

Borne

D'auto-partage

TPE 2010-2011

Émilie CANTERO

Jean CLAISSE

Thibaud LEMAIRE

Comment profiter des libertés permises par la voiture sans pour autant en subir les contraintes actuelles ?



Nous tenons particulièrement à remercier nos professeurs, M. Ryaux et M. Barthélémy pour leur aide précieuse.

L'intégralité de cette étude est disponible sur notre site internet : <http://tpe.tibol.fr>

Table des matières

PARTIE A : Étude préliminaire.....	5
I Présentation de l'auto-partage.....	5
I.1 Qu'est ce que l'auto-partage ?.....	5
I.2 Comment ça marche ?.....	5
I.3 Un transport écocitoyen et écologique	5
II Cahier des charges fonctionnel.....	6
II.1 Expression du besoin.....	6
II.1.a Point de vue utilisateur.....	6
II.1.b Point de vue concepteur.....	6
II.2 Justification du besoin.....	6
II.3 Objectifs et contraintes.....	6
II.3.a Caractéristiques de la borne d'auto partage.....	6
II.3.b Environnement de la borne d'auto partage.....	6
II.3.c Serveur.....	7
II.4 Bête à cornes.....	7
II.5 Diagramme pieuvre et tableau des fonctions.....	8
II.6 Diagramme FAST.....	9
II.7 Étude SADT.....	9
II.8 Organigramme du software.....	10
II.9 Organisation pédagogique.....	10
II.9.a Constitution du groupe :.....	10
II.9.b Répartition des tâches (Diagramme de GANTT).....	10
PARTIE B : Étude mécanique.....	11
I Étude du mécanisme	11
I.1 Principe du disque.....	11
I.2 Étude cinématique.....	11
I.2.a Classes d'équivalences et tableau de liaisons.....	11
I.2.b Schéma cinématique.....	12
I.2.c Nouveau tableau des liaisons.....	13
I.2.d Graphe des liaisons.....	14
I.3 Étude de conception de la liaison pivot	14
I.3.a Réalisation de la liaison rotule sous Inventor.....	15
I.3.b Réalisation de la liaison sphère cylindre sous Inventor.....	15
I.3.c Réalisation des deux liaisons sur la maquette réelle.....	16
II Mise en rotation du disque.....	16
II.1 Les types de moteurs.....	16
II.1.a Le moteur pas à pas.....	16
II.1.b Les moteurs à courant continu..	17
II.1.c Moteur C.C. Brushless.....	18
II.1.d Choix d'une solution technique	19
II.2 Entrainement de l'axe tournant : courroie ou engrenage ?	19
II.2.a Transmission par poulie-courroie	19
II.2.b Transmission par train engrenages.....	20
II.2.c Comparaison entre les deux systèmes :.....	21
II.2.d Choix de la transmission.....	21
III Choix du moteur	21
III.1 Détermination du couple minimal à fournir.....	21
III.2 Sélection du moteur.....	24
III.2.a Recherche et choix du moteur	25
III.2.b Vérification des caractéristiques du moto-réducteur :	26
IV Choix des composants mécaniques.....	27
IV.1 Choix de l'engrenage.....	27
IV.2 Vérification de la conformité du module.....	28
IV.3 Récapitulatif des valeurs remarquables :	29
IV.4 Structure fonctionnelle :	29
IV.5 Illustration du mécanisme.....	29
V Modélisation avec Autodesk Inventor.....	30
VI Présentation de la maquette:.....	31
VI.1 Réalisation sous Inventor:.....	31
VI.2 Mise en plan:.....	31
VI.3 Réalisation de la maquette réelle....	32
PARTIE C : Étude électronique.....	34
I Architecture électronique.....	34
II Étude théorique et choix des composants	35
II.1 Le pont en H	35
II.1.a Description :	35
II.1.b Schéma de principe :	35
II.1.c Principe :	35
II.2 Le module RFID	36
II.2.a Description du module :	36
II.2.b Les tests du module :	36
II.3 Le micro contrôleur	40
II.3.a Introduction.....	40
II.3.b Choix du micro-contrôleur.....	40
II.3.c Programmation.....	41
II.4 Interface utilisateur	41
II.4.a Introduction.....	41
II.4.b L'écran :	41
II.4.c L'acquisition.....	42

II.5 L'écran LCD.....	43	IV.1.a Présentation.....	54
II.5.a Utilisation de l'écran.....	43	IV.1.b Utilisation.....	55
II.5.b Initialisation de l'écran.....	44	IV.1.c Fonctionnement.....	55
II.5.c Tests de l'écran LCD :.....	44	IV.2 Routine de gestion du disque.....	55
II.6 Les touches sensitives	44	IV.2.a Présentation.....	55
II.6.a Choix des touches sensitives.....	44	IV.2.b Utilisation.....	56
II.6.b Présentation du QT320.....	45	IV.2.c Fonctionnement.....	56
II.6.c Fonctionnement.....	46	IV.3 Routine de communication.....	57
II.7 Le circuit MAX232	46	IV.3.a Présentation.....	57
II.7.a Schéma de câblage :.....	47	IV.3.b Fonctionnement.....	57
II.7.b Explications :.....	47	IV.4 Routine de dialogue avec le serveur	58
II.7.c Tests du circuit.....	47	IV.4.a Présentation.....	58
II.8 Contrôle de positionnement du disque	49	IV.4.b Utilisation.....	58
.....	49	IV.5 Routine de dialogue avec le module	58
II.8.a Le signal PWM.....	49	IV.5.a Présentation.....	58
II.8.b Génération des signaux PWM.	49	IV.5.b Utilisation.....	58
III Réalisation pratique et schéma		IV.6 Routine d'interruptions.....	58
électronique.....	50	IV.6.a Présentation.....	58
III.1 Schéma électronique.....	50	IV.6.b Utilisation.....	58
III.2 Réalisation de la carte électronique	50	IV.7 Bibliothèque de fonctions.....	58
III.2.a Conception assistée par		V Logiciel serveur.....	59
ordinateur.....	50	Partie E : Ressources documentaires.....	60
III.2.b Réalisation pratique.....	51	Partie F : Réalisation finale.....	61
Partie D : Étude informatique.....	52	Partie G : Synthèses.....	62
I Présentation du software.....	52	I Synthèse de Jean CLAISSE.....	62
II Architecture du software.....	52	II Synthèse d'Emilie CANTERO.....	63
III Protocole de communication.....	54	III Synthèse de Thibaud LEMAIRE.....	64
IV Logiciel embarqué.....	54		
IV.1 Routine d'affichage.....	54		

PARTIE A : Étude préliminaire

Dans cette section, vous retrouverez l'ensemble du Cahier des Charges Fonctionnel avec les documents d'étude tels que le SADT du système, le diagramme « bête à corne », le FAST ou encore le diagramme pieuvre.

Vous aurez également accès à la description du projet, la problématique dans laquelle il s'inscrit ainsi qu'une description détaillée du comportement attendu.

I Présentation de l'auto-partage

I.1 Qu'est ce que l'auto-partage ?

L'auto-partage est un système par lequel une entité, publique ou privée, propose à la location une flotte de véhicules à ses membres. Le principe est simplisme : au lieu d'utiliser une voiture personnelle qui reste la majeure partie du temps en stationnement ou en garage, l'utilisateur utilise seulement une voiture qu'il loue où et quand bon lui semble sur simple réservation et pour un trajet relativement court. Ce système permet de désencombrer les agglomérations. En effet, à en croire les statistiques à ce sujet, une voiture auto-partagée pourrait dans l'absolu remplacer six voitures personnelles, le système reposant sur les différences de plages horaires d'utilisation.

Ce service et en croissance continue ses dernières années et on compte aujourd'hui plus de six cent villes dans le monde proposant ce service. La plupart du temps, la gestion est opérée par une société privée. On peut citer par exemple les quelques sept cent stations d'auto-partage opérant sur la ville de Paris, ainsi qu'une multitude d'autres exemples français dans certaines villes telles que Montpellier, Marseille, Lille, Lyon, Bordeaux, etc...

I.2 Comment ça marche ?

La réservation s'effectue en ligne, par téléphone ou SMS. L'utilisateur se rend ensuite dans une agence pour récupérer les clés ou directement auprès de la voiture réservée qu'il pourra ouvrir par simple passage de sa carte d'abonné devant le pare-brise, déverrouillant ainsi les portières. L'utilisateur peut ensuite utiliser le véhicule dans une limite temporelle raisonnable, avant de le ramener à sa station d'attache. Il paye un forfait horaire et/ou kilométrique incluant le carburant, l'assurance, le parking et la gestion administrative ainsi qu'un abonnement au service. En pratique, ce mode de déplacement est très attractif pour les usagers utilisant leur véhicule moins de 10 000 km par an.

I.3 Un transport écocitoyen et écologique

Écocitoyen d'abord, car l'auto-partage participe au désencombrement des villes. En effet, en partant du constat qu'un véhicule passe 95% de son temps stationné, il paraît évident qu'il pourrait servir à plusieurs personnes, selon leur emploi du temps respectif. Considérons également qu'un véhicule d'auto-partage peut remplacer jusqu'à dix véhicules personnels, on peut deviner l'effet plus que positif de l'auto-partage sur les difficultés de stationnement. Écologique ensuite car les véhicules d'auto-partage sont sélectionnés pour leurs faibles rejets de CO₂, voire même parce qu'ils sont électriques. D'après une étude Germano-Suisse, l'abandon de la voiture personnelle au profit d'un véhicule auto-partagé va inciter l'utilisateur à utiliser ce mode de déplacement au strict minimum, et donc encourager son usage des transports écologiques et/ou en commun.

II Cahier des charges fonctionnel

Le projet consiste en l'étude et la conception d'un système permettant de gérer les clés d'un parc automobile de type "auto-partage". Le système doit être capable d'identifier un utilisateur inscrit, et de lui délivrer les clés d'un véhicule à proximité. Il doit aussi pouvoir informer le serveur centralisant les données de la location du-dit véhicule. Le système devra respecter les normes en vigueurs ainsi que les contraintes liées à son environnement. Une recherche ergonomique devra être menée afin de faciliter l'utilisation du système.

Le présent cahier des charges détaille l'ensemble des fonctions et contraintes ainsi qu'une pré-étude du projet.

II.1 Expression du besoin

II.1.a Point de vue utilisateur

Pouvoir récupérer ou déposer les clés d'un véhicule auto-partage, c'est à dire en location courte durée sans contrainte de stationnement. Le système doit permettre de restituer le véhicule ailleurs que là où il a été loué pour palier ce défaut de l'actuel système.

II.1.b Point de vue concepteur

Permettre à l'utilisateur de déposer et récupérer les clés d'un véhicule, assurer la gestion des locations de façon centralisé.

II.2 Justification du besoin

- Pouvoir récupérer un véhicule à n'importe quelle heure du jour ou de la nuit.
- Pouvoir louer un véhicule et le restituer à un autre endroit (par exemple, trajet domicile-travail).
- Réduire les dépenses de la société de location en supprimant les aménagements spéciaux actuels des véhicules en location (ouverture des portières par carte) ainsi que de leurs emplacement (dispositif de blocage du véhicule).

II.3 Objectifs et contraintes

L'objectif est de réaliser un système fiable, facile d'emploi, sécurisé et économique, décrivant les spécifications énoncées dans le présent document.

II.3.a Caractéristiques de la borne d'auto partage

- Doit s'adapter au réseau électrique EDF (tension de 230V: +/- 10V).
- Doit pouvoir communiquer avec un serveur.
- Doit afficher des données de contrôle pour l'utilisateur (2 lignes minimum).
- Doit permettre une commande aisée par l'utilisateur.
- Doit résister aux contraintes mécaniques et météorologiques (IP55, sauf prototype).
- Doit s'adapter aux différentes clés et les stocker de façon sécurisée.

II.3.b Environnement de la borne d'auto partage

Dans une rue ou avenue, à disposition du grand public. Le dispositif est donc fortement exposé aux phénomènes météorologiques divers ainsi qu'au vandalisme. Il convient donc de l'en prémunir.

II.3.c Serveur

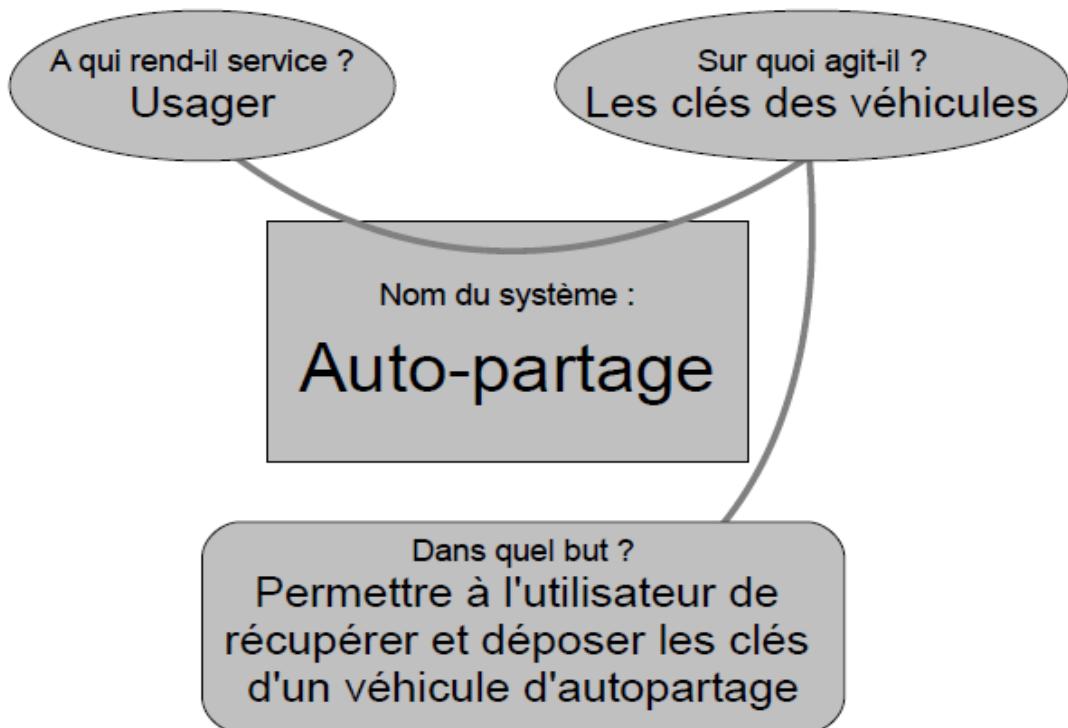
Le serveur sera exclu de l'étude. Cependant, le système étant contrôlé par ce dernier, un logiciel permettant de simuler le serveur sera développé. Ce logiciel devra respecter les caractéristiques du protocole BOSEP (BOrne SErveur Protocol), défini spécialement pour le projet.

La liaison entre la borne et le serveur sera assuré par un module de communication hors étude qui mettra en œuvre les communications longues distances (par radio par exemple). Dans notre étude, les serveurs et le module de communication étant hors étude, nous connecterons directement le PC émulant le serveur à la borne.

La communication avec le serveur sera assurée par une liaison RS-232, facile à mettre en œuvre et utilisable aussi bien avec un PC, qu'avec un module de communication. Pour l'émulation du serveur, les ports série étant rares sur les PC actuels, nous utiliserons un convertisseur USB-Série qui émulera un port COM virtuel sur le PC.

Le port RS-232 de la borne sera classique : les signaux fourni par le micro-contrôleur seront au format TTL, il faudra donc utiliser un circuit MAX232 pour la conversion.

II.4 Bête à cornes

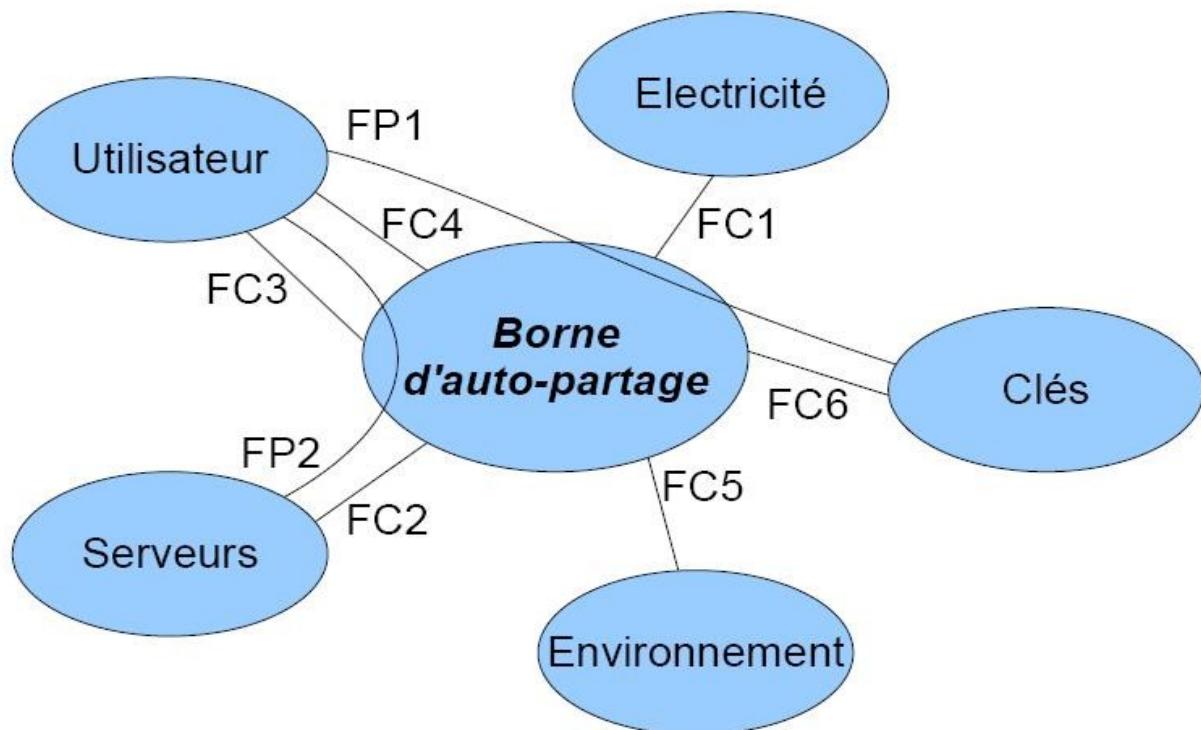


Le diagramme dit "Bête à cornes" permet d'exprimer le besoin de l'utilisateur ou du client. En l'occurrence, il s'agit du point de vue de l'utilisateur.

II.5 Diagramme pieuvre et tableau des fonctions

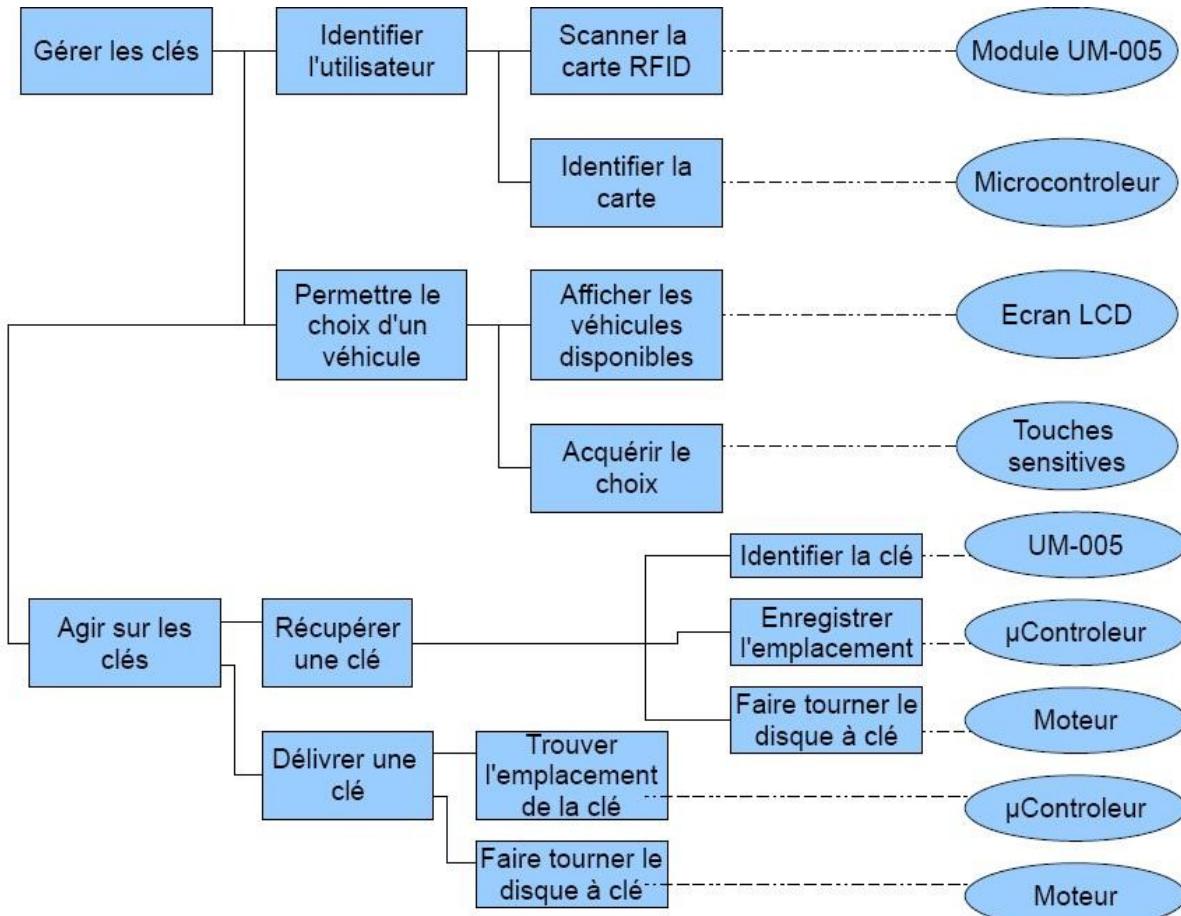
Le tableau des fonctions permet de définir les fonctions principales, complémentaires et contraintes du système. On pourra représenter ces fonctions sous la forme d'un "Diagramme pieuvre" qui met en évidence les relations entre les différents éléments extérieurs au système.

Fonction	Nom	Description	Niveau	Exigence
FP1	Gestion	Gestion des clés en fonction de l'utilisateur	5 clés	minimum
FP2	Identification	Pouvoir identifier l'utilisateur dans une BDD sur le serveur	RFID	
FC1	Alimentation	S'adapter au réseau électrique d'EDF	230 V	+/- 10V
FC2	Communication	Communiquer avec un serveur	RS232	
FC3	Affichage	Afficher des données de contrôle pour l'utilisateur	2 lignes	Ou plus
FC4	Commande	Permettre une commande aisée par l'utilisateur		
FC5	Environnement	Résister aux contraintes mécaniques et météorologiques	IP55	Sauf protot.
FC6	Adaptation et sécurité	S'adapter aux différentes clés et les stocker de façon sécurisée		



II.6 Diagramme FAST

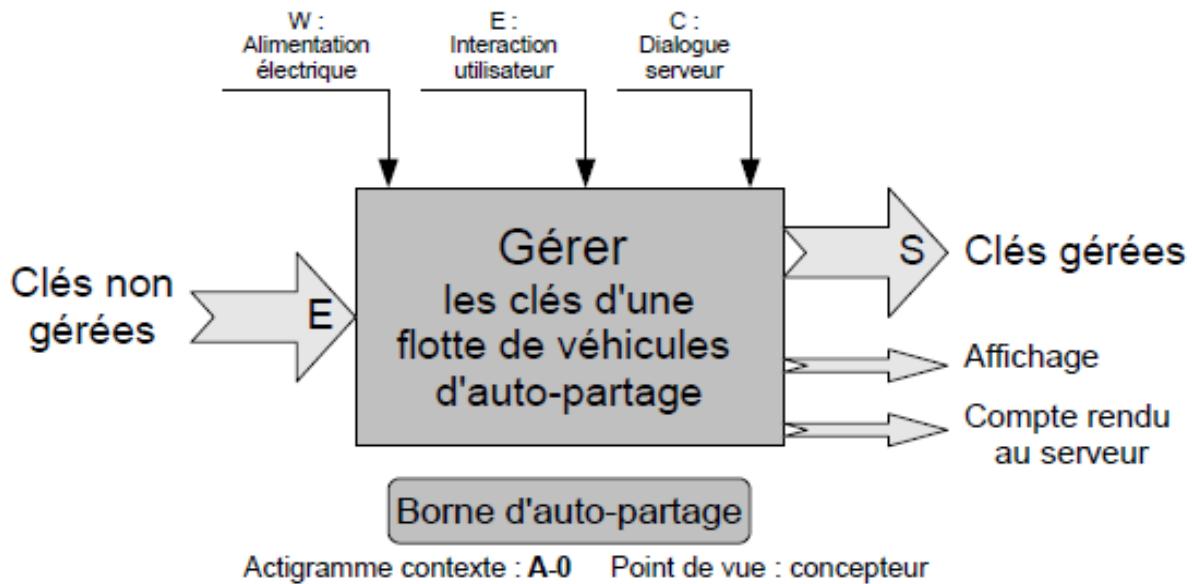
Le diagramme FAST, de par sa construction rigoureuse de gauche à droite et de haut en bas, permet de connaître la liste des fonctions de service, qui définissent le comportement général du système, et des fonctions techniques, qui apportent une solution technique à un comportement voulu. On se déplace donc vers la droite pour obtenir les fonctions techniques réalisant une opération, et vers le bas pour avoir une précision temporelle. Les fonctions techniques sont également liées à des éléments extérieurs au système.



II.7 Étude SADT

L'étude SADT est un outil d'analyse des fonctions de manière hiérarchique en découplant chaque fonction (dite "mère") en sous-fonctions "filles" décomposant ainsi le fonctionnement de chaque bloc. Elle permet en outre de visualiser rapidement la matière d'œuvre du système (ce sur quoi il agit) ainsi que les données de contrôle.

Ci-dessous un extrait de l'étude SADT réalisée pour le projet :



L'intégralité de l'étude SADT est disponible en annexe 1 à la fin de ce dossier.

II.8 Organigramme du software

L'organigramme est un outil qui permet de définir le comportement d'un système. Ici, il est utilisé pour segmenter l'étude de l'informatique embarquée, dite "software". Grâce à l'organigramme, le développeur connaît exactement le comportement du système et peut ainsi le retrancrire sous forme informatique.

L'organigramme est un document volumineux. Il est donc disponible en annexe 2 de ce dossier.

II.9 Organisation pédagogique

L'étude du système s'est déroulée lors des séances de TPE, mais aussi lors du temps libre, lorsque cela a été possible.

II.9.a Constitution du groupe :

THIBAUD LEMAIRE 1S3
EMILIE CANTERO 1S3
JEAN CLAISSE 1S3

II.9.b Répartition des tâches (Diagramme de GANTT.)

Le diagramme de GANTT représente une projection temporelle de l'avancement du projet. Les différents membres de l'équipe de développement se voient attribuer des tâches, auxquelles sont affectées une durée. Ce diagramme a été élaboré par toute l'équipe en début de projet.

Le diagramme de GANTT de notre projet est disponible en annexe 3 de ce dossier.

PARTIE B : Étude mécanique

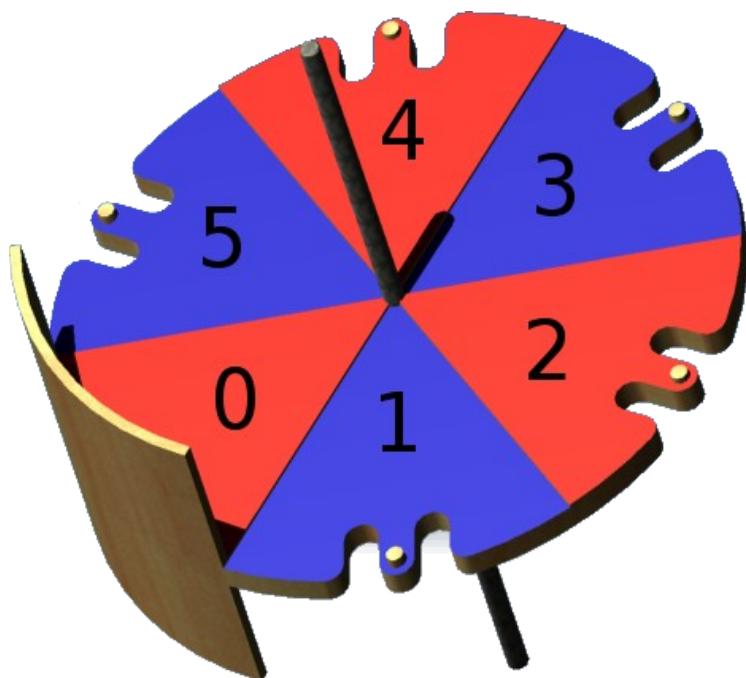
I Étude du mécanisme

Dans cette partie, nous vous présenterons la partie mécanique de notre projet.

S'y trouvent par exemple l'étude du mécanisme (schémas , graphe des liaisons, tableau des liaisons, maquette numérique réalisée avec le logiciel Inventor), des informations sur le calcul des caractéristiques du projet (calcul de la vitesse de rotation du disque, calcul du couple, ...), des choix que nous avons fait au niveau de solutions techniques, et enfin le choix de notre moteur .

I.1 Principe du disque

Pour stocker, restituer et récupérer les clés, nous avons imaginé un système de disque rotatif. Le disque comporte des encoches dans lesquelles viennent se loger les clés. Le disque est directement accessible à l'utilisateur. Cela implique qu'il n'y a pas besoin de mécanisme pour positionner la clé sur le disque. Pour sécuriser les clés, une porte sur le disque empêche l'utilisateur d'accéder à celles-ci.



Ci-dessus les emplacements des clés de 1 à 5. Le 0 étant la porte.

I.2 Étude cinématique

I.2.a Classes d'équivalences et tableau de liaisons

Pour étudier ce mécanisme, nous avons tout d'abord identifié les pièces ayant des mouvements relatifs entre elles : les classes d'équivalences.

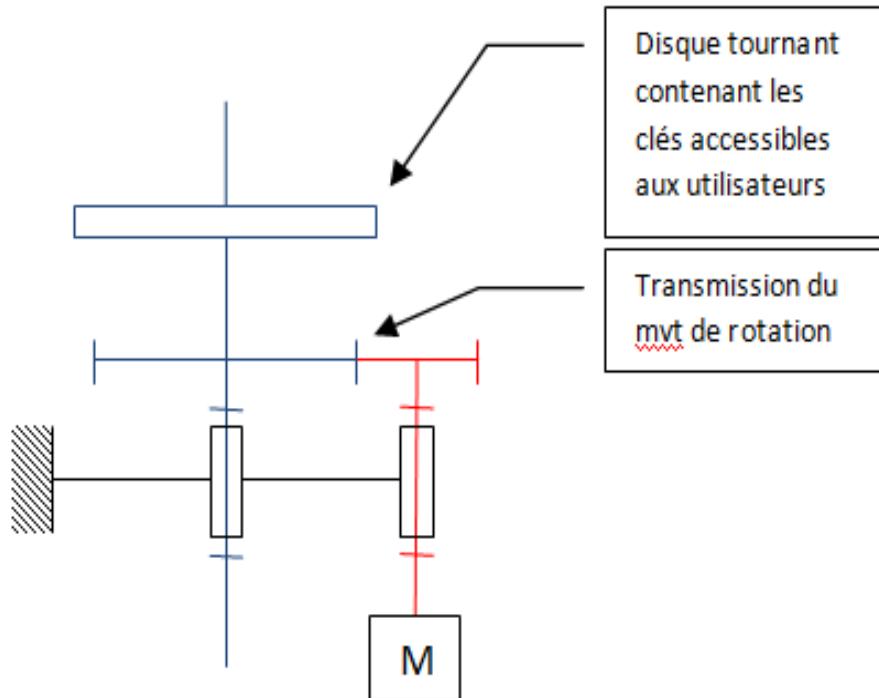
Voici un tableau qui liste ces différentes classes :

<u>Classes d'équivalences</u>	<u>Pièces</u>
S1	Bâti
S2	Axe du disque tournant
S3	Axe moteur

Puis nous avons défini les liaisons entre ces classes :

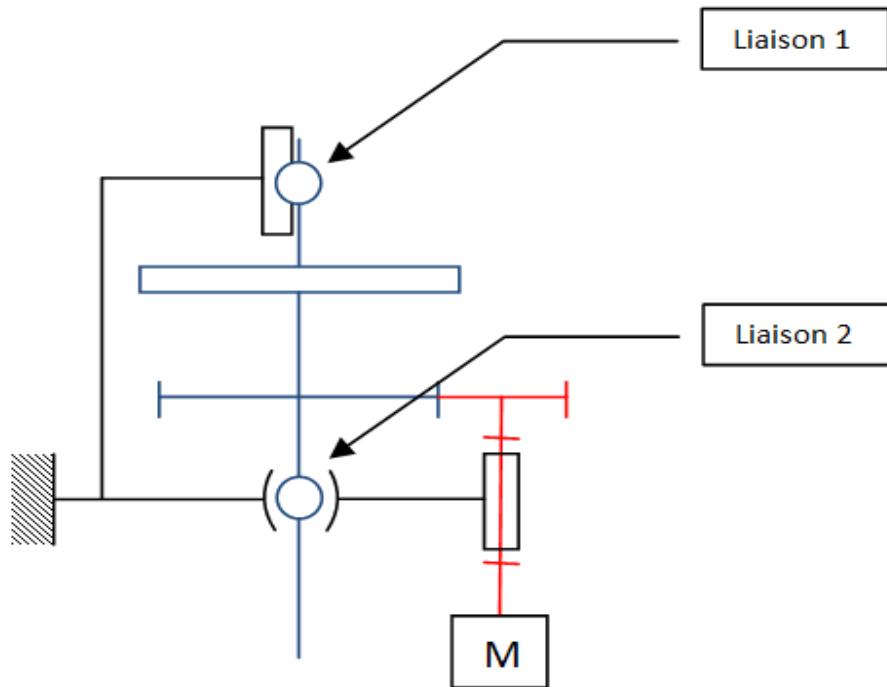
<u>Classe/Classe</u>	<u>Axe principal</u>	<u>Point d'application</u>	<u>Degrés de liberté</u>	<u>Nom de la liaison</u>
S1/S2	Y	A	Ry	Pivot d'axe Ay
S1/S3	y	C	Ry	Pivot d'axe Cy
S2/S3	Y	D	Ty , Ry	Engrenage contact en D

I.2.b Schéma cinématique



La réalisation de la liaison pivot entre l'arbre du disque tournant et le bâti doit permettre un fonctionnement avec le moins de frottements possibles. Pour cela nous avons prévu de réaliser ce guidage en rotation en utilisant un montage de roulements. Cette solution avec éléments roulants étant celle qui correspond le mieux au besoin.

Le montage est constitué de deux roulements dont la mise en position entraîne la modélisation suivante:

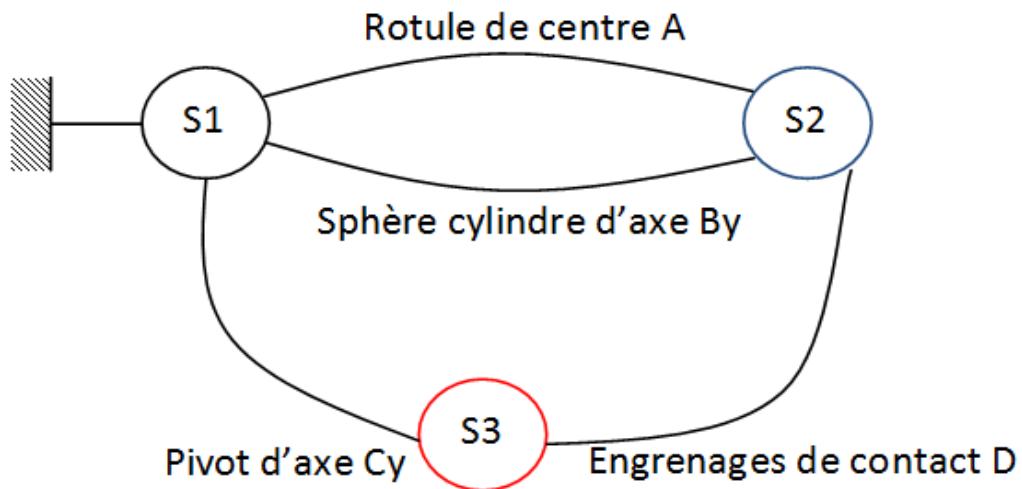


La liaison pivot initiale entre l'arbre du disque tournant et le bâti est remplacée par les 2 liaisons équivalentes 1 et 2.

I.2.c Nouveau tableau des liaisons

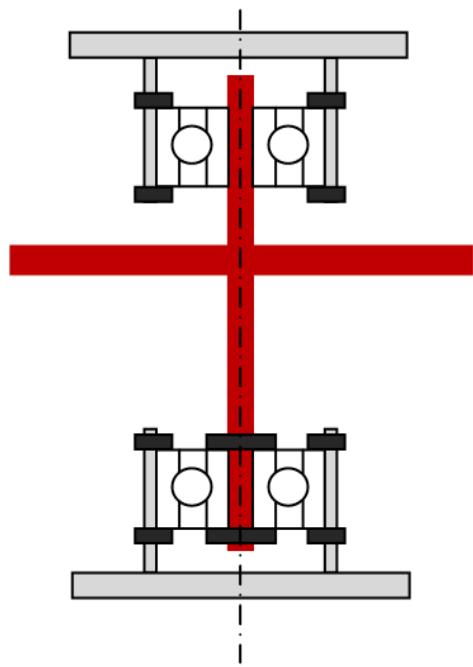
<u>Classe/Classe</u>	<u>Axe principal</u>	<u>Point d'application</u>	<u>Degrés de liberté</u>	<u>Nom de la liaison</u>
S1/S2		A	Rx, Ry, Rz	Rotule
S1/S2	Y	B	Rx, Ry, Rz, Ty	Sphère cylindre d'axe By
S1/S3	y	C	Ry	Pivot d'axe Cy
S2/S3	Y	D	Ty , Ry	Engrenage contact en D

I.2.d Graphe des liaisons



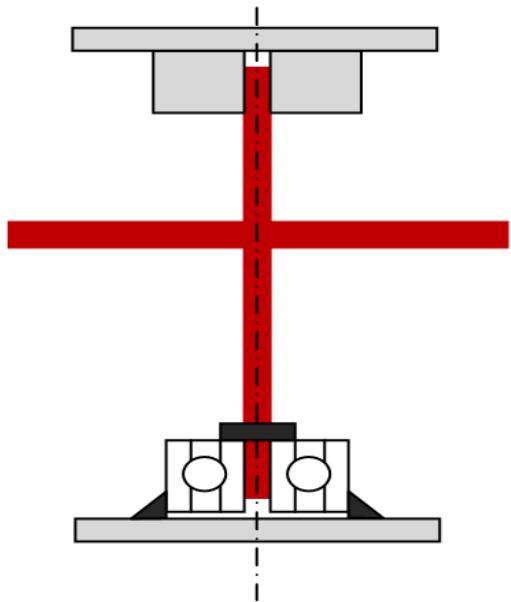
I.3 Étude de conception de la liaison pivot

Le montage de roulements est prévu avec des arrêts en translation tels qu'ils sont représentés sur le schéma suivant:

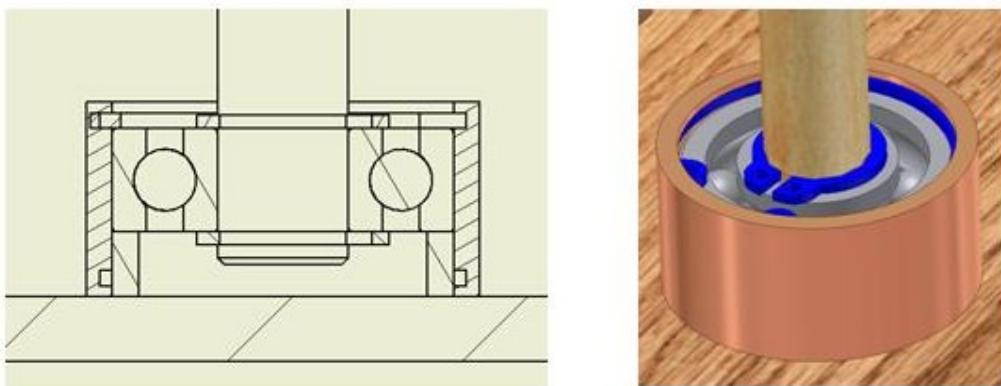


En pratique, la liaison sphère cylindre en partie haute est réalisée par un contact direct entre une bague en PVC et une entretoise collée sur l'axe également en PVC (coefficient de frottements faible).

Le montage de roulements tel qu'il a été réalisé sur la maquette réelle, pour des raisons matérielles, correspond au schéma suivant :

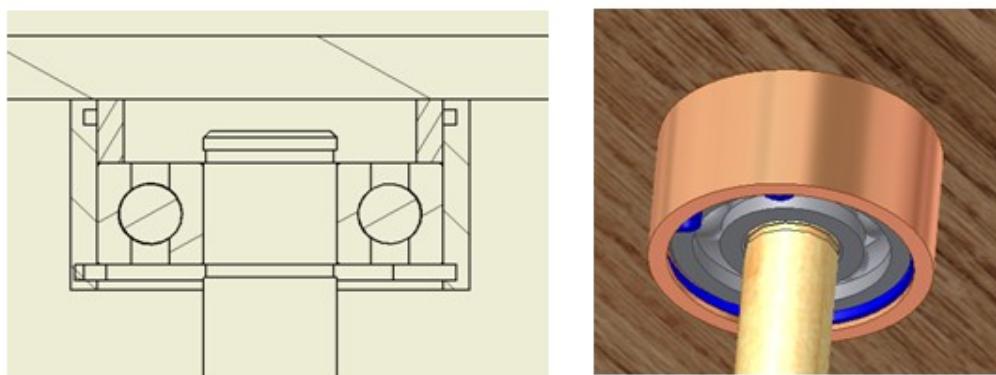


I.3.a Réalisation de la liaison rotule sous Inventor



La mise en plan montre la présence de quatre arrêts (intérieurs et extérieurs)

I.3.b Réalisation de la liaison sphère cylindre sous Inventor



La mise en plan montre la présence de deux arrêts intérieurs uniquement (pas de circlip sur l'axe du disque tournant)

I.3.c Réalisation des deux liaisons sur la maquette réelle



La liaison sphère cylindre est réalisée par un contact direct

II Mise en rotation du disque

II.1 Les types de moteurs

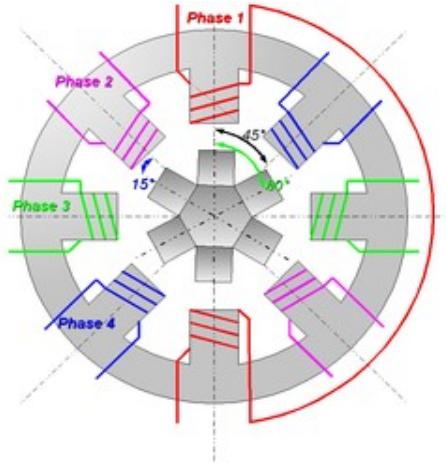
Afin de permettre l'entrainement en rotation du disque à clés par le moteur, plusieurs solutions sont envisageables. Il convient de comparer ces solutions qui présentent toutes des avantages et des inconvénients. Le choix se fera surtout sur des critères de prix et de facilité de mise en œuvre. Voici les solutions qui s'offrent à nous :

II.1.a Le moteur pas à pas

Il permet d'asservir un mécanisme à une position angulaire. Il est contrôlé par des signaux électriques qui permettent de déterminer sa vitesse et sa position. Les moteurs pas à pas sont fréquemment utilisés dans les dispositifs nécessitant un contrôle de vitesse et de position en boucle ouverte, par exemple dans les imprimantes.

Principe de fonctionnement :

Il y a plusieurs types de moteurs pas à pas, mais le plus utilisé est le moteur pas à pas à réductance variable. Il est composé de plusieurs bobines qui sont activées les unes à la suite des autres et qui attirent un barreau de ferrite de forme étoilé. Ces bobines décomposent donc la rotation en pas saccadés.



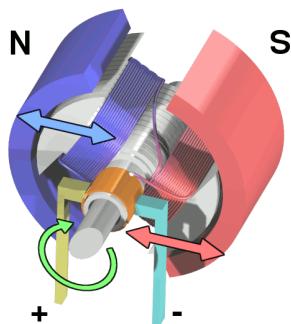
Avantages :

1. Précision de l'asservissement.
2. Pas besoin de capteurs de position.

Inconvénients :

1. Prix.
2. Difficulté de mise en œuvre et de commande.

II.1.b Les moteurs à courant continu



Le moteur à courant continu est un dispositif de conversion de l'énergie électrique en énergie mécanique de rotation.

On distingue deux grands types de moteurs à courant continu (ou DC, ou même CC) : les moteurs avec balais, normaux, c'est les premiers moteurs à courant continu inventés, ou sans balais (brushless en anglais), qui sont une évolution de ce dernier, et qui élimine une grande partie des problèmes rencontrés avec un moteur classique, comme les parasites générés, inertie, durée de vie, refroidissement, etc...

Il est constitué d'une partie mobile appelé le rotor, sur lequel viennent se fixer des bobines, et d'une partie fixe, le stator, sur lequel sont fixés des aimants permanents. Les balais, qui frottent contre l'axe de rotation, activent tour-à-tour les bobines en attraction puis en répulsion.

Pour notre système, la vitesse de rotation est trop importante. Pour ce type de moteur, on aura donc recours à un réducteur de vitesse qui, par ailleurs, augmentera le couple.

Avantages :

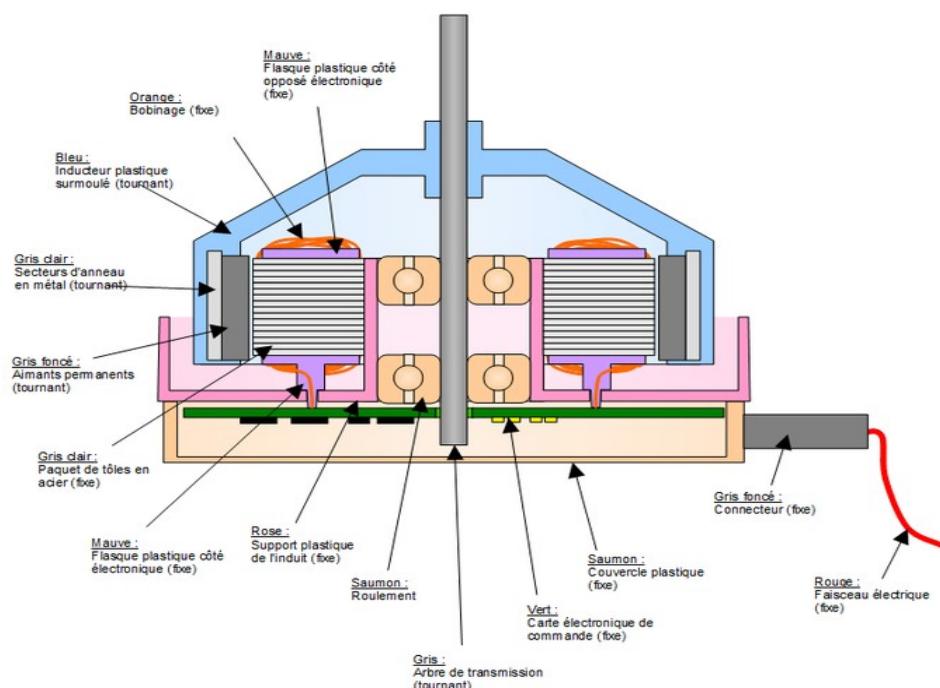
1. Faible coût.
2. Facilité de mise en œuvre.

Inconvénients :

1. Nécessité d'utiliser un réducteur.
2. Inertie d'arrêt.
3. Obligation d'utiliser des capteurs de position sur le disque.

II.1.c Moteur C.C. Brushless

La particularité de ce moteur réside dans le fait que c'est le stator qui abrite les bobines. Il n'y a donc pas besoin de balais. Cependant, pour activer tour à tour les bobines, on a recours à un circuit électronique de commande qui peut être intégré (pour les faibles puissances) ou externe au moteur.



Comme pour son homologue avec balais, la vitesse de rotation du moteur brushless est trop importante. On aura donc également recours à un réducteur de vitesse.

Avantages :

1. Facilité de mise en œuvre.
2. Absence de parasites.
3. Absence d'inertie.

Inconvénients :

1. Prix.
2. Obligation d'utiliser un réducteur.
3. Obligation d'utiliser des capteurs de positions.

II.1.d Choix d'une solution technique

Le moteur pas-à-pas semble une solution intéressante. Cependant, son coût et sa difficulté de mise en œuvre font qu'il est difficilement envisageable de l'utiliser pour la rotation du disque. De plus, l'asservissement de la position du disque ne doit pas être précis, on peut donc se permettre d'avoir une précision de +/- 30° ($360/(6 \times 2)$).

Le moteur à courant continu sans balais et envisageable, cependant l'augmentation de prix par rapport à un moteur DC classique n'est, selon nous, pas justifié. Les problèmes d'inertie peuvent être réduits par le réducteur et les parasites générés par ce dernier sont facilement supprimables avec une bonne alimentation. C'est donc ce dernier (moteur DC classique) que l'on utilisera pour la rotation de notre disque à clé, principalement à cause de son faible coût et de sa commande facile. Il sera utiliser avec un réducteur que nous calculerons. Il faudra également calculer le couple nécessaire.

Pour ce qui est du positionnement, une analyse technologique sera effectué afin de définir la meilleure solution.

II.2 Entrainement de l'axe tournant : courroie ou engrenage ?

La rotation du disque à clé nécessite un système de transmission du mouvement émanant du moteur. Pour cela, deux possibilités s'offrent à nous : la transmission par engrenage ou par système poulie/courroie. Nous allons ici comparer ces deux solutions techniques pour déterminer celle qui correspond le plus à nos attentes. Le système de transmission choisi devra être capable de réduire la vitesse de rotation avec un rapport que nous n'avons, à ce stade, pas encore défini mais pouvant aller de 1:2 à 1:4.

II.2.a Transmission par poulie-courroie



Leur fonction est de transmettre un mouvement de rotation continu par adhérence à l'aide d'un lien flexible : la courroie.

Elles sont silencieuses et permettent une rotation à des vitesses assez élevées. Cependant les courroies doivent être initialement tendues pour garantir l'adhérence et assurer la transmission du mouvement. De ce fait, un système intermédiaire placé entre les deux poulies (galet enrouleur, etc...) est souvent nécessaire pour régler la tension initiale et compenser le fait que les courroies s'allongent au fur et à mesure de leur utilisation.

A l'exception des courroies crantées, en fonctionnement normal, il existe un léger glissement de la courroie sur les poulies amenant une imprécision du rapport de transmission. Cela implique également qu'en cas de pression sur le système, celui-ci bougera sans trop d'efforts.

Avantages :

1. Silence
2. Élasticité

Inconvénients :

1. Usure de la courroie (durée de vie limitée)
2. Glissement (sauf courroies crantées)
3. Tension initiale nécessaire, qui implique l'installation d'un dispositif de réglage de la tension.

Rapport de transmission :

$$r_{trans} = \frac{d}{D} = \frac{\omega_D}{\omega_d} = \frac{C_d}{C_D}$$

II.2.b Transmission par train engrenages



Leur fonction est de transmettre un mouvement de rotation continue par obstacle.

L'engrenage est un système élémentaire constitué de deux roues dentées d'axes différents et dont l'une entraîne l'autre par contact des dents successivement. On les utilise pour transmettre un mouvement et puissance entre deux arbres parallèles.

Pour un prix modéré, ils ont pour avantages un excellent rendement et un encombrement plutôt faible. De plus le couple et la vitesse transmissibles sont très élevés (supérieurs aux courroies) et ils disposent d'une durée de vie élevée. Cependant une lubrification régulière est nécessaire et l'installation suppose un entre-axe précis.

Avantages :

1. Synchronisme
2. Précision
3. Couple et vitesse transmissibles élevés
4. Prix de revient faible
5. Durée de vie élevée

Inconvénients :

1. Lubrification nécessaire
2. Nécessité d'un entre-axe précis
3. Bruit en fonctionnement

Rapport de transmission :

$$r_{trans} = \frac{r_e}{r_s} = \frac{\omega_s}{\omega_e} = \frac{Z_e}{Z_s} = \frac{C_e}{C_s}$$

II.2.c Comparaison entre les deux systèmes :

Les courroies pourraient être intéressantes mais disposent de qualités qui ne sont pas nécessaires au projet et sont toutefois assez difficiles à mettre en œuvre. Le silence du système n'est qu'un atout mineur et l'élasticité est inutile. Les engrenages pourront donc mieux répondre au cahier des charges grâce à leurs avantages incontestables : prix, couple transmissible élevé, précision, etc...

II.2.d Choix de la transmission

Nous choisirons les engrenages pour la liaison entre l'axe du disque à clés et le moteur particulièrement pour la raison suivante :

Les engrenages ont l'avantage de ne pas glisser et de rester à leurs positions jusqu'à l'enclenchement du moteur. Cela implique que si un client force dessus, le disque à clés ne pourra pas tourner, entraînant une sûreté plus grande. De plus, leur prix de revient est bien moindre par rapport au système poulie courroie. Enfin, par leurs caractéristiques intrinsèques, (Synchronisme, précision, et surtout possibilité de réduire le mouvement initial), les engrenages répondent parfaitement au cahier des charges.

III Choix du moteur

III.1 Détermination du couple minimal à fournir

Afin de ne pas dimensionner le moteur qui assurera la rotation du disque "au hasard", nous avons calculé le couple nécessaire à partir de l'accélération angulaire et du moment d'inertie du disque à clé.

Posons ω la vitesse angulaire en rad/s, α l'accélération angulaire en rad.s⁻², C le couple en N.m et J le moment d'inertie en kg.m² :

$$\alpha = \frac{\Delta \omega}{\Delta t} \quad \omega = \frac{2\pi}{t} \quad C = \alpha \times J$$

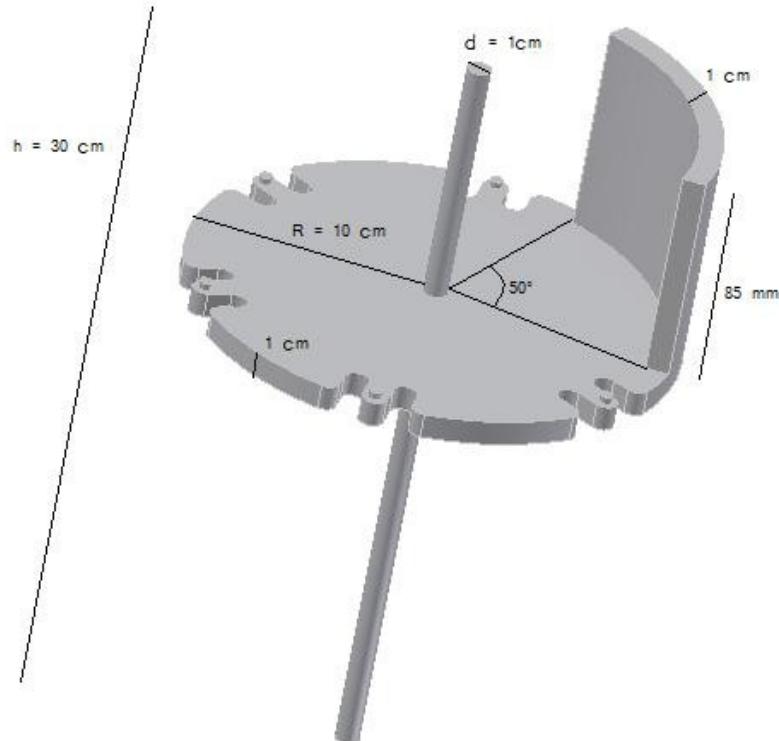
Maintenant, on définit arbitrairement une vitesse angulaire de 1 tour en 5 secondes, ce qui donne 12 tours par minute :

$$\omega = \frac{2\pi}{5} \text{ rad.s}^{-1}$$

On souhaite atteindre cette vitesse de rotation en 250 ms :

$$\alpha = \frac{\frac{2\pi}{5}}{\frac{1}{4}} = \frac{8\pi}{5} \text{ rad.s}^{-2}$$

On souhaite calculer le moment d'inertie de cette pièce faite de bois :



On calculera donc plusieurs moments d'inertie qu'on ajoutera ensuite. La masse volumique du bois varie habituellement entre 350 et 1100 kg.m³. Pour notre calcul de couple, nous avons choisi la valeur critique de 1000kg par m³.

Tout d'abord nous allons calculer le moment d'inertie de l'axe.

$$J_{axe} = \frac{M_{axe} R^2}{2}$$

$$M_{axe} = \rho \pi R^2 h$$

$$M_{axe} = 1000 \times \pi \times 0,005^2 \times 0,29 \approx 0,023 \text{ kg}$$

$$J_{axe} = \frac{0,023 \times 0,005^2}{2} \approx 2,9 \times 10^{-7} \text{ kg.m}^2$$

Le moment d'inertie de l'axe est donc négligeable. On n'en tiendra pas compte dans le calcul du moment d'inertie total de la pièce.

Les clés sont fixées en périphérie du cylindre, cela augmente donc le moment d'inertie du cylindre. Pour calculer le moment d'inertie des cinq clés, nous allons les étudier comme étant de simples points pondérés distants de l'axe de rotation.

Nous avons pesé trois clés et trois tags RFID à l'aide d'une balance domestique (150 g), et en avons déduit que la masse d'une clé est 50 grammes.



$$\begin{aligned} J_{clé} &= M_{clé} \times R^2 \\ J_{clé} &= 0,05 \times 0,1^2 = 5 \times 10^{-4} \text{ kg.m}^2 \\ J_{clés} &= 5 \times J_{clé} = 5 \times 5 \times 10^{-4} = 2,5 \times 10^{-3} \text{ kg.m}^2 \end{aligned}$$

Ensuite, nous avons calculé le moment d'inertie du cylindre supportant les clés :

$$\begin{aligned} J_{cylindre} &= \frac{M_{cylindre} R^2}{2} \\ M_{cylindre} &= \rho \pi R^2 h \\ M_{cylindre} &= 1000 \times \pi \times 0,1^2 \times 0,01 \approx 0,314 \text{ kg} \\ J_{cylindre} &= \frac{0,314 \times 0,1^2}{2} = \frac{3,14 \times 10^{-3}}{2} = 1,57 \times 10^{-3} \text{ kg.m}^2 \end{aligned}$$

Et enfin le moment d'inertie de l'extrusion de matière permettant de sécuriser les clés dans le système (sorte de porte).

$$\begin{aligned} J_{arc\ cylindre} &= \frac{50}{360} \times J_{cylindre\ creux} \\ J_{cylindre\ creux} &= \frac{1}{2} M_{cylindre\ creux} \times (R_{intérieur}^2 + R_{extérieur}^2) \\ M_{cylindre\ creux} &= \rho \pi h (R_{extérieur}^2 - R_{intérieur}^2) \\ M_{cylindre\ creux} &= 1000 \times \pi \times 0,085 \times (0,1^2 - 0,09^2) = 1000 \times \pi \times 0,085 \times 0,0019 = 0,51 \text{ kg} \\ J_{cylindre\ creux} &= \frac{1}{2} \times 0,51 \times (0,09^2 + 0,1^2) = \frac{1}{2} \times 0,51 \times 0,0181 = 4,6 \times 10^{-3} \text{ kg.m}^2 \\ J_{arc\ cylindre} &= \frac{50}{360} \times 4,6 \times 10^{-3} = 6,3 \times 10^{-4} \text{ kg.m}^2 \end{aligned}$$

Pour calculer le moment d'inertie total de la pièce, il suffit d'additionner les différents moment d'inertie trouvés précédemment :

$$J_{total} = J_{cylindre} + J_{clés} + J_{arc cylindre} = 1,57 \times 10^{-3} + 2,5 \times 10^{-3} + 6,3 \times 10^{-4} = 4,7 \times 10^{-3} \text{ kg.m}^2$$

On en déduit le couple nécessaire pour mettre en mouvement la pièce à partir de l'accélération angulaire :

$$C = \frac{8\pi}{5} \times 0,0047 \approx 0,024 \text{ N.m}$$

$$0,024 \text{ N.m} = 2,4 \text{ N.cm} \approx 240 \text{ g.cm} \approx 0,24 \text{ kg.cm}$$

Cette valeur n'est pas totalement exacte. Tout d'abord car la valeur de la masse volumique du bois a été définie arbitrairement. Cependant, nous avons choisi une valeur critique, le couple nécessaire est donc, en réalité, certainement moindre. Cela n'altère donc pas le fonctionnement du système. Ensuite parce que les frottements, certes minimes, de l'axe de rotation ne sont pas pris en compte dans le calcul.

III.2 Sélection du moteur

Comme défini précédemment, notre système nécessite de mettre en mouvement le "disque à clés" sur lequel reposent ces dernières. Pour cela, nous utiliserons un moteur électrique à courant continu car ce sont les moteurs les plus faciles à mettre en œuvre et à alimenter en énergie. Nous avons défini précédemment le couple minimal nécessaire pour la mise en rotation du disque. La vitesse de rotation avait également été définie. Nous nous sommes donc mis à la recherche d'un moteur, ou plus précisément d'un moto-réducteur capable de fournir le couple nécessaire à la vitesse de rotation définie. Cette page détaille notre démarche de recherche et la vérification par le calcul de la conformité du moteur choisi à notre cahier des charges.

Notre recherche s'est immédiatement orientée vers un moto-réducteur qui inclue le moteur et le réducteur adéquat dans une même enceinte car nous avons jugé que la mise en place d'un moteur et d'un réducteur séparés serait source de dysfonctionnements et d'élévation du coût.

Démarche pour la sélection du moteur :

A vrai dire, la recherche d'un moto-réducteur pour une application donnée s'effectue par tâtonnements car les fournisseurs indiquent seulement un moteur de référence et les différents rapports de réduction disponible. Il faut ensuite calculer pour chaque moteur sélectionnés la vitesse angulaire et le couple après rotation. Notre principal critère de recherche a été le prix. En effet, les moto-réducteurs peuvent rapidement coûter très cher.

Pour chaque moto-réducteur que l'on veut étudier, on applique la formule suivante :

$$P_{moteur} = C \cdot \omega \Leftrightarrow C = \frac{P_{moteur}}{\omega}$$

Or, la puissance du moteur et la puissance du réducteur à étudier ne diffèrent pas (en théorie) aux pertes près. On a donc :

$$P_{réducteur} = P_{moteur} \cdot \eta$$

Pour le calcul du couple en sortie du réducteur :

$$C_{réducteur} = \frac{\eta_{réducteur} \times C_{moteur}}{R_{réduction}}$$

Et celle-ci pour le calcul de la vitesse de rotation en sortie du réducteur :

$$\omega_{réducteur} = \omega_{moteur} \cdot r_{réduction}$$

III.2.a Recherche et choix du moteur.

On a tout d'abord fixé la vitesse de rotation du moto-réducteur. Nous avions choisi, lors du calcul du couple nécessaire, que le disque ferait un tour en 5 secondes, soit 12 tours par minute. Nous avons également choisi l'engrenage entre le moteur et le disque; celui-ci affiche un rapport de réduction de 1:2. Nous devons donc choisir un moteur dont la vitesse de rotation est environ 24 tours par minute.

Pour convertir une vitesse de rotation (en tour par minute) en vitesse angulaire, on peut utiliser la relation suivante :

$$\omega = \frac{\pi \cdot N}{30}$$

On peut donc facilement calculer la vitesse angulaire en sortie du moto-réducteur :

$$\omega = \frac{\pi \cdot 24}{30} \approx 2,5 \text{ rad.s}^{-1}$$

Ensuite nous avons chercher dans un catalogue les moto-réducteurs disponibles. Nous en avons repérés chez Sélectronic qui utilisent un moteur de base, auquel est couplé un réducteur. Notre travail a donc été de choisir le réducteur adéquat.

Le moteur sur lequel sont basés les moto-réducteurs de la gamme 918D de MFA a pour caractéristiques : une vitesse de rotation en charge de 6300 tr/min sous 12V, et un couple de 25 g.cm. Tout d'abord, il faut convertir ces données dans les unités du Système International. Cela donne donc pour la vitesse angulaire :

$$\omega = \frac{\pi \cdot 6300}{30} \approx 659,7 \text{ rad.s}^{-1}$$

Pour la conversion du couple, on peut utiliser cette relation :

$$C[N.m] = \frac{C[g.cm]}{10197.16}$$

Donc pour le moteur en question :

$$C_{moteur} = \frac{25}{10197.16} \approx 2,45 \times 10^{-3} \text{ N.m}$$

Cette valeur représente le couple en sortie du moteur, il n'est donc pas gênant qu'elle soit si petite.

Il faut maintenant calculer le rapport de réduction théorique que l'on recoupera avec la valeur la plus proche proposée par le catalogue. Pour cela, on utilise la relation suivante :

$$\frac{\omega_{réducteur}}{\omega_{moteur}} = R_{réduction}$$

Appliqué à notre cas, cela donne :

$$R_{réduction} = \frac{2,5}{659,7} \simeq 3,79 \times 10^{-3} \simeq \frac{1}{264}$$

Après comparaison avec les valeurs standards, nous avons choisi un réducteur 1:250.

III.2.b Vérification des caractéristiques du moto-réducteur :

Ce rapport de réduction appliquée au couple, on obtient un couple en sortie du moto-réducteur tel que :

$$C_{réducteur} = \frac{0,8 \times 2,45 \times 10^{-3}}{\frac{1}{250}} \simeq 0,5 \text{ N.m}$$

Ce couple est largement supérieur au couple minimal calculé précédemment et permettra sans doutes de compenser les pertes dues aux différents étages de transmission ainsi qu'aux frottements.

Nous choisirons donc le moto-réducteur basé sur un moteur RE-280 de MFA, de vitesse de rotation 6300 tr/min et de couple 2,45E-3 N.m, associé à un réducteur de 1:250. La référence fabriquant 918D-251/12 et la référence Sélectronic est 11.1834-12. Ce moteur à une tension nominale de 12V et consomme en charge moins d'un ampère.



IV Choix des composants mécaniques



Nous avons précédemment calculé la vitesse de rotation et le couple en sortie du réducteur. Pour rappel, ces valeurs sont $C=0,5 \text{ N.m}$ et $N=25 \text{ tr.min}^{-1}$. Nous avons également posé le rapport de réduction de l'étage de transmission entre le moto-réducteur et le disque à clé, 1:2 ou supérieur. Nous allons maintenant choisir des roues dentées capables de transmettre le mouvement du moteur au disque.

IV.1 Choix de l'engrenage

On choisira nos engrenages en résine Acétale, communément appelés "HostaForm", car ce matériau est peu onéreux. Le module est fixé arbitrairement, pour le moment, à 0,5 mm. Nous vérifierons par la suite

L'étage de transmission sera composé d'un pignon fixé à l'axe du moteur de $Z=20$ dents, et d'une roue fixée au disque à clé de $Z=48$ dents. On obtient donc un rapport de réduction tel que :

$$r_{réduction} = \frac{Z_{menante}}{Z_{menée}} = \frac{20}{48} \approx 0,417$$

Nous calculons ensuite les diamètres primitifs de ces roues dentées grâce à la formule :

$$D_{primitif} = m \times Z$$

Ce qui nous donne pour le pignon :

$$D_{pignon} = 0,5 \times 20 = 10 \text{ mm}$$

Et pour la roue :

$$D_{roue} = 0,5 \times 48 = 24 \text{ mm}$$

IV.2 Vérification de la conformité du module

On pose, R_{pe} la résistance pratique à l'extension, pour un coefficient de sécurité de 2 et une résistance à l'extension de la résine Acétale Re=65MPa :

$$R_{pe} = \frac{R_e}{k_{sécurité}} = \frac{65}{2} \approx 32,5 \text{ MPa}$$

On rappelle que F_t, l'effort tangentiel appliqué à la roue tel que :

$$F_{tangential} = \frac{C}{r} \quad \text{et} \quad r = \frac{D}{2}$$

Enfin, après mesure de la largeur du pignon b, on pose k le coefficient de largeur de denture tel que :

$$k = \frac{b}{m} = \frac{3}{0,5} = 6$$

On sait que :

$$m \geq 2,34 \times \sqrt{\frac{F_{tangential}}{k \times R_{pe}}}$$

On peut en déduire que l'effort tangentiel maximal transmissible est obtenu grâce à la formule :

$$F_{tangential\ max} = \frac{k \times R_{pe} \times m^2}{2,34^2}$$

Appliquée à notre cas, l'effort tangentiel transmis ne doit pas être supérieur à :

$$F_{tangential\ max} = \frac{6 \times \frac{65}{2} \times 0,5^2}{2,34^2} \approx 8,9 \text{ N}$$

Ce qui correspond à un couple maximal transmissible au niveau du pignon de :

$$C_{motoréducteur\ max} = F_{tangential} \times r_{pignon} = 8,9 \times 0,005 = 0,045 \text{ N.m}$$

Soit au niveau de la roue dentée, donc du disque à clé :

$$C_{disque\ max} = F_{tangential} \times r_{roue} = 8,9 \times 0,012 = 0,107 \text{ N.m}$$

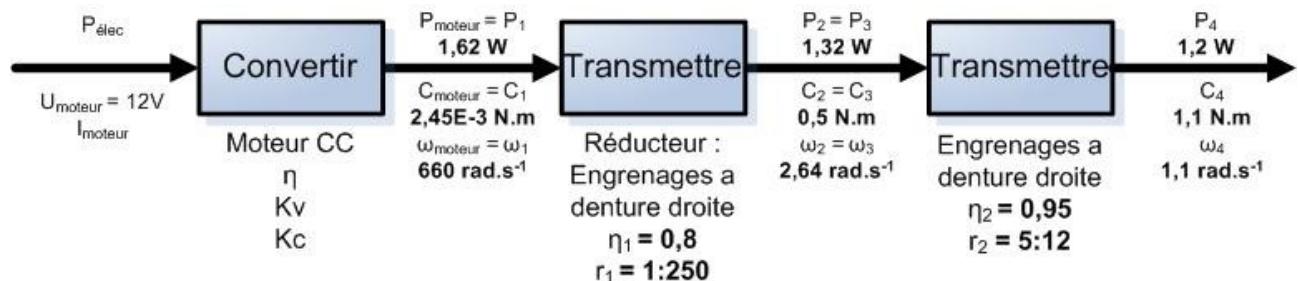
Le module 0,5 suffit donc pour notre application. En cas de blocage du système, le moto-réducteur étant capable de fournir un couple bien supérieur à celui que peut transmettre l'engrenage, c'est ce dernier qui viendrait à rompre.

IV.3 Récapitulatif des valeurs remarquables :

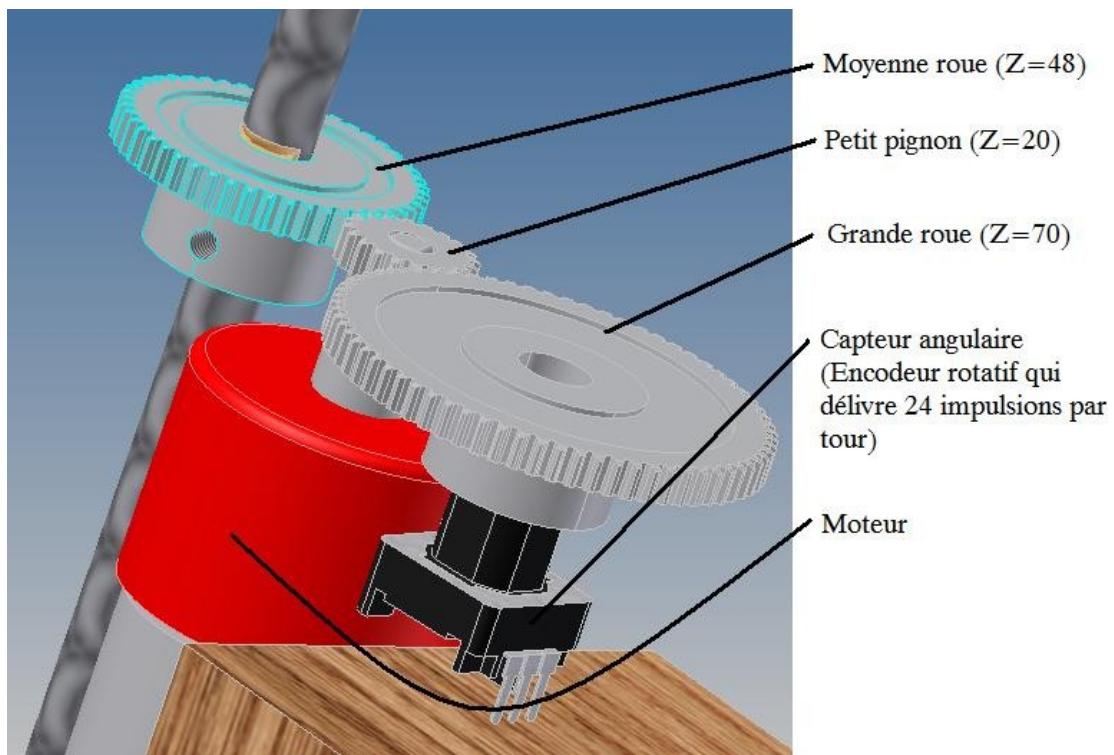
Le système est composé d'un pignon solidaire de l'axe moteur ($Z_1=20$), d'une roue fixée au disque à clé ($Z_2=48$) et d'une dernière roue qui engrène avec le pignon et qui, solidaire d'un capteur, permet de renvoyer une information angulaire sur le positionnement du disque ($Z_3=70$).

<u>Nom</u>	<u>Nombre de dents</u>	<u>Diamètre primitif (mm)</u>	<u>Vitesse de rotation (tr/min)</u>	<u>Vitesse angulaire (rad/s)</u>	<u>Couple théorique (mN.m)</u>	<u>Couple pratique maximal (mN.m)</u>
Z_1	20	10	25	2,6	490	45
Z_2	48	24	10,4	1,1	1176	107
Z_3	70	35	7,1	0,7	1715	156

IV.4 Structure fonctionnelle :

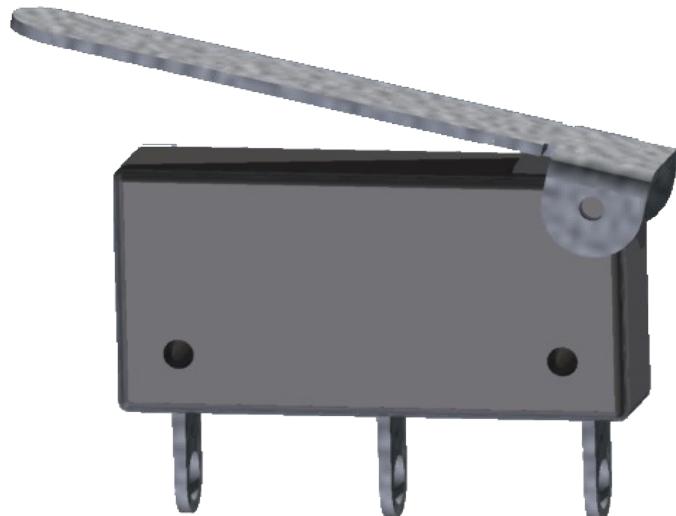


IV.5 Illustration du mécanisme

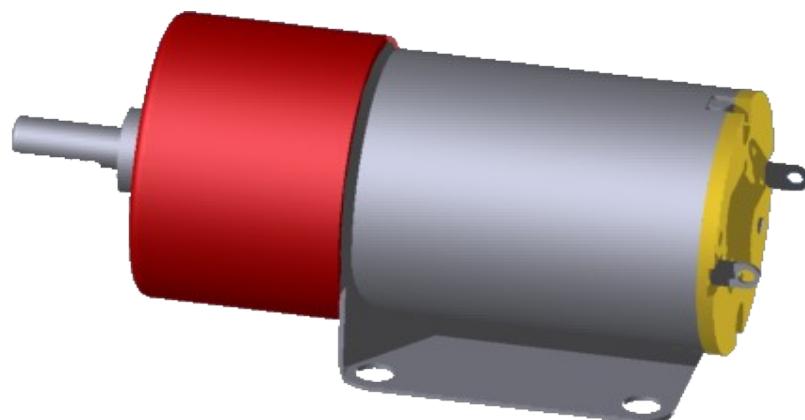


V Modélisation avec Autodesk Inventor

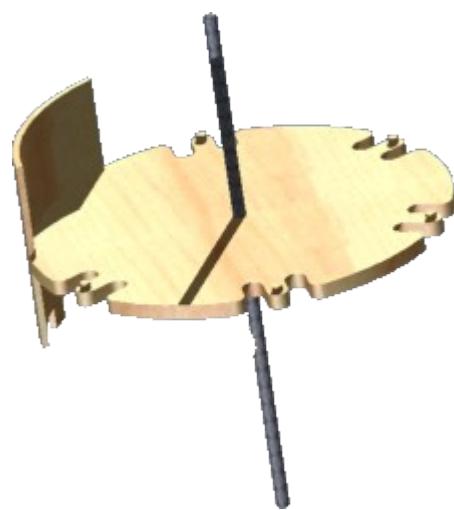
Toutes les pièces, à l'exception de la carte électronique, ont été modélisée puis assemblée sous Inventor afin de s'assurer que la réalisation réelle se ferait sans problèmes. La plupart de ces pièces vous ont déjà été présentée plus haut, voici ceux que vous n'avez pas encore vu :



L'encodeur rotatif et le microswitch



Le moteur



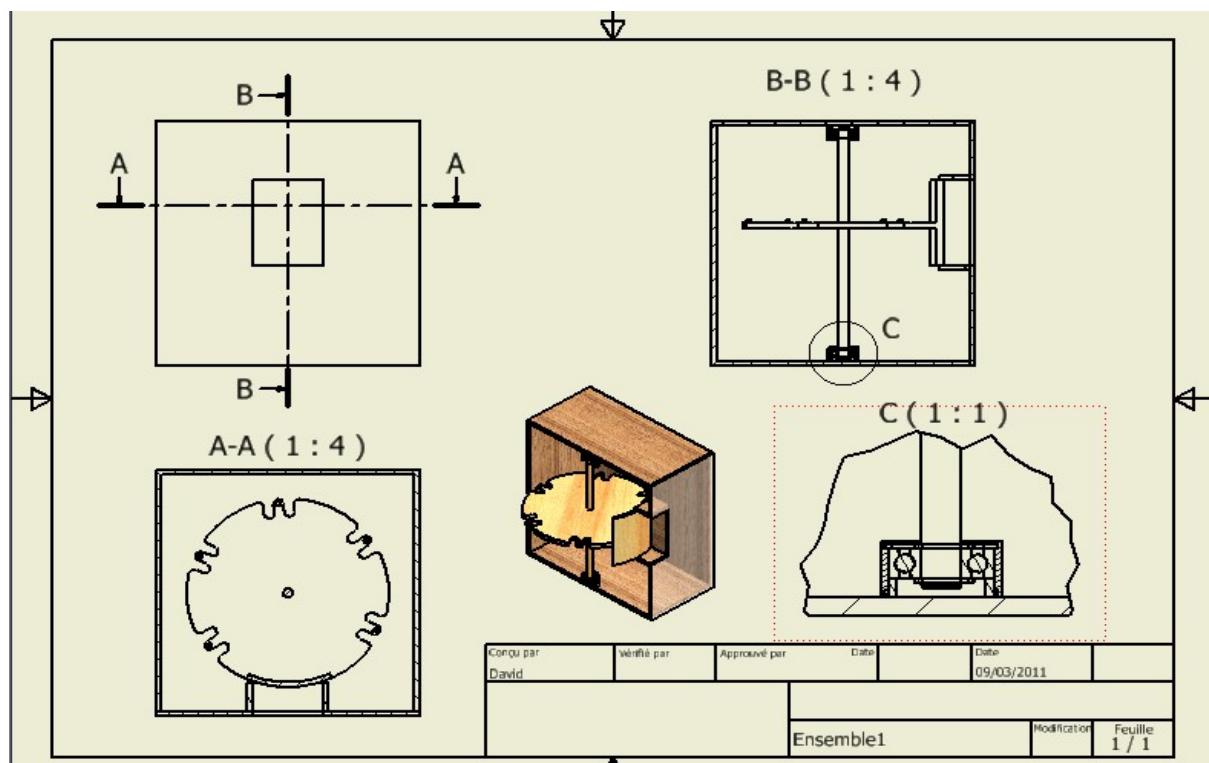
Le disque à clés

VI Présentation de la maquette:

VI.1 Réalisation sous Inventor:



VI.2 Mise en plan:



VI.3 Réalisation de la maquette réelle

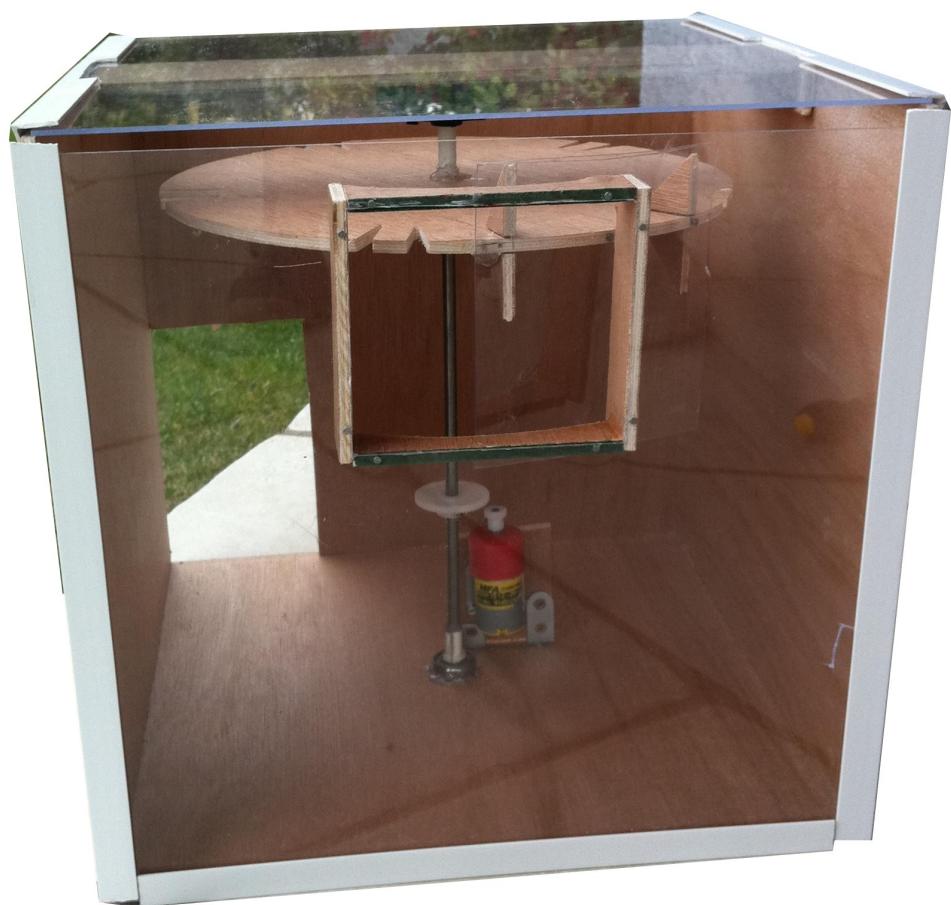


Système d'entrainement de l'axe par trains d'engrenages

Pour que la maquette soit entièrement démontable, nous avons réalisé deux systèmes d'équerres dans la maquette pour que l'on puisse enlever ou mettre en position précisément la partie supérieure afin d'assurer le bon guidage de la roue tournante.



Vue de dessus



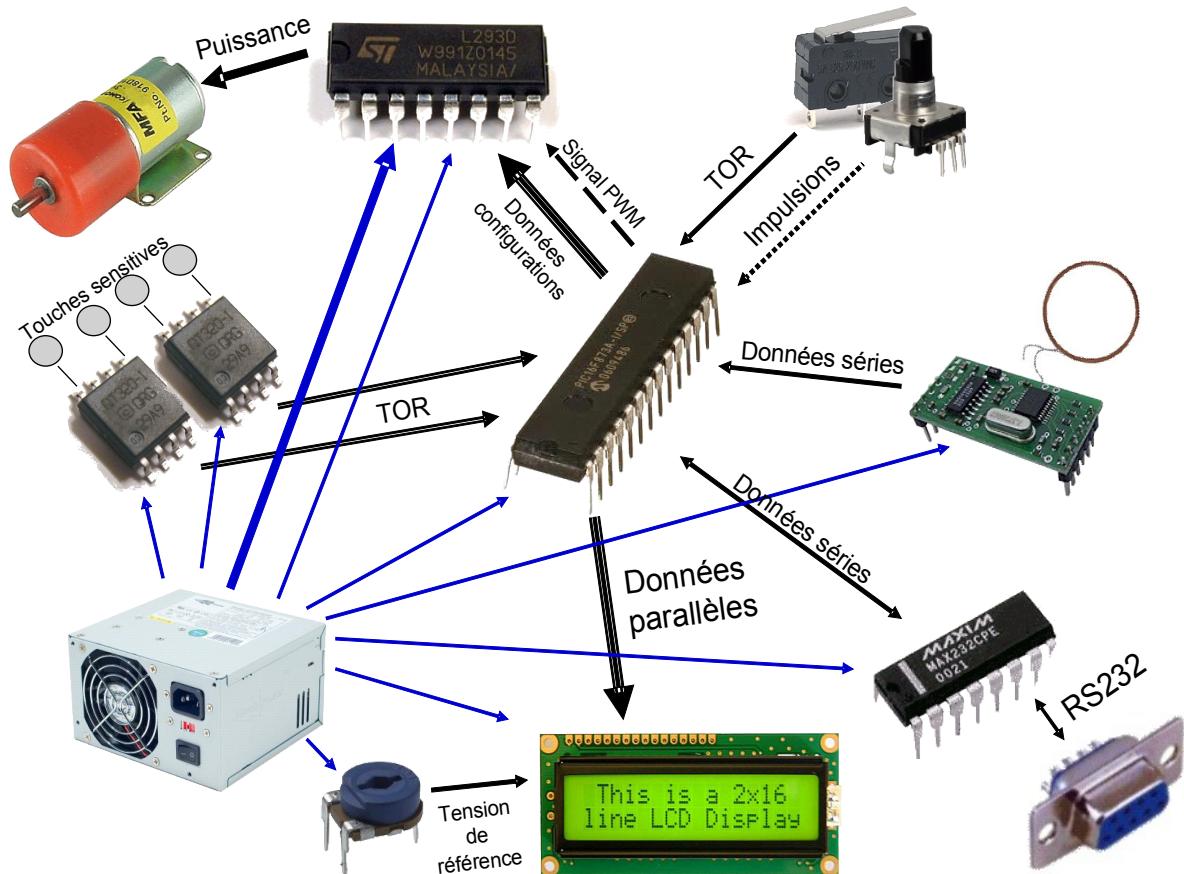
Vue de face

PARTIE C : Étude électronique

Dans cette section, la partie électrotechnique du TPE vous sera présentée.

Nous y présentons les circuits que nous utiliserons (Pont en H, module RFID, le circuit MAX232, le circuit QT320), ainsi que notre micro contrôleur le PIC 18F24K22 après avoir choisi le PIC qui répondrait le mieux à notre demande.

I Architecture électronique



Sur le schéma ci-dessus, on remarque l'architecture en étoile des composants électroniques. En effet, ils sont tous articulés autour du micro-contrôleur et communiquent avec lui. Il y a différents types de signaux qui circulent : des signaux séries, entre le module RFID et le PIC et entre le CI MAX232 et le PIC, des données en parallèles, entre l'écran et le PIC, des informations en Tout Ou Rien, et des impulsions en provenance du capteur angulaire. Le fonctionnement de tous ces composants est détaillé dans les sous-parties suivantes. Une alimentation à découpage de PC permet de fournir au composants électroniques logiques le 5V dont ils ont besoin, et aux composants de puissances le 12V.

II Étude théorique et choix des composants

II.1 Le pont en H

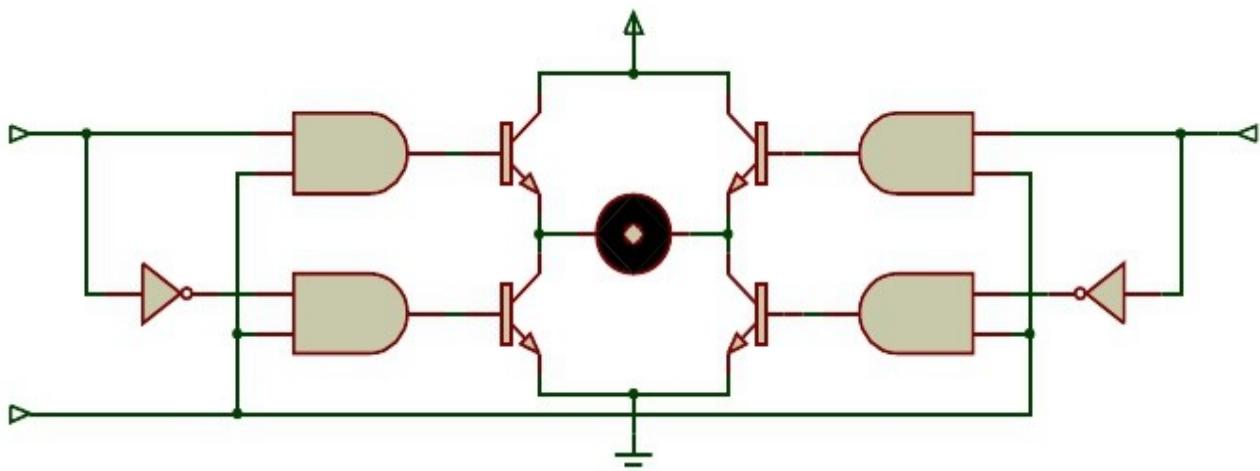
Notre système utilise un moteur pour mettre en mouvement le disque à clés. Afin de contrôler ce moteur, sa vitesse et son sens, nous avions besoin d'une interface permettant de faire le lien entre l'électronique logique (le micro-contrôleur) qui peut débiter un courant très faible, et l'électronique de puissance, qui a besoin pour fonctionner d'un fort courant. Après plusieurs recherche sur internet, nous avons opté pour un circuit de type "Pont en H", qui permet non seulement de contrôler le sens de rotation, l'arrêt et la marche du moteur, mais également, et c'est un grand avantage, de moduler la vitesse de rotation au moyen de signaux PWM.



II.1.a Description :

Le pont en H est une structure électronique qui permet de contrôler la polarité aux bornes d'un dipôle. Il est souvent utilisé pour les moteurs car il permet d'inverser le sens du courant les traversants, et donc d'inverser le sens de rotation de ces derniers. Nous avons opté pour un circuit intégré de type pont en H appelé L293. Il intègre deux ponts en H pour la commande de deux moteurs.

II.1.b Schéma de principe :



II.1.c Principe :

On peut distinguer trois entrées : IN1, IN2 et ENABLE.

ENABLE permet d'activer le circuit. Quand ENABLE est au niveau haut, les niveaux de IN1 et IN2 sont envoyés aux bornes du moteur. À l'inverse, pour ENABLE au niveau bas, toutes les portes logiques AND renvoient un niveau bas et bloquent les transistors. Donc ENABLE permet de mettre le moteur en "roue libre". A noter que l'on peut ajouter quatre diodes entre les bornes du moteur et la masse et le potentiel positif afin d'éviter d'éventuelle sur-tension dues à la rotation du moteur (en mode générateur).

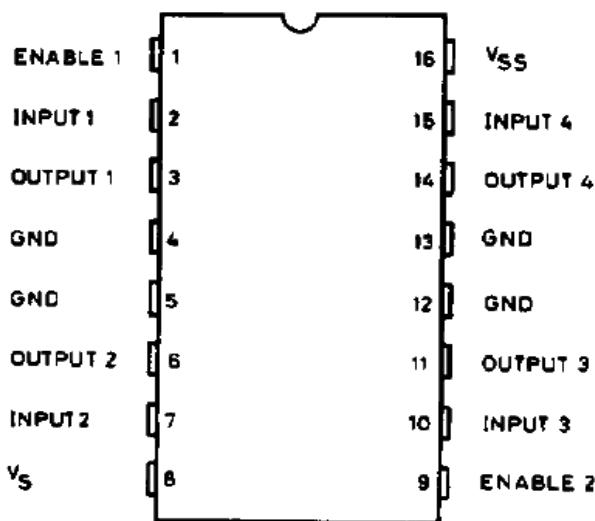
Si IN1 est au niveau logique haut et IN2 au niveau logique bas, les transistors T1 et T4 seront

passant, ce qui permettra au courant de passer et donc d'alimenter le moteur (ou dipôle). La polarisation sera donc la suivante : plus du coté gauche du moteur et moins du coté droit du moteur.

Le circuit se comporte à l'inverse si l'entrée IN2 est au niveau haut et l'entrée IN1 au niveau bas. Ce qui permet donc d'inverser le sens de rotation du moteur.

Si IN1 et IN2 sont au même niveau logique. Le potentiel aux bornes du moteur sera identique des deux cotés. Cela aura pour effet de "bloquer" le moteur ou du moins d'exercer un couple résistif sur celui-ci. On peut utiliser cette configuration pour faire un "frein moteur". Cependant, l'usage prolongé et la mise en rotation forcée du moteur malgré ce frein moteur risque de démagnétiser les aimants permanents du moteur.

La variation de vitesse est possible en modulant l'entrée ENABLE. Si on envoi un signal PWM, en faisant varier le rapport cyclique de celui-ci, on fait varier la valeur moyenne de la tension aux bornes du moteur. Nous utiliserons le module Compare Capture & PWM du PIC pour générer le signal PWM. Il faudra peut être envisager de placer un condensateur en sortie du L293 pour aplatisir le signal. Des tests nous révélerons l'utilité ou non de ce condensateur.



II.2 Le module RFID

II.2.a Description du module :



Le module UM-005 de Nextronic, permet de mettre en œuvre facilement l'identification par radio-fréquences dans un montage électronique. Il est connecté d'une part à une antenne qui génère un champs magnétique qui alimente par induction les TAGS RFID à proximité, et d'autre par, d'une sortie série qui délivre les signaux véhiculant les informations sur les TAGS RFID scannés.

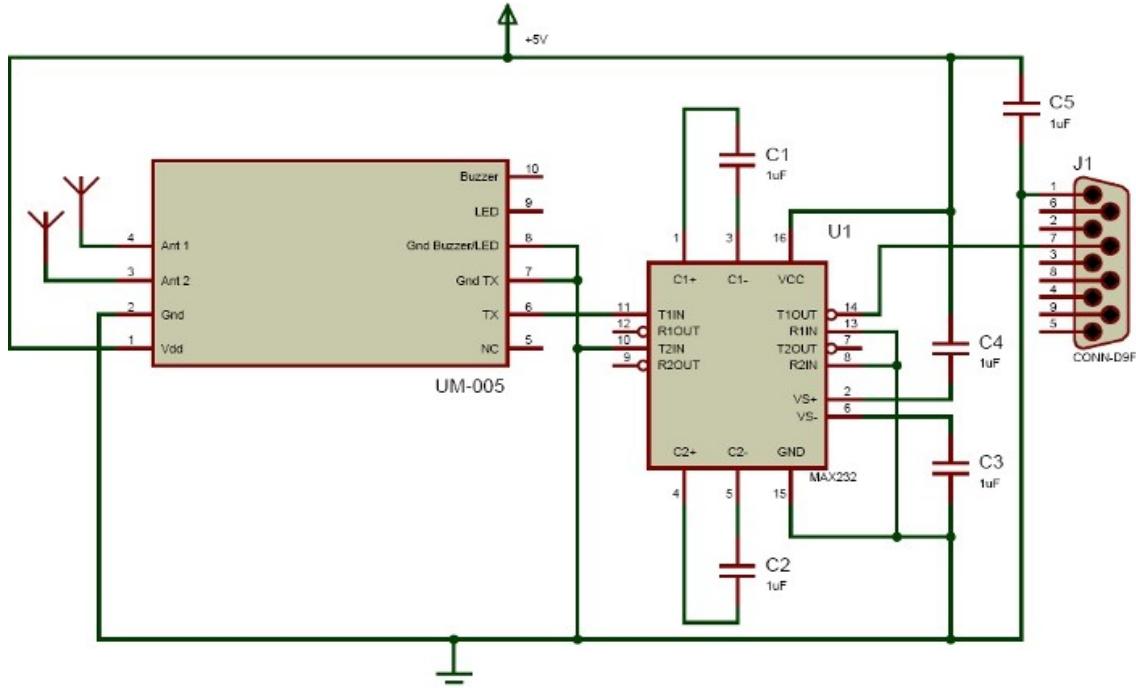
Les Tags RFID qui se trouvent à proximité du module sont alimentés par le champs magnétique généré par son antenne et dès lors, renvoient un identifiant unique qui est capté par le module. Celui-ci génère alors des signaux que l'on récupère avec un micro contrôleur ou un ordinateur.

II.2.b Les tests du module :

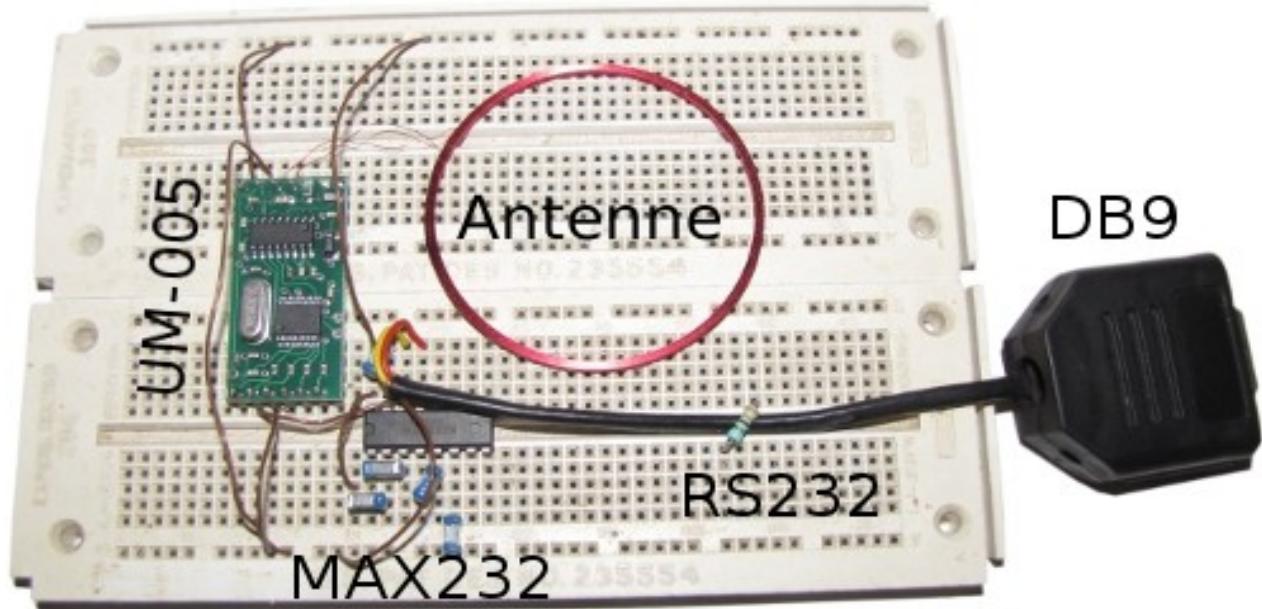
Pour tester le module et analyser les trames qu'il transmet, nous avons réalisé un petit montage électronique comportant un circuit MAX232 et quelques condensateurs. Le module délivre des signaux série TTL (0 à 5V) alors que le PC utilise le format RS232, pour lequel les signaux doivent

varier de -12V à 12V. Pour effectuer la conversion, on utilise un circuit spécialisé qui, couplé à 5 condensateur de 1µF, s'occupe d'inverser et de convertir les signaux.

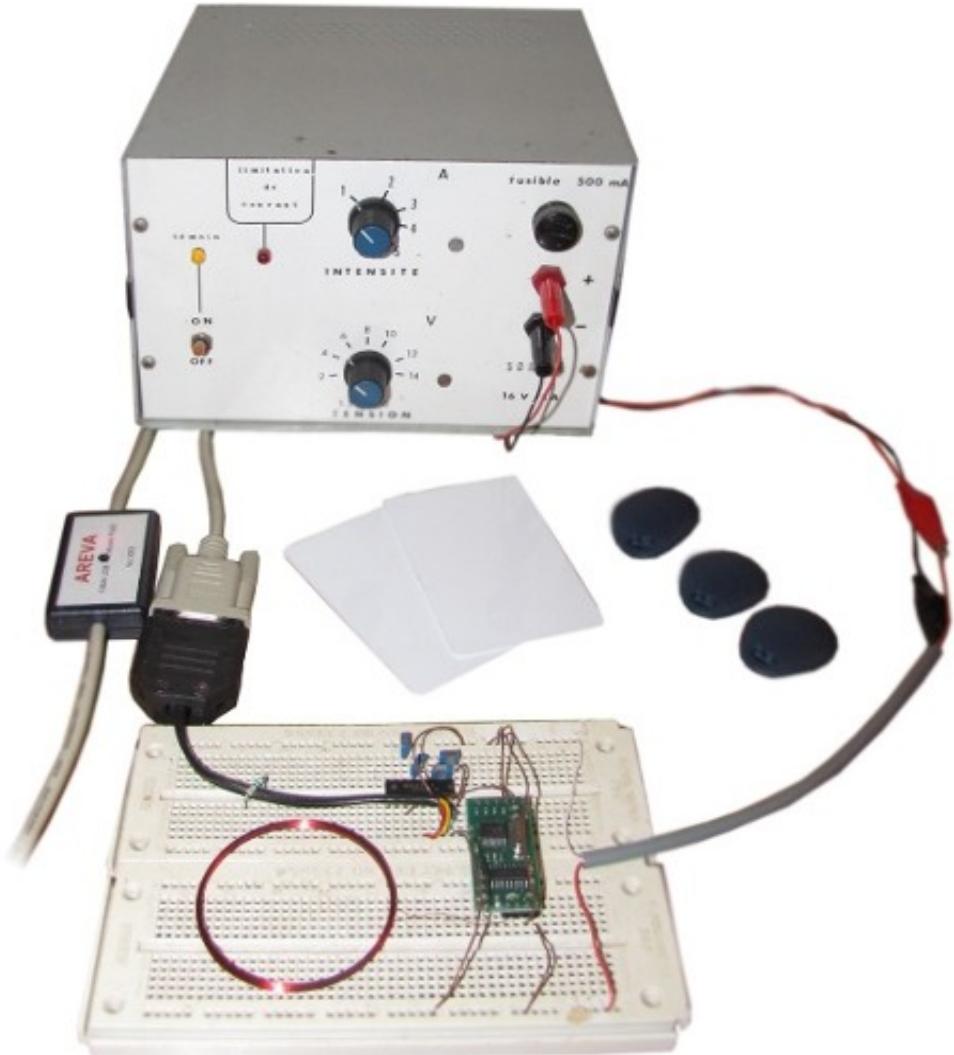
Le schéma mis en œuvre est le suivant :



Après prototypage sur plaquettes d'expérimentation, cela donne :

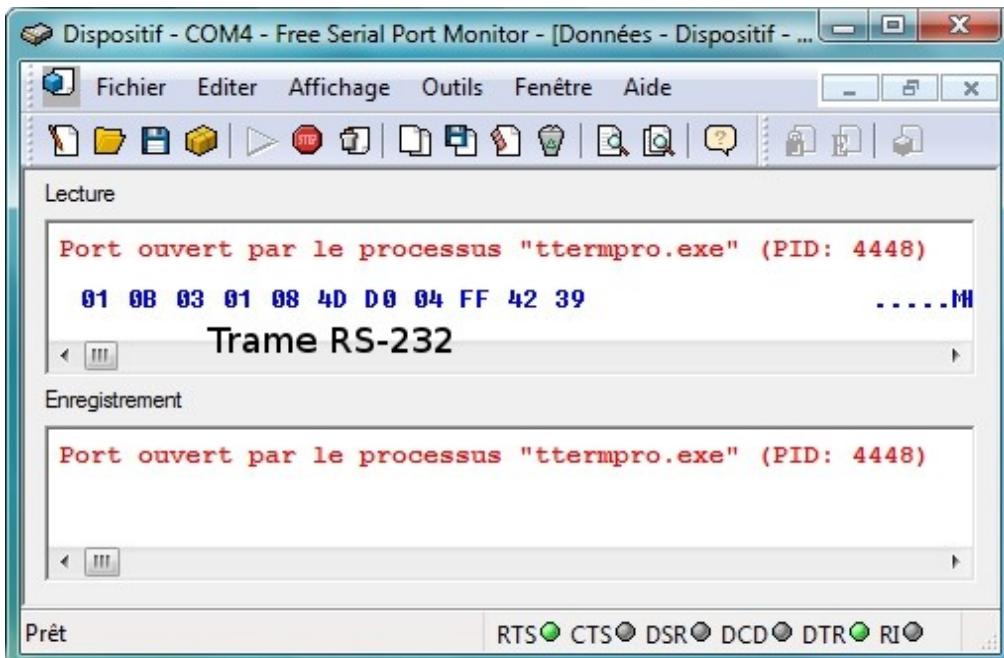


Le montage est connecté au PC par l'intermédiaire d'un convertisseur USB-Série :



Après avoir alimenté et connecté le montage au PC. Le logiciel d'analyse de trames RS232 est lancé. Nous passons ensuite un TAG RFID à proximité de l'antenne du module. Celui-ci renvoi alors sur son port série l'identifiant du TAG. Le circuit intégré MAX232 se charge de convertir les signaux au format RS232. Ils ont alors envoyé à un convertisseur USB-Série car le PC utilisé pour les tests ne disposent pas de la connectique appropriée.

Les trames sont alors exploitables dans le logiciel Free Serial Port Monitor :



La description de trames et la signification des différents octets sont donnés dans la datasheet :

Frame format for serial transmission

Module address	Frame width	Response	Data	Operation code	CRCH	CRCL
1 byte	1 byte	1 byte	n bytes	1 byte	1 byte	1 byte

Where during read-out from transponder:

Module address - 0x01 always

Frame width – total number of response frame bytes = 0x0b

Response - 0x01

Data - ID1...5 – transponder ID (5 bytes)

Operation code – 0xff

CRCH, CRCL - CRC16 MSByte and LSByte respectively

Le module doit être alimenté en 5V et la liaison série est à configurer à 9600BPS, 1 Bit de STOP, pas de bit de parité, comme le précise la datasheet du module :

Dane techniczne

Supply voltage Vdd:	4.5...5.5V
Supply current:	5...55 mA
Module rated operating radio frequency:	125 kHz
Modulation type of data received from transponder:	Manchester
Baud rate of data received from transponder:	RF/64 (1953 b/s)
Maximum read-out frequency:	2 read-outs/sec
Output current capacity buzzer, LED and RS-TX:	5 mA
Transponder read-out distance (depending on used antenna):	up to 12 cm
Antenna inductance	1 mH +5%
RS232 transmission:	9600 b/s, 8 data bits, 1 stop bit, no parity bit, with voltage levels TTL format compatible.

II.3 Le micro contrôleur

II.3.a Introduction

Notre système devra gérer des communications avec un serveur, des moteurs, des capteurs, les utilisateurs et les clés... Pour cela, il nous faut une carte électronique capable de gérer ces aspects. Nous ne pouvons pas réaliser des fonctions (communication avec un serveur, l'identification d'utilisateurs, etc...) de manière simple avec des portes logique : la mise en œuvre serait titanique. Il nous faut donc un circuit intégré, capable de gérer des signaux logiques complexes (binaires) et de les analyser dans le temps : ce circuit est appelé micro-contrôleur.

Le fonctionnement d'un micro-contrôleur, sans rentrer dans les détails, est comme ceci :

1. Il effectue des calculs très rapidement qui définissent son fonctionnement.
2. Sa vitesse de calcul, appelé fréquence (donc exprimé en Hertz), est définie par un oscillateur : à chaque impulsion de l'oscillateur, le micro-contrôleur effectue un calcul.
3. Il comporte des modules qui effectue des actions spécifiques utiles au concepteur (timers pour la gestion du temps, port série pour les communications, convertisseur analogique/numérique, etc...)
4. Ses pattes (hormis les pattes d'alimentation) sont réparties en ports de 8 bits qui (dans la plupart des cas) peuvent être configurés en entré, en sorti ou affecté à un module.
5. Le comportement du micro-contrôleur est défini dans son code source, écrit en assembleur par le programmeur sur un PC
6. Le code source est transféré dans le micro-contrôleur (que l'on écrira dorénavant μ C) l'aide d'un programmateur relié au PC.



Il existe beaucoup de micro-contrôleurs, pour notre application nous avons choisi de restreindre notre comparaison à ceux commercialisé par la société américaine Microchip, spécialisée et leader dans ce domaine, pour deux raison : leurs simplicité et leurs coût modique. Les micro-contrôleurs qu'ils proposent sont appelés PIC. Ils se présentent en plusieurs gammes. Nous nous intéresserons aux μ C de milieu de gamme, dénommée chez Microchip PIC18. Pour faire un choix dans cette vaste gamme, nous allons définir les caractéristiques minimales que notre PIC devra vérifier.

II.3.b Choix du micro-contrôleur

Caractéristiques :

1. Alimentation de 5V
2. 20 entrées/sorties logiques minimum
3. 2 modules de communication série (USART)
4. Un timer, peu importe la résolution, minimum
5. Être programmable en circuit (ICSP) par un PICKit3, et même reprogrammable (technologie Flash)
6. Une mémoire programme (Flash) de 8Ko minimum
7. Une mémoire RAM de 500 octets minimum

Après avoir entré ces données dans le comparateur fourni par Microchip, nous avons constaté que

les µC de la série K22, respectivement 18F23K22, 18F24K22 et 18F25K22, correspondaient à nos attentes, avec par ordre croissant de la mémoire Flash et RAM supplémentaire. Nous avons donc opté pour le PIC intermédiaire : le PIC18F24K22

Nota : Malheureusement, ce PIC n'est pas simulable avec ISIS. Nous devrons donc tester la carte électronique en simulant nous même les réactions du PIC et en testant sur prototype les communications séries...

Le fonctionnement d'un µC est très complexe. Il est décrit dans un document qui est propre à chaque µC, fourni par le constructeur, appelé datasheet. On y trouve la description de toutes les caractéristiques, tous les modules, etc... En définitive elle renferme toutes les informations dont le développeur a besoin, tant sur le plan logiciel (Software) qu'électronique (Hardware). Il convient donc de s'y reporter régulièrement lors de l'étude pour ne pas faire d'erreurs.

Dans le cas de notre PIC18F24K22, la datasheet fait 496 pages.

II.3.c Programmation

L'écriture du code source se fera avec le logiciel de Microchip MPLAB. Et la programmation du PIC à l'aide d'un PICKit3 qui fait également office de débugger.

Le langage assembleur, compris par l'µC, est de moins en moins utilisé par les développeurs. On lui préfère le langage C, plus flexible et plus structuré. Afin de convertir le code en C en assembleur, on utilise un compilateur. Dans notre cas il s'agira du compilateur C18, toujours de Microchip.

II.4 Interface utilisateur

II.4.a Introduction

Dans l'étude fonctionnelle de notre système, et plus particulièrement dans le diagramme pieuvre, on peut constater que les interactions avec l'utilisateur tiennent une place prépondérante. En effet, c'est en fonction de l'utilisateur et de ses choix que la borne va réagir. Il convient donc de permettre cet échange homme-machine, le plus instinctif et ergonomique possible.

Pour cela, il faut tout d'abord un moyen pour communiquer des informations à l'utilisateur. On peut imaginer plusieurs solutions mais dans notre cas, il s'agira d'un écran. Ensuite, il faut pouvoir récupérer les ordres de l'utilisateur. Pour cela nous avons plusieurs choix : la molette rotative, les boutons poussoirs ou les touches sensitives. Nous détaillerons ces solutions techniques plus bas.

Ces périphériques (écran, touches ou mollette) seront évidemment reliés au micro-contrôleur.

II.4.b L'écran :

Pour choisir l'écran de notre borne d'auto-partage, il faut prendre en considération plusieurs contraintes :

1. Lisibilité
2. Prix
3. Facilité de mise en œuvre

Il existe un grand nombre d'écran différents, nous nous intéresserons particulièrement aux écran LCD dit "alphanumériques", simplement car ils permettent d'afficher uniquement des caractères appartenant à l'alphabet latin, des chiffres, et quelques caractères spéciaux. Ils sont à différencier des écrans graphiques qui eux permettent d'afficher des images "point-par-point" : on parle de pixels.

Les écran alphanumériques sont un bon compromis entre lisibilité et prix. Leur mise en œuvre est relativement aisée avec un µC. De plus, compte tenu de leur faible encombrement, il peuvent parfaitement s'intégrer dans notre borne. Nous définirons d'ailleurs un nombre de caractères que l'on souhaite afficher : deux lignes de 16 caractères semble un choix judicieux si on prend en compte le rapport lisibilité/prix.



II.4.c L'acquisition

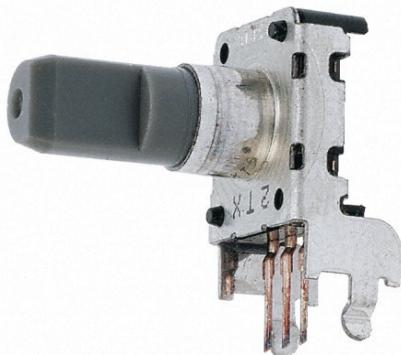
Nous avons besoin d'acquérir quatre ordres différents : précédent, suivant, valider, appel.

Comme indiqué plus haut, nous avons envisagé trois solution techniques répondant à notre problématique.

La molette rotative

Son intérêt réside dans le fait qu'elle centralise les trois ordres (précédent, suivant, et valider) sur un même bouton. En effet, on tourne la molette sur la droite -> suivant, on tourne sur la gauche -> précédent, et on presse la molette -> valider. Son prix est faible, d'autant plus qu'elle opère pour trois ordres à transmettre.

Ses inconvénients sont : sa mise en œuvre nécessite presque toujours l'emploi d'une bascule D pour détecter le sens de rotation. Il faut rajouter un bouton poussoir pour la touche "Appel". Très utilisée dans les années 80, elle est désormais "ringarde" et on lui préfère les touches sensitives.



Les boutons poussoirs

Ils ont l'avantage d'être peu coûteux, facile à mettre en œuvre car ne nécessitant aucun autre composant, et peuvent même être esthétiques. Cependant, ils sont trop "classiques", et on considère que désormais, les boutons poussoirs tendent à disparaître au profit des touches sensitives, bien plus esthétiques.



Les touches sensitives

Elles permettent de détecter l'appui ou l'effleurement d'une surface, sans mise en mouvement de pièces mécaniques et sans forte pression sur la surface. Il existe plusieurs technologies qui permettent de réaliser ce type de touches mais la plus au point et sans doute la détection de variation de capacité électrique. L'avantage est que le capteur se place derrière la surface isolante que l'on souhaite transformer en touche sensible.

Les touches sensitives étaient avant utilisées lorsque les boutons poussoirs ou touches mécaniques ne pouvaient l'être (par exemple : environnement explosif, médical, aquatique, ou encore anti-vandalisme), mais elles sont désormais de plus en plus utilisées à des fins esthétiques ou ergonomiques. On en trouve par exemple sur les plaques de cuisson, les baladeurs, certaines chaînes hi-fi, etc...

Naturellement, c'est donc ce type de touches que nous choisirons pour notre système, bien que la mise en œuvre soit plus complexe qu'avec les boutons poussoirs.

II.5 L'écran LCD

Pour dialoguer avec l'utilisateur, nous avons besoin d'un dispositif capable de restituer des informations. Pour cela nous envisageons d'utiliser un écran LCD. Nous aurions pu utiliser des voyants ou DEL mais ceux-ci limitent les informations à afficher à de simples états. Hors, nous avons besoin d'afficher des messages, des chiffres, etc...

II.5.a Utilisation de l'écran

Il existe dans le commerce des écrans LCD capables d'afficher plusieurs caractères pré-définis. Ces écrans ont un contrôleur intégré qui sert d'interface entre l'écran physique et la partie commande. Il existe deux grandes familles d'écran LCD : les écrans dits "graphiques" et les écrans qui ne peuvent afficher que des caractères alphanumériques. Ces derniers sont plus faciles à contrôler.

La plupart de ces écrans communiquent par port parallèle, c'est à dire que les bits de commandes sont

envoyés au même moment par le biais de plusieurs liaisons. Nous connecterons l'écran directement au micro contrôleur.

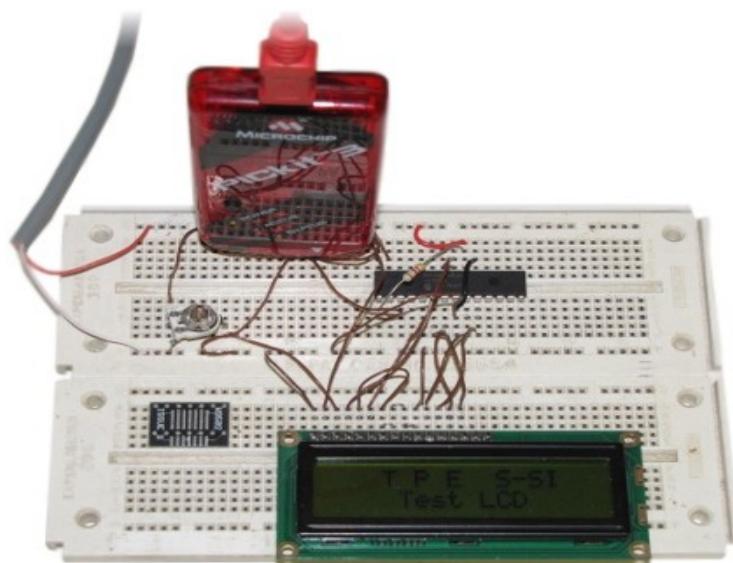
Ces écrans peuvent être commandés soit avec huit bits de commandes, soit avec deux fois quatre bits de commande, auquel cas les bits les moins significatifs sont envoyés les premiers. Une fois les bits de données positionnés, on passe la patte d'activation à l'état haut, ce qui à pour effet de prendre en compte la commande. La patte d'activation (appelée Enable) doit être activée au minimum 600nS.

II.5.b Initialisation de l'écran

Une séquence de démarrage est nécessaire pour indiquer à l'écran qu'on l'utilise en mode 4 bits, sur 2*16 caractères, etc... Cette séquence est la suivante :

1. Forçage en mode 8 bits
2. Passage en mode 4 bits
3. Configuration du nombre de ligne
4. Configuration du curseur
5. Passage en mode entrée, activation du défilement du curseur
6. Effacement de l'écran

II.5.c Tests de l'écran LCD :



II.6 Les touches sensitives

II.6.a Choix des touches sensitives

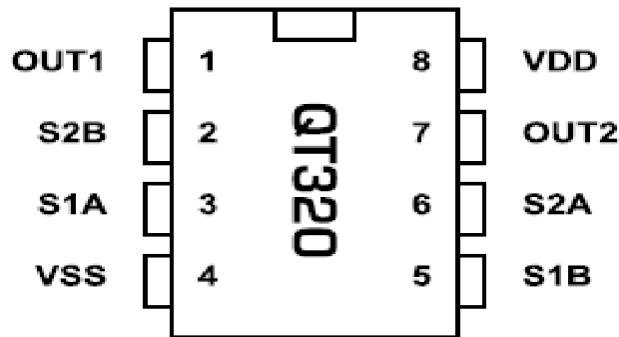
Nous avions décidé préalablement que nous utiliserions des touches sensitives pour l'interface utilisateur de notre projet. Tout d'abord car elles sont similaire, dans le fonds, aux boutons poussoirs et surtout à des fins esthétiques. C'est d'ailleurs une solution technique choisie par de plus en plus de systèmes dans la vie de tous les jours (plaques chauffantes, baladeurs, etc...). Ce système est relativement complexe à mettre en œuvre. Cependant, il existe dans le commerce des solutions "clés en main" pour utiliser des touches sensitives. Il existe plusieurs technique pour déployer des touches sensitives. La plus fiable et répandue détecte les variations de capacitance par le biais d'une électrode. C'est cette technique que nous utiliserons.

Pour déployer les touches tactiles, nous utiliserons un circuit nommé QT320 de chez Atmel qui

permet de gérer deux touches sensitives. Notre TPE nécessite l'emploi de 4 touches, nous utiliserons deux circuits QT320.

A noter que nous aurions pu utiliser un périphérique du PIC qui mesure la capacitance. Cependant, cette solution était (beaucoup) plus difficile à mettre en œuvre et, outre le fait que la fiabilité est moindre, nécessitait un travail très important au niveau logiciel pour déceler, calibrer et corriger les variations de capacités, qu'elles soit intempestives ou volontaires.

II.6.b Présentation du QT320



Le QT320 est un circuit intégré qui permet de gérer des touches sensitives. Il peut utiliser gérer 2 touches sensitives.

Il dispose de huit pattes :

- 2 Pattes pour l'alimentation (VDD et VSS)
- 4 pattes pour la réception des informations en provenance des électrodes (2 par canaux)
- 2 pattes de sorties logique "Output"

Voici un schéma de mise en œuvre :

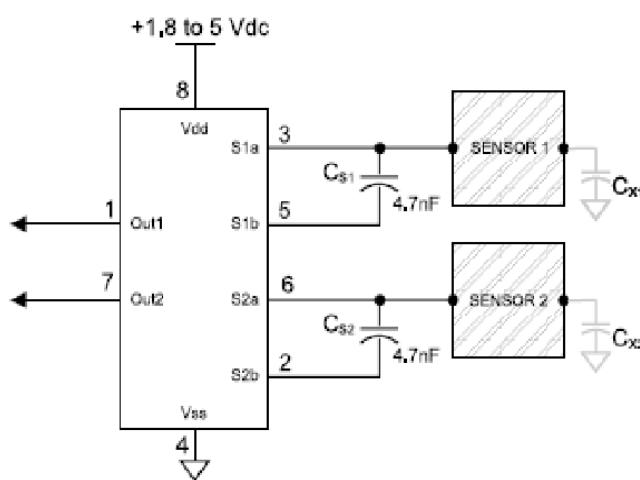


Figure 1-1 Basic QT320 circuit

Comme vous pouvez le constater, la mise en œuvre est relativement simple car le seul composant externe à rajouter pour rendre opérationnel le CI est un simple condensateur. La conception des électrodes est très ouverte : toutes les formes sont permises, de toutes tailles, et le choix du matériau entre l'électrode et l'utilisateur sont quasi-libre.

II.6.c Fonctionnement

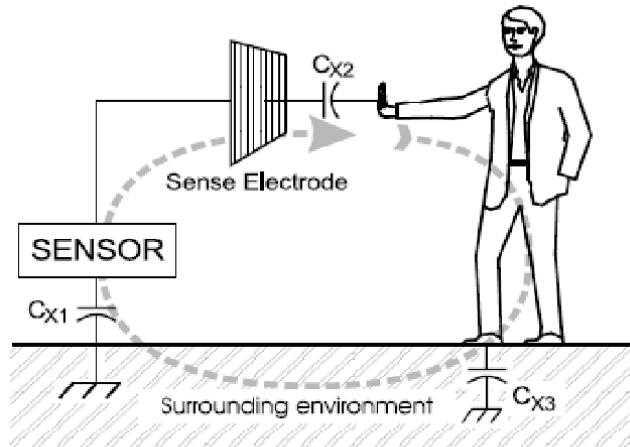


Figure 1-4 Kirchoff's Current Law

Nous pouvons observer que lorsque l'utilisateur touche avec son doigt le capteur, la variation de capacitance est détectée par le capteur et celle-ci est transmise de manière logique au PIC.

Dans certains cas, il peut être souhaitable de faire varier la sensibilité, par exemple lors de l'utilisation de surface isolante très épaisse devant le capteur, ou encore pour capter des objets à faible capacité électrique. Nous n'aurons vraisemblablement pas à utiliser cette option. Si toute fois nous le voulions, on pourrait jouer sur la valeur de seuil du signal qui est sur une échelle de 1 à 16. Par défaut la valeur est définie à 10.

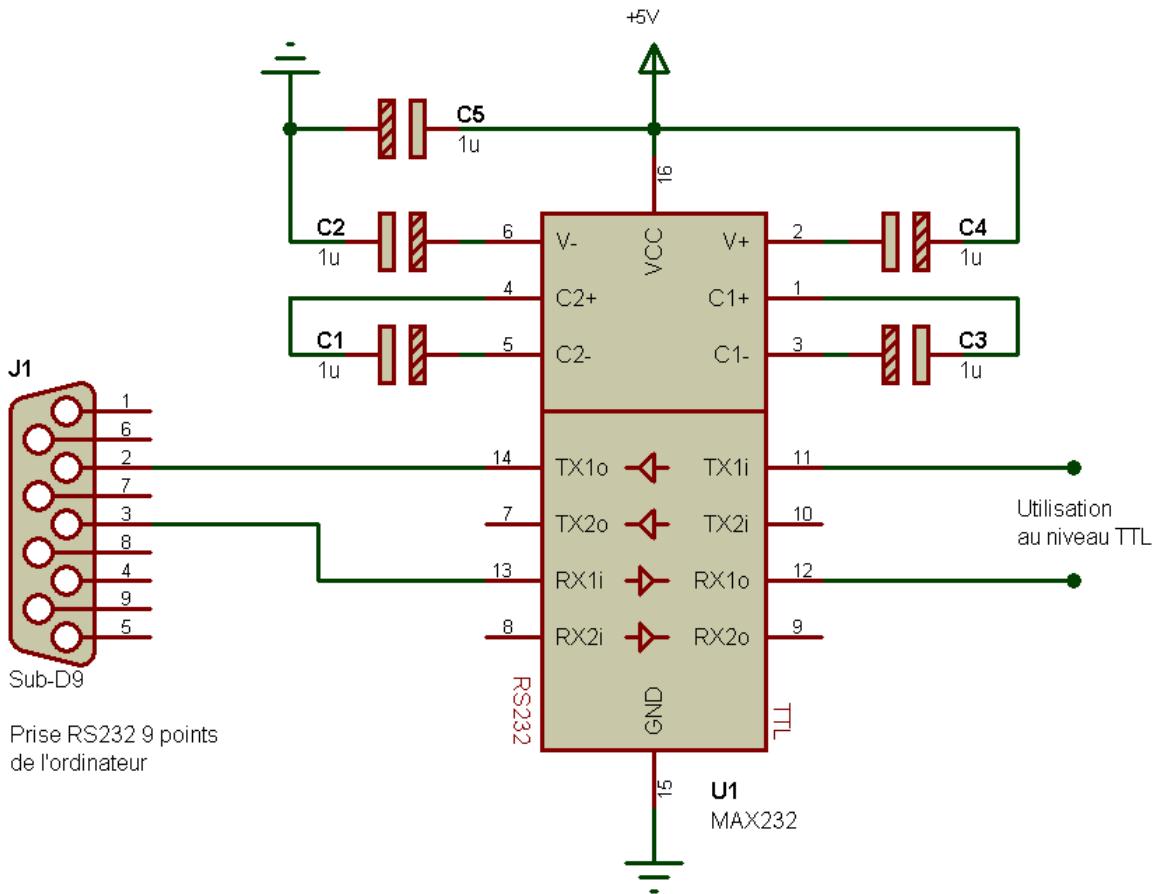
Le composant se calibre automatiquement au démarrage et lors d'une modification prolongée de la capacitance. Il n'y a donc, normalement, pas à intervenir sur le CI pour un quelconque réglage. C'est là que réside l'intérêt d'utiliser un tel composant.

II.7 Le circuit MAX232

Dans notre projet, le PIC communique avec le serveur via une liaison série. Les signaux générés par le module USART du micro-contrôleur sont de type TTL, soit de 0 à 5V. Cependant, cela n'est pas compatible avec la norme RS232 qui, d'un point de vue électrique, utilise des signaux logiques entre -10 et +10V. Il faudra donc transformer cette liaison série TTL en liaison série RS232. Pour ce faire nous utiliserons le très connu MAX232 créé par MAXIM qui permet la conversion. Il ne nécessite qu'une alimentation de 5V et cinq transistors pour être opérationnel.



II.7.a Schéma de câblage :



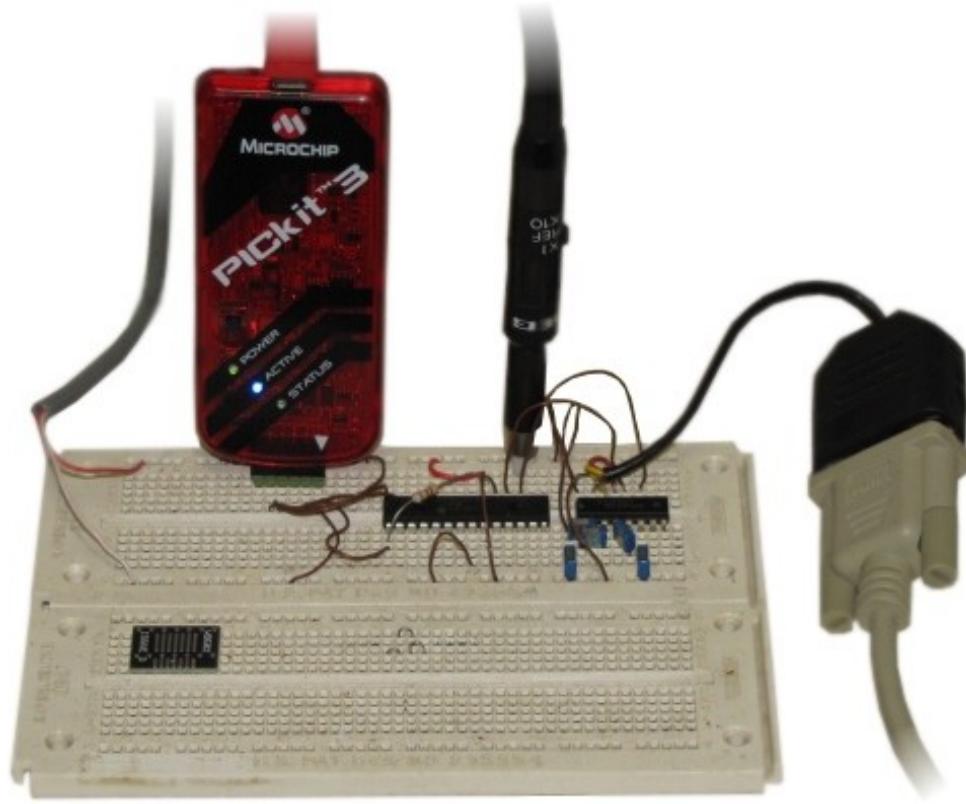
II.7.b Explications :

Comme on peut le constater sur le schéma ci-dessus, le MAX232 fonctionne grâce aux condensateurs qui lui permettent d'élever la tension de 5V à 10V et -10V. En plus d'adapter la tension, le circuit inverse les niveaux logique car la norme RS232 l'impose.

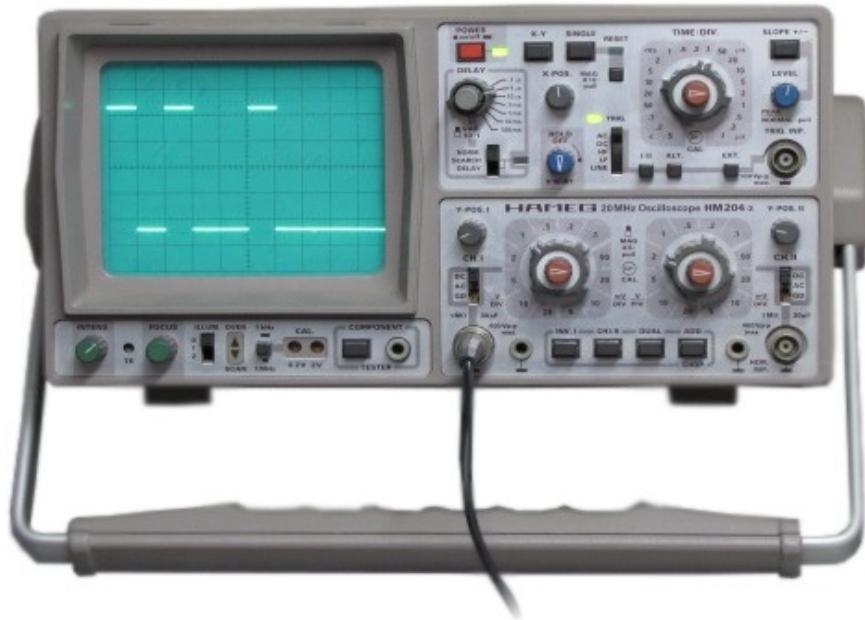
Ce circuit sera connecter au PC serveur par l'intermédiaire d'une prise DB9 et éventuellement grâce à un convertisseur USB-Série.

II.7.c Tests du circuit

Nous avons réalisé les tests avec un ordinateur, le logiciel Free Serial Port Monitor et un boîtier de conversion RS232-USB. Le MAX232 était connecté à un micro-contrôleur qui envoyait des signaux série TTL.



Montage sur plaquette de prototypage



Visualisation des trames RS232

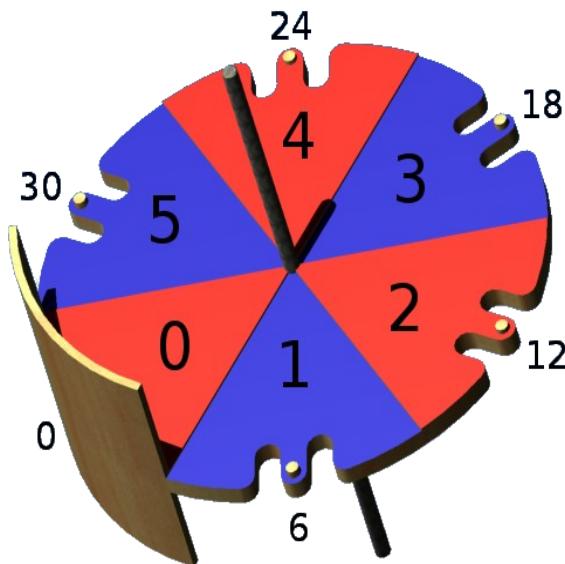
II.8 Contrôle de positionnement du disque

Comme nous l'avons vu précédemment, le moteur est commandé par un pont en H de type L293D. Ce circuit imprimé permet de moduler la vitesse de rotation du moteur en envoyant sur le signal d'activation du CI (patte EN) un signal PWM.

II.8.a Le signal PWM

Il s'agit d'un signal logique périodique rectangulaire à rapport cyclique variable. Le fait de moduler le rapport cyclique fait par conséquent varier la valeur moyenne de la tension aux bornes du moteur. Il faudra éventuellement prévoir un condensateur aux borne du moteur pour "aplatir" ce signal carré, de façon à récupérer uniquement la tension moyenne.

Pour connaître la position du disque et commander le moteur en conséquence, nous utiliserons deux capteurs, un microswitch qui nous indiquera le disque en position "zéro" c'est à dire clés inaccessibles, et un encodeur rotatif. Il s'agit d'un capteur qui s'apparente à un potentiomètre. Sa différence avec ce dernier réside dans le fait qu'il délivre des impulsions. Il possède trois pattes, une commune et deux phases. Ces sorties sont légèrement décalées (on appelle cela le déphasage) ce qui permet de connaître le sens de rotation de l'encodeur, nous utiliserons qu'une seule phase. Le modèle retenu pour notre projet est un encodeur à 24 impulsions par tour. Par un simple calcul, on peut déterminer une correspondance impulsions/angle. Sur l'image ci-dessous, les emplacements du disque de 0 à 5 et le nombre d'impulsions délivrer par le capteur pour une rotation à partir de la position "zéro" dans le sens direct.

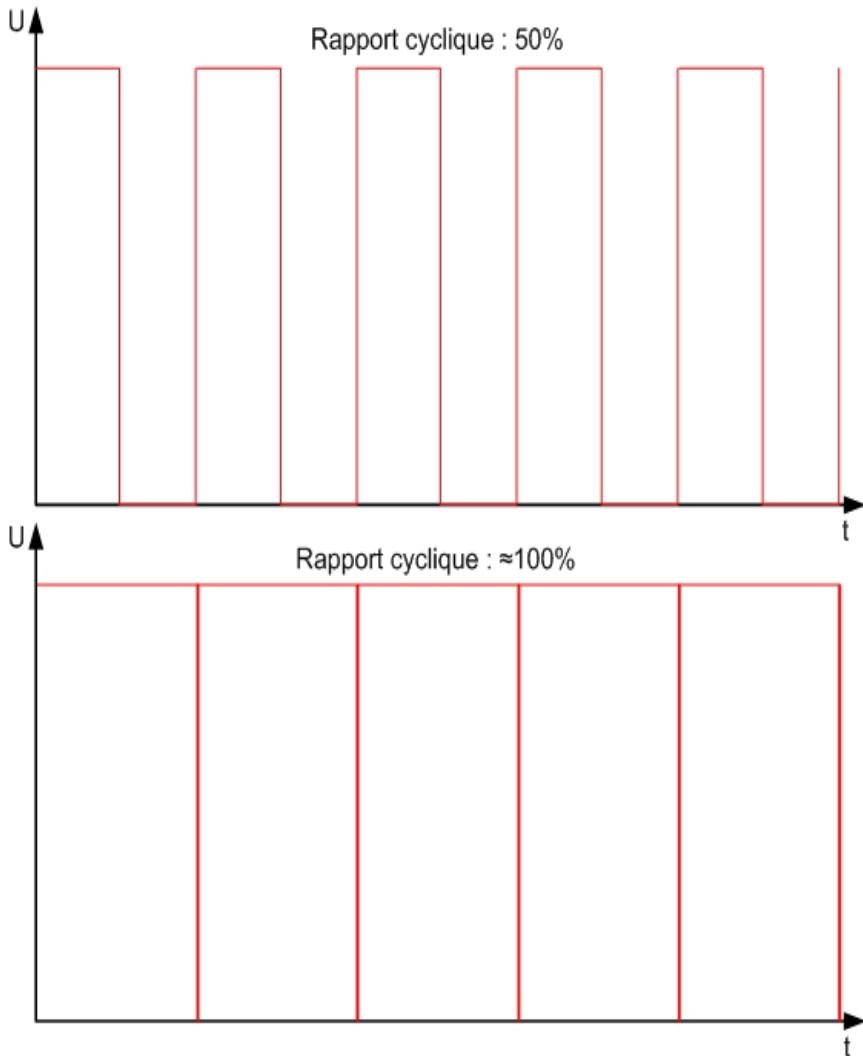


II.8.b Génération des signaux PWM

Pour générer les signaux PWM, nous utiliserons le module CCP (Compare Capture & PWM) du micro-contrôleur. Il est facile à mettre en œuvre et flexible. On peut paramétriser la fréquence, la période et par conséquent le rapport cyclique. Pour notre application, la vitesse du moteur variera de 50 à 100% en phase d'accélération et de 100 à 50% en décélération.

Le disque entrera toujours en rotation en accélérant. Environ 30° avant d'atteindre sa position de consigne, le disque décélérera de façon progressive pour éviter tout variation brutale de vitesse. La détection de la zone de décélération s'effectuera conformément aux informations transmises par l'encodeur rotatif.

Voici un exemple de signaux PWM transmis par le PIC :



Dans le premier cas, le moteur tourne à 50% de sa vitesse nominale car dans le cas d'un moteur C.C., celle-ci est proportionnelle à sa tension d'alimentation. Dans le second cas, sa vitesse est normale.

III Réalisation pratique et schéma électronique

III.1 Schéma électronique

Le schéma électronique a été réalisé grâce au logiciel ISIS.

Le schéma est disponible en annexe 4 de ce dossier.

III.2 Réalisation de la carte électronique

III.2.a Conception assistée par ordinateur

Nous avons conçu les fichiers typons grâce au logiciel Ares. Les NetLists ont été importées de ISIS puis nous avons effectué le placement routage manuellement.

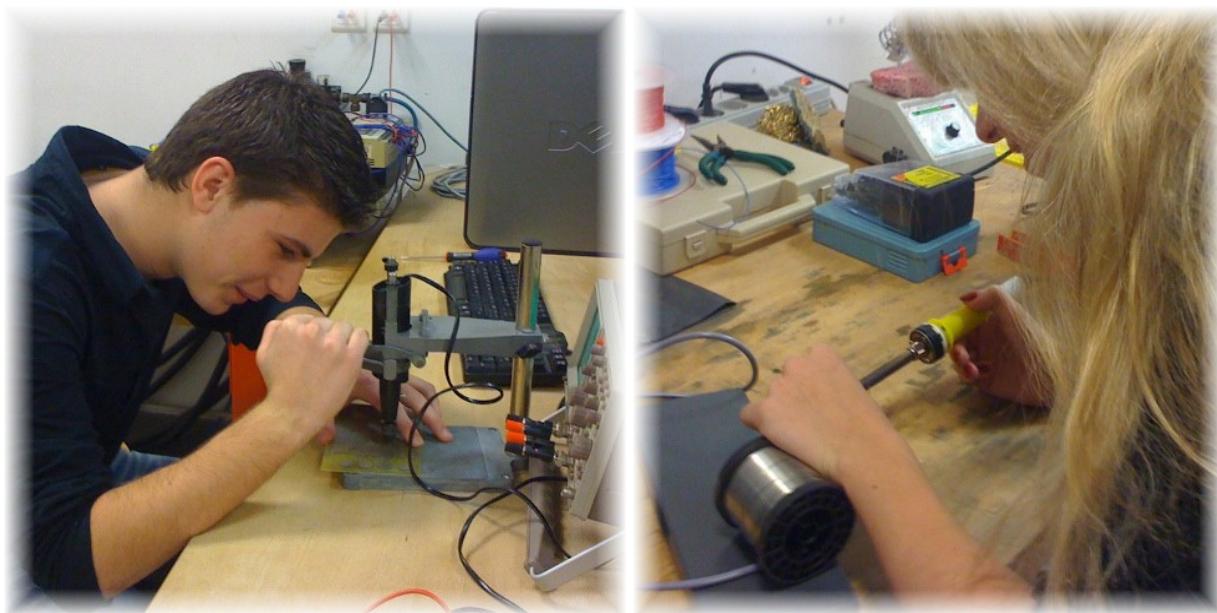
Les typons sont disponibles en annexe 5 de ce dossier

III.2.b Réalisation pratique

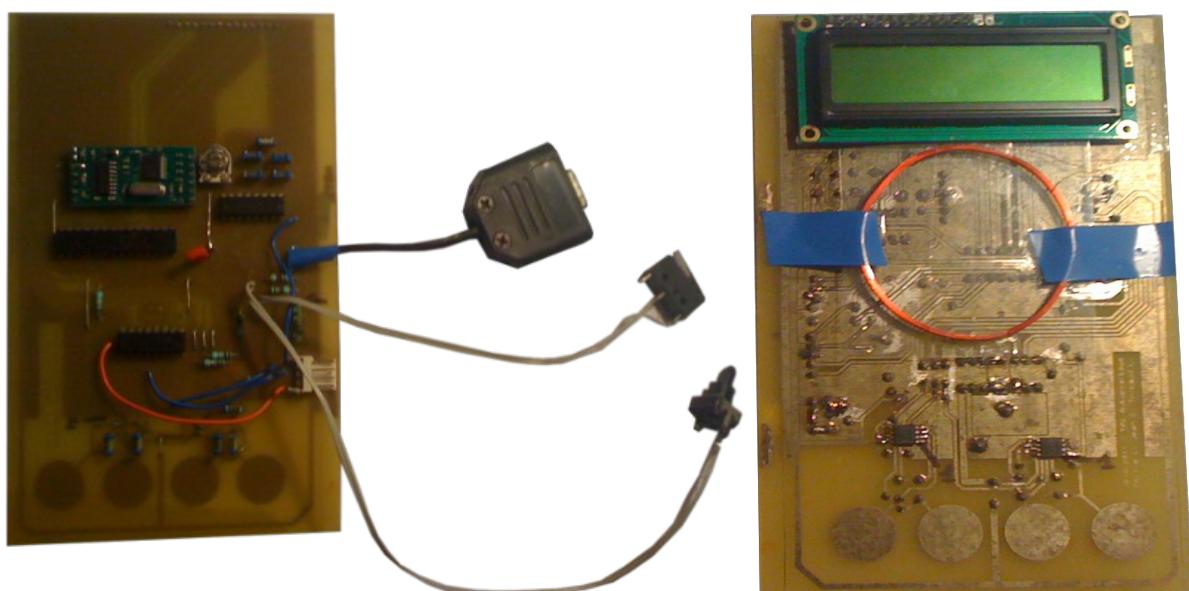
Afin de réaliser le circuit imprimé, nous avons imprimé le typon sur du papier transparent. Nous avons ensuite scotché ce papier transparent sur un circuit imprimé pré-sensibilisé en résine époxy et placé le tout dans une insoleuse à Ultras-Violets. Ensuite, nous avons révélé les piste grâce à une solution de Soude concentrée.

Après cela, nous avons placé le circuit imprimé dans un bain de perchlorure de fer chauffé pendant 5 minutes afin de ronger les espaces entre les pistes. Après cela, nous avons trempé le circuit imprimé dans une solution d'étamage à froid.

Après une courte recherche, l'étamage à froid consiste en une réaction d'oxydoréduction. La solution d'étamage à froid n'est autre qu'une solution d'ions étain Sn^{2+} qui au contact de cuivre se transforment en étain Sn. Les atomes de cuivres, eux, se transforment en ion cuivre Cu^{2+} ce qui bleui la solution.



Perçage et soudure de la carte



Carte électronique recto et verso

Partie D : Étude informatique

Dans cette section, vous retrouverez l'ensemble de l'étude informatique réalisée. Qu'elle soit embarquée ou sur PC, l'informatique fait partie intégrante du projet et tient un rôle très importante car c'est elle qui configure le comportement de la borne.

Vous retrouverez également le protocole de communication BOSEP développé pour notre application, la description du serveur et l'architecture du software.

Les fichiers sources étant trop volumineux, ils ne sont pas fournis en annexe. Cependant, vous pouvez les télécharger sur notre site internet : <http://tpe.tibol.fr>

I Présentation du software

Le code source de la borne décrit le comportement de celle-ci. Il contient toutes les fonctions qui permettent de faire fonctionner entre eux les différents composants.

Dans le code source, on accède aux périphériques du PIC18F24K22 pour communiquer avec l'extérieur. Voici la liste des applications pour les périphériques.

- Port A : Port parallèle de gestion de l'afficheur
- Port B : Port parallèle de gestion du moteur et des capteurs
- Port C : Port parallèle des touches sensitives
- USART 1 : Port série de communication avec le module RFID
- USART 2 : Port série de communication avec le serveur

On utilise également des timers pour la gestion du temps. Ces timers génèrent une interruptions qui exécute une fonction spécifique. On peut donc faire une action spécifique contrôlée par le temps. Mais les timers ne sont pas la seule source d'interruption. Les modules USART par exemple génèrent des interruption lorsqu'ils reçoivent des données ou lorsqu'ils sont prêt à en émettre.

L'informatique embarquée de la borne, c'est plus de 2500 lignes de code commenté, répartis sur 16 fichiers et 56 fonctions.

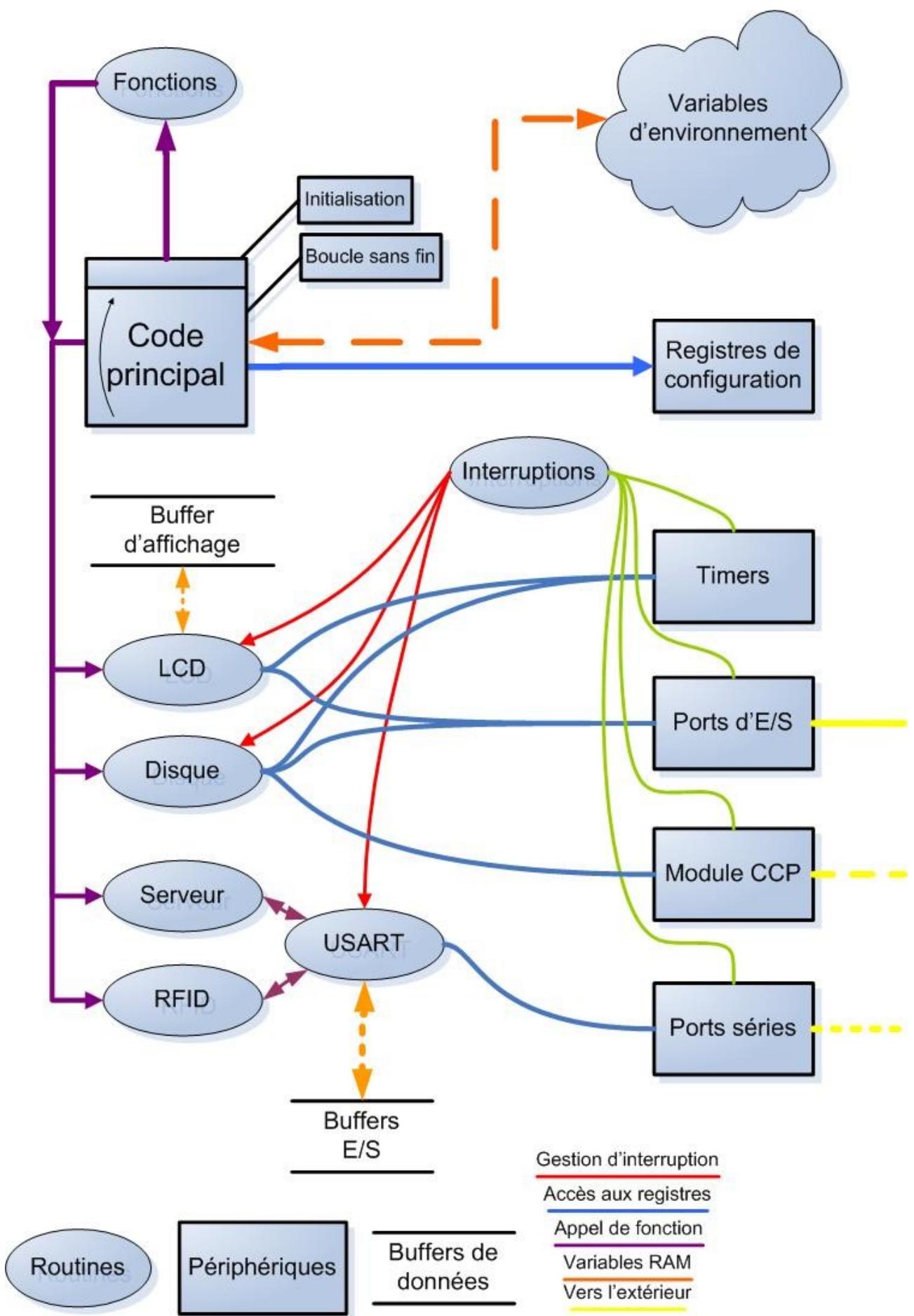
Les fichiers sont compilés à l'aide du compilateur MCC18 de Microchip.

II Architecture du software

Le code source est écrit en C et, après compilation, est téléchargé dans le micro-contrôleur. Il s'articule en routines de différents niveau qui permettent de faciliter la compréhension et le codage. En effet, la couche applicative se sert des différentes routines de bas niveau pour communiquer avec le serveur, avec l'utilisateur, avec le module RFID, etc...

Sur le schéma ci-dessous, vous pouvez observer les relations des routines entre elles. Plus bas, nous décrirons le fonctionnement interne de ces routines.

A savoir qu'il s'agit de schéma simplifié censé donner une vision simple du code source et de l'informatique embarquée.



III Protocole de communication

Comme décrit dans le cahier des charges, nous allons mettre en œuvre une liaison RS232 qui permettra à la borne de communiquer avec un serveur. On pourra y raccorder soit directement un serveur en local, soit un module intermédiaire qui gèrera la communication avec un serveur distant. En effet, en pratique le serveur ne sera pas sur place, et les données seront centralisées sur un serveur qui pourra se trouver à plusieurs kilomètres de la borne. Le module (hors étude) qui sera connecter un port RS232 à la borne, établira la connexion avec le serveur distant en utilisant une technologie propice à son environnement (liaison radio, internet, lignes spécialisées, etc...).

Dans le cadre de nos test, nous nous connecterons directement au port RS232 pour émuler un serveur local. Cette émulateur sera un logiciel PC, conçu pour cette application. La plupart des PC ne possédant plus de port série, il faudra utiliser un convertisseur USB/série, ainsi qu'un circuit MAX232 pour la mise à niveau du signal provenant de la borne.

A noter que le développement du logiciel serveur ne sera pas traité dans cette étude.

Afin de définir et d'organiser les communications entre la borne et le serveur, les données doivent être structurée. Pour décrire cette structure, on utilise un protocole de communication...

Il en existe plusieurs sortes, notamment dans le domaine des communications réseau. On peut citer par exemple les protocoles TCP, UDP, IP, HTTP, FTP, RS232, etc... Ces protocoles ne structurent pas les mêmes "types" de données. Ils sont regroupés par couches, de la plus concrète à la plus abstraite, qui constituent le modèle OSI. Dans notre système, les communications sont régies par le protocole RS232 de couche 1 (physique) qui s'occupent de transférer les trames de données. Mais ce protocole ne suffit pas... Il se contente de transférer les données.

Les données transmises par trames, il convient maintenant de définir un protocole permettant de structurer les informations de façon spécifique à notre application. Nous avons donc développé un protocole de couche 5 (Session), permettant de faire transiter les données relatives aux différents véhicules et utilisateur, ainsi que d'actionner des évènements. Nous avons nommé ce protocole BOSEP (BOrne - SERveur Protocole).

Ce protocole est dit Client-Serveur, c'est à dire que la borne se connecte à un serveur qui est maître de la communication. Les demandes sont formulées par la borne et le serveur répond ou collationne les informations.

La borne peut par exemple demander les informations d'un utilisateur, d'un véhicule, demander une location, ou signaler une alerte. Le serveur peut quant à lui répondre à ces demandes, signaler une erreur ou demander l'identification d'un TAG RFID.

La description du protocole de communication est disponible en annexe 6 du présent dossier.

IV Logiciel embarqué

IV.1 Routine d'affichage

IV.1.a Présentation

La routine d'affichage, appelé « lcd.c », gère l'écran LCD, son initialisation et son utilisation. L'écran LCD est, on l'a vu, connecté sur un port parallèle du PIC, la routine gère ce port. Elle comporte des fonctions de positionnement, d'affichage et d'initialisation propres à l'afficheur. L'écran LCD a besoin de quelque micro-secondes pour afficher des données, afin de créer des fonctions non-blocantes, les données à envoyer à l'afficheur sont placées dans un buffer et envoyées quand l'afficheur est libre.

IV.1.b Utilisation

Son utilisation est aisée. En début de programme, il faut appeler la fonction d'initialisation qui place un séquence spécifique dans le buffer. Ensuite, l'affichage s'effectue selon le positionnement-écriture. Des fonctions sont prévues pour afficher des chaînes de caractères présentent en mémoire RAM ou en ROM.

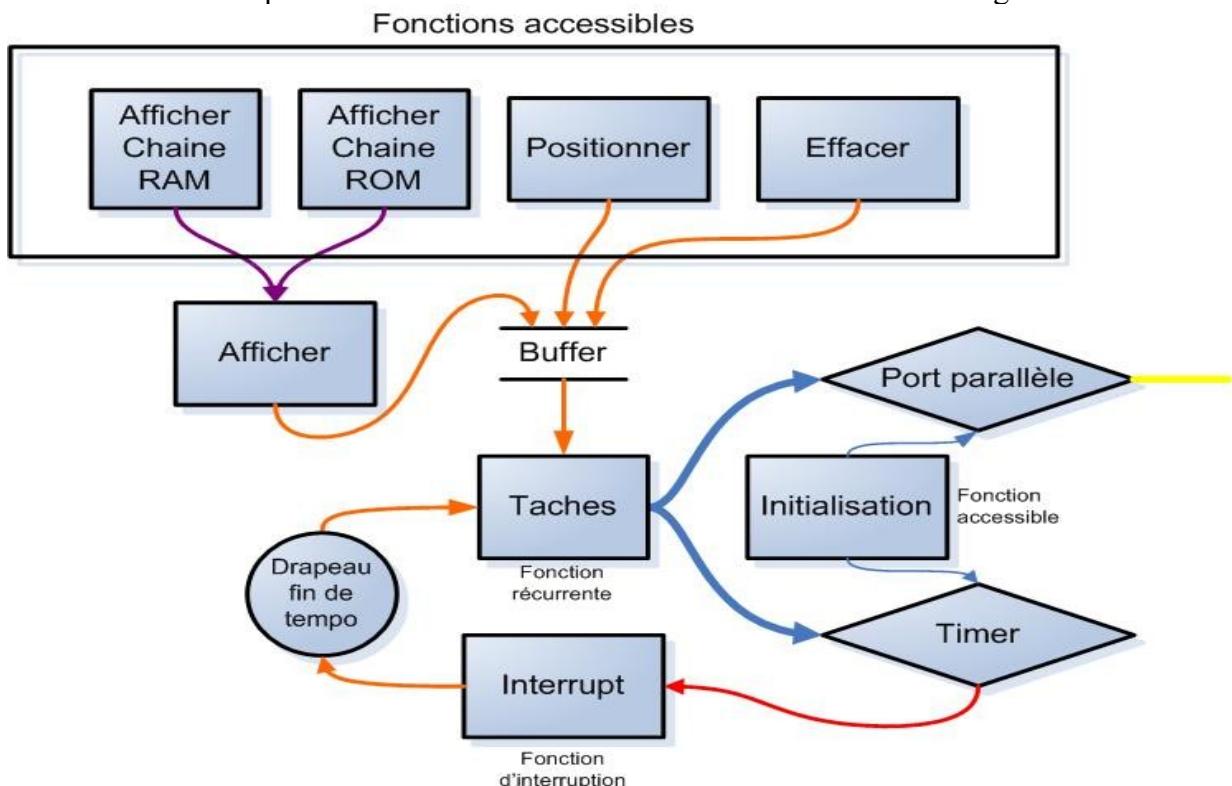
IV.1.c Fonctionnement

Chaque opération à effectuer sur l'afficheur correspond à un code envoyé à celui-ci. Un temps d'attente est associé à chaque opération.

Les codes d'opérations sont disponibles en annexe 7 de ce dossier.

Les opérations que l'on souhaite effectuer placent le code voulu dans le buffer d'affichage. A chaque opération dans le buffer est associé un temps d'attente. A la lecture du buffer, l'opération est envoyée à l'afficheur et un timer est configuré selon le temps d'attente associé. La fonction « Tâches » est appellée de manière récurrente et vérifie si le buffer est vide, s'il ne l'est pas et que l'afficheur est libre, elle envoie la donnée à l'afficheur via le port parallèle. Une fonction « effacer » permet au programmeur d'effacer l'écran LCD. Lorsqu'il déborde, le timer génère un interruption qui configure un drapeau (flag) pour signaler que l'afficheur est libre.

Le schéma ci-dessous présente le fonctionnement interne de la routine d'affichage :



IV.2 Routine de gestion du disque

IV.2.a Présentation

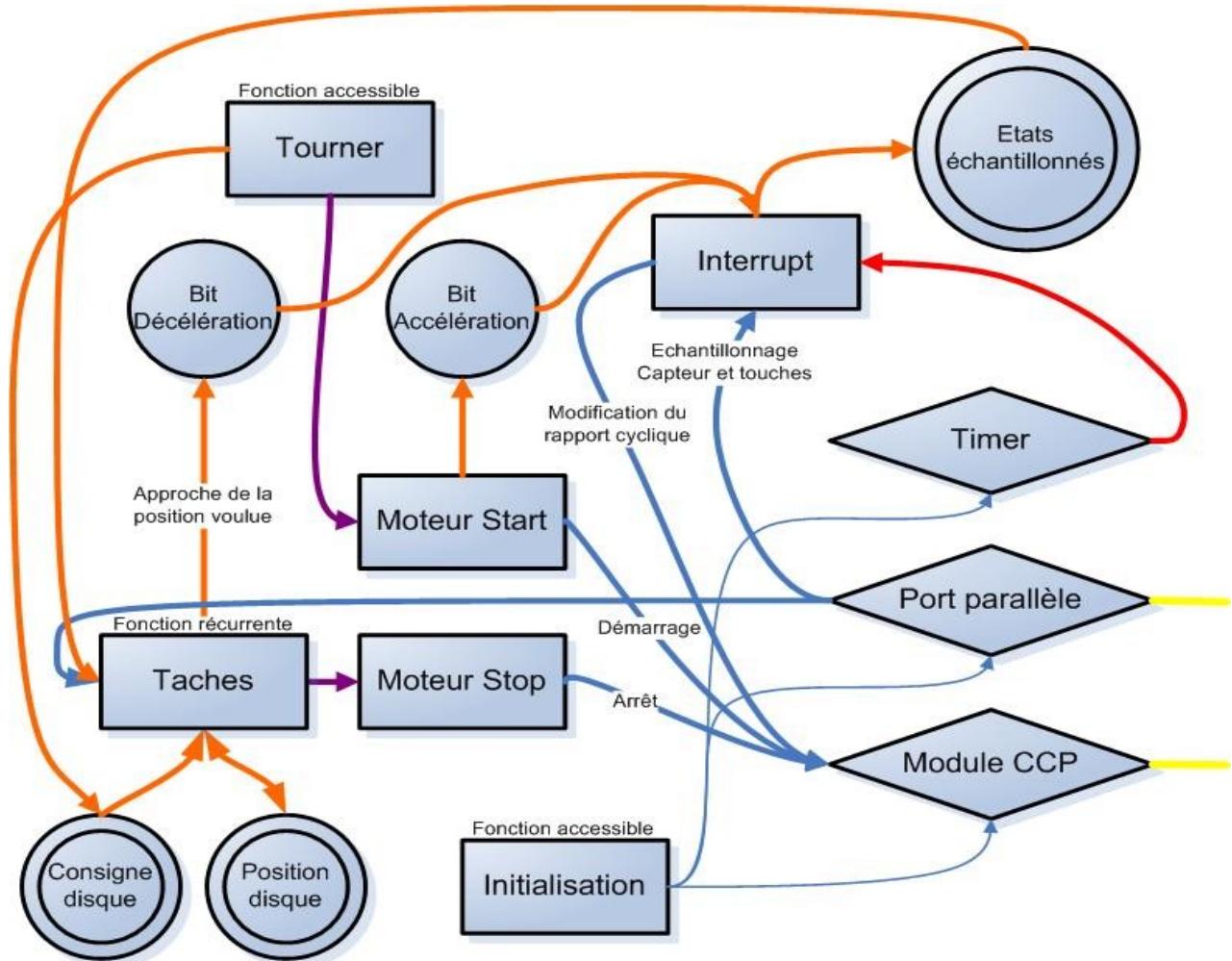
La routine de gestion du disque s'occupe d'asservir le disque à une position et de l'échantillonnage des touches sensitives. Elle gère la vitesse du moteur, le positionnement du disque par le biais des capteurs et les touches. Le capteur angulaire subit aussi un échantillonnage pour obtenir un état stable, utilisable par le programme. Elle utilise un timer, le module CCP et un port parallèle pour l'acquisition des signaux TOR des touches et des capteurs.

IV.2.b Utilisation

Comme toutes les routines du soft, elle doit être initialisée. Puis son utilisation est très simple car il n'y a que la fonction « tourner » qui est utilisable. Il suffit de lui passer en paramètre une position du disque entre 0 et 6 et la routine se charge de positionnement. Bien entendu, il faut appeler de façon régulière la fonction « taches » afin que le code fonctionne correctement.

IV.2.c Fonctionnement

Lors de la demande d'asservissement du disque à une position (via la fonction « tourner »), la consigne est stockée et convertie en nombre d'impulsions. Ce nombre sera utilisé pour suivre l'avancement du disque. Ensuite, un algorithme se charge de calculer le sens de rotation le plus rapide pour atteindre la position souhaitée, puis demande la mise en rotation à la fonction « Moteur Start ». Cette fonction place le moteur en rotation à 50% de sa vitesse nominale puis active le bit d'accélération. A chaque interruption du timer, le rapport cyclique est augmenté de façon à créer une accélération progressive. Quand le disque atteint sa vitesse maximale, le bit d'accélération est mis à zéro. Pendant ce temps, la fonction d'interruption échantillonne les valeurs des capteurs et des touches pour fournir un état stable. Cet état est utilisé par la fonction « taches » pour connaître la position du disque et demander son ralentissement. Une fois que le disque a atteint sa position de consigne, la fonction « Moteur stop » est appelée et le disque est arrêté.



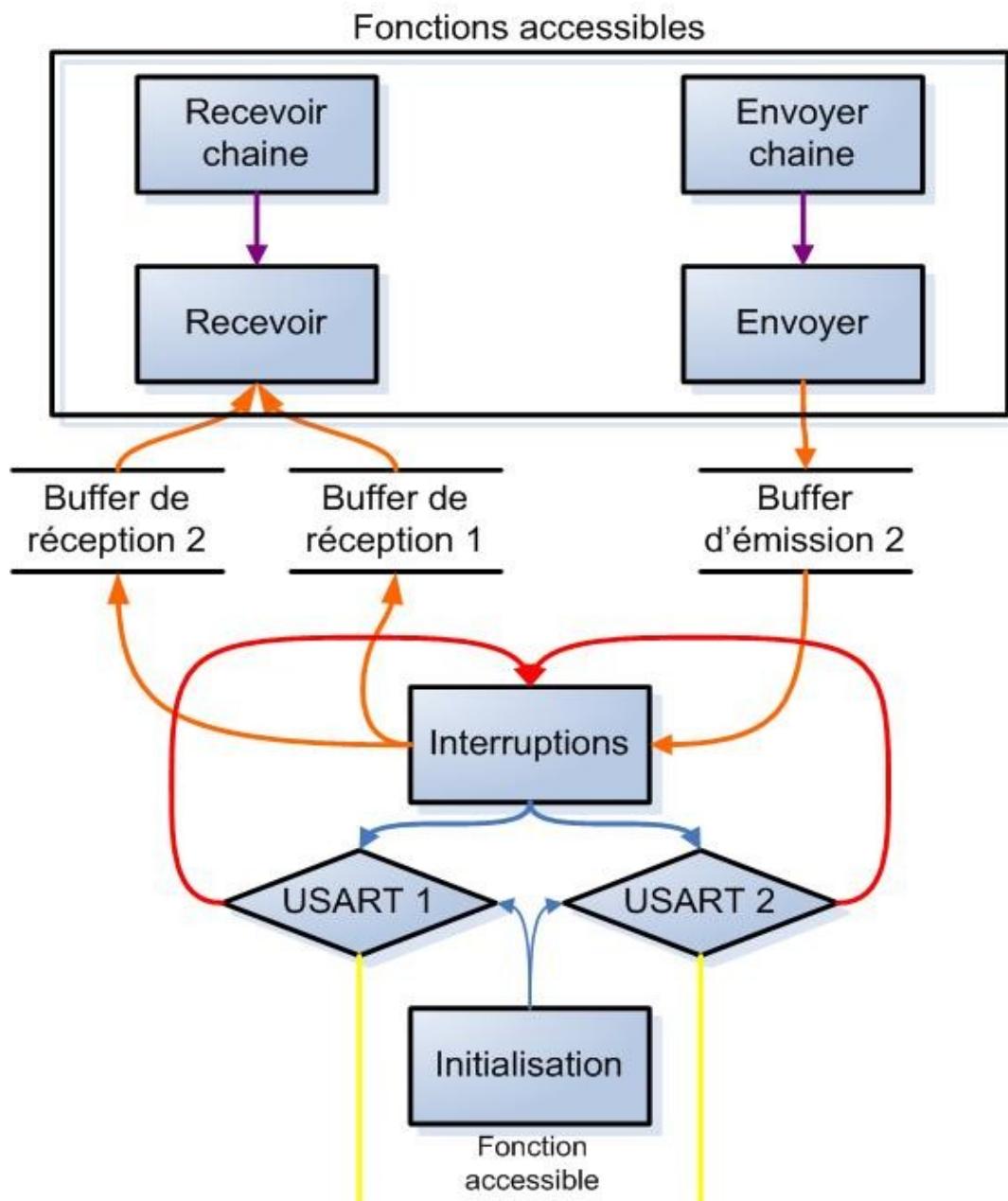
IV.3 Routine de communication

IV.3.a Présentation

La routine de communication gère les module USART, c'est à dire les communication séries du micro-contrôleur. Elle effectue la gestion de bas niveau des dialogue entre la borne et le serveur et entre la borne et le module RFID. Cette routine se trouve dans le fichier « `uart.c` ».

IV.3.b Fonctionnement

Deux fonctions d'émission et de réceptions sont disponible. Pour émettre, on envoi un caractère ou une chaîne et la fonction « `envoyer` » les place dans le buffer d'émission. Quant le port série est libre, une interruption est générée et la fonction d'interruption se charge d'émettre l'octet le plus ancien présent dans le buffer. Pour la réception, on lit avec la fonction « `recevoir` » les octets reçus dans le buffer. Quand le port série reçoit des données, il génère une interruption. Dans la fonction d'interruption, l'octet reçu est placé dans le buffer de réception.



IV.4 Routine de dialogue avec le serveur

IV.4.a Présentation

Cette routine n'est autre que l'intégration du protocole BOSEP dans le code. Une fonction par action du protocole. Elle est nommée « serveur.c ».

IV.4.b Utilisation

Il suffit d'appeler la fonction correspondante à l'action souhaitée et éventuellement de fournir un ou plusieurs paramètres si nécessaire. La routine met en forme la trame et la place dans le buffer d'émission grâce à la routine « usart.c ». Lors de la réception de données du serveur, la routine de communication « usart.c » appelle la fonction de traitement dès réception de la trame entière. La fonction de traitement va ensuite mettre à jour les variables d'environnements (nom de l'utilisateur, nom du véhicule en cours, etc...) et lever ou baisser des drapeaux.

IV.5 Routine de dialogue avec le module RFID

IV.5.a Présentation

Comme la routine « serveur.c », la routine de gestion du module RFID, « rfid.c » permet de décoder les trames reçues du module RFID.

IV.5.b Utilisation

A la réception d'une trame via le module USART, celle-ci est décodée pour extraire uniquement le TAG RFID. Celui-ci est placé dans une variable d'environnement et un drapeau est levé. Pour indiquer si on attend un tag RFID d'utilisateur ou de véhicule, on va configurer des bits qui permettront à la fonction de traitement de placer les 5 octets du codes RFID dans la bonne variable d'environnement.

IV.6 Routine d'interruptions

IV.6.a Présentation

La routine d'interruption est appelé par le « hardware » sur demande des périphériques. Il existe deux sortes d'interruptions : les interruptions de haute et de basse priorité. Cette routine permet de connaître le responsable de l'interruption et d'appeler sa fonction de traitement.

IV.6.b Utilisation

Il n'y a aucune action nécessaire de la part de l'utilisateur hormis l'appel de la fonction d'initialisation qui configure les interruptions.

IV.7 Bibliothèque de fonctions

Il s'agit d'un fichier (« fonctions.c ») qui contient toutes les fonction récurrentes dans le code principal afin de l'alléger. On y trouve des fonctions de gestions des erreurs, de transfert et de copie de tags RFID, etc... Elle permet aussi de gérer des temporisation d'environnement.

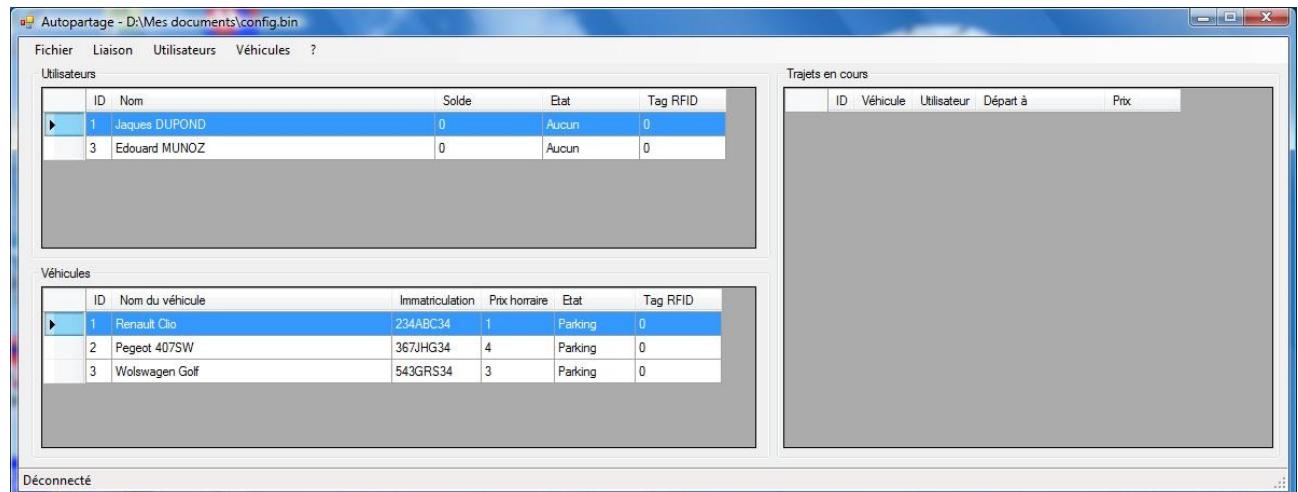
C'est également dans ce fichier que se trouvent les fonctions d'initialisation et de configuration du micro-contrôleur (vitesse de travail, activation des resets, etc...)

V Logiciel serveur

Afin de centraliser les données, nous avons développé un logiciel serveur qui gère les utilisateurs, les véhicules et les locations. C'est lui qui donne les autorisations de location. Ce logiciel est volontairement incomplet car il n'est pas le but premier de notre étude. Cependant il permet d'exploiter pleinement les capacités de la borne.

Il est développé en Visual Basic .NET et communique avec la borne au moyen d'une liaison RS232.

Voici une capture d'écran du logiciel serveur :



Partie E : Ressources documentaires

- <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en547749>
- <http://ww1.microchip.com/downloads/en/DeviceDoc/41412D.pdf>
- <http://www.datasheetcatalog.org/datasheet/texasinstruments/l293d.pdf>
- <http://claude.dreschel.free.fr/composants/max232.htm>
- <http://pdf1.alldatasheet.com/datasheet-pdf/view/87207/QUANTUM/QT320.html>
- <http://matthieu.benoit.free.fr/affi-lcd.htm>
- <http://www.brodeurelectronique.com/index.php?2007/03/15/9-afficheurs-lcd>
- <http://fribotte.free.fr/bdtech/Drivers/DrivesPontH.html>
- http://fr.wikipedia.org/wiki/Moment_d%27inertie
- <http://www.ntn-europe.com/fr/fr/documentation/Needlerollerbrg/pdf/A3Calculdescharges.pdf>
- http://www.lextronic.fr/~lextronic_doc/UM-005.pdf
- <http://www.table-ascii.com/>
- <http://www.commentcamarche.net/contents/c/ctab.php3>
- http://fr.wikipedia.org/wiki/Machine_%C3%A0_courant_continu
- <http://www.tracepartsonline.net>
- http://daniel.robert9.pagesperso-orange.fr/Liaisons_serie_Liaisons_parallel.html
- <http://sitelec.org/cours/abati/flash/codeur.swf>
- <http://www.aix-mrs.iufm.fr/formations/filières/ge/data/PIC/PICC/MCC18%20v14.pdf>
- <http://www.laboelectronique.be/lcd6100.html>
- http://www.lyceedupaysdesoule.fr/microtec/construction/cours/ch6_rdm.pdf
- http://fr.wikipedia.org/wiki/Analyse_fonctionnelle_descendante
- <http://fr.wikipedia.org/wiki/Engrenage#Vocabulaire>

Partie F : Réalisation finale



Partie G : Synthèses

I Synthèse de Jean CLAISSE

L'étude de ce TPE dont le sujet est je le rappelle, une borne d'auto-partage, aura été bénéfique pour ma part non-seulement du point de vue éducatif, mais également sur le plan relationnel. J'ai pu observer en temps réel les conflits d'idées entre les membres d'un groupe, ce qui pourtant n'a pas altéré le bon déroulement de cette (demie) année de TPE. Puisqu'évidemment les trois personnes sont différentes, les idées ne peuvent pas toujours les mêmes. C'est alors que l'on apprend à s'organiser en groupe pour la réussite du projet. C'est pourquoi, je garde une expérience personnelle satisfaisante du travail en groupe.

Bien sûr, l'étude de ce TPE m'aura également ouvert l'esprit sur les sciences de l'ingénieur. J'ai eu la chance de me trouver dans un groupe composé de personnes qui avaient déjà plus ou moins de l'expérience, ce qui m'a permis d'aborder certains sujets que je n'aurais sûrement pas pu utiliser tout seul. Par exemple j'ai pu apprendre à manier le langage de programmation : C, découvert plusieurs circuits électriques aux diverses fonctions (L293 : pont en H, QT320 : système gérant les touches sensitives, etc....). J'ai également pu aider à la création d'une carte électronique, une expérience très enrichissante et qui m'a bien montré un aperçu du travail quotidien d'un ingénieur.

Ce TPE aura été enfin un moyen de pouvoir appliquer les connaissances grâce aux cours, notamment en mécanique où l'on a pu appliquer les formules de calculs de vitesses, couple, énergies, et autres moments d'inerties ainsi que bien d'autres valeurs qui nous ont permises d'arriver au résultat final qu'est la maquette de notre système sans que celui-ci ne s'écroule sur lui-même. Evidemment les cours d'électronique nous ont bien aidé également dans la réalisation du projet, par exemple lorsque nous avons remarqué qu'un changement de rapport cyclique dans la tension nominale du moteur entraînait un changement de vitesse de rotation.

Bref, plus qu'un simple projet à rendre à la fin de l'année, le TPE m'a été profitable non seulement au niveau des sciences de l'ingénieur, mais aussi au niveau humain.

Merci à Thibaud LEMAIRE, Emilie CANTERO, ainsi qu'aux professeurs Monsieur RYAUT et Monsieur BARTHELEMY qui nous ont soutenu tout au long du projet.

II Synthèse d'Emilie CANTERO

CANTERO Emilie

1°S3

Synthèse personnelle

TPE 2011

Comment profiter des libertés permises par la voiture sans pour autant en subir les contraintes actuelles ?

Notre équipe est constituée de trois personnes, Thibaud LEMAIRE, Jean CLAISSE et moi-même.

Pour notre TPE nous avons choisi le thème Contraintes et Libertés. Au travers de ce thème nous nous sommes posés la problématique suivante: « Comment profiter des libertés permises par la voiture sans pour autant en subir les contraintes actuelles ? ». La première solution à laquelle nous avons pensé est l'utilisation d'un système d'auto-partage. Nous en sommes venus rapidement à nous poser la question : « Comment concevoir ou améliorer un système d'auto-partage ? ».

Ce sujet d'actualité permet la réduction de l'encombrement des villes entraîné par le nombre important de véhicules. La pollution, elle aussi, pourrait être réduite grâce au développement d'un tel système.

Le mécanisme étudié, se présente sous la forme d'un distributeur de clés de voitures stockées sur un disque tournant avec une commande électronique permettant la reconnaissance du conducteur et le choix du véhicule.

Pour guider notre conception et nous aider dans le choix des solutions techniques que nous allions mettre en place, Thibaud et moi-même avons réalisé une analyse fonctionnelle afin de définir le cahier des charges complet. Par la suite, je me suis consacrée à la partie « technologie mécanique » du problème. La première étape a été la conception du modèle en trois dimensions sous Inventor, puis le choix des systèmes de transmission de l'énergie mécanique (poulie-courroie ou train d'engrenages par exemple). Une fois les choix technologiques faits, le travail s'est poursuivi par le dimensionnement des composants choisis, et enfin, par la réalisation de la maquette réelle.

Le TPE m'a permis de mieux percevoir le travail d'équipe, d'apprécier ce qu'apporte la collaboration de personnes motivées, de faire des recherches et se cultiver sur un thème intéressant. Mais ce qui a été le plus difficile est la gestion de chaque idée des membres du groupe. Beaucoup étaient intéressantes mais pas toujours exploitables dans cette application.

Le plus passionnant a été la construction de la maquette réelle et de voir concrètement l'aboutissement de notre projet.

III Synthèse de Thibaud LEMAIRE

Pour mener à bien notre projet, mes camarades et moi avons du mettre en place une répartition et une gestion rigoureuse des tâches. Il en allait de la réussite du TPE et bien que cette organisation n'est pas toujours été respectée, je suis globalement satisfait du travail que nous avons accomplis.

Quand il nous a été demandé de choisir un sujet, nous n'avions pas envisager la création d'une « Borne d'auto-partage ». Nous étions certains de vouloir réaliser une maquette, témoignage de l'aboutissement de six mois de travail, mais nous ne savions pas quel thème aborder. En plus de cela, et comme de nombreux groupes, nous voulions concevoir un système encore inexistant et à tout prix éviter les traditionnels « Robots suiveurs de ligne » et autres « Voitures électriques ». C'est en prenant des cours de code de la route en auto-école, lors desquels revenaient souvent des questions traitant de l'auto-partage, que l'idée m'est venue de traiter ce thème novateur. J'en ai alors parlé avec Émilie, qui elle aussi suivait des leçons de code, puis expliqué à Jean en quoi cela consistait. Restait à trouver une façon de mettre les sciences de l'ingénieur au service de ce thème. Après avoir analyser le fonctionnement actuel des réseaux d'auto-partage et en avoir dégagé les inconvénients, nous avons convenu de l'étude de ce qui allait devenir « La borne d'auto-partage ».

En premier lieu, nous avons travaillé de concert afin de définir le cahier des charges. Nous voulions un projet conséquent, qui soit le plus réaliste possible, c'est à dire qui puisse être utilisé dans la réalité. Cela implique une prise en compte de nombreux facteurs, détails techniques et technologiques comme le moyen de gestion des utilisateurs, l'architecture du parc de bornes, le moyens de communication entre ces bornes, etc... Nous ne devions réaliser qu'une seule maquette mais devions réfléchir au possible déploiement de dizaines de bornes. Ce fut un travail long et fastidieux, bien que je trouvais intéressant chaque obstacle technique qui se dessinait au fur et à mesure de l'élaboration du cahier des charges. J'avoue avoir passé plus de quinze jours à réfléchir sur la manière de faire transiter les données entre les bornes et le serveur. Le cahier des charges terminé et les tâches distribuées au moyen d'un diagramme de GANTT, nous avons entrepris l'étude technique du projet.

Nous avons commencé par une étude préliminaire, réalisée plus ou moins tous ensemble. Il fallait définir de manière exacte et précise le comportement de la borne. Nous avons également mis en place des outils qui nous aideraient dans la conception et auxquels nous pourrions nous référer au besoin. Il s'agissait du protocole de communication entre la borne et le serveur, de l'organigramme du software, de l'étude SADT, etc... Cela terminé, nous pouvions (enfin) commencer l'étude proprement dite de la borne.

Chacun dans son domaine de prédilection, nous commençâmes à rechercher, analyser, comparer les solutions techniques envisageables. Émilie à la mécanique, Jean et Moi à l'électronique, faisant le point régulièrement. Enfin, ça c'est la théorie ! En pratique, je suis plutôt du genre à vouloir tout superviser. Je reconnais que c'est un défaut peu enviable, mais je ne peut m'empêcher, lors de travaux en groupe, de contrôler, donner mon avis, voire même compléter les productions de mes camarades. Cela m'a d'ailleurs valu quelques heurts, notamment avec Émilie. Cependant, si les TPE étaient à refaire, je me comporterais de la même manière, car je pense que cela ne nous a pas tant pénaliser et que les divergences de points de vues et d'opinions sont une force qu'il faut mettre à profit afin de tirer le meilleur de chacun.

Donc tout en supervisant de loin les travaux de mes camarades, je me concentrais à concevoir la carte électronique, du schéma jusqu'aux typons. Je me renseignais, prototypais, testais, tout en m'appuyant sur les remarquables travaux que Jean publiait sur le site Internet du TPE.

Dès le départ, nous avions décidé de retranscrire l'intégralité de nos travaux sur un site Internet, formidable outil de communication, disponible à l'adresse suivante : <http://tpe.tibol.fr>. En effet, j'ai

une grande admiration pour les gens qui publient leur savoir faire sur internet pour aider la communauté et ce, bénévolement. Je m'étais toujours dit que, quand j'en aurais l'opportunité, je publierai à mon tour mes recherches et projets. C'était là là l'occasion idéale. Et le site allait être bien rempli, au delà de ce que nous imaginions.

Pendant ce temps, Émilie réfléchissait au mécanisme de gestion des clés. Calculs, contraintes, schémas, on peut dire qu'elle avait du travail. Quand il a fallu définir le couple nécessaire pour mettre en mouvement le disque, un problème s'est posé. Nous n'avions pas les compétences pour l'évaluer... Porté de curiosité, je m'obstinais à vouloir le calculer. Nous ne pouvions pas prendre un moteur « au hasard » alors que tous les autres points étaient tant recherchés. Après plusieurs jours de recherches, pas à temps plein rassurez-vous, je suis tombé sur « la » formule de calcul du couple en fonction du moment d'inertie et de l'accélération angulaire. Restait à calculer le moment d'inertie du disque à clé, et la encore, Internet est une source intarissable de renseignements... Wikipédia pour l'occasion.

Le moment que j'ai le plus apprécié, c'est la réalisation de la carte électronique. Une expérience riche. Quelle joie de voir se révéler les pistes de cuivre sur lesquelles nous travaillions depuis plusieurs semaines. Et quel étonnement : alors que nous venions d'employer une solution d'étamage à froid qui transforme le cuivre en étain, nous apprîmes le lendemain, totalement par hasard, l'explication de cette transformation en cours de chimie : oxydoréduction. Voilà qui satisfit ma curiosité.

Peu après, les choses s'accélérèrent. Un mois avant l'oral, l'heure était aux perçages, soudures et tests de la carte. Par chance tout fonctionna du premier coup et l'on pu programmer le micro-contrôleur.

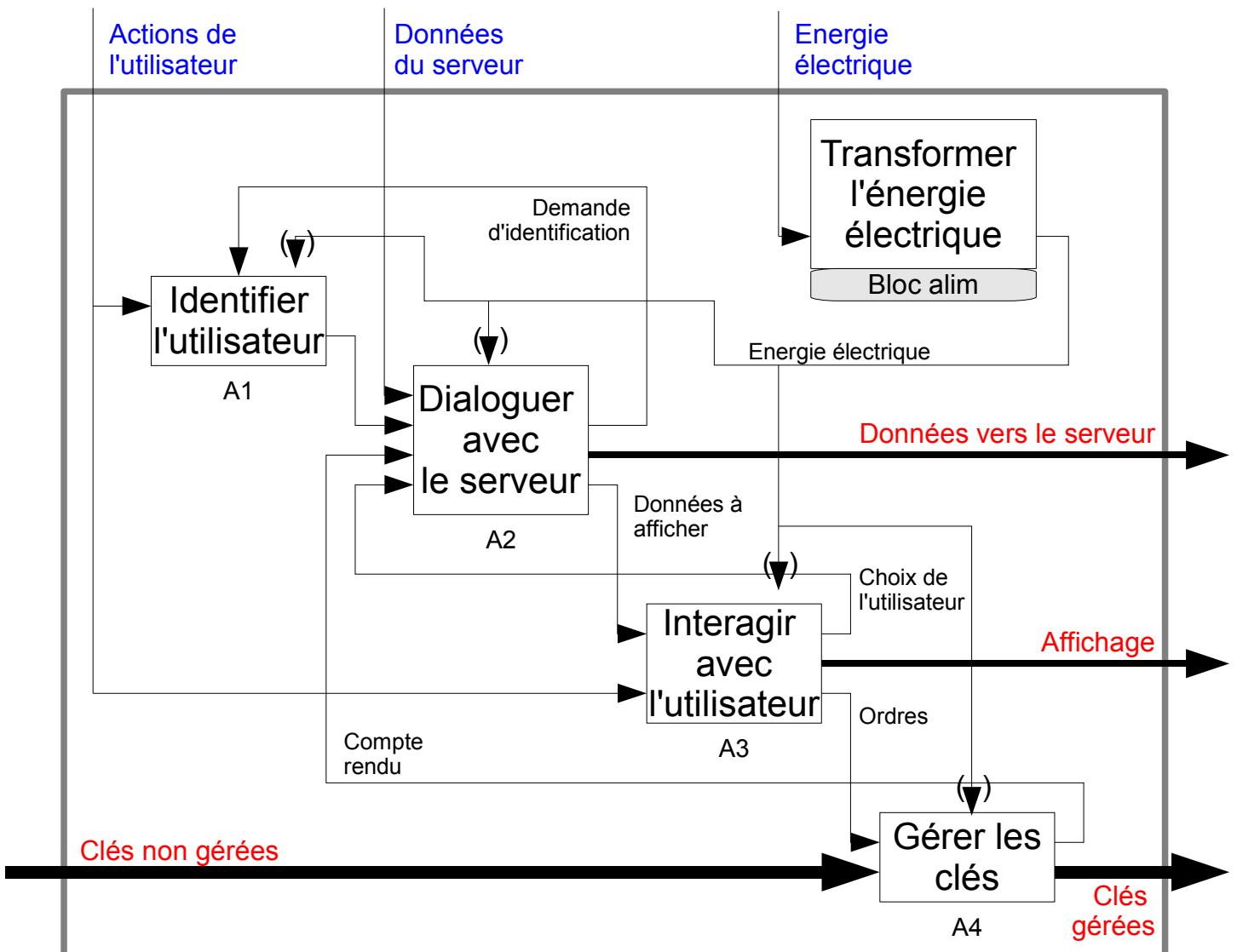
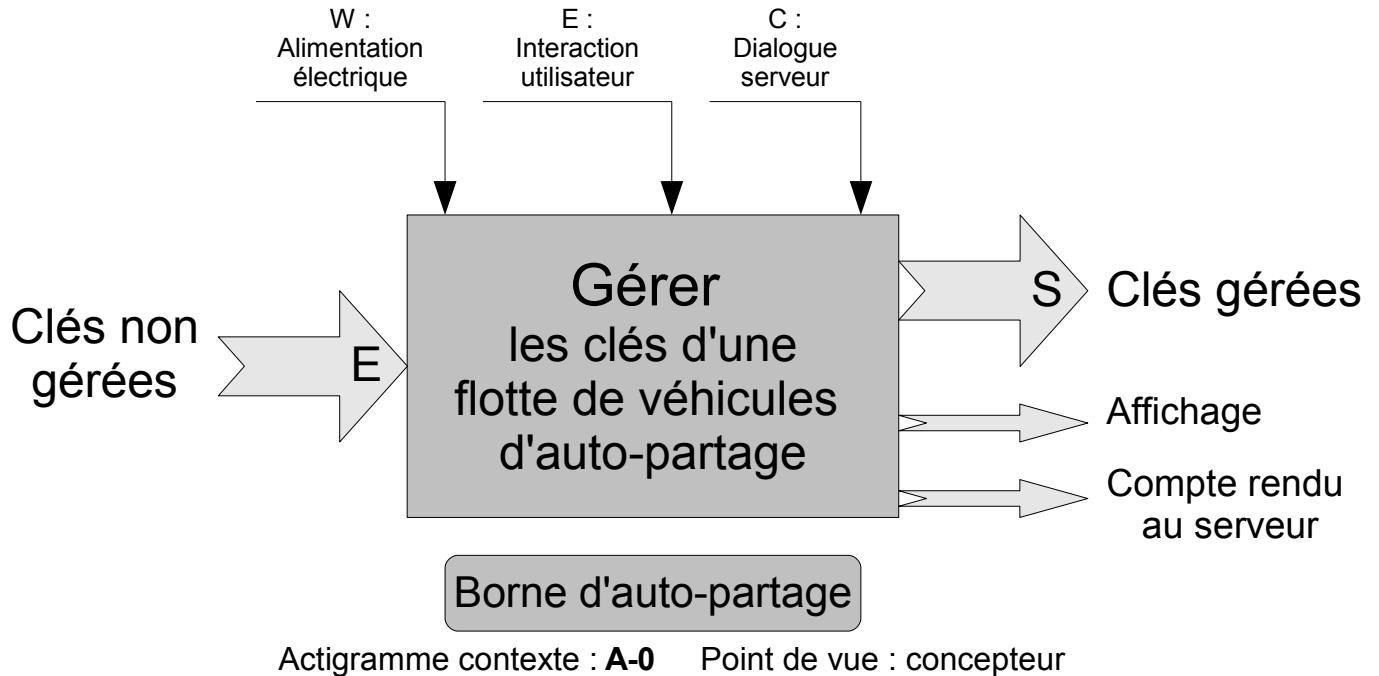
Nous avions commencer à coder les routines de gestions de la borne. Par « routine » j'entends un petit code informatique qui gère une fonction en particulier, l'affichage par exemple. Jean appris même le langage C pour m'aider à programmer. Il restait alors 15 jours avant l'oral, et nous n'avions toujours pas de maquette.

Au départ en vacances, celles de février, les taches était distribuées. Émilie fabriquerait la maquette, et je finirais la carte électronique de façon à ce qu'à la rentrée, nous n'ayons qu'à assembler le tout. Et c'est encore une fois ce qui se passa, à merveille. Je dois dire que, sinon de la chance, nous avons été grandement avantagé par le fait que tout ce que nous entreprenions ce déroulait comme prévu. Tout ? Enfin presque. Ce n'est jamais tout rose en électronique, ni en mécanique d'ailleurs. Des défauts de programmations, petites erreurs de constructions, mais nous voilà quelques jours après la rentrée avec un maquette en fonctionnement, un dossier, certes un peu long mais précis, et l'agréable sensation du travail (devoir ?) accompli.

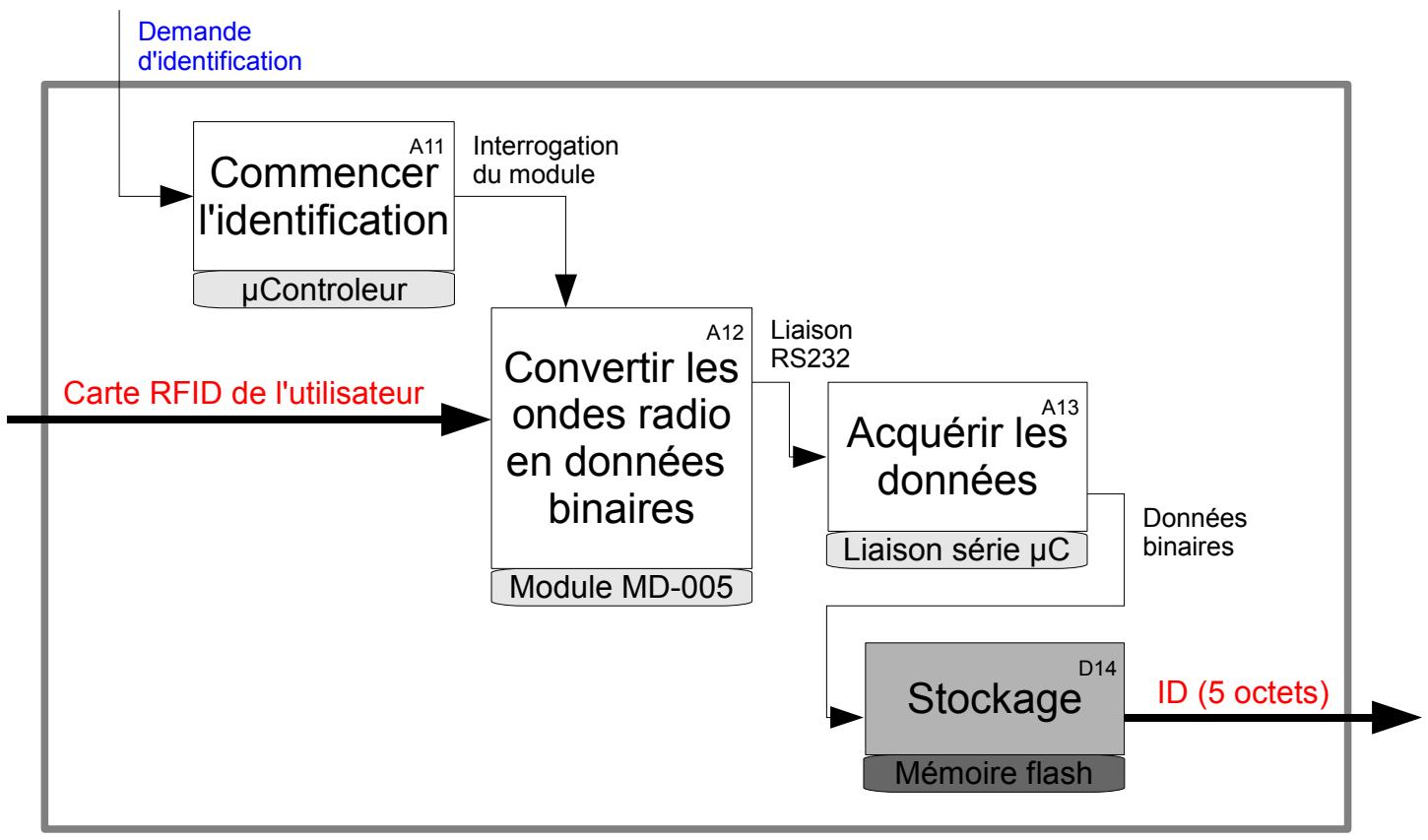
J'ai longuement hésiter avant de tout brancher. Moteur, capteurs, alimentation. J'avais déjà tout testé séparément, mais je sais par expérience que les surprises en électronique arrivent à la fin, quand il faut faire cohabiter tous les composants ensemble. Et d'autant plus quand il s'agit de logique jouxtant la puissance. Ma plus grande crainte était de voir le moteur générer tant de parasites que le micro-contrôleur aurait adopté un comportement inadéquat. Fort heureusement, il n'en fut rien.

Je me rends compte désormais de la quantité colossale de travail que cela nous a demandé. Le sujet était ambitieux, le résultat dépasse nos espérances. Ce m'a fait prendre conscience de ce qu'est la réalisation à moyen terme d'un projet, et m'a beaucoup apporté sur le plan relationnel, notamment sur la façon de s'investir en équipe. Hormis la divergences sur certains sujets, tout s'est passé remarquablement bien et je ressens maintenant une grande fierté, un soulagement aussi. Si c'était à refaire, ce serait avec joie. Peut-être choisirions nous un sujet un peu moins imposant, car le rythme de travail commençait à peser.

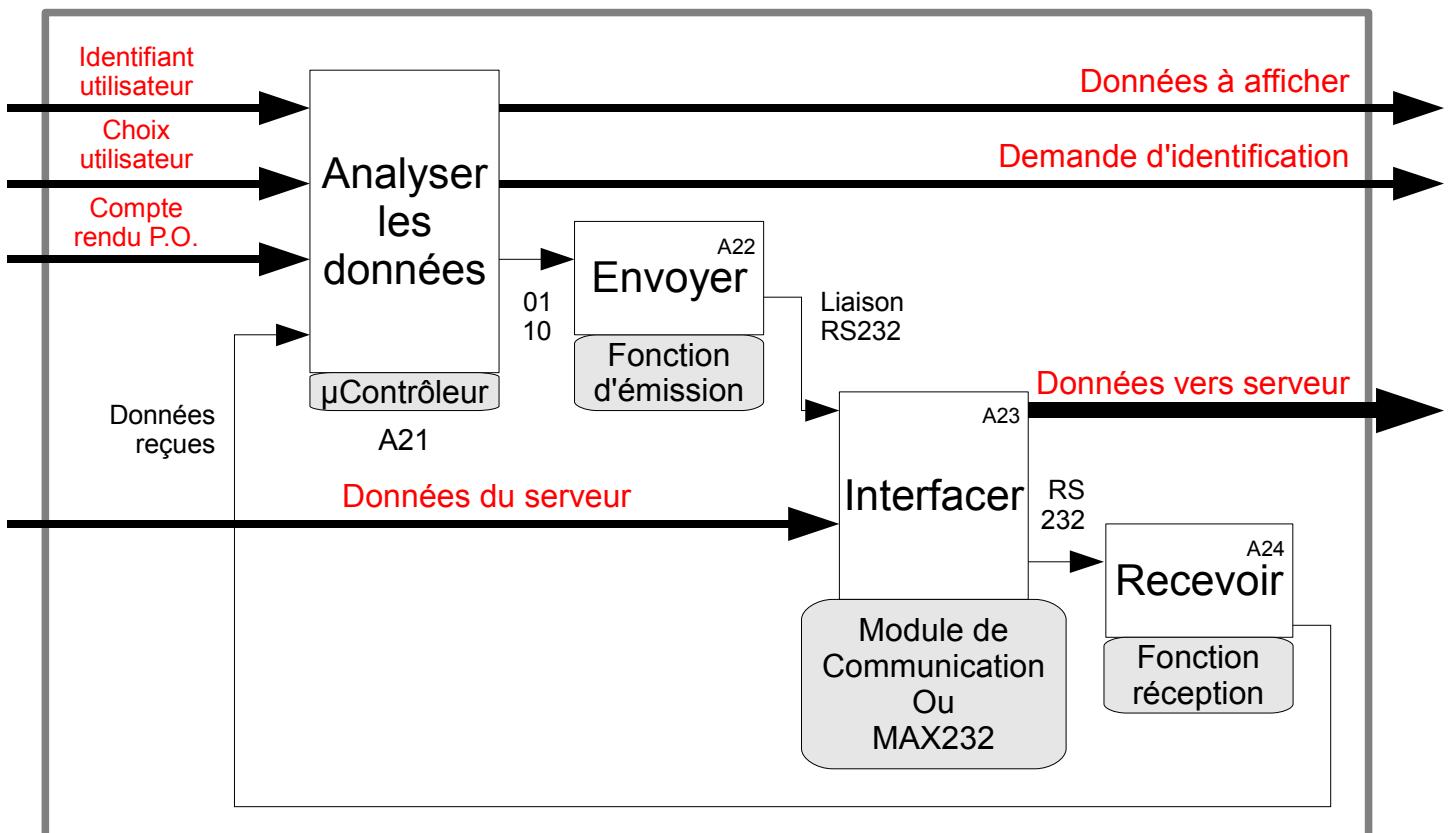
Annexe 1



Annexe 1

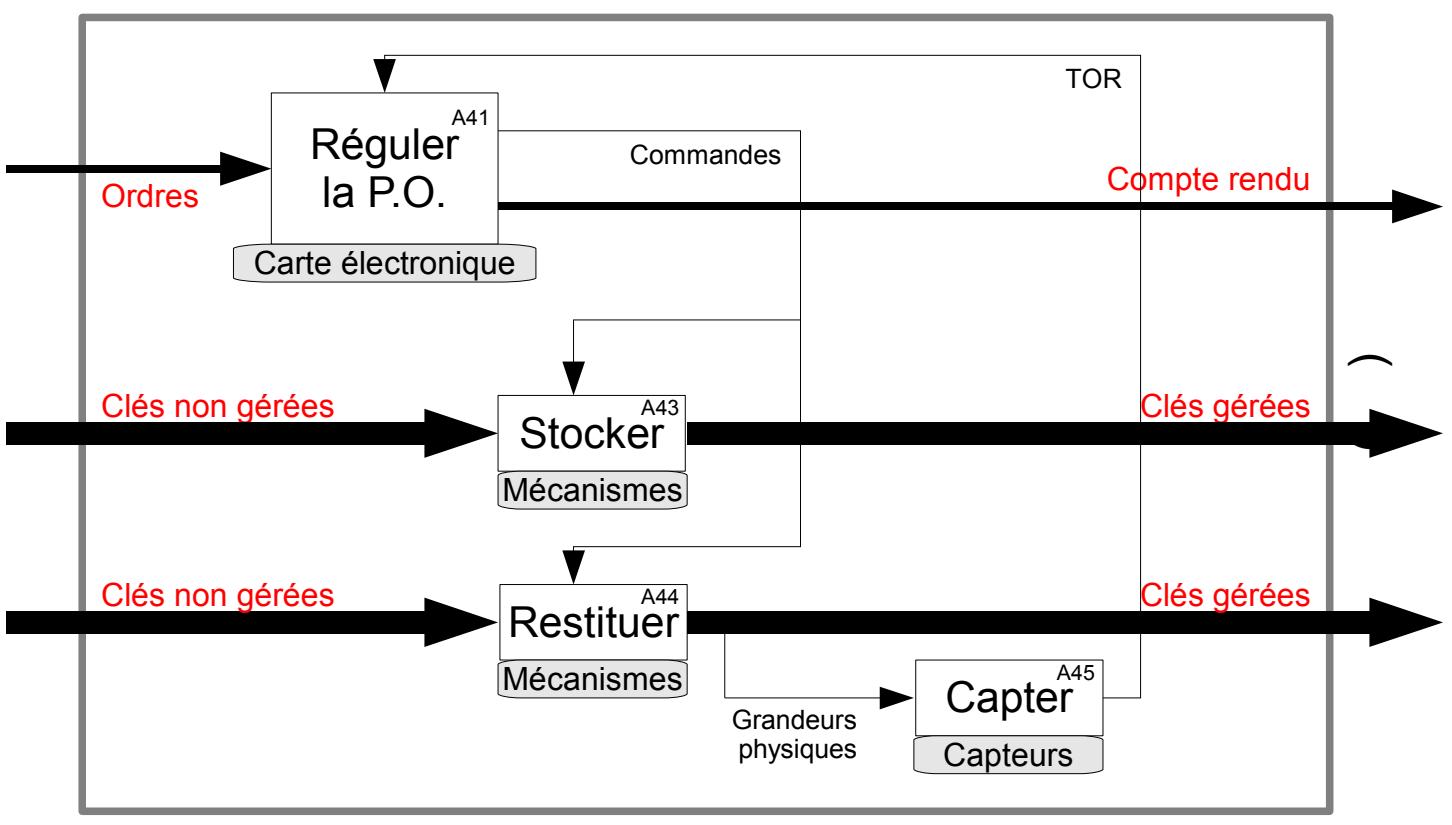
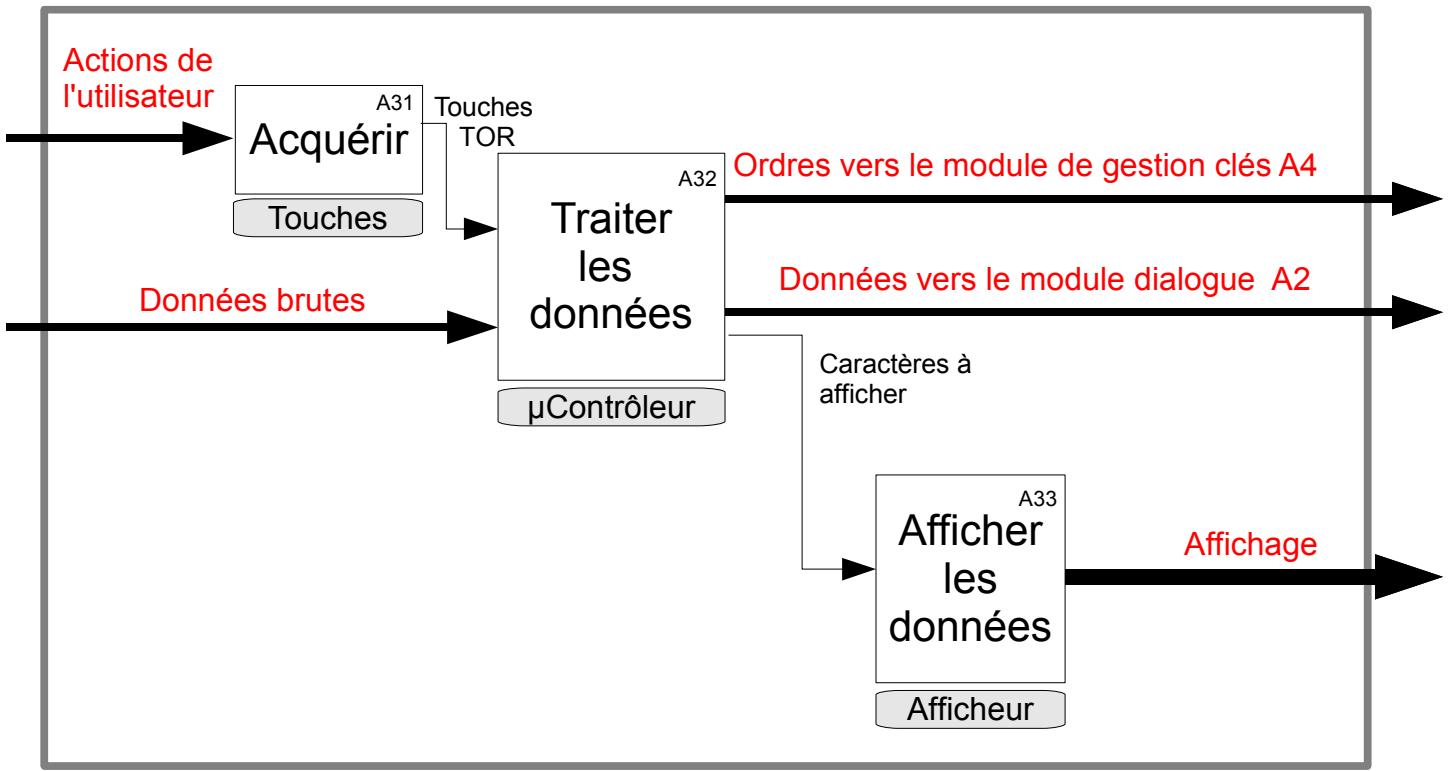


Actigrammes et datagramme niveau 2 : A1 Point de vue : concepteur

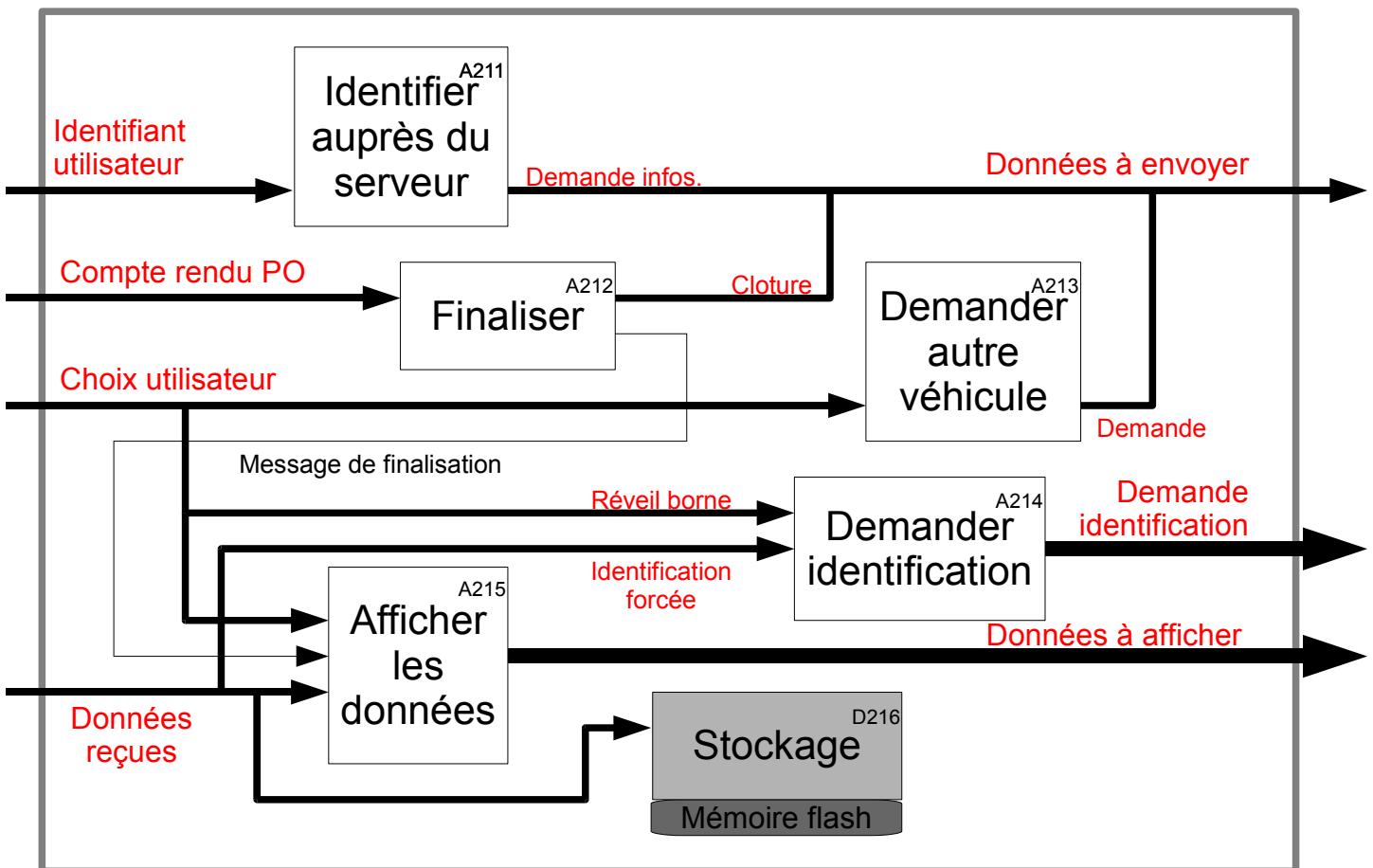


Actigrammes niveau 2 : A2 Point de vue : concepteur

Annexe 1

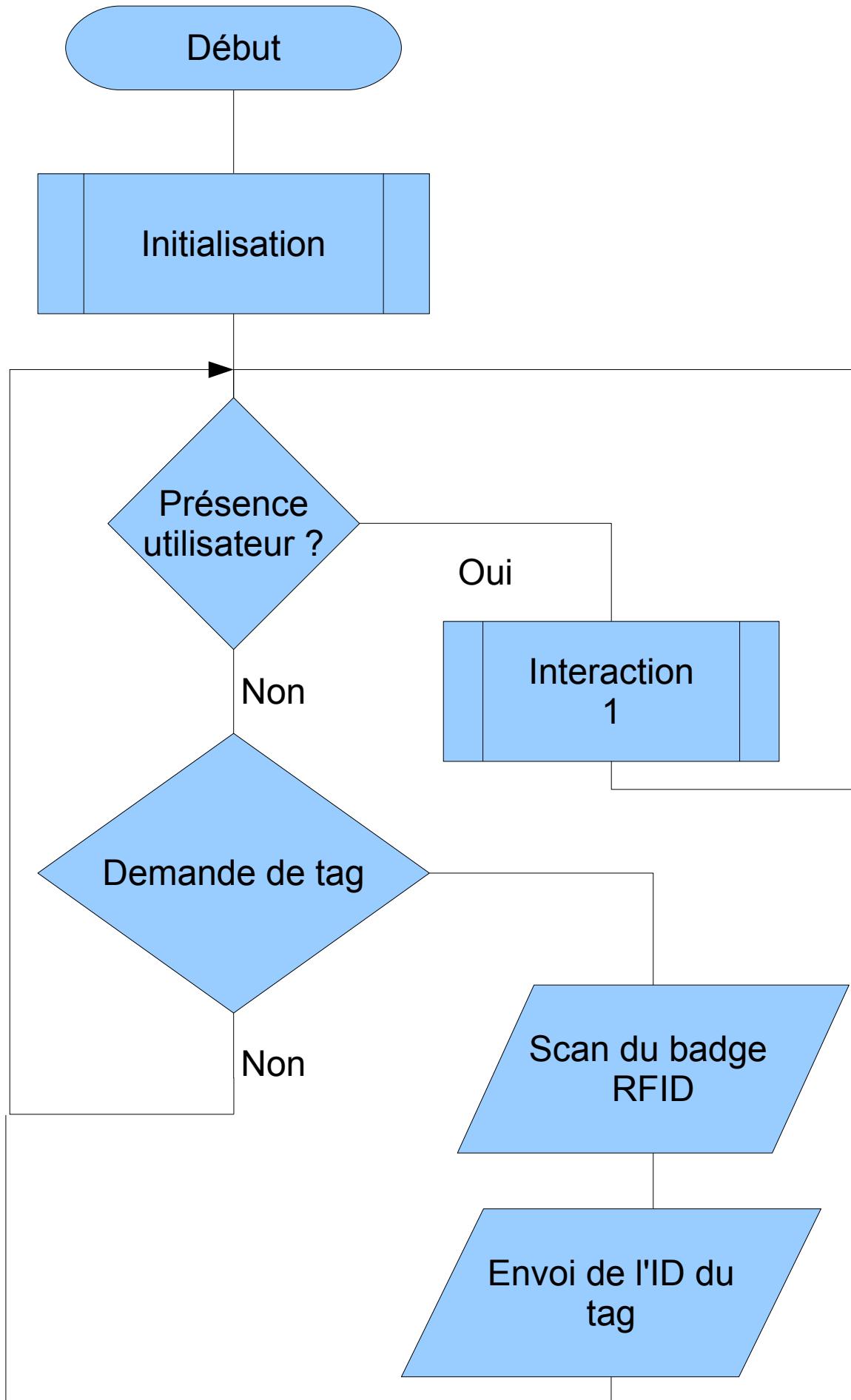


Annexe 1

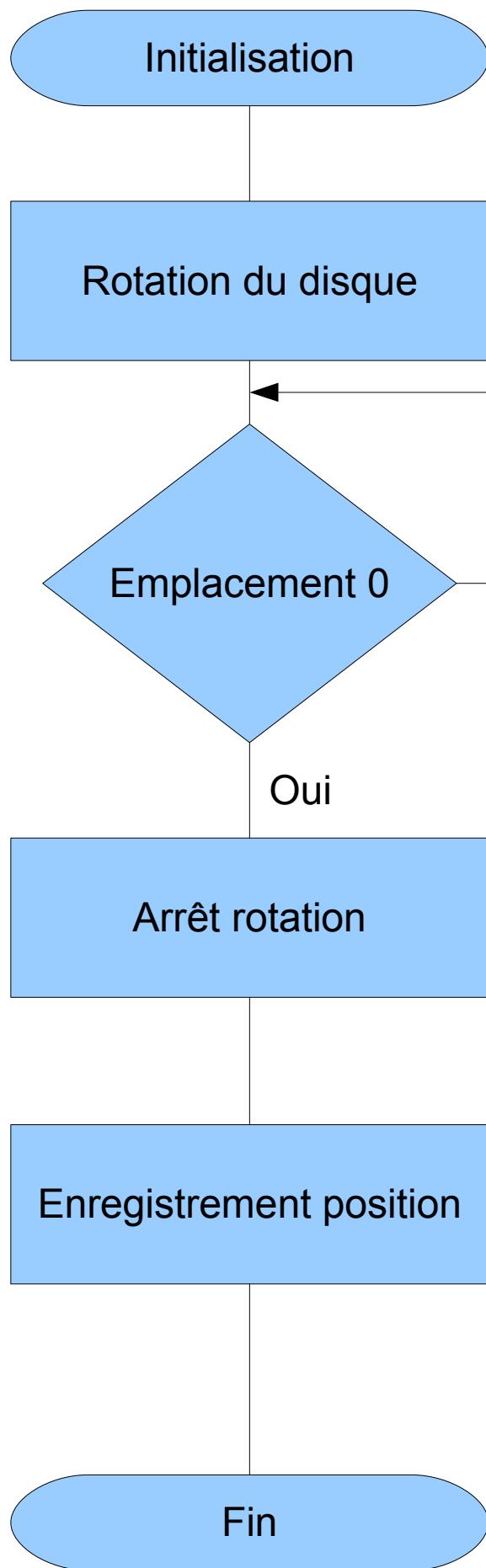


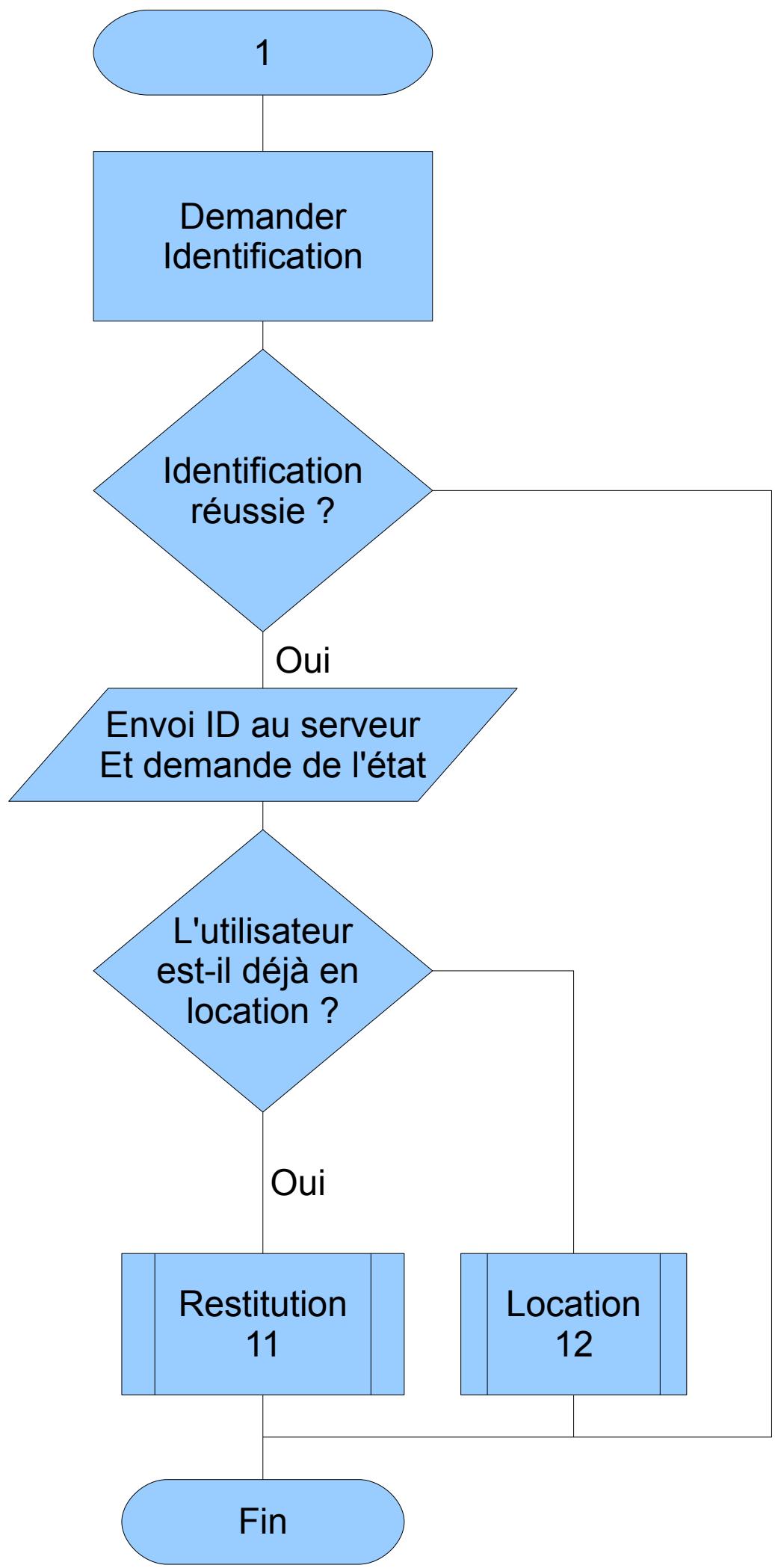
Actigrammes et datagramme niveau 3 : A21 Point de vue : concepteur

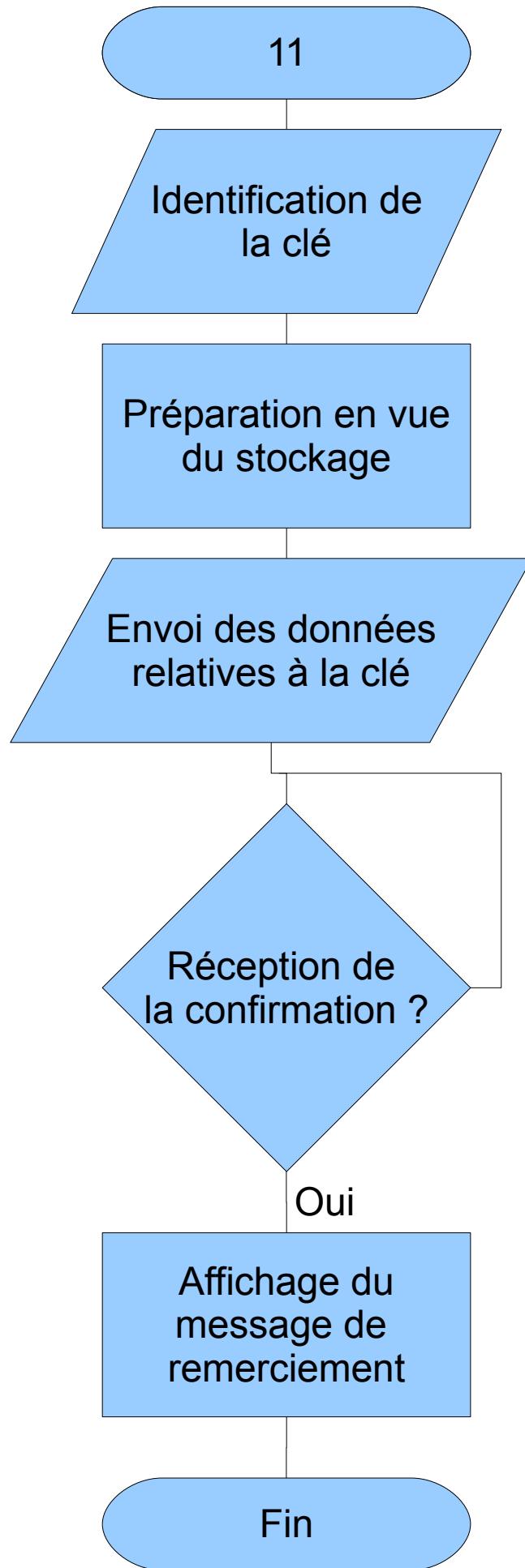
Annexe 2

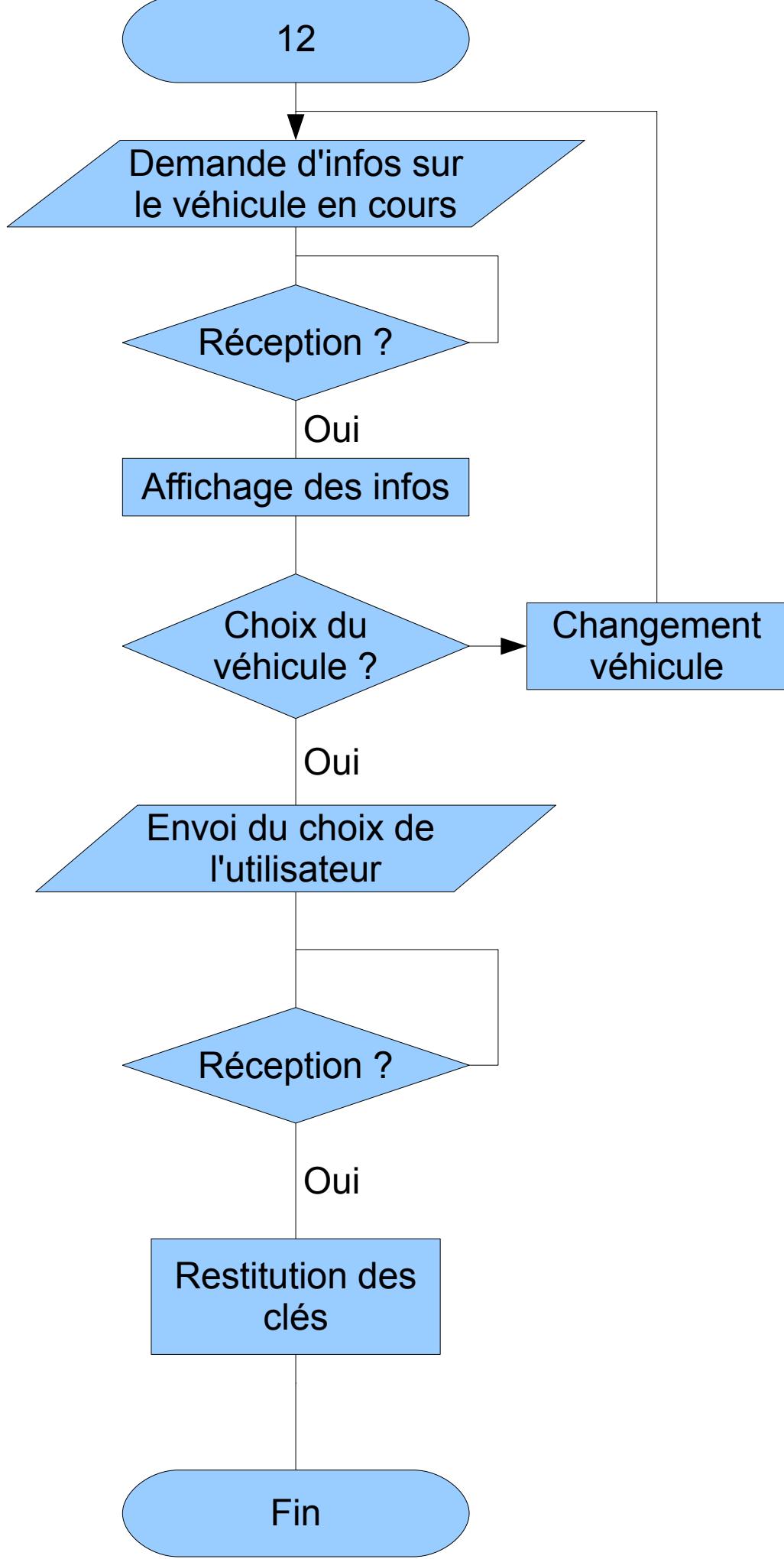


Annexe 2









Annexe 3

The Gantt chart illustrates the project timeline across 16 weeks, starting from week 39 and ending at week 16. The tasks are categorized by color:

- Chercher**: Red bars, including "Chercher un thème et une problématique" (weeks 39-40) and "Présentation du projet et finalisation du CdCF" (weeks 41-42).
- Présentation**: Blue bars, appearing in the first two weeks of the chart.
- Actigramme**: Yellow bars, appearing in the first four weeks of the chart.
- Pieuvre**: Green bars, appearing in the first five weeks of the chart.
- F.A.S.T.**: Light blue bars, appearing in the first six weeks of the chart.
- Algorithme simplifié du soft.**: Orange bars, appearing in the first seven weeks of the chart.
- Etudier**: Purple bars, appearing in weeks 45-52.
- Modéliser**: Pink bars, appearing in weeks 45-52.
- Réaliser**: Brown bars, appearing in weeks 52-16.
- Animer**: Grey bars, appearing in weeks 45-16.
- Etude**: Teal bars, appearing in weeks 45-52.
- Ecriture**: Light teal bars, appearing in weeks 45-16.

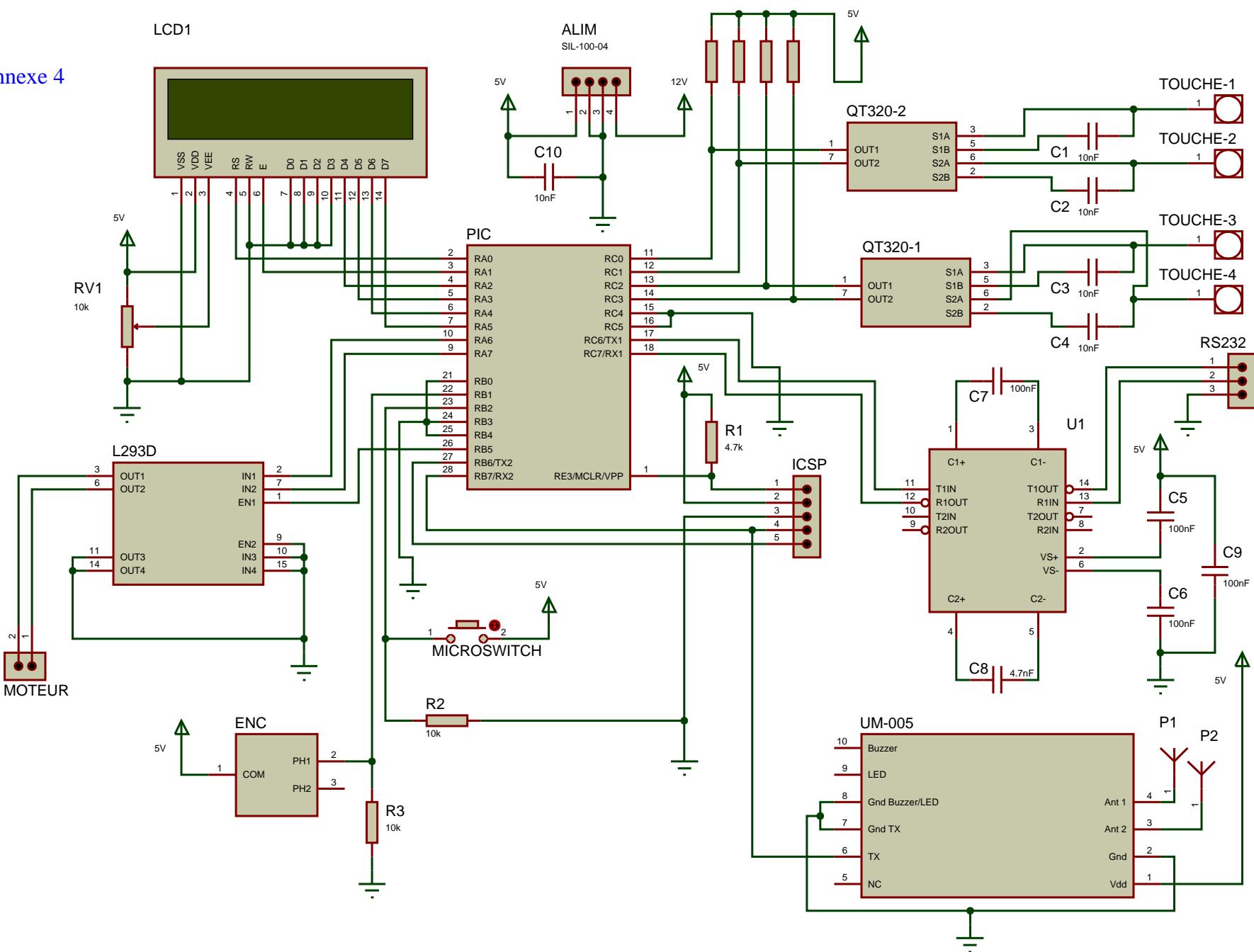
The chart shows a significant amount of overlap between tasks, particularly in the later weeks where multiple categories like "Etudier", "Modéliser", "Réaliser", and "Animer" are active simultaneously.

Annexe 3

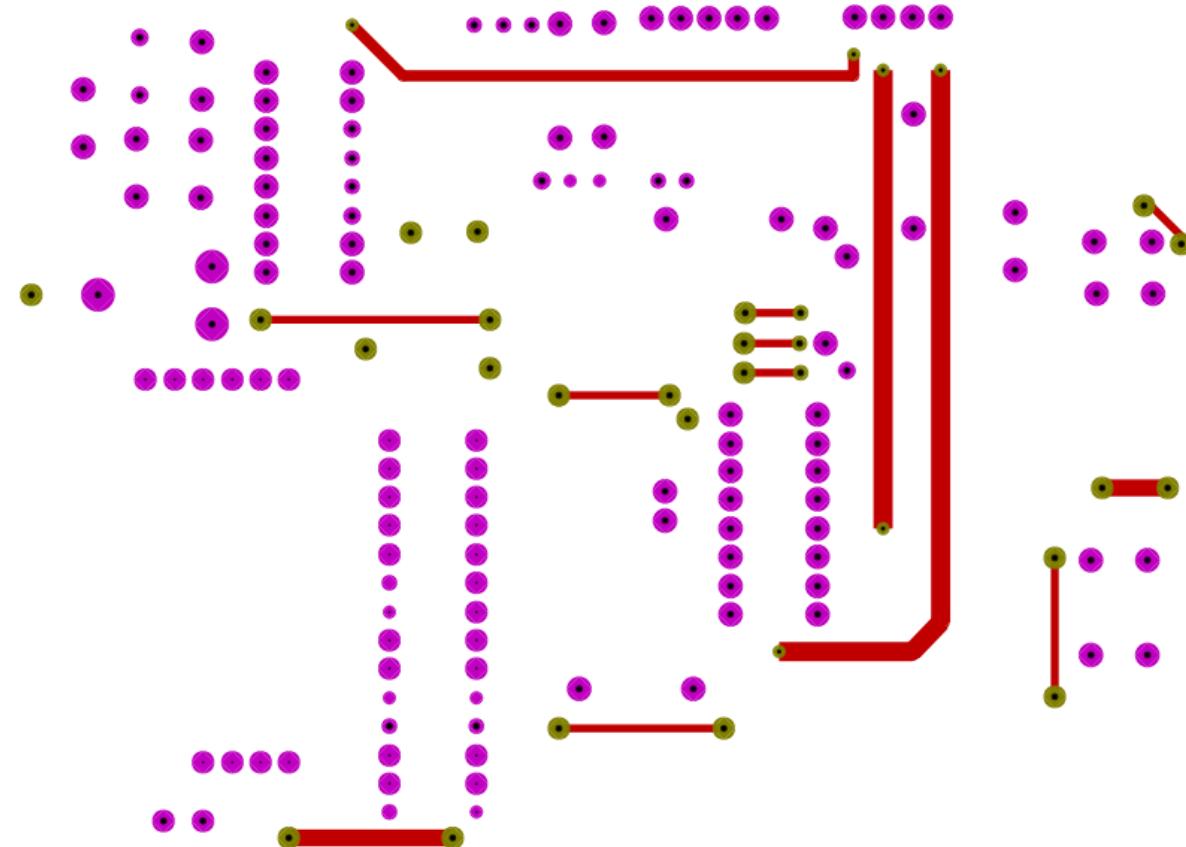
Taches et Semaines	39	40	41	42	43	44	45	46	47	48	49	50	51	52	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Ecriture des fonctions de la P.O.																														
Rédaction de tests																														
Réalisation d'un prototype de carte																														
Test de la carte sur le prototype																														
Réalisation de la carte électronique définitive																														
Réalisation du mécanisme définitif																														
Correction des dysfonctionnements																														
Entrainement à l'oral des TPE																														
Finalisation du dossier et de la présentation																														

Emilie
Jean
Thibaud
Tous

Annexe 4



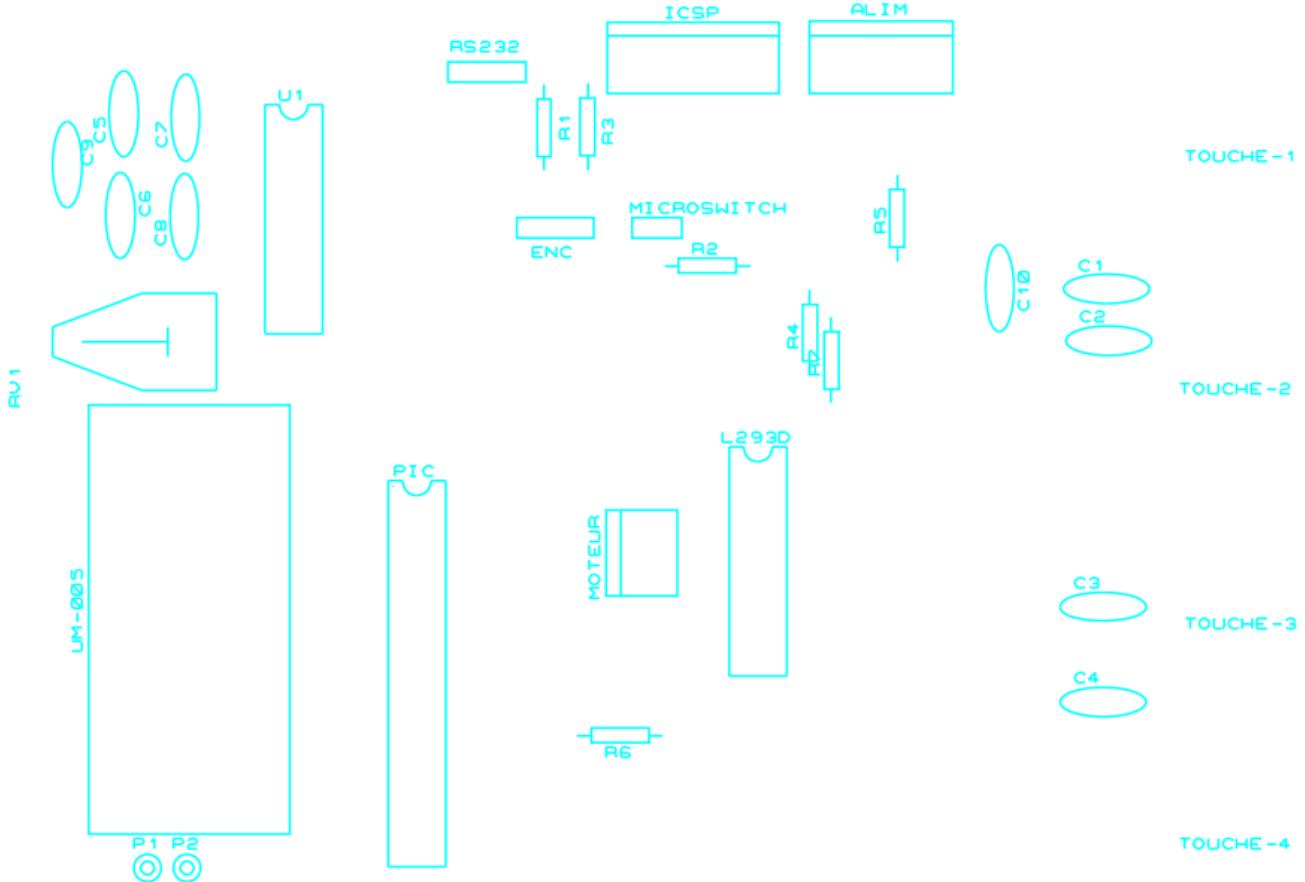
Annexe 5



Annexe 5

SO10V5011 - TPE Autopowerpass
Emilia C. Nezu C. Triplaud L.

Annexe 5



Annexe 5

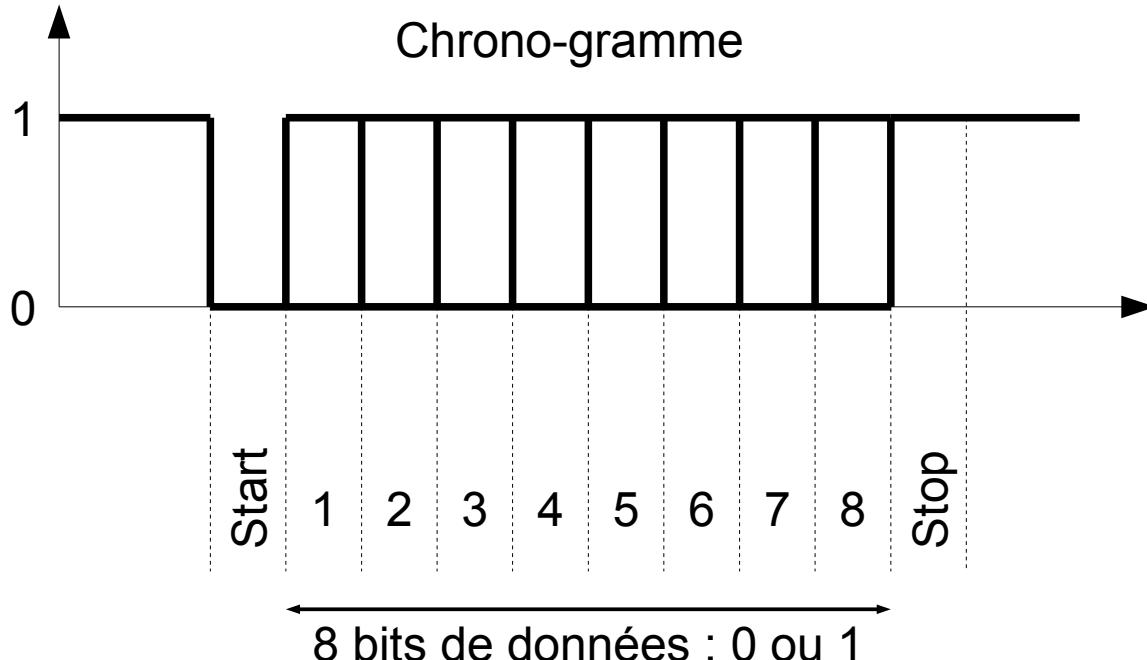
ΓCD 1



O1350-5



O1350-1



Caractéristiques de la liaison RS232 utilisée :

- 1 bit de START
- 1 bit de STOP
- 8 bits de données
- 0 bit de parité

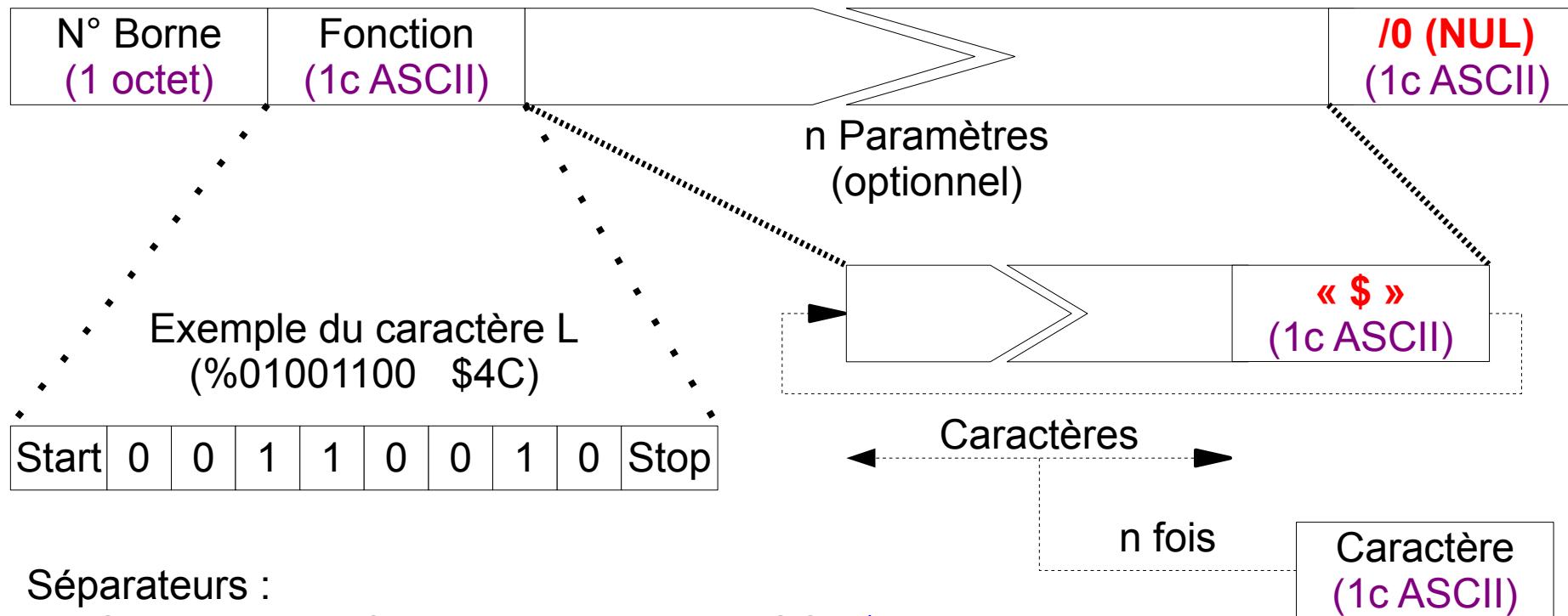
Vitesse : 9600bps

Format des trames RS232



Le LSB est transmit en premier !

Protocole BOSEP : Structure des paquets (Borne-SErveur Protocole)



Séparateurs :

/0 : Séparateur de fin de paquets, code ASCII **\$00**

« \$ » : Séparateur de paramètre, code ASCII **\$24**

Les paramètres, de longueurs supérieures à 8 bits, sont transmis octets les moins significatifs en premiers.

Les paramètres sont optionnels mais le dernier caractère **/0** est obligatoire. Il indique la fin de la trame. Soit pour une trame sans paramètre : N°borne – Fonction – **/0**

Protocole BOSEP - Description des fonctions :

Sens Serveur → Borne :

« **I** » : Identification, demande d'identification d'un tag RFID (cf. « T »), sans paramètre.

« **R** » : Réponse utilisateur, renvoi le nom et le statut d'un utilisateur, 2 paramètres :

- Nom de l'utilisateur, ASCII, **n octets**
- Statut de l'utilisateur, octet 1bit significatif, **\$00 ou \$01**

« **J** » : Envoi les informations sur un véhicule, 4 paramètres :

- Nom du véhicule, ASCII, **n octets**
- Prix du véhicule, ASCII, **n octets**
- Immatriculation du véhicule, ASCII, **n octets**

« **V** » : Validation d'un message ou d'une demande, sans paramètre

« **E** » : Erreur, sans paramètre

Sens Borne → Serveur :

« **T** » : Tag, envoi l'ID du tag RFID scanné, 1 paramètre :

- ID du tag RFID, **5 octets**

« **U** » : Utilisateur, demande d'infos sur un utilisateur (cf. « R »), 1 paramètre :

- ID du tag RFID de l'utilisateur, **5 octets**

« **D** » : Demande d'info, réclame les information sur un véhicule (cf. « J »), 1 paramètre :

- ID du tag RFID de la clé, **5 octets**

« **L** » : Location, signifie la location d'un véhicule, 2 paramètres :

- ID du tag RFID de la clé, **5 octets**
- ID du tag RFID de l'utilisateur, **5 octets**

« **C** » : Clé, Signifie la restitution d'une clé, 1 paramètre :

- ID du tag RFID de la clé, **5 octets**

« **A** » : Appel, demande d'intervention sur la borne, sans paramètre.

Annexe 7

Instructions	Code										Description	Durée
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Efface l'ensemble de la mémoire de donnée sans toucher au générateur de caractères. Ramène le curseur en position « home », à l'adresse 00.	1,64 ms
Return home	0	0	0	0	0	0	0	0	1	X	Ramène le curseur en position « home », à l'adresse 00. Si l'affichage était décalé, il est remis à sa position d'origine : l'adresse 00 se trouve à nouveau en haut à gauche.	1,64 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Définit le sens de déplacement du curseur après l'apparition d'un caractère (vers la gauche si I/D=1, vers la droite si I/D=0) et si l'affichage accompagne le curseur dans son déplacement ou non (S).	40 µs
Display on/off control	0	0	0	0	0	0	1	D	C	B	Met l'affichage en ou hors fonction l'affichage (D), le curseur (C), le clignotement du curseur (B).	40 µs
Cursor and display shift	0	0	0	0	0	1	S/C	R/L	X	X	Deplace le curseur (S/C=1) ou l'affichage (S/C=0) d'une position vers la gauche (R/L=1) ou la droite (R/L=0) sans changer la DD RAM.	40 µs
Function set	0	0	0	0	1	DL	N	F	X	X	Définit la taille de l'interface (DL=0 pour mode 4 bits, DL=1 pour mode 8 bits), le nombre de lignes (NL=0 pour 1 ligne, N=1 pour 2 ou 4 lignes), et la taille des fontes (F=0 pour des caractères 5x7, F=1 pour des caractères 5x10).	40 µs
Set CG RAM address	0	0	0	1	A5	A4	A3	A2	A1	A0	Définit l'adresse de la CG RAM. Les données de la CG RAM sont envoyées après cette commande.	40 µs
Set DD RAM address	0	0	1	A6	A5	A4	A3	A2	A1	A0	Définit l'adresse de la DD RAM. Les données de la DD RAM sont envoyées après cette commande.	40 µs
Read busy flag & address	0	1	BF	A6	A5	A4	A3	A2	A1	A0	Lit le flag busy (BF), et l'adresse de la position du curseur. BF vaut 0 si l'afficheur accepte une instruction, 1 s'il est occupé	1 µs
Write data to CG or DD RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Ecrit des données dans la DD RAM ou la CG RAM.	40 µs
Read data	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Lit les données de la DD RAM ou de la CG RAM.	40 µs

Annexe 8

