

Rapport : Deuxième Soutenance
Projet : C

Les Phénoménoïdes

Guillaume Goussard
Sylvain Piot
Joël Lafaille

4 mars 2004

Table des matières

1	Présentation du projet	3
1.1	Origine du projet	3
1.2	But du projet	4
2	Réalisation du projet	5
2.1	Aspect matériel	5
2.1.1	Structure	5
2.1.2	Electronique	9
2.2	Aspect logiciel	12
2.2.1	Partie Commune	12
2.2.2	Windows	13
2.2.3	Unix	15
2.2.4	Pilote USB	16
2.3	Partie Théorique	17
2.3.1	Protocole Client Serveur	17
2.3.2	Protocole Serveur Machine	18
2.4	Site	19
3	TODO	20
3.1	Aspect matériel	20
3.1.1	Structure	20
3.1.2	Electronique	20
3.2	Aspect logiciel	20
3.3	Plannings	21
3.3.1	Planning des soutenances	22
3.3.2	Planning des tâches	23
A	Références	25
B	Schémas préliminaires	26

Introduction

Ce rapport à pour but de vous présenter le projet scolaire d'informatique de l'équipe dite des "Phénoménoïdes". Le projet étant libre (dans la limite que ce ne soit pas un jeu) nous avons décidé de présenter un projet de machine à café automatique et contrôlée à partir d'un ordinateur. Ce projet nous apprendra ainsi à manier aussi bien fer à souder que le clavier, car portant aussi bien sur une partie électronique/mécanique que sur une partie logicielle. Nous allons vous présenter dans ce rapport l'avancement de ce projet et les différentes parties que nous avons réalisées durant ce premier trimestre.

Chapitre 1

Présentation du projet

L'idée de contrôler une machine à café à partir d'un ordinateur peut paraître saugrenue de prime abord, mais elle découle de l'observation que la boisson préférée de l'informaticien est le café (rapport aux nombreuses nuits blanches). L'implantation de nombreux cyber-cafés (et non cyber-bars ou autres) qui dénote l'addiction dont souffre les personnes pratiquant l'informatique à haute dose.

1.1 Origine du projet

Nous allons vous décrire la façon dont nous est venue cette idée de machine à café "télécommandée" pour cela, remontons un peu dans le temps. A l'origine, il n'y avait rien. Que du vide, quand une spectaculaire explosion se produit... Je suis peut-être remonté un peu loin, descendons le temps jusqu'au moment où plusieurs cerveaux de génie ont unis leurs forces cognitives pour avoir une idée de génie, on peut le dire. Nous parlions d'Unix et de sa commande `man` très utile quand notre majestueux chef de projet s'est rappelé qu'il avait vu sur une Mandrake le résultat de la commande `man C`. Cette commande, au lieu d'expliquer le fonctionnement du `C` comme on pourrait à priori le penser, donne les instructions de contrôle en ligne de commande d'un distributeur à café. Il se rappela également du `httpc`, un protocole réseau basé sur l'`http` pour le contrôle d'une machine à café. D'où l'idée logique d'assembler la rfc 2324 (`htpcipc`) et le `man C` pour créer réellement cette machine à café.

1.2 But du projet

Le but du projet est de réaliser physiquement un distributeur de café qui puisse produire réellement un liquide caféiné chaud non toxique. Et aussi accessoirement nous apprendre les mécanismes des réseaux et entrée/sortie physique dans un système. Ainsi que le contrôle d'appareils externes au pc par l'intermédiaire du port USB.

Chapitre 2

Réalisation du projet

Nous allons réaliser un distributeur à café créé de toutes pièces (armature, doseurs, ainsi que toute pièce s'avérant nécessaire à toute bonne cafetière qui se respecte), de ce fait nous allons vous présenter les deux grandes parties du projet : la partie logicielle et la partie matérielle, c'est par cette dernière partie que nous allons commencer.

2.1 Aspect matériel

Une partie du projet sera physiquement présente : la machine en elle-même. Comme il s'agit de la première chose que l'on verra quand le projet sera fini, bien que nous n'y avons pas consacré la plus grande partie de notre temps pour cette soutenance, nous allons la décrire en premier.

2.1.1 Structure

Cette soutenance comprenait la création de la structure dans sa version finale, ce que nous avons réalisé. On peut diviser cette partie en trois paragraphes : les plans, la construction proprement dite et les anecdotes liées à celle-ci.

Plan

Nous nous sommes pour cette soutenance lancés dans la réalisation de la structure de notre distributeur à café. Nous n'avons pas voulu bâcler cette partie qui sera très importante pour la suite et nous avons donc réfléchi avant de commencer quoi que ce soit, c'est pourquoi nous avons commencé par faire des plans de ce que nous voulions. Ces plans ont permis à tout le groupe de bien voir à quoi pourrait ressembler notre machine, mais surtout à corriger les erreurs qui ont pu apparaître sur nos schémas et qui nous auraient bloqué pendant la construction. Chacun a proposé ses idées et nous avons pu décider ensemble de la meilleure solution à apporter à chaque problème. Nous avons

réalisé de nombreux schémas sur toutes les parties de notre distributeur, nous avons essayé de ne rien laisser au hasard, afin de repérer les futurs problèmes. Nous n'utilisons pas de systèmes très complexes mais il fallait réfléchir au cas par cas à chacun de ces systèmes. Nous avons donc réalisés des plans pour une structure simple, solide et permettant pour la suite d'être adaptée à la partie mécanique et électronique qui y sera intégrée. Ce travail préliminaire nous a grandement facilité la suite du travail et nous a permis de gagner un temps précieux. Nous avons donc fait des plans de la structure en général, ainsi que de chaque étage en particulier. Nous avons cependant modifié le premier étage (celui des réservoirs a café et des moteurs) car nous avons trouvé un moyen bien plus efficace et plus rentable de fixer le tout lors d'une de nos innombrables visites au bricorama du coin. Certaines modifications ont été apportées sans l'aide préliminaire que peuvent fournir des plans, mais en général nous y avons eu recours avant pour se mettre d'accord.

Construction

Nous sommes allés acheter les différentes pièces de ce qui devra plus tard être la structure définitive de notre distributeur à café, en suivant les plans que Sylvain a fait (cf. annexe). Ces achats se sont élevés à plus d'une centaine d'euro (nous mettrons une partie dépenses/revenus dans le rapport final), et peuvent être résumés ainsi :

- 5 cornières acier d'un mètre de long servant aux arrêtes externes. Dimensions : 30x30x1000, peint blanc.
- 4 cornières aluminium d'un mètre servant au support en longueur du bac du deuxième étage, et au support des conteneurs de cafés et de sucre. Dimension : 15x15x1000, laqué noir.
- 4 barres plates d'acier étiré d'un mètre servant au support en largeur du bac du deuxième étage, et au renforcement et support des moteurs et de l'électrovanne du premier étage. Dimension : 20x1000, acier verni.
- 3 conteneurs a cafés et sucre. Nous avons trouvé des solides en plastique pouvant nous être utiles, initialement destiné à l'usage sanitaire (tuyauterie). Ce sont en fait des raccords d'un diamètre de 5,7cm vers 4,7cm.
- 3 pinces en plastique servant à tenir les conteneurs, eux aussi initialement destiné a la tuyauterie.
- 1 grille en zinc permettant de poser le gobelet, soutenu par un assemblage de cornières et barres diverses.
- 2 coulisses servant au support et déplacement du réservoir, vissé sur les barre plates. Dimensions : 27x278, acier zingué.
- 1 réservoir a eau + lait, c'est en fait une boîte à outil en plastique de 33x23x3,5cm, divisé en plusieurs conteneurs à l'intérieur (un peu à la

manière du Titanic). Nous avons donc du évider toutes les séparations sauf une, de manière à pouvoir mettre eau et lait dans le même réservoir. Les capacités sont de 1914 cm³ pour l'eau, et de 493cm³ pour le lait.

- 3 boîtes de boulons de 12, une de boulons de 8. - 3 mèches a perceuse, spécial acier renforcé dont une en titane. Dimensions : une de 6mm, une de 4mm, et une de 3,5mm finalement non utilisée.
- 2 lames de scie à métaux, usage " pro ".
- Quelques autres choses, tel que des bouchons, cutters...

Nous avons aussi utilisé le matériel du père a guillaume, à savoir la perceuse et la scie a métaux, les deux dotés de mèches et de lames inutilisables, expliquant les achats cités ci-dessus. Lors des parcours de reconnaissance au bricorama du coin pour se donner des idées avant de passer à la réalisation des plans, nous avons imaginé des systèmes de remplissage/vidange du réservoir et des conteneurs " avec les moyens du bord ". Ainsi, nous avons pensé à un système de coulisse du réservoir, permettant, avec un clapet coté carrosserie, de faire sortir à moitié le réservoir par l'arrière. Ce système permet le remplissage par le couvercle, et la vidange par deux trous (eau + lait) pratiqués dans le fond et possédant des bouchons. Il faudra travailler un peu l'étanchéité du réservoir, car au départ, les boîtes à outils ne sont pas prévues pour cet usage... Pour les conteneur de cafés et de sucre, nous avons prévu de mettre un système d'entonnoir (avec les cotés très raides de manière à ne pas donner au café en poudre l'occasion de se coincé sur la pente) et un moteur pas à pas dessous régulant le débit grâce a une sorte de roue bloquant la sortie de entonnoir, et entraînant le café/sucre lors de la rotation. De cette manière, nous pourront satisfaire presque n'importe quelle commande, car les moteurs pas à pas sont très sensibles et nous pourrons les faire tourner très progressivement. Il a cependant fallu positionner les conteneurs le plus verticalement possible par rapport au gobelet, car si le café ne pourra pas stagner dans le réservoir, rien ne dit qu'il ne le fasse pas sur la glissière entre moteur et gobelet. Coté dimension de la structure, nous avons choisis des faire suffisamment grand, de manière a parer aux éventuels problèmes de place que pourrait poser un ajout de réservoir ou une modification de l'agencement des pièces. Le solide ainsi formé par les barres externes à une dimension de 50cm de hauteur, 40 de profondeur et 30 de largeur. Ceci peut paraître excessif lors d'un premier abord, mais le résultat final ne choque pas, la structure parait simplement solide.

Anecdotes

Plusieurs problèmes se sont posés lors de la construction de la structure, dont certain assez amusants. Nous avons tout d'abord commencé en coupant les cornières en acier à la scie à métaux. Ces cornières devaient servir aux arrêtes du parallélépipède. La lame de la scie à métaux gracieusement fournie par le père à guillaume étant totalement usée, nous avons forcé comme des brutes (moi et sylvain, guillaume boulonnait) pour réussir à couper deux barres. Complètement crevé, nous avons ensuite décidé de couper les petites cornières en alu, chose plus simple que de couper l'acier, mais tout de même pas rapide. Encore plus vanné, nous nous sommes attaqué aux lames d'acier étiré (nous n'avions pas vu le type d'acier, juste que ça avait l'air assez mou comme métal). 20mn après, on était toujours dessus, alors on a décidé de percer les trous dans mes cornières qu'on avait réussi à couper. Percer un trou dans celle en alu nous ayant prit près de 5mn, nous avons décidé d'essayer l'acier galvanisé, avec la mèche spéciale acier fournie avec la perceuse. Le premier essai n'ayant pas abouti, j'ai décidé de prendre les choses en main et j'ai mis la perceuse sur la puissance maximale. Après 20 secondes, la pièce étant envahie par une fumée tenace et puante, nous avons regardé la mèche : complètement fondue, et la barre d'acier : même pas percée. Là on a craqué, on a décidé d'aller chercher des mèches dignes de ce nom au arrêtes, et on a prit les plus cher (spécial acier renforcé, doublé titane...). Moyenne pour percer les barres après achat : 3 secondes... Le même rapport est valable pour couper à la scie à métaux.

2.1.2 Electronique

La partie électronique est la partie qui nous a donné le plus de fil à retordre et ce pour les raisons que nous allons vous expliquer.

Electronique numérique

L'électronique numérique est le coeur de notre système électronique embarqué dans la machine, en effet c'est cette partie qui sert d'interface entre le monde numérique issu de l'ordinateur via le câble USB et les différentes parties mécaniques. C'est pour cette raison que nous y avons consacré une grande partie de notre temps.

Le composant central de cette partie est le microcontrôleur PIC16F877 de Microchip. Nous avons choisi ce circuit car le laboratoire d'électronique microcontrôleur/microcontrôleur possède tous les outils nécessaire à sa programmation, ces outils sont hors de nos moyens financiers car le circuit de développement ainsi que toutes la connectique et les logiciels coûtent environ 150 euros. L'autre raison de ce choix a été le rapport qualité/prix de ce composant.

Le microcontrôleur PIC16F877, cadencé à 20Mhz, possède un jeu d'instructions de type RISC (Reduced Instruction Set Computer), c'est à dire qu'il possède un nombre d'instructions restreint (trente cinq instructions) ce qui lui permet d'avoir un câblage interne optimisé et de moins chauffer. Il possède une mémoire flash de 8ko pour le programme, de 368 octets de mémoire RAM, et de 256 octets de mémoire EEPROM. La communication avec celui-ci s'effectue au moyen des 33 ports d'entrée-sortie, il dispose également de 2 timers internes, et de port de communication série RS232 ce qui nous est très utile pour le brancher avec le module de conversion USB vers Série à base de FT232BM (de FTDIChip).

La programmation de ce composant se fait par l'intermédiaire de l'ICSP (In-Circuit Serial Programming) et le ICD (In-Circuit Debugging) qui sont un modules de débogage relier au PC par un câble USB et un circuit de démonstration et nécessite l'emploi d'un logiciel de développement MPLAB IDE. L'ensemble nous est prêté par le Lab-elec.

Pour nous familiariser avec la programmation sur PIC nous avons tout d'abord fait le premier TP de Tek1 sur les pic (alias le TP K2000) en assembleur. Ensuite pour nous facilité la programmation nous sommes passés au C sur PIC. Le logiciel de développement IAR Embedded Workbench de IAR System fourni un très bon compilateur C pour PIC. C'est là que les problèmes ont commencé car n'ayant aucune expérience de programmation sur un tel système le passage du C "classique" de PC au C du pic a été brutale. Nous n'avons plus accès à des fonctions évoluées pour accéder au système. Nous avons donc décidé dans un premier temps de réussir une connection série avec un PC ce qui n'est pas compliqué sur un PC grâce au

system d'exploitation deviens alors plus ardu car il faut connaître parfaitement le microcontrôleur, c'est donc armé de la documentation que nous avons commencé à programmer le PIC. Pour initialiser une connexion série il faut non seulement savoir où se trouve le "port série" du PIC, ce port se résume dans notre cas à deux pattes, une pour la réception et une pour la transmission, mais aussi savoir comment initialiser les registres qui activent ou non la communication série RS232.

Une connexion série RS232 se fait octet par octet par l'intermédiaire de deux fils RX (pour la réception) et TX (pour la transmission), la vitesse de transmission doit être identique à celle d'acquisition du récepteur. Nous utilisons le mode asynchrone de ce type de transmission (pas d'horloge commune entre l'émetteur et le récepteur ce qui oblige à utiliser deux bits supplémentaires : les bits de start et de stop. Pour pouvoir utiliser l'interface RS232 du PIC16F877 il suffit de connecter le RX de l'interface série sur le bit 6 du PORTC et le TX sur le bit 7 de ce même port. Ensuite il faut paramétrer la transmission sur le PIC en activant deux registres (RCSTA et TXSTA) avec des valeurs spécifique à la connexion qui sont pour nous respectivement 0x90 et 0x26 ensuite on choisi une vitesse de transmission et on calcul la valeur du registre SPBRG d'après la formule donnée par le constructeur $Baud = \frac{FOSC}{16(SPBRG+1)}$ où FOSC correspond à la valeur du quartz qui sert à cadencer la connexion. Dans la plateforme de développement nous utilisons un quartz cadencé à 16Mhz ce qui nous fait un SPBRG à 103.

Electronique Analogique

Une fois la connexion série via USB établie et les informations transmises par le protocole décrit plus tard il nous faut commander les différents moteurs et électrovannes.

Pour les moteurs notre choix s'est arrêté sur les moteur pas à pas à 48 pas car ils sont très bon marché. Les moteur pas à pas sont de 3 type : aimant permanent, reluctance variable, ou hybride.

Les moteurs à aimant permanent possèdent un aimant permanent solidaire de l'axe du moteur (rotor) et des bobines excitatrices placées sur les parois du moteur (stator) qui sont alimentés chronologiquement. Le rotor s'oriente suivant le champ magnétique crée par les bobines.

Les moteurs à reluctance variable comporte un rotor à encoche se positionnant dans la direction de la plus faible reluctance¹. Le rotor, en fer doux, comporte moins de pôles qu'il n'y a de pôle au stator. Le fonctionnement du moteur est assuré par le pilotage du type unipolaire et l'avance du rotor est obtenu en excitant tour à tour une paire de pôle du stator.

Les moteurs hybrides sont simplement un mélange entre les deux précédents types.

¹La reluctance est le quotient de la force électromotrice d'un circuit magnétique par le flux d'induction qui le traverse

Dans notre cas nous avons des moteurs bipolaires ce qui implique la fabrication d'une interface entre la commande donnée par le PIC (sur un bit continu) et le moteur qui doit recevoir un courant alternatif bipolarisé. Cette carte d'interface n'est pas très compliquée à fabriquer et on trouve de nombreux schémas sur Internet car ces moteurs servent principalement en astronomie amateur pour éviter des effets néfastes de la rotation de la terre sur les observations de longues durées et les photos à long temps de pose. Pour les électrovannes notre choix n'a pas encore été fait car la plupart des électrovannes du marché sont des électrovannes à débit trop élevé pour notre usage mais leur fonctionnement est assez simple : il suffit de mettre l'électrovanne sous tension pour l'ouvrir ce qui assure de fait une sécurité contre l'inondation en cas de coupure de courant.

Connectique

Depuis le début du projet nous avons décidé que l'appareil sera relié à l'ordinateur par l'intermédiaire d'un câble USB car cette interface est maintenant très commune et surtout très facile d'emploi. Pour gérer la connection nous utilisons un composant assez répandu dans l'industrie : le FT232BM, après l'installation d'un driver sous Windows, il ne reste plus qu'à utiliser une dll fournie par FTDI Chip, sous linux le driver est déjà intégré dans le noyau à partir de la version 2.4. Le FT232BM est monté dans un circuit qui permet d'obtenir une information en série via USB.

Une fois que la connexion série sera fonctionnelle, ce qui ne saurait tarder vu nos récents progrès en programmation sur PIC, il nous restera à créer un montage définitif en remplaçant le port série de la plaque de montage par le composant référencé DLP-232M chez FTDIChip mais produit aussi par d'autres fabricants. Il nous reste aussi à créer les mécanismes de dosage mais pour cela nous avons des idées qu'il ne reste qu'à mettre en pratique ainsi que les montages de contrôle des moteurs pas à pas.

2.2 Aspect logiciel

Après moult débats dans le groupe, la partie logicielle tournera sur les deux principales familles de système d'exploitation : Unix et Windows. Pour obtenir une plus grande interopérabilité entre ces systèmes, nous testerons sur de multiples versions : FreeBSD, Linux (kernel 2.4.x et 2.6.x), et Windows XP, 98 Pour les OS n'ayant pas de portage réussi (autre que Windows et Linux) nous aurons un système de réseau permettant quand même le contrôle de la machine mais via un OS plus commun.

2.2.1 Partie Commune

Le système de serveur est actuellement porté sur deux systèmes d'exploitations (Windows et Linux). Le principe est commun au deux systèmes, il possède deux "têtes" : une tête qui s'occupe de traiter les données avec la machine, comme nous n'avons pas encore les moyen techniques fonctionnelle de tester de la communication avec le module USB FT232xx via le port USB cette partie est actuellement développée sous Windows et en cours sous Linux. En revanche le serveur est assez bien avancé il accepte la connexion et prend en compte le temps de latence du a la "fabrication du café". Mais comme cette partie est dépendante du système d'exploitation nous les présenterons dans les parties correspondantes.

Les clients eux fonctionnement exactement pareil en se connectant au serveur et en lui envoyant la commande selon une structure que nous développerons dans la partie Protocole codé sous forme d'un entier. Le serveur Unix est capable de communiquer avec le client Windows, et inversement car nous utilisons des deux cotés le port 4242. Nous simulons le traitement de la commande par une pause du serveur, qui lorsqu'il est dans cet état, bufférise les commandes suivantes et les exécute à la file. Pour communiquer en réseaux nous utilisons le protocole TCP car actuellement nous n'envoyons qu'un entier mais ultérieurement nous gèrerons une identification sécurisée qui nécessite de garantir l'intégrité des paquets envoyés, ce que n'offre pas le protocole UDP. Nous allons maintenant détailler les systèmes de serveurs sur les deux OS supportés.

2.2.2 Windows

Pour la partie Windows, la version que nous avons faite pour la première soutenance a été totalement modifiée et revue

Serveur

L'ancienne version du serveur, bien que entièrement fonctionnelle, n'avait eu pour but avoué que de mettre en pratique nos connaissances des entièrement et des api windows. Dans cette nouvelle version (définitive normalement), nous avons réglé les quelques problèmes restant, à savoir le lancement en processus système du serveur, et le log des commandes reçues. Nous avons réussi à mettre un bouton minimiser qui permet d'envoyer la fenêtre réduite dans la barre système (entièrement). Il suffit de double cliquer sur l'icône (fait pour l'occasion, a savoir une superbe tasse de café avec un "C" dessus) pour relancer la partie graphique du serveur. L'interface graphique n'étant aucunement utile vu que l'on ne peut commander que depuis le client, nous avons mit pour remplir une fenêtre de log inscrivant toutes les commandes reçues (et les éventuelles erreur lorsque la partie électronique/mécanique sera opérationnelle), log aussi enregistré dans un fichier log.txt crée dans le même répertoire que l'exécutable. Bien sur, les nouveau log sont rajouté à la fin des anciens, de manière à avoir un log du mois si nécessaire. Histoire de remplir un peu plus l'interface, nous avons rajouté un résumé des connexions reçues, a savoir le nombre de commandes reçues, et le nombre de commandes traitées. Le serveur est agencé en plusieurs thread, car lors de la première soutenance, nous étions tombé sur un problème de rafraîchissement de la fenêtre : en effet, la fonction qui écoute sur le port précisé et accepte les éventuelles connexions est bloquante, ce qui implique l'arrêt du rafraîchissement de la fenêtre, car ce rafraîchissement est fait par une procédure système présente dans tous les programmes : `OnPaint()`. Pour régler ce problème, nous avons lancé cette fonction dans un thread séparé du reste du programme, de manière a écoute en permanence sur le écoute, et a accepter les "incoming connections". Ce thread, lorsqu'il reçoit une commande, l'enfile dans une file passée en paramètre global, avec toutes les infos nécessaires a l'envoi de la commande a la future machine. Ici un autre problème s'est posé, à savoir défiler la file dès qu'un élément y est présent. Pour ce faire, nous avons créé un autre thread qui défile en permanence et envoie la commande à la cafetière. Bien évidemment, pour ne pas utiliser 99% du processeur avec ces threads dans leur boucle `for(;;)`, un `sleep(1)` a été rajouter, ce qui a provoqué le passage du 99% a 0. Ainsi, coté serveur, tout est en place pour permettre l'envoi des données a la future machine (nous compilons en effet avec la arrêt fournie par FTDI, ce qui permet d'utiliser toutes les fonctions de transfert et d'initialisation de l'électronique, mais pas celles d'initialisation du port COM virtuel nécessaire pour l'utilisation du conver-

tisseur usb/série que nous utilisons. Il reste peut être un dernier problème... lors du plantage d'explorer.exe, au relancement de ce processus, l'icône ne réapparaît pas dans le systray... cependant plusieurs autres programmes ne gèrent pas cette exception, pour ne pas le citer : décoche bien sur...

Client

Le client a été entièrement revu, de manière à être plus attirant, plus simple d'utilisation, et bien plus optimisé. Nous avons remplacé les x boutons "Commande n°x" par un moyen de commander a peu près n'importe quel dosage de café, sucre, lait : Des check box ont été mise pour permettre le choix du café (bien évidemment en cocher une décoche toutes les autres), et une barre de progression permettent de choisir le dosage voulu (avec pour l'instant une limite de 4 choix, mais si l'électronique marche bien, nous pourront en mettre plus d'une dizaine si nécessaire). Deux autres check box sont utilisée pour choisir entre café long et café court, ainsi que deux pour les choix sans sucre et sans lait. Deux autres barres de progression sont disponibles pour choisir les dosage de sucre et lait, qui sont grisés si le choix "sans sucre" et "sans lait" est fait. Lors de chaque choix, la commande est envoyée au serveur sous la forme d'un int passé en chaîne de caractère, qui est en fait une structure remplie en fonction des différents choix castée en int lors de l'envoi, et recastée en structure lors de la réception. De cette manière la structure est recrée, sans aucune perte d'information, et coté serveur nous disposons de toutes les informations nécessaires au traitement de la commande. Ici aussi nous avons créé un thread pour envoyer la commande : en effet si le serveur n'est pas trouvé sur l'ip spécifiée, la fonction send sera bloquante pendant plusieurs secondes avant de renvoyer un code d'erreur. L'ip du serveur est a choisir avant la commande, (par défaut nous avons mit 192.168.0.2 car nous étions en réseau local lors du développement), et il est possible de la modifier entre chaque commande sans relancer le client.

2.2.3 Unix

A la dernière soutenance le client était un simple socket qui envoyait les deux arguments placé sur la ligne de commande. Pour cette soutenance nous avons décidé de nous pencher sur l'un des points de départ du projet : la concordance du client Unix en mode console avec le man C et avec le protocole de communication avec le serveur que nous avons établi.

La partie de concordance avec le man C n'a pas été techniquement très compliquée, le plus dur a été de choisir les arguments que nous pouvons utiliser et ceux qui sont techniquement irréalistes comme l'argument `-other=drink`. Pour le reste un parsing de la ligne de commande a suffit à régler le problème. Ensuite nous avons décidé d'agrémenter le client de graphiques en mode texte à la manière des progressbar du client windows.

De plus il a fallu nous adapter au nouveau protocole de communication que nous avons établi de la même manière que sous Windows.

2.2.4 Pilote USB

Une documentation complète existe a propos du DLP-USB232, ainsi qu'un driver pour Windows et linux comprenant une bonne centaine de fonctions diverses et variées permettant de communiquer avec ce dernier. Ces fonctions sont utilisables en langage C/C++, et permettent non de programmer l'EEPROM (un logiciel permettant de le faire est livré avec le microcontrôleur FT232BM), mais de lui envoyer les données que le serveur reçoit, d'initialiser les circuits, ou utiliser certaines des fonctions hardware. De cette manière le pilote envoie le code des commandes au circuit électronique, grâce par exemple à la fonction FTSend, puis le circuit électronique s'occupe d'abord de convertir les données du port USB vers le port serie, puis de les interpréter de manière à commander l'activation des différents composants de la machine au bon moment et surtout pendant la durée prédéterminée. Cette partie sera complétée lorsque nous aurons les circuits de manière à tester les différentes commandes. Nous avons réussi à compiler un programme utilisant les fonctions de la librairie téléchargeable sur les site www.ftdichip.com.

2.3 Partie Théorique

Le fonctionnement de notre projet est basé sur une communication entre différentes parties : Client -> Serveur -> Machine. Pour faciliter cette communication il nous a fallu définir des protocoles de communication.

2.3.1 Protocole Client Serveur

Nous avons voulu choisir un protocole de transfert de données du client au serveur le plus clair possible en restant complet et ayant une possibilité d'évolution par la suite. Nous nous sommes inspirés du protocole RIP. Nous avons tout d'abord pensé à utiliser la RFC2324 mais cela n'était pas intéressant car mal adapté et le principe utilisé par RIP est extrêmement simple et facile à implémenter, c'est un protocole de routage, au sein d'un réseau, où les routeurs s'envoient de proche en proche la totalité de leur table de routage toutes les 30 secondes, jusqu'à ce qu'ils se connaissent tous. Ce système est efficace sur un petit réseau. Nous avons choisi de coder la commande du client au serveur sur 16 bits, nous pouvons envoyer toutes les informations nécessaires à la commande effectuée par le client. Il y a un bit pour le type de café, ensuite trois bits pour donner la force du café. Le bit suivant dit si l'utilisateur veut du sucre ou non, si oui la quantité de sucre voulue sera codée sur les 3 prochains bits, ce qui laisse la possibilité de définir 8 quantités possibles, pour le moment nous n'avons défini que 4 niveau de sucre différents car cela pourrai être suffisant mais nous avons la possibilité d'augmenter le nombre de choix sans grande difficulté. Les 4 bits suivants gèrent le niveau de lait de la même façon que pour le sucre, avec un bit de validation et 3 bits pour le niveau. Ensuite un bit pour dire si c'est un café long ou court, et nous avons gardé 3 bits pour une possible extension, comme un type de café différent ou autre chose, mais nous ne voulons pas être bloqué par la suite pour un problème comme cela, il faut mieux voir grand.

Exemple : pour un café capuccino dosage moyen, long avec beaucoup de sucre et sans lait nous aurons cette commande 1010 1011 0000 1000.

Ce code binaire est ensuite lu comme un entier et envoyé au serveur. Le serveur décode l'information et va ensuite séparer chaque partie pour les traiter séparément, puis il va envoyer les informations au PIC pour que celui-ci lance enfin la préparation du café.

2.3.2 Protocole Serveur Machine

Sachant que la transmission USB ne permet pas de savoir exactement le temps où la trame USB sera accessible par la machine, nous avons décidé de gérer toute la temporisation à l'intérieur du PIC selon le protocole suivant : Les moteurs nécessaires pour la commande de l'utilisateur seront gérés un par un, on transmet au pic le numéro du moteur à activer ainsi que la durée pendant laquelle le moteur doit fonctionner. Le serveur envoie donc au pic un code sur 8 bits pour chaque moteur qui sera utilisé, un bit pour la gestion d'erreur, 3 pour la sélection du moteur et ensuite 4 bits pour donner la durée de fonctionnement du moteur.

Exemple : pour activer le moteur du sucre pour une durée courte : 0011 0010
Ce protocole est donc très facile à comprendre et donc à utiliser, cela permet de pouvoir, si nous le voulons, modifier ou rajouter des commandes sans le moindre problème.

2.4 Site

Le site a été créé pour cette soutenance, il est fait en html et en java. Nous avons décidé de le mettre en ligne vu l'importance potentielle du projet (ainsi que pour faciliter la recherche de sponsors, nous en sommes déjà à près de 150 euros...), il est disponible à l'adresse <http://projectc.free.fr>. Nous nous sommes inspiré pour la trame d'un site open source (<http://www.oswd.org/>), et avons modifié et rajouté des fonctions, comme l'heure et la date, ou l'index de la page (par exemple Accueil ↵ Téléchargement). Le site a été rempli en grande partie par des extraits modifiés du cahier des charges, modifiés car à l'époque nous ne connaissions rien en électronique, et ne nous savions pas tout à fait les types de moteurs et d'électrovannes que nous allions utiliser... Ce site a pour but de présenter notre projet au reste du monde, pour ce faire nous l'avons référencé sur les principaux moteurs de recherche (google, yahoo...), cependant le référencement n'étant pas instantané, il va falloir attendre plusieurs semaines avant de le voir apparaître dans les résultats. Nous avons créé une partie Projet contenant l'origine et la nature du projet, une section description du projet, le cahier des charges et un historique contenant toutes les dates clés, depuis la création du groupe à la seconde soutenance. Une partie spéciale soutenance a été créée, elle contient les liens vers les rapports, avec un "Non disponible" pour les rapports 3 et 4. La section Téléchargement contient quand à elle des liens vers les rapports, les programmes tels que le client et le serveur, et les sources des rapports. Une section Page perso a aussi été créée, mais elle sera remplie pour la soutenance finale. Pour finir, une page liens est disponible, elle contient les liens que nous avons utilisés pour nous aider à coder, ainsi que des liens divers et variés que tout le monde apprécie.

Nous avons bien finalisé le site dès sa création, de cette manière pour le mettre à jour nous n'aurons qu'à mettre les rapports de soutenance avec le bon nom et ça sera ok pour le téléchargement. Nous avons aussi rajouté quelques fonctions inutiles mais amusantes, comme le temps de chargement de la page, ou encore des liens pour ajouter la page dans les favoris ou en page de démarrage, ainsi que la date et l'heure.

Ce site fait très professionnel, ce qu'il nous faudra lorsqu'on recherchera des sponsors chez Nestlé par exemple...

Chapitre 3

TODO

Voilà ce que nous comptons réaliser pour la prochaine soutenance qui sera cette fois plus technique.

3.1 Aspect matériel

3.1.1 Structure

Cette soutenance était placée sous le signe de la structure, la prochaine sera sous celui de l'électronique. Côté structure, nous n'avons plus grand chose à faire : il ne nous reste qu'à rajouter la mécanique et l'électronique pour donner un distributeur à café entièrement fonctionnel. Nous n'aurons alors qu'à créer la carrosserie stylisée pour rendre un produit fini et fonctionnel.

3.1.2 Electronique

Pendant cette période inter-soutenance nous avons beaucoup progressé en électronique numérique et analogique. Si nous continuons de cette manière pour la troisième soutenance, nous aurons une partie électronique, si ce n'est pas parfaitement fonctionnelle, du moins dans sa forme définitive.

3.2 Aspect logiciel

La section logicielle a été faite dans sa forme finale pour cette soutenance, le client est définitif, aux codes envoyés au serveur près, et côté serveur il ne reste plus que la partie commande de l'électronique à rajouter dans le thread déjà créé précédemment, ce qui risque de s'avérer le plus difficile à faire.

3.3 Plannings

Pour réaliser ce projet nous avons découpé le planning en plusieurs parties ponctuées par différentes soutenances. Nous avons donc deux plannings : l'un représentant l'avancée globale du projet en fonction des différentes soutenances, un autre présentant la répartition des tâches entre les membres du groupe.

3.3.1 Planning des soutenances

Soutenance	Mécanique-Structure	Electronique-Connectique	Informatique
Soutenance 1	Prototype de la structure avec l'appareillage	Plan du système électronique de contrôle par USB	Interfaces utilisateur Unix et Windows, début du serveur, élaboration du protocole de communication via USB
Soutenance 2	Réalisation de la structure définitive en acier, aluminium, plastique et autre matériaux	Réalisation de la partie électronique et électrique ainsi que du système de contrôle USB, contrôle manuel des différentes parties	Finalisation du système Client Serveur de contrôle interne au PC. Début de l'implémentation du contrôle USB
Soutenance 3	Vérification de l'étanchéité et de l'aération de l'électronique	Vérification et ajout de système de contrôle supplémentaire si besoin est	Finalisation du système de contrôle USB
Soutenance 4 Finale	Création d'une carrosserie stylisée qui camoufle l'appareillage du système un peu à la manière d'un appareil électroménager. Création d'une documentation détaillée de la structure du système	Cartographie détaillée de l'architecture électronique du système	Finalisation totale du système informatique de contrôle et mise au point définitive des réglages

3.3.2 Planning des tâches

Soutenance	Guillaume	Joël	Sylvain
Soutenance 1	Développement de la partie logicielle sous Unix et cartographie du système électronique embarqué	Elaboration du système de contrôle USB (protocoles et pilotes) et création de la structure en matériaux provisoires.	Création d'une interface graphique pour Windows ainsi que du système Client Serveur
Soutenance 2	Réalisation de l'électronique embarquée (partie interface USB) et Début de communication Périphérique-PC via USB (Unix)	Réalisation de l'électronique embarquée (partie contrôle des parties mobiles. Début de communication Peripherique-PC (Windows)	Réalisation de la structure définitive et implémentation des protocoles de communication
Soutenance 3	Contrôle de la résistance de la structure aux contraintes et à la corrosion et Communication Peripherique-PC (Unix)	Mise au point de la transmission (redondance - temps ...) et des dosages et du système Client-Serveur	Finalisation du protocole de communication en liaison avec les mises au point et système Client-Serveur
Soutenance 4 Finale	Debug Partie Unix, développement client(s) pour OS alternatif(s) , Documentation client Unix et OS alternatif(s)	Carrossage de la structure et tests. Documentation de la Structure et de l'électronique	Création des systèmes d'installation et d'aide générale et tests. Communication sur le produit

Le Site Web présentant l'évolution du projet sera créer et mis à jour au fil des soutenances par l'ensemble du groupe.

Conclusion

Cette soutenance a marqué le véritable lancement de la réalisation de notre distributeur à café. Après une soutenance placée sous le signe de la programmation, celle-ci est sous celui de la mécanique et de l'électronique. En effet notre machine a vu ses premières pièces s'assembler, la structure ainsi que les principaux systèmes de fonctionnement ont été établis, et les premiers circuits électroniques ont fait leur apparition. Tout en continuant à avancer ces différentes parties, la prochaine soutenance sera en grande partie basée sur le rassemblement de la programmation, l'électronique et de la mécanique. En effet après avoir fixé les moteurs nous allons maintenant les faire fonctionner, et les gérer coté ordinateur directement à partir de notre logiciel. Notre distributeur prenant peu à peu forme nous nous rapprochons de la préparation de notre premier café.

L'équipe de C (Les Phénoménoïdes) :

- Guillaume GOUSSARD : Chef de projet, Responsable de la partie Unix-OS alternatifs
- Jeël LAFAILLE : Vice Chef de projet, Responsable de la partie matérielle (Electronique - Structure)
- Sylvain PIOT : Sous-chef de projet, Responsable de la partie Windows et logistique

Annexe A

Références

- **man c**
<http://internet.ls-la.net/man-pages/c.html>
Page de manuel Unix de la version GNU de C , un distributeur de boisson.
- **RFC2324**
<http://www.faqs.org/rfcs/rfc2324.html>
RFC traitant de l'Hyper Text Coffee Pot Control Protocol.
- **USB et driver sous linux**
Linux Magazine Hors série 17 page 40
Article décrivant le protocole USB et la création de driver USB sous Linux.
- **PIC16F873/4/6/7 Datasheet**
<http://www.microchip.com>
Documentation du microcontrôleur PIC16F877.
- **FT232BM Application Schematics**
<http://www.ftdichip.com>
Schema d'utilisation du FT232BM.
- **Asservissement d'une electrovanne**
http://perso.wanadoo.fr/olivier.granier/electro/ex_og/asservi/asservi.htm
Schema et explication concernant l'utilisation d'une electrovanne.
- **Le moteur pas à pas et son circuit de commande**
<http://asso.nordned.fr/carl/moteurpas2.htm>
Fonctionnement d'un moteur pas à pas et sa logique de controle.

Annexe B

Schémas préliminaires