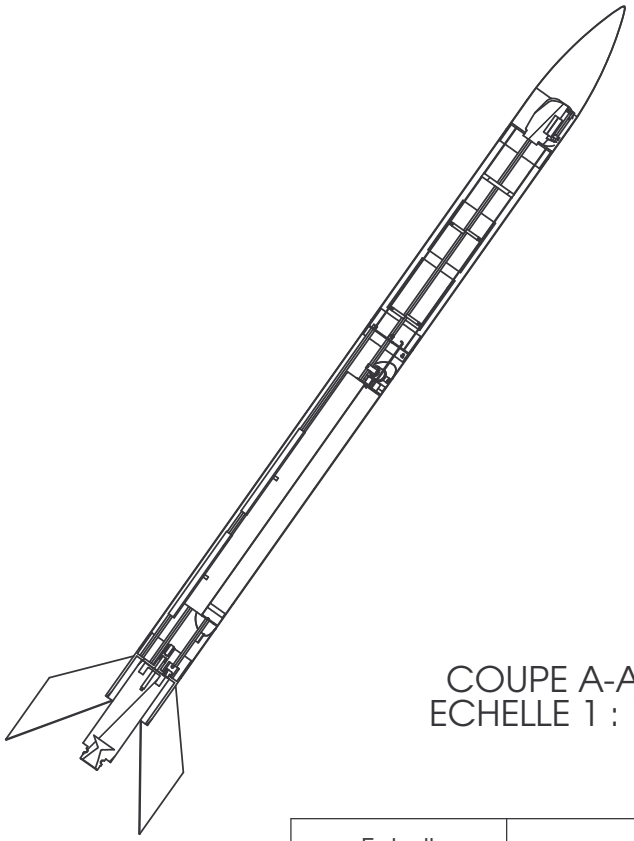


Echelle 1:1	Support du Laser	
A4		
Le 23/06/06	Fusée AMER	ESTACA

5 4 3 2 1



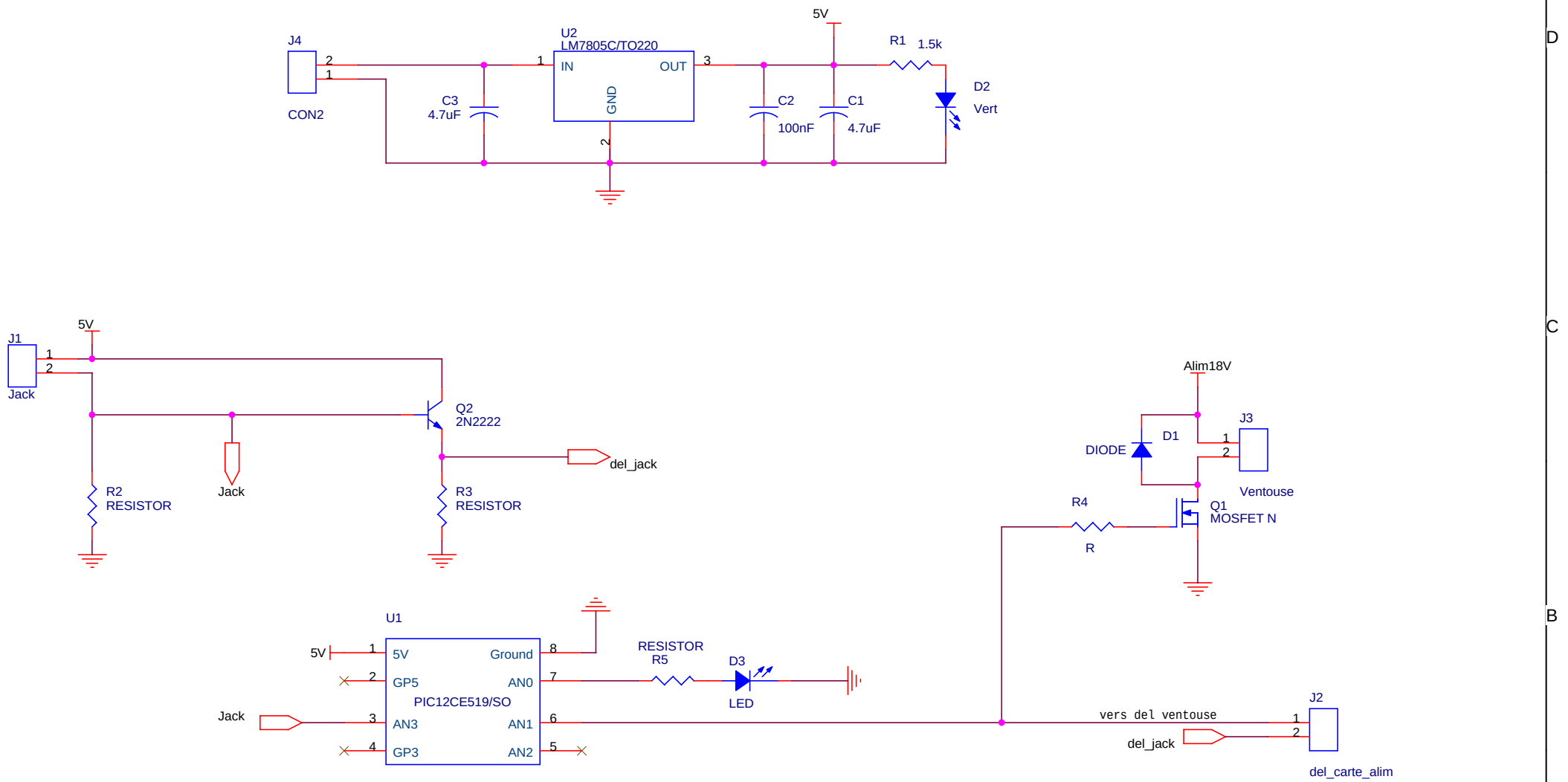




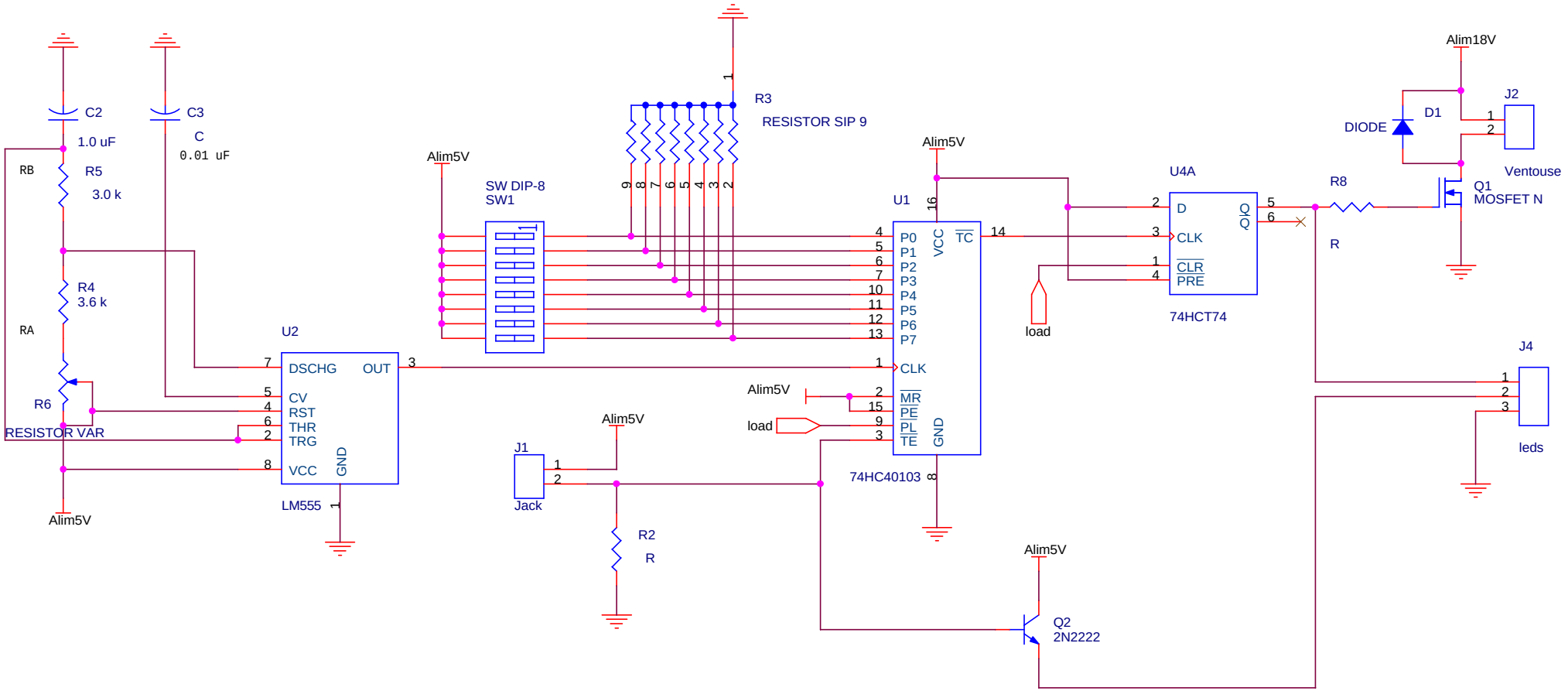
COUPE A-A
ECHELLE 1 : 15

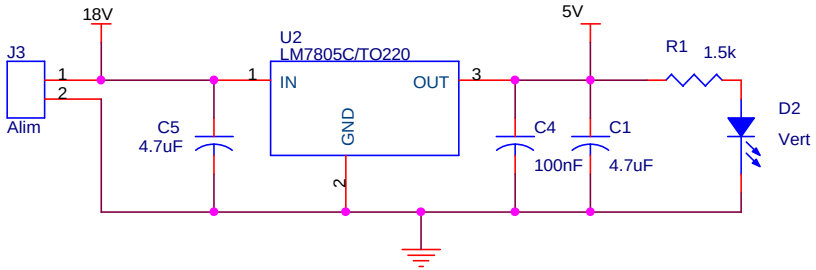
Echelle 1:8	Vue d'ensemble	
A4		
Le 23/06/06	Fusée AMER	ESTACA

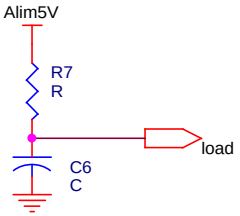
5 4 3 2 1



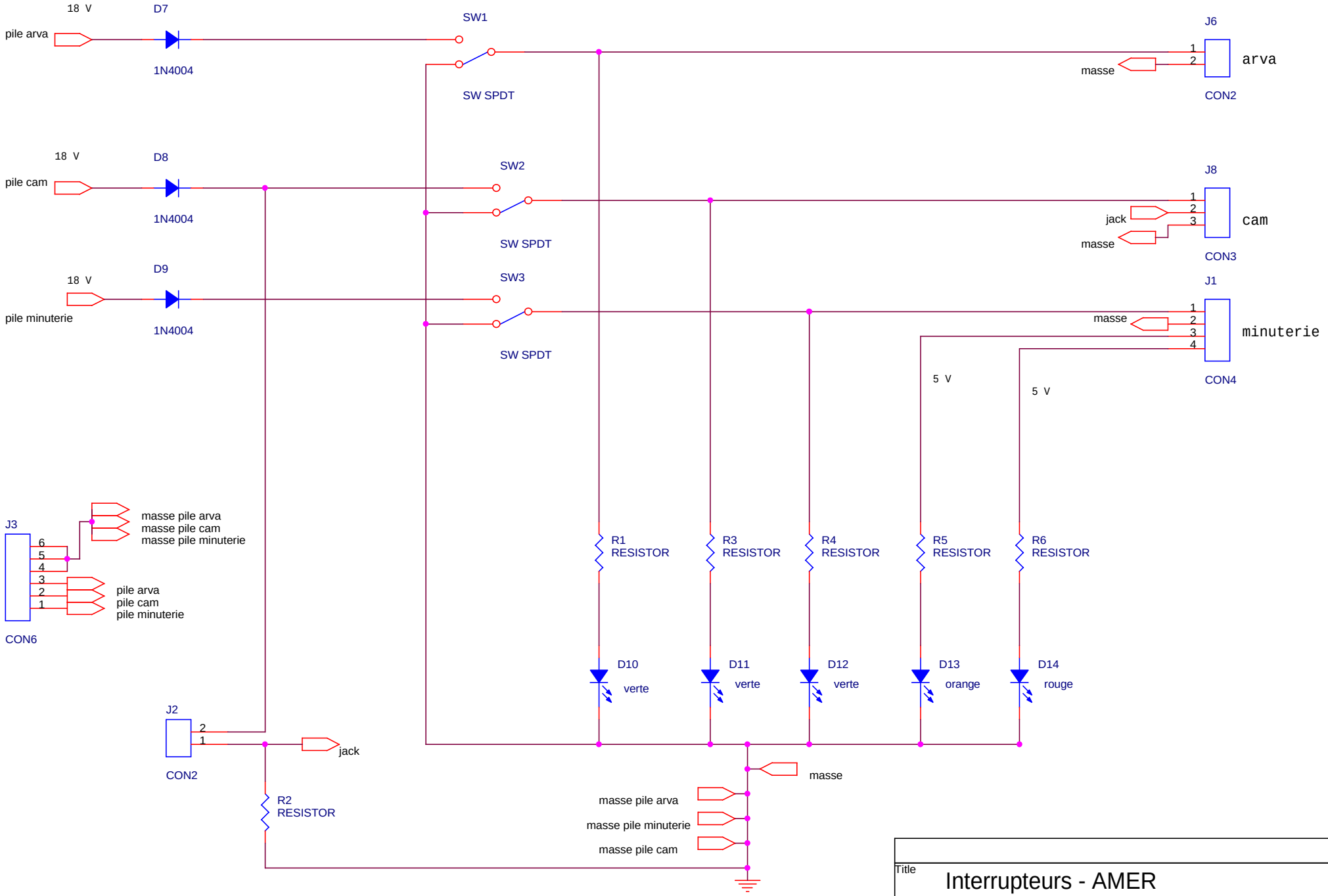
Title	Minuterie Pic - AMER	
Size A4	Document Number <Doc>	Rev <RevCode>
Date: Wednesday, June 21, 2006	Sheet 1 of 1	



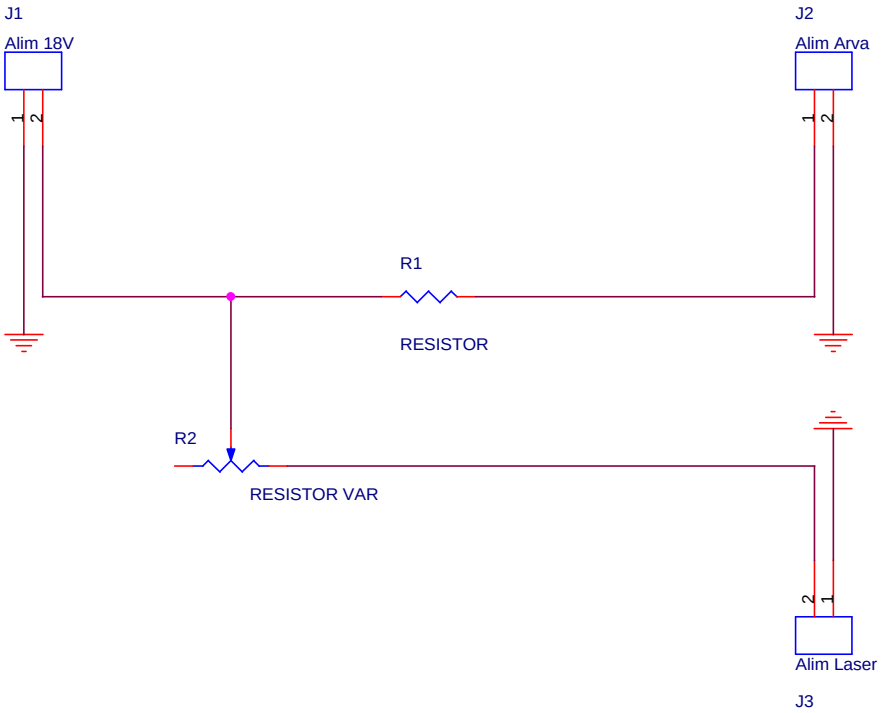




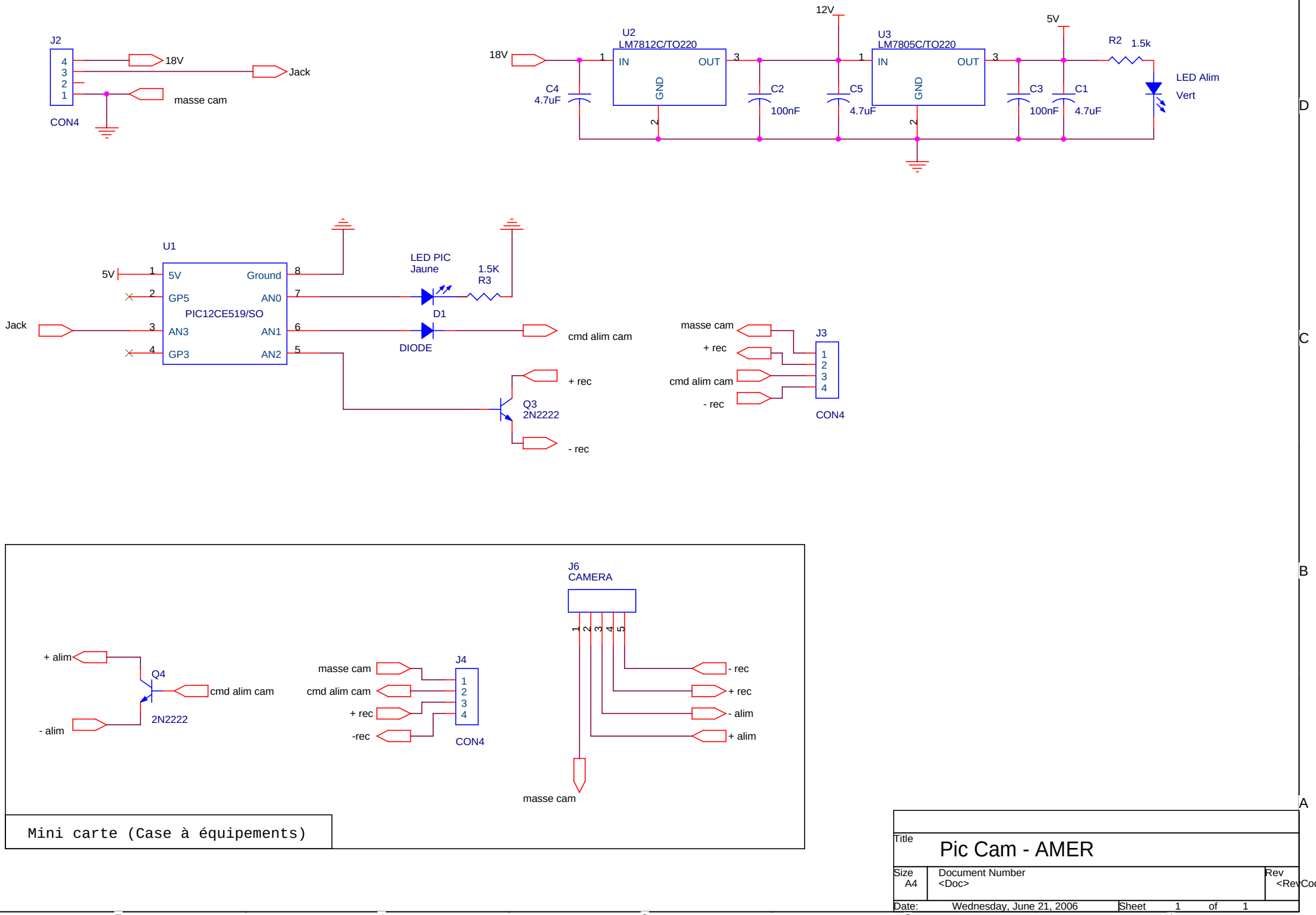
Title	Minuterie Numérique - AMER	
Size A4	Document Number <Doc>	Rev <RevCode>
Date: Wednesday, June 21, 2006	Sheet 1 of 1	



Title	Interrupteurs - AMER	
Size A4	Document Number <Doc>	Rev <RevCode>
Date: Friday, June 23, 2006	Sheet 1 of 1	



Title <Title>		
Size A4	Document Number <Doc>	Rev <RevCode>
Date: Wednesday, June 21, 2006	Sheet 1 of 1	


J2
4
3
2
1
18V
Jack
masse cam
CON4
12V
U2
LM7812C/TO220
IN
OUT
GND
U3
LM7805C/TO220
5V
R2
1.5k
LED Alim
Vert
18V
C4
4.7uF
C2
100nF
C5
4.7uF
C3
100nF
C1
4.7uF
U1
PIC12CE519/SO
5V
Ground
GP5
AN0
AN3
AN1
GP3
AN2
LED PIC
Jaune
1.5K
R3
D1
DIODE
Jack
cmd alim cam
+ rec
Q3
2N2222
- rec
masse cam
+ rec
cmd alim cam
- rec
J3
CON4
+ alim
Q4
2N2222
- alim
cmd alim cam
masse cam
cmd alim cam
+ rec
-rec
J4
CON4
J6
CAMERA
masse cam
- rec
+ rec
- alim
+ alim
Mini carte (Case à équipements)
Title
Pic Cam - AMER
Size
A4
Document Number
<Doc>
Rev
<RevCode
Date:
Wednesday, June 21, 2006
Sheet
1
of
1