



PROJET FUSEX 2006 : SATOURNE V

Compte rendu d'expérience



Membres du club

Florent Bouchet florent.bouchet@supelec.fr

Jean Charvet jean.charvet@supelec.fr

Philippe Ullah philippe.ullah@supelec.fr

Antoine Vallée antoine.vallee@supelec.fr

Aline Meuris aline.meuris@supelec.fr

Supélec Espace

Supélec - Plateau du Moulon - 91192 Gif-sur-Yvette Cedex
espace@supelec.fr



Sommaire

A. OBJECTIFS EXPERIMENTAUX	4
1 Expérience principale : production d'énergie électrique	4
1.1 But de l'expérience	4
1.2 Principe de mise en œuvre	4
1.3 Choix des capteurs	5
1.4 Etalonnage et prédéterminations des résultats	6
1.4.1 Etalonnage sur une voiture	6
1.4.2 Extrapolation au vol de la fusée	6
2 Expérience secondaire : tension subie par les sangles parachute	7
2.1 But de l'expérience	7
2.2 Mise en œuvre	7
2.2.1 Théorie des jauges de contraintes	7
2.2.2 La pièce mécanique	7
2.2.3 Conditionnement du signal	8
B. REALISATIONS	9
1 Vue d'ensemble de la fusée expérimentale	9
1.1 Plan d'intégration	9
1.2 Plan de câblage	Erreur ! Signet non défini.
1.3 Structure mécanique	9
1.4 Ailerons	10
2 Systèmes électroniques	11
2.1 Carte centrale d'acquisition	11
2.2 Caractérisation de la chaîne de mesure	11
2.3 Télémessure numérique	12
3 Système de récupération	13
3.1 Séquenceur	13
3.2 Parachute	13
C. MISE EN ŒUVRE DU PROJET	14
1 Caractéristiques du vol	14
1.1 Estimation des paramètres de vol par Trajec	14
1.2 Chronologie	14
1.3 Résultat de vol	15
2 Résultats d'expérience	16
3 Bilan du projet	18
3.1 Difficultés rencontrées	18
3.1.1 Aspect mécanique	18

3.1.2	Aspect électronique	18
3.2	Améliorations à apporter dans les futurs projets de fusée expérimentale	18
3.2.1	Finitions mécaniques	18
3.2.2	Mise en œuvre de l'expérience secondaire	18
3.2.3	Mise en œuvre de la mesure de l'état du séquenceur	18
D.	ANNEXES	19
1	Plans mécaniques	19
2	Schémas électroniques	25
2.1	Carte centrale d'acquisition	Erreur ! Signet non défini.
2.2	Séquenceur	Erreur ! Signet non défini.
3	Codes microcontrôleur	19
3.1	Code du séquenceur (PIC16F)	19
3.2	Code de la carte centrale d'acquisition (PIC18F)	19
3.2.1	Notes pour la programmation	19
3.2.2	Logiciel embarqué	19



A. Objectifs expérimentaux

1 Expérience principale : production d'énergie électrique

1.1 But de l'expérience

L'expérience *Turbine* consiste à produire du courant à l'aide d'une hélice entraînant, grâce à un arbre, un moteur à courant continu fonctionnant en générateur. Le but de cette expérience est de mesurer la puissance fournie par un tel dispositif, afin d'envisager la possibilité de l'utiliser pour alimenter des circuits dans un prochain projet.

1.2 Principe de mise en œuvre

Le système de génération d'électricité est composé d'un moteur à courant continu utilisé en génératrice, relié à l'hélice par un arbre et un cardan. Le cardan a été jugé nécessaire pour pouvoir s'affranchir des contraintes en torsion au niveau de la transmission. Le moteur est fixé derrière la plaque supérieure de la structure (maintenu à l'arrière par une plaque vissée à la structure) et l'arbre traverse la coiffe de part en part.

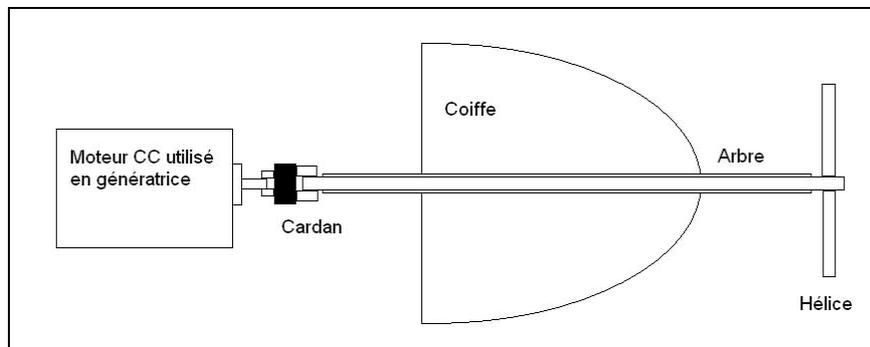


Figure 1 – Principe d'intégration de la génératrice au sommet de la fusée

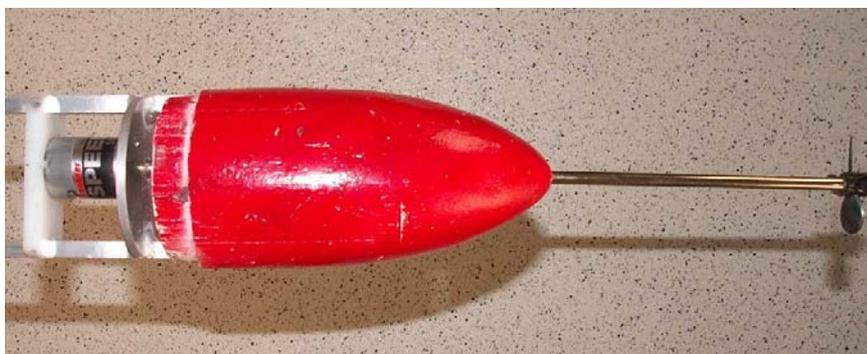


Figure 2 – Intégration réelle du système de mesure dans la coiffe

Le principe est de mesurer la tension aux bornes de résistances de charge stables pour en déduire la puissance électrique générée. On effectue ensuite un conditionnement du signal (tension entre 0 et 5 V) pour arriver sur un convertisseur analogique numérique en liaison SPI avec la carte centrale d'acquisition.

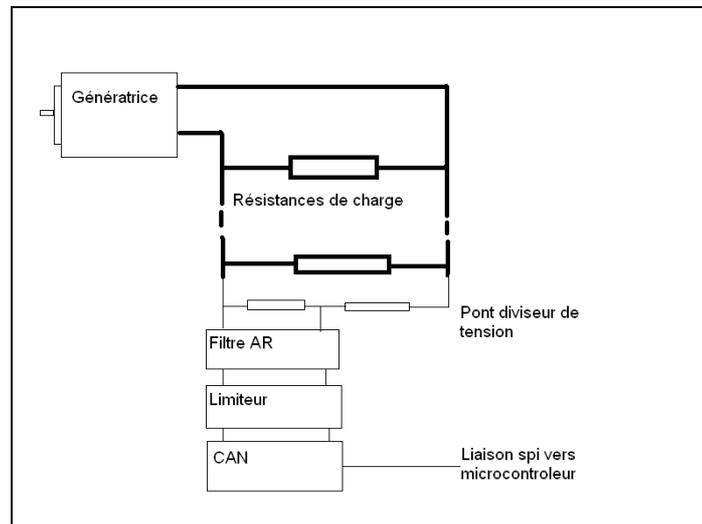


Figure 3 – Chaîne de mesure de l'expérience principale

1.3 Choix des capteurs

Capteur	Type	Référence, caractéristiques	Lieu d'achat
Hélice	Pour maquette de bateau, en fibre de carbone		EOL modélisme
Génératrice	Moteur de maquette de bateau	Speed 600, 12V	EOL modélisme
Résistances de charges	Résistances haute puissance	220 ohm, 25W	

1.4 Etalonnage et prédéterminations des résultats

1.4.1 Etalonnage sur une voiture



Afin de vérifier que le vent entraînait bien l'hélice et produisait une puissance intéressante, nous avons fixé les équipements mécaniques à une voiture et mesuré la tension produite sur une charge de 100ohms. Toutefois, la valeur de cette charge est à prendre avec précaution, car nous ne disposons pas encore des résistances de puissances et avons donc utilisé des résistances 1/4W hors de leur régime linéaire.

Les résultats se sont avérés satisfaisants, avec une tension maximale de 25V à 130km/h. De plus, le système a montré qu'il supportait bien la vitesse et n'entraînait pas l'apparition de vibrations importantes.

1.4.2 Extrapolation au vol de la fusée

Il est difficile d'extrapoler un résultat à une vitesse 7 fois supérieure mais non constante : en effet le moteur est assez massif et dispose donc d'une forte inertie, il est donc peu probable qu'il atteigne une vitesse équivalente à un 700km/h stabilisé. De plus, il ne faut pas négliger l'augmentation des frottements avec la vitesse.

Toutefois, nous avons pu remarquer que la tension évoluait peu en fonction de la charge résistive, et que la tension augmentait de moins en moins vite avec la vitesse. Nous avons donc estimé qu'au maximum, la tension ne devrait pas dépasser 50-60V quelle que soit la charge, et nous avons choisi d'utiliser 4 résistances de 220ohms en parallèle pour obtenir une puissance d'environ 55W. Obtenir une telle puissance pourrait à la fois être très intéressant comme source d'énergie de la fusée, tout en restant sans danger.

2 Expérience secondaire : tension subie par les sangles parachute

2.1 But de l'expérience

Cette expérience consiste à quantifier la force qui s'exerce sur la sangle du parachute. Il s'agira ici de mesurer non seulement la force s'exerçant sur le parachute lors du choc consécutif à son ouverture, mais aussi de réaliser des mesures lors de la phase de descente de la fusée. Etant donnée la violence du choc (de l'ordre de plusieurs milliers de Newtons), le point critique de l'expérience est la réalisation d'un capteur capable de mesurer une telle force et de la supporter.

2.2 Mise en œuvre

Après avoir étudiés plusieurs solutions techniques (à base de ressorts et de potentiomètres linéaires), nous avons finalement opté pour un système à base de jauges de contraintes, permettant de suivre la déformation d'un « mousqueton », aux formes étudiées pour répondre à notre étude.

2.2.1 Théorie des jauges de contraintes

Il s'agit du coeur du capteur. Ces pièces sont à la base de la mesure des déformations des matériaux, ou extensiométrie. La jauge de contrainte repose sur le principe d'un fil que l'on déforme. Le fil, très fin, est placé préférentiellement longitudinalement par rapport à la déformation. En agissant par traction ou compression sur le fil, celui-ci devient plus ou moins long par rapport à son état repos. Cette variation de longueur modifie la résistance électrique du fil. On mesure alors cette variation de résistance entre l'état repos et l'état sous contrainte.

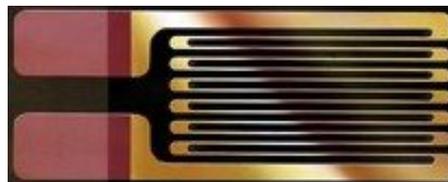


Figure 4 – Schéma d'une jauge de contrainte

Si on note $\varepsilon = \frac{\Delta \ell}{\ell}$ la déformation d'un matériau soumis à une contrainte, on observera une variation de résistance Δr de la jauge $\Delta r = k \frac{\Delta \ell}{\ell}$ où k est le facteur de jauge. De plus, on peut quantifier la force ayant conduit à la déformation :

$$F = E\varepsilon S$$

où S représente la surface de la section qui se déforme et E le module d'Young du matériau contraint.

2.2.2 La pièce mécanique

Réalisée en aluminium, les dimensions de la pièce ont été calculées pour qu'un choc de 3000 Newtons reste dans le domaine de fonctionnement linéaire des jauges. Agissant comme un mousqueton, elle permet une bonne intégration dans la fusée. A ses deux extrémités sont prévus des emplacements pour fixer des attaches permettant de la relier à la sangle.



Figure 5 – Jauges de contraintes collées sur la pièce mécanique subissant les efforts

2.2.3 Conditionnement du signal

Le problème qui apparaît lors du dimensionnement des jauges de contrainte est la très faible variation de résistance que l'on est censé mesurer. On va tout d'abord monter les jauges en pont de Wheatstone, qui sera déséquilibré lors de l'application d'une contrainte sur la pièce. Par ce biais, on pourra récupérer une tension exploitable.

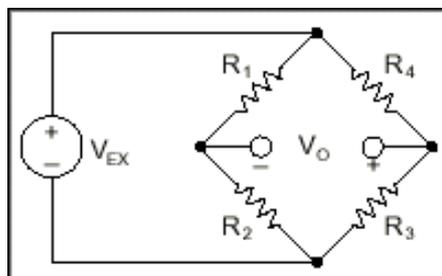


Figure 6 – Schéma d'un pont de Wheatstone (les résistances XXX sont celles des jauges)

La tension ainsi créée sera amplifiée (par un amplificateur type INA 118, permettant de forts gains), puis échantillonnée à l'aide d'un convertisseur analogique-numérique à deux voies, car deux mesures sont prévues. Une jauge se charge de la plage 1000-3000N et l'autre se charge de la plage 500-5000N. Ainsi, on a une mesure générale et une mesure plus précise. Cela permet en plus de rejeter les résultats incohérents.



B. Réalisations

1 Vue d'ensemble de la fusée expérimentale

1.1 Plan d'intégration

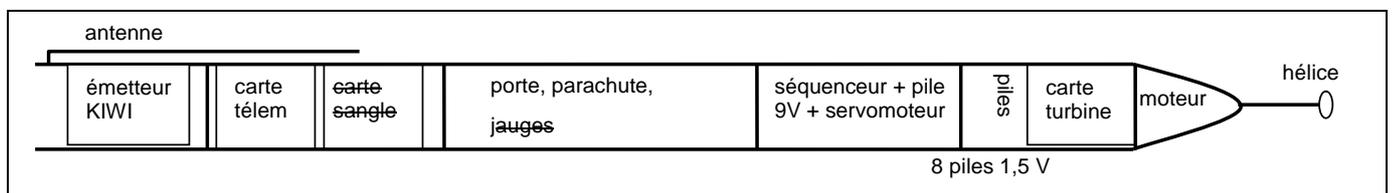


Figure 7 – Synoptique de la fusée expérimentale Saturne V

L'expérience secondaire a été abandonnée en fin de projet car la carte électronique associée n'était pas prête. La mesure de tension par jauges de contraintes nécessite une attention particulière pour le test des cartes et l'étalonnage des capteurs.

Le conditionnement du signal de l'expérience principe se fait au plus près de la génératrice. Les signaux sont transmis sous forme numérique à travers la fusée pour être traitée par la carte centrale d'acquisition qui effectue la mise en forme de la télémessure vers l'émetteur.

1.2 Structure mécanique

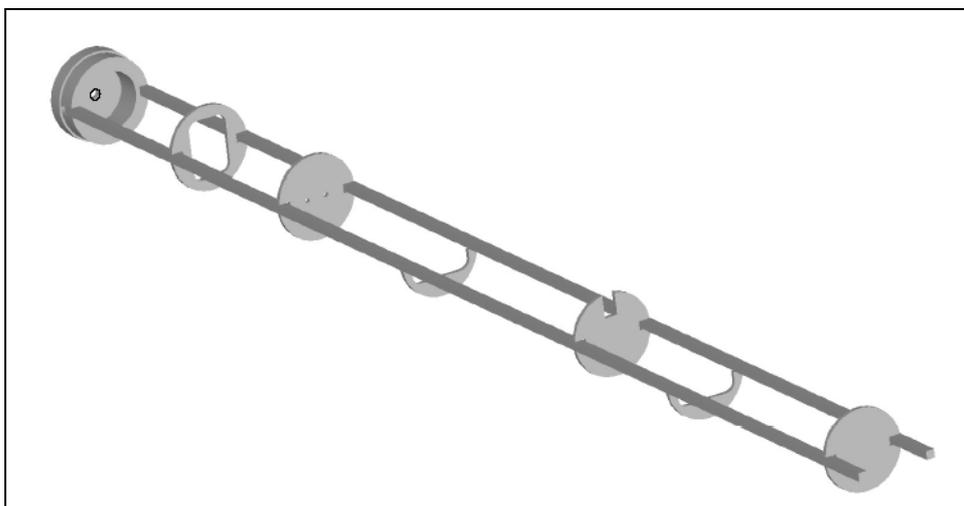


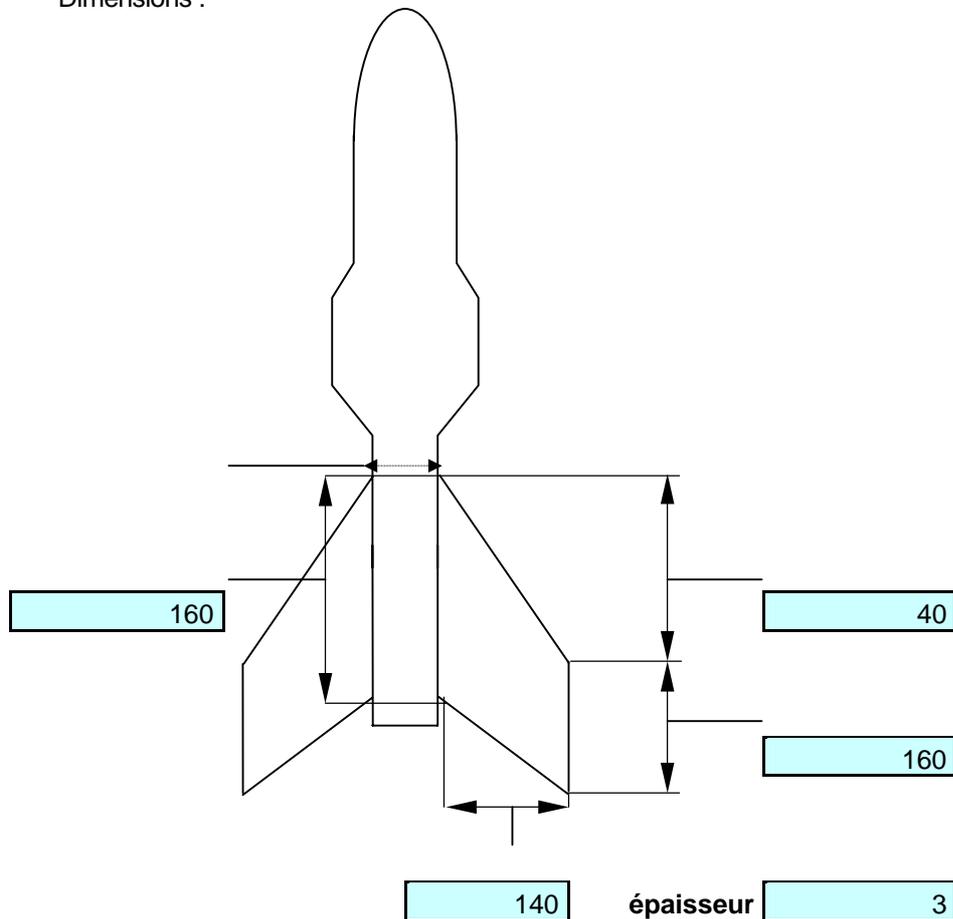
Figure 8- Structure interne de la fusée expérimentale Saturne V

Les schémas techniques de l'ensemble des pièces mécaniques réalisées sont donnés en annexe¹.

1.3 Ailerons

Les ailerons ont été taillés dans une plaque d'aluminium de 3mm d'épaisseur, et ont été fixés à la virole grâce à des équerres en aluminium et des vis de diamètre 3mm.

Dimensions :



Systemes électroniques

1.4 Carte centrale d'acquisition

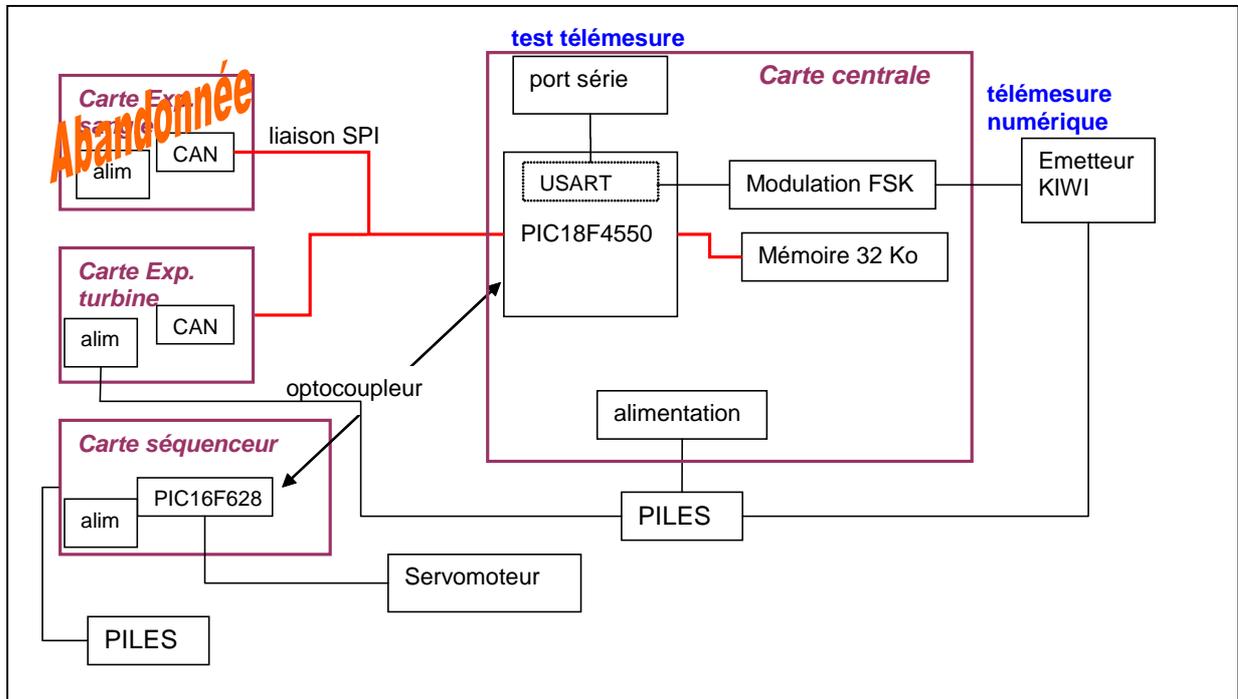


Figure 9 – Architecture des systemes électroniques

La carte centrale commande l'échantillonnage au niveau des convertisseurs analogiques-numériques, sauvegarde les données échantillonnées en mémoire et les envoie par télémesure numérique.

Les optocoupleurs entre la carte séquenceur et la carte centrale permettent d'avoir des informations sur les phases de vol (déclenchement de la minuterie, déclenchement de l'ordre d'ouverture de la porte...), tout en isolant les deux circuits électriques, conformément aux cahiers des charges. Ce système devait avant tout servir de synchronisation pour la mesure de la tension de la sangle parachute

1.5 Caractérisation de la chaîne de mesure

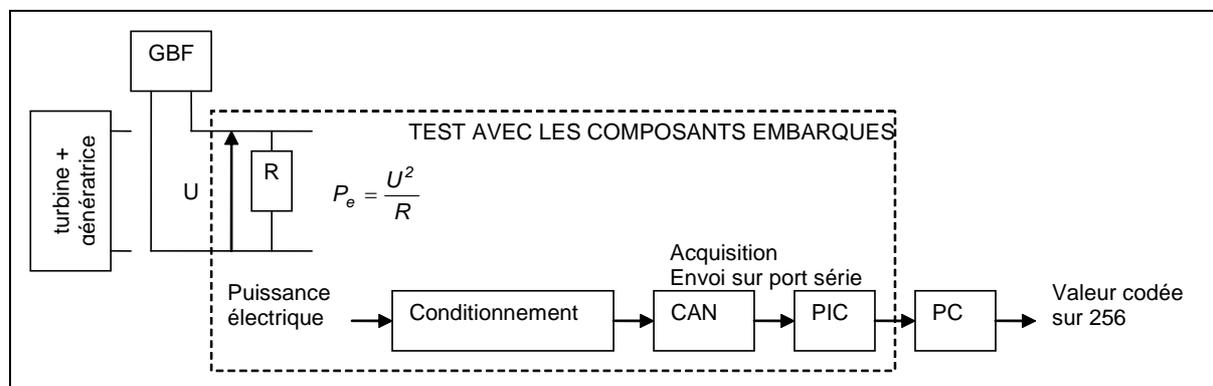


Figure 10 – Principe de l'étalonnage de la chaîne de mesure

Pour caractériser la chaîne de mesure de la puissance électrique fournie par la turbine, nous avons fait l'acquisition des données numériques, en remplaçant la génératrice par une tension connue aux bornes de résistances de charge (cf. Figure 10).

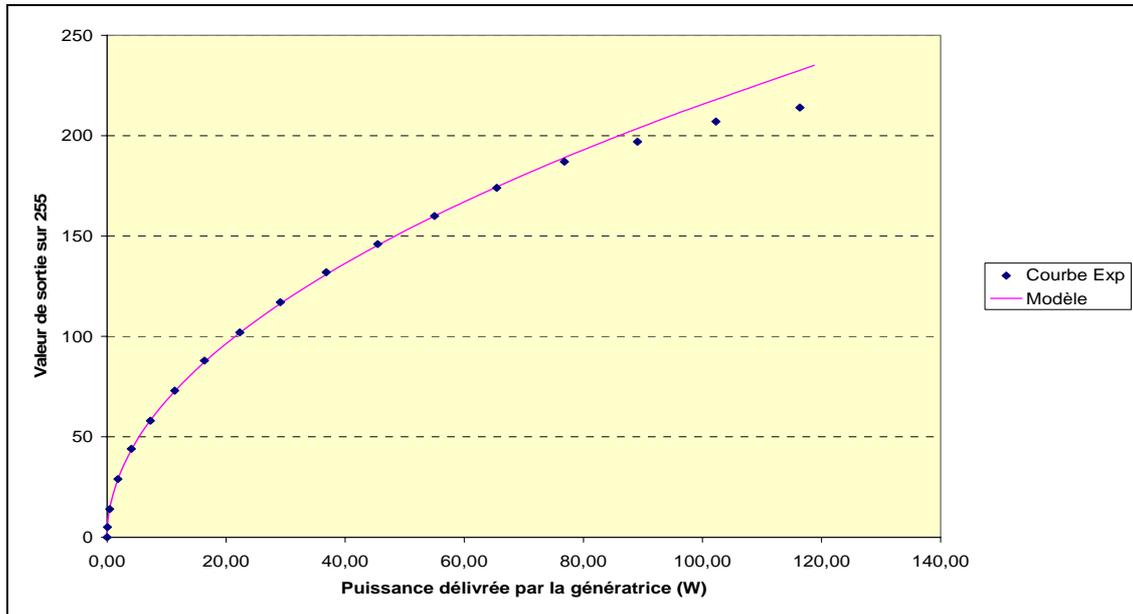


Figure 11 – Résultats de l'étalonnage

Le conditionnement avant l'échantillonnage sur 8 bits permet d'exploiter au moins la dynamique sur 256. Le modèle ajusté est adapté aux puissances inférieures à 80 W car on suppose que cette puissance ne sera pas atteinte pendant le vol. Il est tel qu'on retrouve la puissance à partir de la mesure, par la formule

$$Puissance = 2,15 \cdot 10^{-3} \text{Sortie}_{256}^2$$

Remarque : l'expérience réalisée n'intègre pas l'étage de télémessure. Néanmoins, les mesures avec et sans émetteur allumé à proximité sont cohérentes.

1.6 Télémessure numérique

Les données sont envoyées par télémessure FSK sous forme de trames à 4800 bauds. La modulation est effectuée sur la carte centrale d'acquisition avant d'être envoyée vers l'émetteur Kiwi. L'antenne est fixée via une prise BNC le long du corps de la fusée, recouverte de gaine thermo-rétractable, pour l'isoler du corps de la fusée qui sert de masse pour l'électronique.

octet 1	octet 2	octet 3	octet 4	octet 5	octet 6	octet 7	octet 8
Start	N°trame MSB	N°trame LSB	séquenceur	T1	0	0	Checksum

1 trame = 8 octets de 10 bits = 80 bits
4800 bauds ⇔ 60 trames / s au maximum

Figure 12 – Format de la trame SNR envoyée par télémessure FSK

- octet 1 : FF (synchronisation du début de trame)
- octet 2 : numéro de trame (MSB) modulo 255
- octet 3 : numéro de trame (LSB) modulo 255
- octet 4 : état du séquenceur (cf. Figure 13)

- octet 5 : mesure issue de l'expérience turbine T1
- octets 6 et 7 : inutilisés
- octet 8 : checksum (octet2+octet3+octet4+octet5)&(0111 1111)

	t_0	t_1	t_2	fin
Phase de vol	P0	P1	P2	
Etat séquenceur	00	01	11	

t_0 mise sous tension de la carte centrale
 t_1 décollage
 t_2 ouverture de la porte parachute (durée déterminée par le séquenceur)

Figure 13 – Définition de l'état du séquenceur

2 Système de récupération

2.1 Séquenceur

Les particularités du séquenceur mis en œuvre dans le projet pour l'ouverture de la porte parachute sont les suivantes :

- Mode manuel / mode automatique : En mode manuel, la minuterie ne peut pas se déclencher (mode utilisé pour les manipulations de la porte et du parachute) et le servomoteur peut être commandé par un interrupteur en position ouverte ou fermée. En mode automatique, la minuterie se déclenche sur l'accélération, grâce à un accélérocontact.
- Réglage digital de la durée de la minuterie : le séquenceur est équipé d'un microcontrôleur qui réalise un compteur à partir de son horloge interne. Le programme est unique et figé, quelle que soit la durée de la minuterie désirée, ce qui assure l'intégrité du système. Le réglage du temps d'ouverture se fait via un bornier qui permet de configurer la valeur binaire en demi secondes.

Le schéma électronique ainsi que le programme du microcontrôleur sont donnés en annexe.

2.2 Parachute

Le parachute est un modèle en croix, constitué de 12 suspentes rassemblées dans un émerillon. La partie opposée de l'émerillon est fixée au cours de la fusée. Un disque anti-torche permet de séparer les suspentes afin d'éviter qu'elles s'emmêlent.

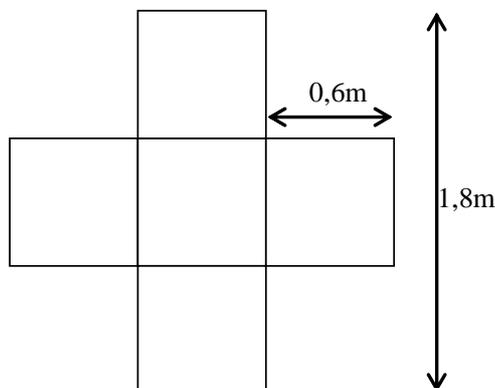


Figure 14 – Dimensions du parachute



C. Mise en œuvre du projet

1 Caractéristiques du vol

1.1 Estimation des paramètres de vol par Trajec

Simulation de vol

Site utilisé	70	80°
Temps de culmination	13,8 sec	14,7 sec
Altitude de culmination	911 M	1 037 M
Vitesse culmination	47 m/s	25 m/s
Portée balistique	1 298 M	726 M
Vitesse maximale	161 m/s	161 m/s
Accélération maximale	82 m/s ²	82 m/s ²
Cx utilisé	0,8	

1.2 Chronologie

Date	Durée	Action	Responsable	Lieu	Matériel
		Enlever la peau de la fusée Mettre des piles neuves Pistocoller le bloc de piles Installer l'antenne et la fixer à la glue Fermer la fusée avec les vis sur la plaque de poussée et sur la coiffe		R3	Piles neuves Pistocolle Vis fusée Tournevis Glue
M-70	10 min	Allumer l'interrupteur d'alimentation expériences →Vérifier LED verte Carte Acquisition →Vérifier LED rouge Carte Centrale →Vérifier bonne réception des trames Eteindre l'interrupteur. Allumer l'interrupteur d'alimentation séquenceur →Vérifier interrupteur mode « manuel » et interrupteur position « ouverte » →Vérifier petite LED verte Séquenceur →Vérifier LED rouge Carte Séquenceur	Florent	R3	Petit tournevis plat Lampe de poche

M-60	20 min	Mettre la coiffe. Plier le parachute avec le talc et l'insérer dans la case parachute. Basculer interrupteur Séquenceur en position « fermée ». Eteindre l'interrupteur d'alimentation séquenceur.	Florent + Aline Florent	R3 Tente club	Talc
	0 min	Emmener la fusée, la virole, le matériel (cf. liste) et la chronologie sur l'aire de lancement	Aline		La feuille
		Attente en tente club	Equipe		
M- 40	5 min	Descendre en zone rampe avec la fusée et la virole	Equipe		
M -35		Vérifier la compatibilité rampe	Florent	Rampe	
		Prendre une photo de nous	Accompagnateur	Rampe	
M -25		Mettre en place le propulseur Eriger la rampe Mettre la canne d'allumage	Pyrotechnicien	Aline en tente pyro, Florent en zone rampe	
		Demander l'autorisation d'émettre Allumer l'interrupteur Expériences Vérifier la réception télémétrie Allumer l'interrupteur Séquenceur →Vérifier petite LED verte Basculer interrupteur mode « auto ». →Vérifier LED verte Séquenceur Allumer les 2 interrupteurs pour les buzzers Evacuer la rampe	Florent		
M-5		Mettre en place la ligne de mise à feu Retour à la tente pyro	Pyrotechnicien		
T-10s		Décompte final	Tente Jupiter		
		Préparer prise de vue photo			
T=0		Appuyer sur le bouton de mise à feu	Florent		

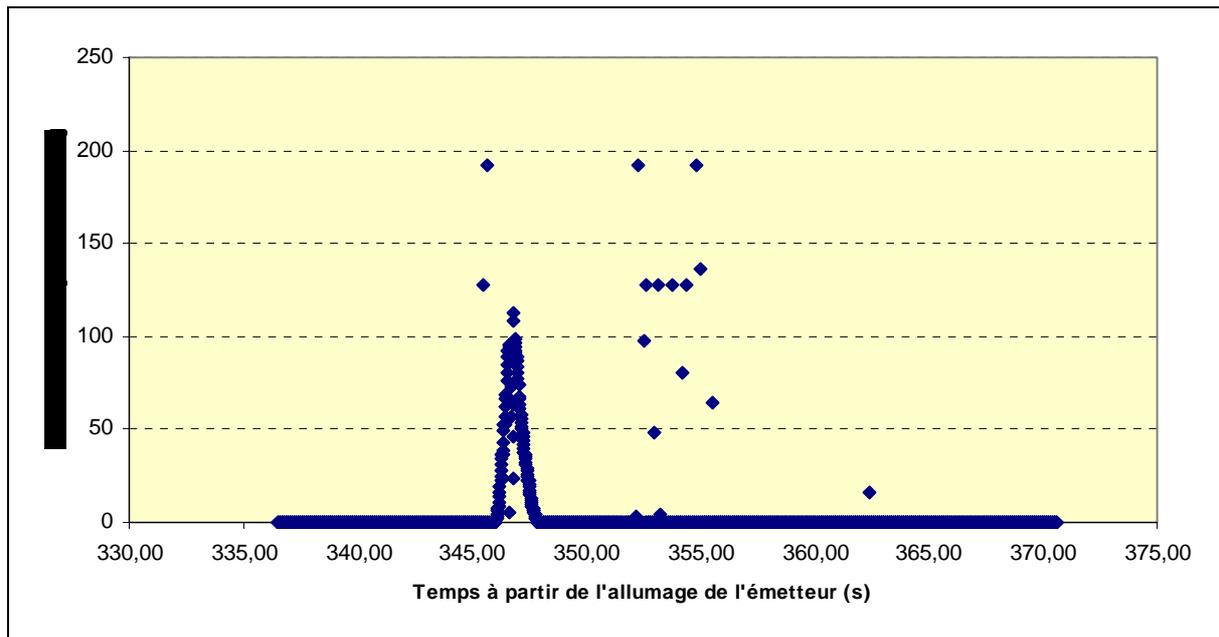
1.3 Résultat de vol

La fusée a fait un vol balistique. La porte latérale ne s'est pas ouverte. La fusée a été retrouvée avec la porte et le parachute plié. Il est difficile d'expliquer pourquoi la porte ne s'est pas ouverte. Le système mécanique d'ouverture a toujours permis une ouverture franche lors des contrôles au sol et le séquenceur a été testé de nombreuses fois avec toujours la même réponse.

De plus, les signaux transmis par le séquenceur via l'optocoupleur étaient toujours configurés pour prévenir la carte centrale de l'ouverture imminente (afin de permettre à la mesure de tension du câble parachute de démarrer) et ne permettaient pas d'obtenir un suivi précis de l'état du séquenceur. Enfin, nous avons eu des problèmes de fiabilité avec l'optocoupleur.

Les deux hypothèses envisagées sont principalement le non déclenchement de l'accélérocontact, même si cela nous semble peu probable, et un problème plus grave comme l'arrachement d'un composant ou des piles lors du décollage. Nous avons en effet remarqué que les supports des piles 9V étaient détruits après le crash, même celui situé à l'arrière dans la partie qui a peu souffert.

2 Résultats d'expérience



La Figure 15 montre les données obtenues par télémétrie. Le temps a été calculé à partir du numéro de trame sur deux octets de mesure. Seules les trames correspondant à des numéros de trame aberrants ont été supprimées (une dizaine sur 2035) sur cette figure. On peut remarquer que :

- La télémétrie a très bien marché pendant toute la durée du vol (34 s). Très peu de trames sont fausses (environ 3%).
- L'origine des temps correspond à moment où la télémétrie a été allumée et non au décollage. Les données sont représentées à partir de la date de décollage estimée par le fichier audio de télémétrie.
- La mesure de puissance n'a donnée des résultats que pendant les toutes premières secondes de vol, ce qui laisse supposer que les pales de l'hélice se sont arrachées lorsque la fusée a atteint sa vitesse maximale, environs 2 secondes et demi après le décollage. Nous pensons que seules les pales ont cédé car nous avons retrouvé le pas de vis de l'hélice au bout de l'arbre après le crash. Dans cette hypothèse, la mesure est continue car l'arbre a continué à tourner pendant quelques secondes en raison de son inertie, avant de s'immobiliser définitivement.

En prenant pour hypothèse que la date de décollage est bien à 336 s et en utilisant la fonction de conversion établie au cours de l'étalonnage, on obtient les résultats définitifs de la Figure 16.

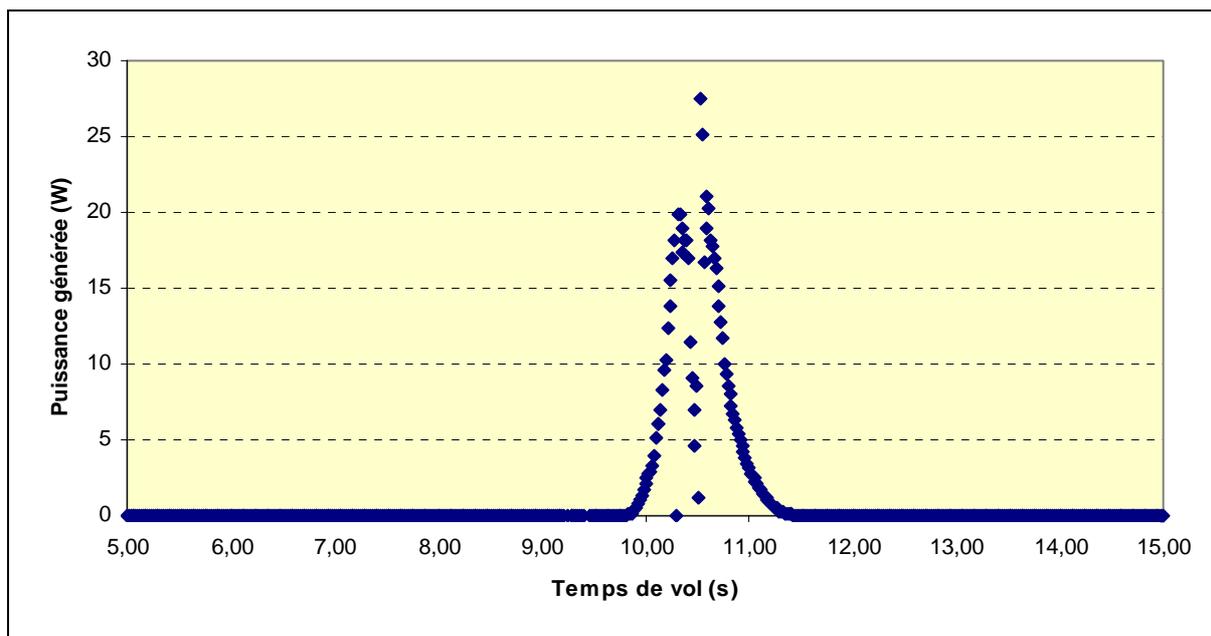


Figure 16 – Exploitation des résultats de télémessure

En extrapolant la courbe de mesure, on peut conclure que notre système avec hélice a permis de générer au bout de 10 secondes de vol des puissances allant jusqu'à 30 W. Nous ne savons pas quelle énergie totale nous aurions pu générer si l'hélice n'avait pas été détruite, mais la forte inertie du moteur et le fait que la puissance de 25W corresponde à nos essais à 130km/h nous permet de penser que notre génératrice n'avait pas encore atteint sa puissance maximale.

3 Bilan du projet

3.1 Difficultés rencontrées

3.1.1 Aspect mécanique

Les ailerons ont dû être retailés à plusieurs reprises pendant la campagne de lancement pour satisfaire les contraintes de stabilité. La première fois, il a fallu réduire leur dimension car nous avions omis de prendre en compte la valeur maximale autorisée pour le produit marge statique $\times C_n$, et la deuxième fois nous avons dû les placer plus haut sur la virole pour permettre un accès aisé au pyrotechnicien.

3.1.2 Aspect électronique

Nous n'avons pas réussi à mettre au point l'écriture et la relecture de données en mémoire EEPROM par le protocole SPI. Nous ne savons pas s'il s'agit d'un problème logiciel ou matériel. Plusieurs tests ont été effectués avec le matériel embarqué, en modifiant plusieurs fois le programme, sans succès. Nous avons dû abandonner ce système d'acquisition de données expérimentales.

3.2 Améliorations à apporter dans les futurs projets de fusée expérimentale

3.2.1 Finitions mécaniques

Les interrupteurs à actionner étaient situés sur des cartes électroniques fixées sur la structure interne. Ils n'étaient accessibles que par quelques trous dans la peau de la fusée via un petit tournevis, ce qui a compliqué sérieusement la mise en œuvre du projet en zone rampe. Il faut repenser l'intégration des systèmes électroniques pour rendre apparents les interrupteurs à actionner.

3.2.2 Mise en œuvre de l'expérience secondaire

L'expérience de mesure de la force subie par les sangles sur parachute nous semble une idée intéressante à développer. La plaque de déformation a été conçue ainsi que la carte électronique pour le conditionnement du signal au plus près des jauges de contraintes. Le principe d'acquisition des données a également été mis au point : il s'agit d'échantillonner à une fréquence plus grande que celle disponible pour la télémétrie, d'effectuer une moyenne par la carte centrale pendant les phases où les mesures nous intéressent le moins (avant ouverture et à la fin du vol) et de sauvegarder toutes les mesures pendant les premières secondes après ouverture. Pour transmettre toutes les données par télémétrie, les données sont relues en fin de vol dans la mémoire embarquée et envoyées en différé vers l'émetteur.

3.2.3 Mise en œuvre de la mesure de l'état du séquenceur

Relier des signaux du séquenceur par optocoupleur à la carte centrale est une idée à conserver et à mettre en œuvre car elle permet de mieux diagnostiquer les problèmes en cas de vol balistique, mais aussi de mieux analyser les résultats expérimentaux en fonction des phases de vol, en identifiant par exemple précisément la date de décollage sur les données obtenues par télémétrie.

	D. Annexes	
	Responsable	Florent Bouchet

1 Codes microcontrôleur

1.1 Code du séquenceur (PIC16F)

Voir annexe

1.2 Code de la carte centrale d'acquisition (PIC18F)

1.2.1 Notes pour la programmation

Logiciels nécessaires à la programmation PIC :

- MPLAB avec le compilateur C18 (cocher toutes les cases lors de l'installation pour définir les PATHs)
- USB bootloader Setup :
http://www.microchip.com/stellent/groups/picmicro_sg/documents/devicedoc/en021827.exe

Mise en œuvre pour la programmation :

- Compiler le projet sous MPLAB
- Lancer MCHPFSUSB\Pc\Pdfsusb\PDFSUSB.exe
- Connecter par le port USB la platine de programmation avec le pic dessus, attention au sens ! (schéma : <http://pic18fusb.online.fr/wiki/wikka.php?wakka=UsbBootload>)
- Faire un reset du PIC en mode Bootloader : [appuyer sur le bouton reset (/MCLR) ; appuyer sur le bouton de boot (RB4) ; relâcher le bouton reset ; relâcher le bouton de boot]
- Ding dong : Le PC reconnaît un périphérique USB
- Dans PDFSUSB, sélectionner le programmeur qui vient de s'ajouter au menu déroulant.
- Faire un « Load Hex » du fichier compilé
- Faire un « Program device »
- Un fois que la programmation est terminée (message après quelques secondes) appuyer sur le bouton reset.
- Ding dong : Le PC reconnaît que le périphérique a été éteint.
- Le pic est bon pour être mis sur la carte centrale.

1.2.2 Logiciel embarqué

La gestion de la mémoire par le microcontrôleur via la liaison SPI n'a pas pu être au point avant le vol. Seuls le formatage et l'envoi des données au modulateur FSK sont correctement réalisés par le programme.

```

/*****
/*-----
/ CODE DE LA CARTE CENTRALE D'ACQUISITION : TELEMESURE ET SAUVEGARDE MEMOIRE
/*-----
/*****

/** I N C L U D E S *****/
#include <pl8f4550.h>
#include <delays.h>

/** V A R I A B L E S *****/
#pragma udata

/** P R I V A T E P R O T O T Y P E S *****/

/** V E C T O R R E M A P P I N G *****/

extern void _startup (void); // See c018i.c in your C18 compiler dir
#pragma code _RESET_INTERRUPT_VECTOR = 0x000800
void _reset (void)
{
    _asm goto _startup _endasm
}
#pragma code

#pragma code _HIGH_INTERRUPT_VECTOR = 0x000808
void _high_ISR (void)
{
    ;
}

#pragma code _LOW_INTERRUPT_VECTOR = 0x000818
void _low_ISR (void)
{
    ;
}
#pragma code

#define ETAT_SEQ_H PORTBbits.RB5
#define ETAT_SEQ_B PORTBbits.RB4
#define ETAT_SEQ (ETAT_SEQ_H*2+ETAT_SEQ_B)
#define TOCON_0b10000000 // Timer0 overflow à 45,78Hz avec Fosc=24Mhz
#define CAN_turbine 0
#define CAN_sangle 1
#define CS_memoire LATDbits.LATD7
#define CS_sangle LATDbits.LATD5
#define CS_turbine LATDbits.LATD6
#define READ 0b00000011
#define WREN 0b00000110
#define WRITE 0b00000010
#define TAILLE_MEM 256 //32768 // en octets

typedef struct {unsigned char debut; unsigned char numeroH; unsigned char numeroL;
    unsigned char type; unsigned char donnee1; unsigned char donnee2; unsigned char
    donnee3; unsigned char checksum;} TRAME;
TRAME trame;
unsigned char adresseH_mem=0;
unsigned char adresseL_mem=0; //adresse de la mémoire où doit être écrit la prochaine
    donnée
unsigned char adresseE_mem_lecture=0;
unsigned char temp;

void envoi_portserie(unsigned char c);

//-----
// INIT SPI : initialisation des E/S SPI
//-----

void init_SPI(void){
    TRISDbits.TRISD5 = 0; //CS_mémoire
    TRISDbits.TRISD6 = 0;
    TRISDbits.TRISD7 = 0;
    TRISCbits.TRISC7 = 0; //SDO

```

```

    TRISBbits.TRISB0 = 1; //SDI
    TRISBbits.TRISB1 = 0; //SCK
    CS_memoire=1;
    CS_sangle=1;
    CS_turbine=1;
    SSPCON1 = 0b00100010; //envoi à Fosc/64 ,vitesse OK ( <2.8Mhz pour le MCP3001, <10Mhz
pour le 25AA256 )
    SSPSTATbits.SMP=0;
    SSPSTATbits.CKE=0;
return;
}

//-----
// INIT SERIE : initialisation du port série
//-----

void init_serie(void)
{
    TRISCbits.TRISC6=1;
    //TRISCbits.TRISC7=1;
    //Vitesse d'émission ~4800bauds ( Fosc=24Mhz)
    TXSTAbits.BRGH=1;
    SPBRGH=0x04;
    SPBRG=0xE1;
    BAUDCONbits.BRG16=1;
    //Mode asynchrone
    TXSTAbits.SYNC=0;
    RCSTAbits.SPEN=1;
    //Emission sur 8bits
    TXSTAbits.TX9=0;
    //Activation du port serie
    TXSTAbits.TXEN=1;
    envoi_portserie('I');
    envoi_portserie('N');
    envoi_portserie('I');
    envoi_portserie('T');
return;
}

//-----
// ETAT_SEQUENCEUR : lit RB4/RB5
//-----
unsigned char etat_sequenceur(void) {
    return ETAT_SEQ;
}

//-----
//INCREMENTE_NTRAME (sur 16 bits)
//-----
void incremente_ntrame(void){
    trame.numeroL++;
    if (trame.numeroL==0xFF)
    {
        trame.numeroL=0;
        trame.numeroH++;
        if(trame.numeroH==0xFF){trame.numeroH=0;}
    }
return;
}

//-----
// ENVOI_SPI : envoie un octet sur SDO
//-----
//L'écriture d'un octet dans SSPBUF provoque l'envoi sur SDO bit par bit et parallèlement
la réception d'un octet via SDI qui s'écrit bit par bit dans SSPBUF au fur et à mesure
qu'il se vide de l'autre côté.
//Avant de commencer un nouvel envoi il faut s'assurer 1-que l'octet a été entièrement
envoyé (flag SSPSTATbits.BF) 2-que l'octet reçu a été lu (purge).
//L'écriture nécessite donc une lecture inutile.
void envoi_SPI(unsigned char c){
    temp=SSPBUF; //lecture purge
    SSPBUF=c;
    while(SSPSTATbits.BF==0) {} // réception paralelle à l'émission
}

```

```

return;
}

//-----
// LECT_CAN : renvoie la donnée du CAN sur 16 bits
//-----
unsigned char lect_CAN(void){ // CAN MCP3001
    unsigned char val_h;
    unsigned char val_b;
    unsigned char mesure;

    LATD &= 0b10111111; //mise à zéro du CS du CAN
    // CS_turbine = 0;
    envoi_SPI(0); // juste pour déclencher le CLK
    val_h = (SSPBUF<<3)&(0b11111000); //récupère les 5 bits de poids forts de la
    conversion (B9 B8 B7 B6 B5), BF automatiquement RAZ
    envoi_SPI(0); // juste pour déclencher le CLK
    val_b = (SSPBUF>>5)&(0b00000111); //récupère B4 à B2

    LATD |= 0b01000000; //déactivation du CS du CAN
    // CS_turbine = 1;
    mesure = val_h + val_b;
    if (mesure==0xFF) {
        mesure=0xFE;
    }

    return mesure;
}

//-----
// ECRIT_TRAME_MEM : écrit tous les octets de la trame dans la mémoire
//-----
void ecrit_trame_mem(unsigned char adresseH, unsigned char adresseL){
    // LATD &= 0b01111111; //mise à 0 du CS
    CS_memoire = 0;
    envoi_SPI(WREN);
    // LATD |= 0b10000000; //mise à 1 du CS
    CS_memoire = 1;
    Delay10TCY();
    //write sequence
    // LATD &= 0b01111111; //mise à 0 du CS
    CS_memoire = 0;
    envoi_SPI(WRITE);
    envoi_SPI(adresseH);
    envoi_SPI(adresseL);

    envoi_SPI(trame.debut);
    envoi_SPI(trame.numeroH);
    envoi_SPI(trame.numeroL);
    envoi_SPI(trame.type);
    envoi_SPI(trame.donnee1);
    envoi_SPI(trame.donnee2);
    envoi_SPI(trame.donnee3);
    envoi_SPI(trame.checksum);

    // LATD |= 0b10000000; //mise à 1 du CS
    CS_memoire = 1;
    return;
}

//-----
// INCREMENTE_ADRESSEE_MEM : ajoute 8 à l'adresse 15 bits, sauf pour 0x7FF8 (dernière
// adresse trame)
//-----
void incremente_adresseE_mem(void){
    if (adresseL_mem==0xFF8){
        if(adresseH_mem==0x7B){
            return;
        }
        adresseL_mem=0;
        adresseH_mem++;
        return;
    }
    adresseL_mem += 8;
    return;
}

```

```

}
//-----
// ENVOI PORTSERIE : envoie un octet sur TXREG
//-----

void envoi_portserie(unsigned char c){
while(!TXSTAbits.TRMT){}
TXREG=c;
return;}

//-----
// ENVOI_TELEM : envoie tous les octets de la trame sur TXREG pour la FSK
//-----
void envoi telem(void){
    envoi_portserie(trame.debut);
    envoi_portserie(trame.numeroH);
    envoi_portserie(trame.numeroL);
    envoi_portserie(trame.type);
    envoi_portserie(trame.donnee1);
    envoi_portserie(trame.donnee2);
    envoi_portserie(trame.donnee3);
    envoi_portserie(trame.checksum);
return;
}

//-----
// LECT OCTET MEM : lit un octet en mémoire
//-----
//Le raisonnement est analogue pour la lecture : elle nécessite une écriture inutile qui
    déclenche la CLK et qui provoquera la réception d'un octet bit à bit sur SDI à fur et
    à mesure qu'on envoie notre donnée sur SDO.
//Dans le chronogramme de la datasheet mémoire, on voit que la donnée arrive après l
    'adresse et non pas en parallèle, d'où la nécessité d'un nouvel envoi.
unsigned char lect_octet_mem(unsigned char adH, unsigned char adL){
    unsigned char lecture;
// LATD &= 0b01111111; //mise à 0 du CS
    CS_memoire = 0;
    envoi_SPI(READ);
    envoi_SPI(adH);
    envoi_SPI(adL);
    envoi_SPI(0xAA); //caractère inutile. A la sortie de la fonction, notre donnée est
    dans SSPBUF (puisque'on fait une attente dans envoi_SPI)
// LATD |= 0b10000000; //mise à 1 du CS
    CS_memoire = 1;
    lecture=SSPBUF;
    return lecture;
}

//-----
// MAIN : programme principal
// Si RA0 = 0, lance programme télém (écriture en mémoire et envoi sur port série)
// Si RA1 = 1, lance programme relecture (lecture mémoire à partir de 0)
//-----
void main (){
    unsigned char cnt;
// if (PORTAbits.RA0==0){
    trame.debut=0xFF;
    trame.numeroL=0;
    trame.numeroH=0;
    TOCON=TOCON_;
    init_SPI();
    init_serie();
    while(1){
        trame.type=etat_sequenceur();
//        trame.donnee1=lect_CAN(CAN_turbine);
        trame.donnee1=lect_CAN();
        trame.donnee2=0;
//        trame.donnee3=lect_mem(adresseE_mem-7); // (test) relire le numéro précédent
        (test)
        trame.donnee3=0xAA;
        trame.checksum=trame.numeroL+trame.numeroH+trame.donnee1+trame.donnee2+trame.
        donnee3;
        trame.checksum &= 0b01111111; // ne jamais avoir 0xFF

```

```

        ecrit_trame_mem(adresseH_mem, adresseL_mem);
        incremente_adresseE_mem();
        envoi_telem();
    //    }

    while (INTCONbits.TMR0IF==0) {}
    INTCONbits.TMR0IF=0;
    incremente_ntrame();
}

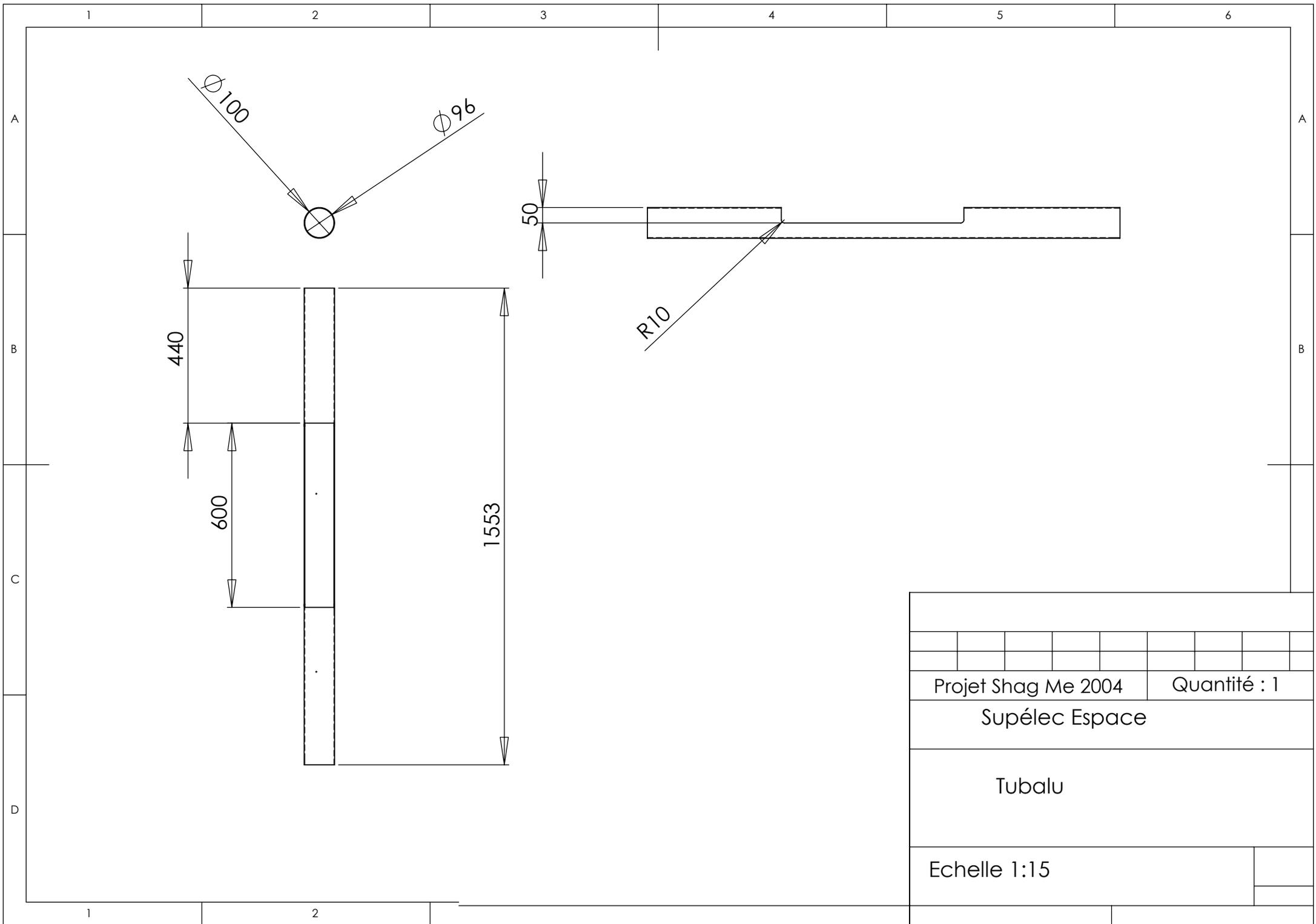
// else{
//     init_SPI();
//     for(cnt=0;cnt<16;cnt++){
//         lect_octet_mem(0,cnt);
//     }
// }
}

```

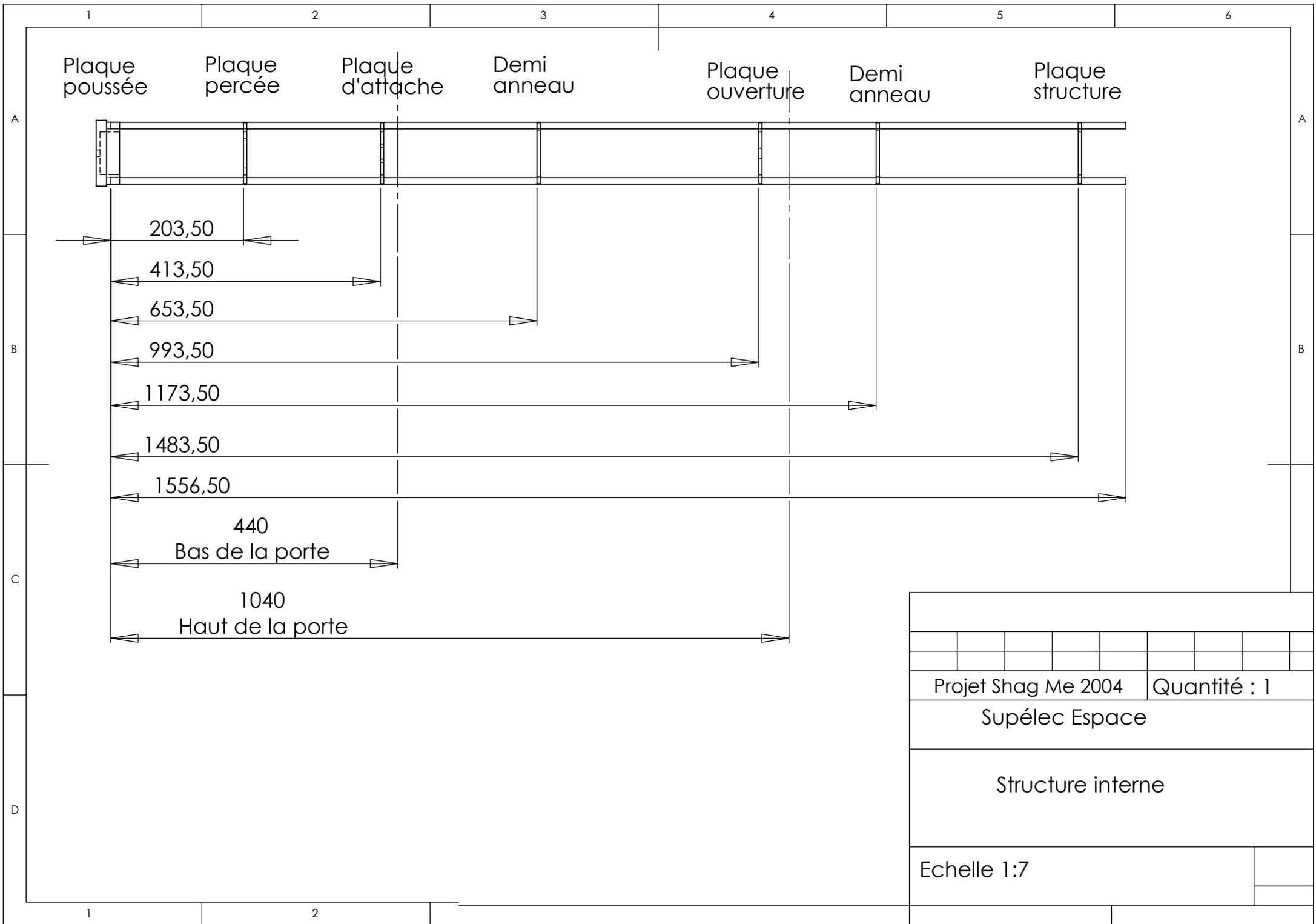
2 Plans mécaniques

Le dossier comprend les dessins techniques des pièces mécaniques suivantes :

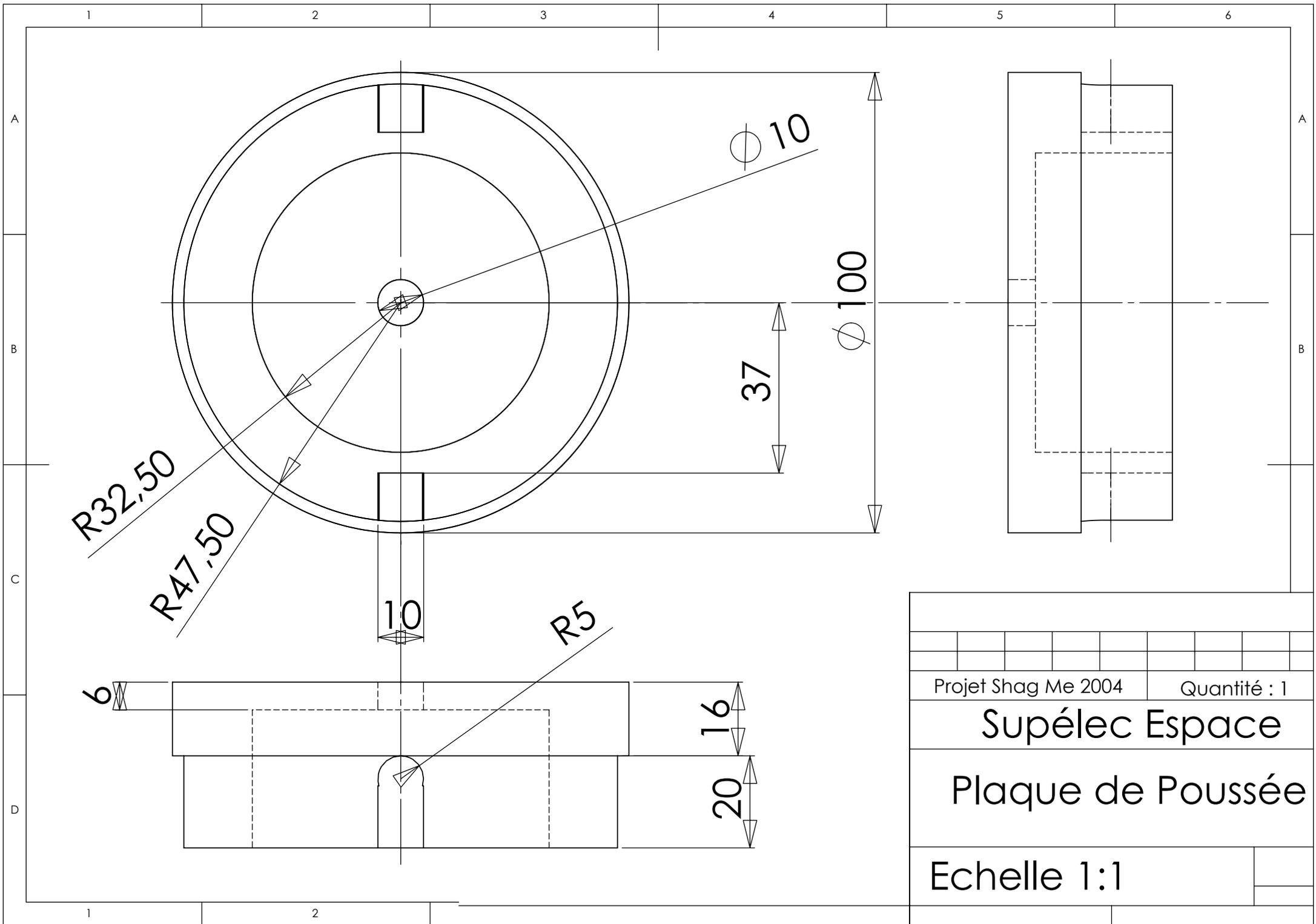
- Tube aluminium (corps de la fusée)
- Structure interne (décrivant la position de toutes les plaques de structure)
- Plaque de poussée
- Plaque de structure percée
- Plaque de structure pour l'attache du parachute
- Demi-anneau
- Plaque de structure pour l'ouverture / fermeture de la porte
- Dernière plaque de structure
- Porte



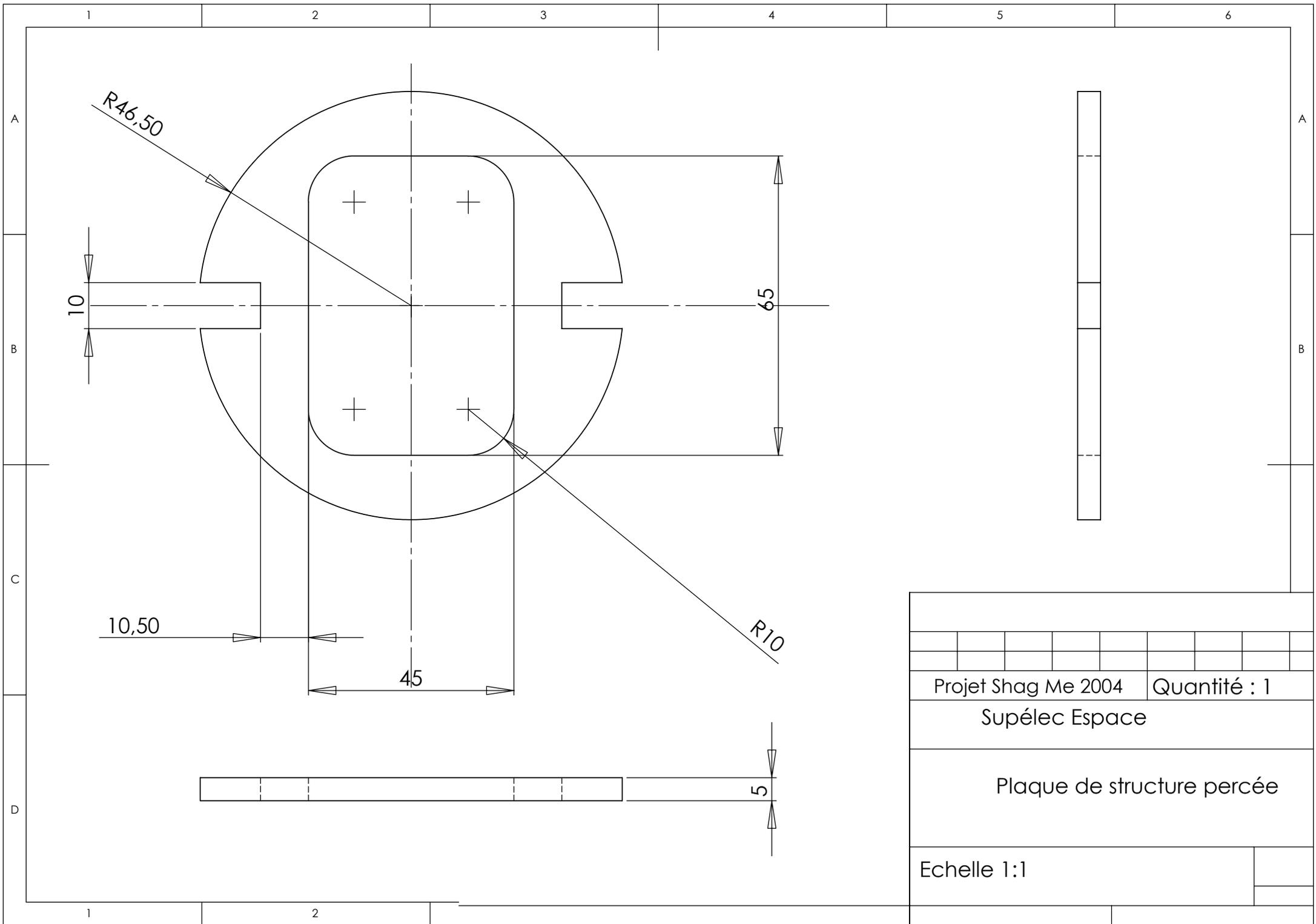
Projet Shag Me 2004					Quantité : 1				
Supélec Espace									
Tubalu									
Echelle 1:15									



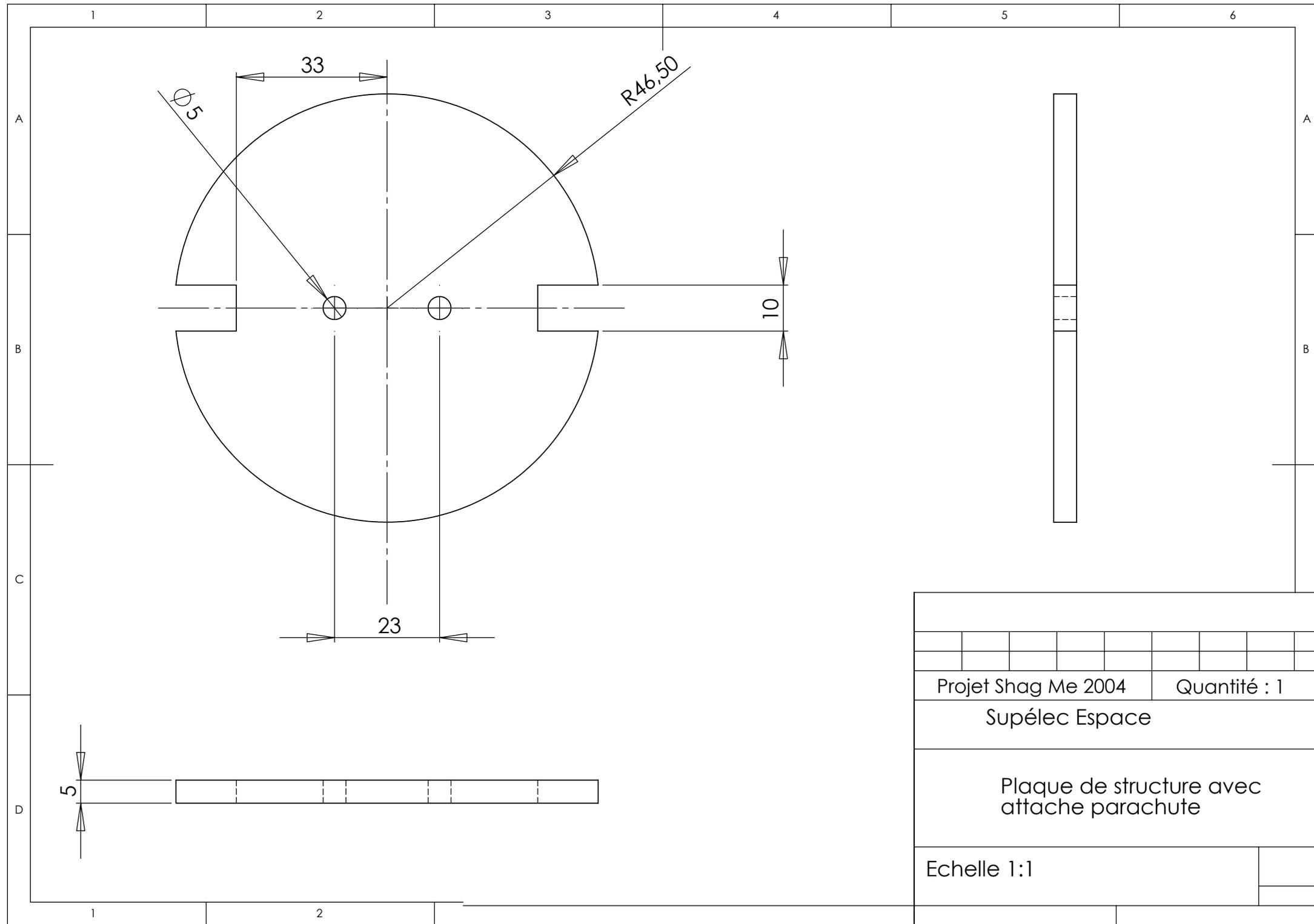
Projet Shag Me 2004						Quantité : 1			
Supélec Espace									
Structure interne									
Echelle 1:7									

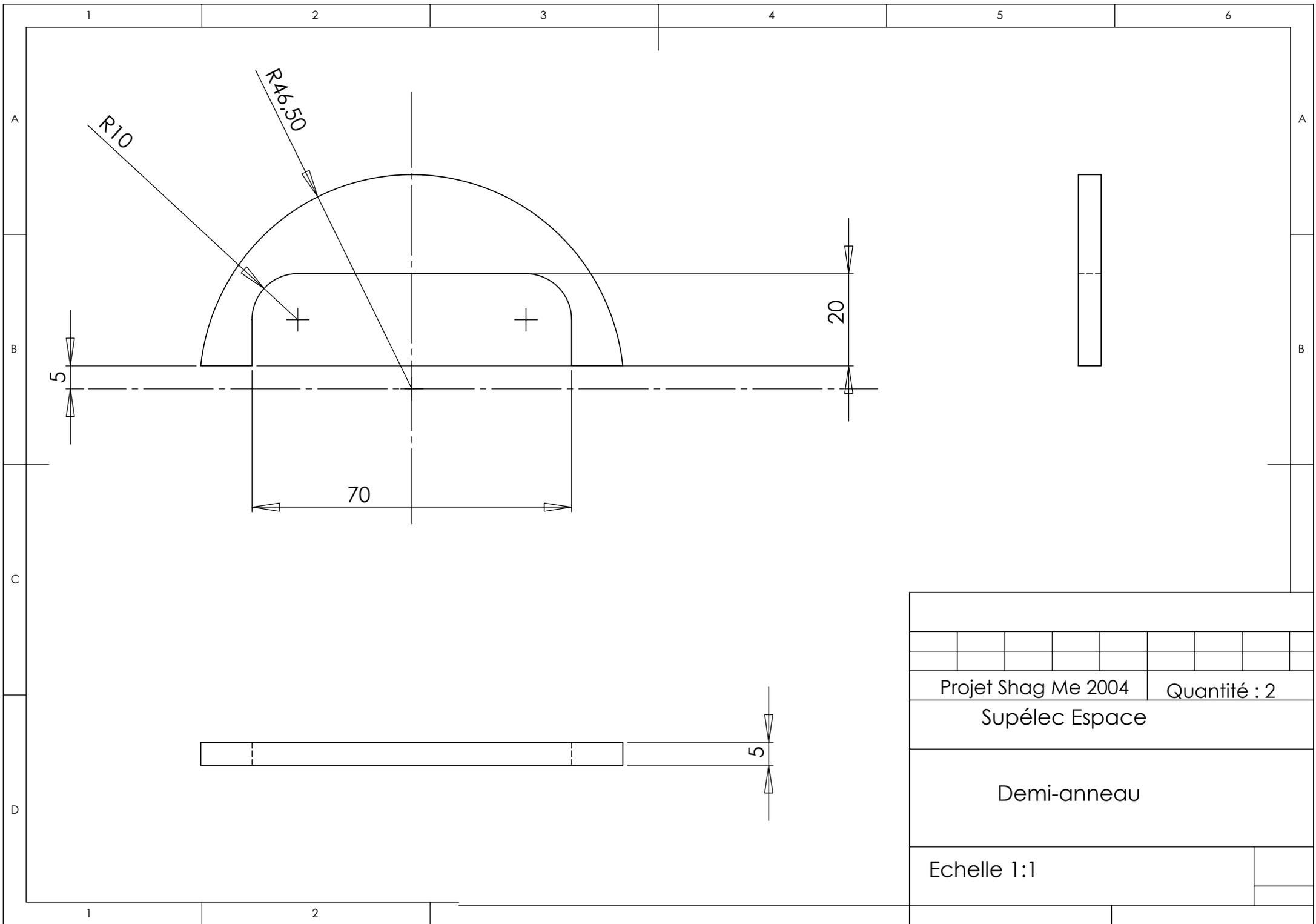


Projet Shag Me 2004					Quantité : 1				
Supélec Espace									
Plaque de Poussée									
Echelle 1:1									

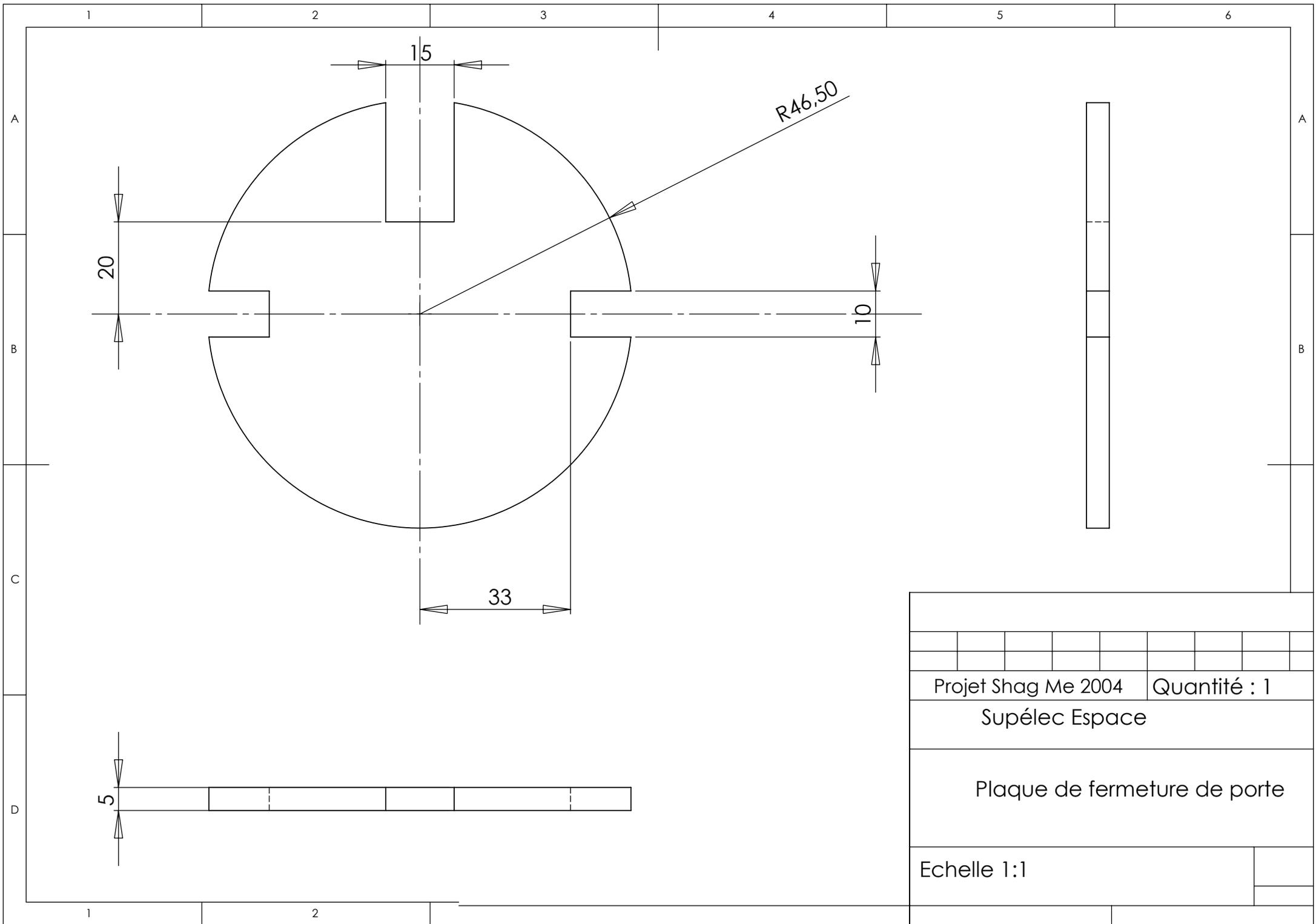


Projet Shag Me 2004					Quantité : 1				
Supélec Espace									
Plaque de structure percée									
Echelle 1:1									

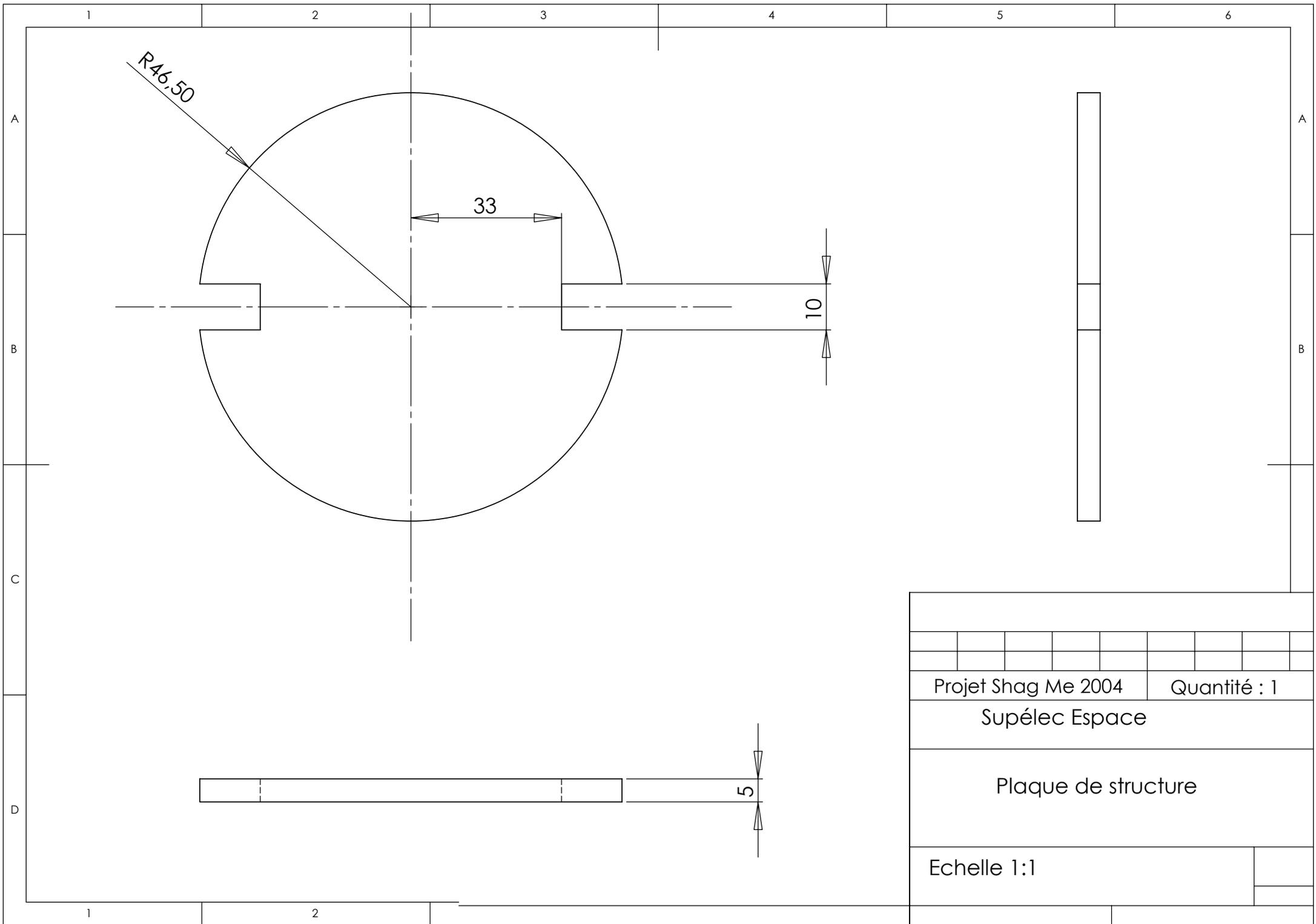




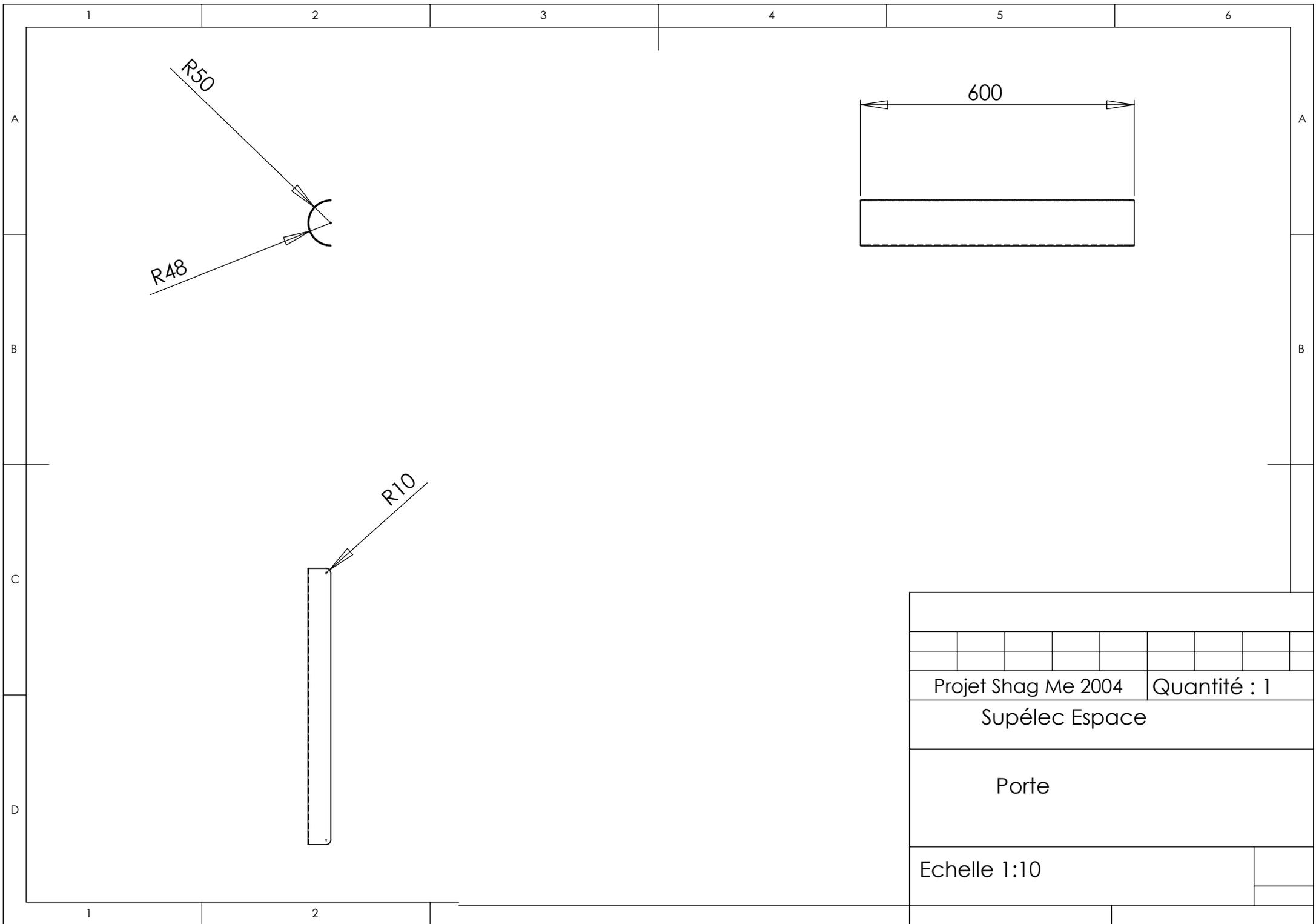
Projet Shag Me 2004					Quantité : 2				
Supélec Espace									
Demi-anneau									
Echelle 1:1									



Projet Shag Me 2004					Quantité : 1				
Supélec Espace									
Plaque de fermeture de porte									
Echelle 1:1									



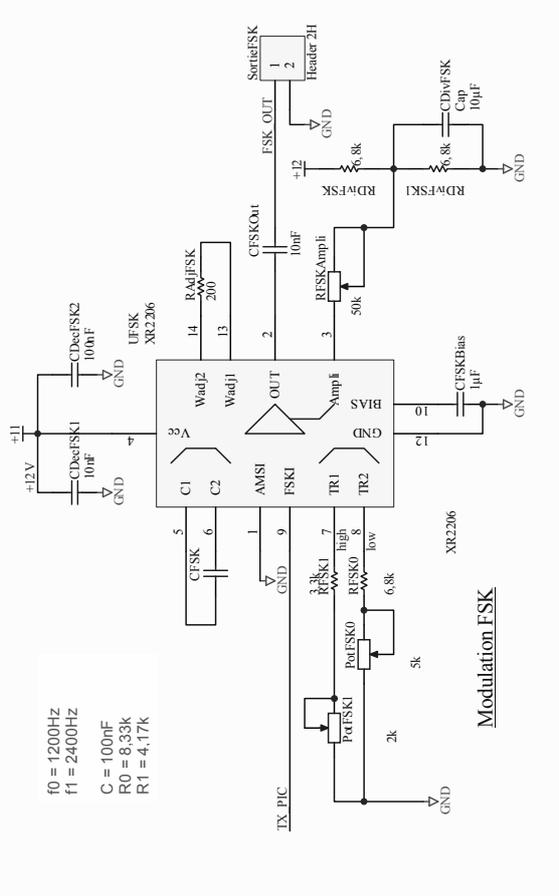
Projet Shag Me 2004					Quantité : 1				
Supélec Espace									
Plaque de structure									
Echelle 1:1									



Projet Shag Me 2004					Quantité : 1				
Supélec Espace									
Porte									
Echelle 1:10									

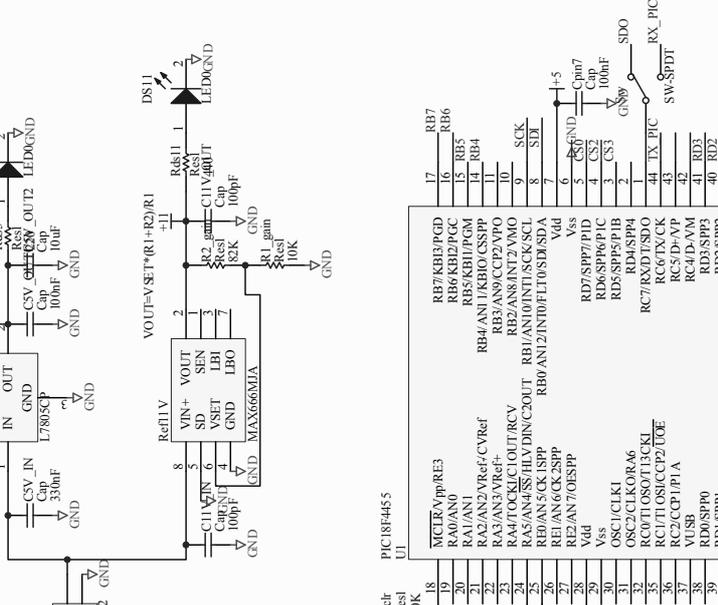
3 Schémas électroniques

- Carte centrale d'acquisition
- Séquenceur



$f_0 = 1200\text{Hz}$
 $f_1 = 2400\text{Hz}$
 $C = 100\text{nF}$
 $R_0 = 8,33\text{k}$
 $R_1 = 4,17\text{k}$

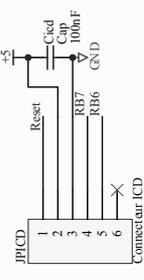
Modulation FSK



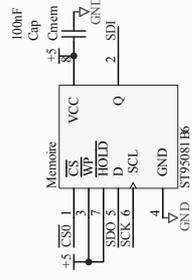
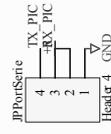
Alimentation

CS0 : memoire
 CS2 : exp turbine
 CS3 : exp single
 RA1 : séquenceur
 RA5 : séquenceur

Interface programmation



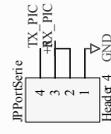
Interface port série



Expériences

configuration TX et RX séparées = utilisation des fonctions USART
 sinon configuration série des pins RC6, RC7

Interface port série



Title	
Size	Number
B	
Date:	Sheet of
File:	Revision
C:\mp\Carte acquisition SatV v1.Sch\Babawon.Bx	

Séquenceur
 Header 3

Séquenceur
 Header 4

CARTE SEQUENCEUR

