

Projet IMA 4 2016/2017

3615 Facebook

Rapport



LOPEZ Tristan
M'CHARFI Marouan

IMA 4 SC
Tuteur : VANTROYS Thomas

Remerciements

Premièrement, nous tenons à remercier l'équipe pédagogique du département Informatique, Microélectronique et Automatique de Polytech Lille qui nous a enseigné les connaissances de base pour mener à bien ce projet de 4^{ème} année.

Nous remercions également notre tuteur de projet, Monsieur Thomas VANTROYS, pour son aide et ses conseils tout au long du projet.

Enfin, nous remercions Monsieur Xavier REDON et Monsieur Alexandre BOE qui ont aussi pu nous aider sur certains points du projet.

Sommaire

Remerciements	1
Introduction.....	3
I – Présentation du projet	4
1. Objectif et cahier des charges	4
2. Choix techniques : matériel et logiciel	4
3. Calendrier prévisionnel et organisation	6
II – Déroulement du projet.....	7
1. Schéma de principe et organisation modulaire du code.....	7
2. Gestion de l’affichage sur le Minitel avec l’Arduino.....	8
2.1 Prise en main du Minitel.....	8
2.2 Fabrication du câble pour connecter le minitel et l’Arduino	9
2.3 Communication série entre le Minitel et l’Arduino	10
2.4 Fonctions d’affichage et de mise en page de l’écran du minitel.....	10
2.5 Création de l’interface graphique du 3615 Facebook	12
3. Création du proxy	14
3.1 Navigation sur Facebook.com en mode texte.....	14
3.2 Installation d’un serveur Apache 2.....	14
3.3 Connection à un compte Facebook depuis le serveur avec les fonctions cURL.....	14
3.4 Filtrage du texte des pages Facebook à afficher sur le minitel	15
4. Programmation de l’ethernet shield	16
4.1 Fonctionnement du module.....	16
4.2 Configuration réseau.....	18
4.3 Fonctionnement en tant que client TCP.....	20
4.4 Programme.....	21
5. Résultats	23
III – Difficultés rencontrées et améliorations possibles	25
1. RAM limitée sur l’Arduino Uno.....	25
2. Améliorations possibles.....	26
Conclusion	27
Bibliographie.....	28

Introduction

Dans le cadre de notre quatrième année de formation à Polytech Lille en Informatique, Microélectronique et Automatique, nous devons réaliser un projet de grande envergure tout au long du semestre 8. C'est l'occasion de mettre en pratique une partie des connaissances acquises tout au long de notre formation à Polytech Lille et plus généralement depuis notre entrée dans l'enseignement supérieur, mais aussi de les améliorer et de développer de nouvelles compétences. Ce projet nous permet aussi d'apprendre la gestion d'un projet sur une période relativement longue, de l'élaboration du cahier des charges au test final du produit réalisé en passant par les différentes étapes de conception.

Nous avons choisi le sujet « 3615 Facebook » dont l'idée générale est de donner une seconde vie aux minitels en les transformant en terminaux d'accès à internet, et plus précisément d'adapter le site internet Facebook.com à l'affichage sur minitel.

Dans ce rapport, nous présenterons tout d'abord de façon détaillée le projet et son cahier des charges, puis nous détaillerons le déroulement du projet et ses différentes parties. Enfin, nous expliquerons quelles ont été les difficultés rencontrées et quelles sont les améliorations possibles.

I – Présentation du projet

1. Objectif et cahier des charges

Le Minitel (Médium interactif par numérisation d'information téléphonique) est un type de terminal informatique qui était destiné à la connexion au service français de Vidéotex baptisé Télétel. Lancé en 1980 en France, il permettait d'accéder à de nombreux services (annuaire, jeux, vente par correspondance, tchat ...) à une époque où peu de foyers possédaient un ordinateur. Il a connu un véritable essor dans les années 1980 et 1990 car internet, son concurrent direct était encore balbutiant. Le service a atteint un pic d'utilisateurs à la fin des années 1990 avec 25 millions d'utilisateurs et 9 millions de terminaux en service avant de décliner progressivement face au développement d'internet dans les années 2000. Le service a été arrêté en 2012, les minitels sont maintenant inutilisables et ne servent donc plus à rien, c'est pourquoi il peut être intéressant de leur donner une seconde vie afin d'éviter le gaspillage.

L'objectif du projet est donc d'utiliser le minitel comme terminal d'accès à internet. Nous avons choisi de nous concentrer sur l'adaptation de Facebook.com qui est un des symboles de l'essor d'internet pour les particuliers. Le défi est d'arriver à créer une interface la plus ergonomique possible sur un terminal avec des capacités d'affichage plutôt limitées et ne comportant pas de souris, mais seulement un petit clavier.

2. Choix techniques : matériel et logiciel

La réalisation de ce projet nécessite peu de matériel : un minitel, une plateforme microcontrôleur avec un contrôleur ethernet et une prise DIN 5 broches. Aucun logiciel particulier n'est nécessaire, la programmation du microcontrôleur sera réalisée en langage C.

- Minitel : **Minitel 1 Bistandard Alcatel Telic**

Les minitels disposent généralement d'une prise série appelée prise péri-informatique permettant de communiquer avec le monde extérieur. En connectant une plateforme microcontrôleur à cette prise série, on peut réaliser l'accès à des pages web et les afficher à l'écran. Il est donc indispensable pour ce projet de choisir un modèle de minitel qui comporte cette prise. Nous avons donc choisi le Minitel 1 Bistandard de marque Alcatel Telic qui, en plus d'être un des modèles les plus répandus, permet un d'afficher 24 lignes de 80 caractères à l'écran au lieu de 24 lignes de 40 caractères comme sur les modèles plus anciens. Cela nous permet donc d'afficher plus d'information à l'écran, et donc une meilleure ergonomie. De plus, ce modèle permet la transmission par voie série jusqu'à 4800

bauds sur sa prise péri-informatique. Nous aurions aussi pu utiliser les minitel 1 Bistandard d'autres marques ou encore le Minitel 2, car leur fonctionnement est identique, ce qui fait que notre projet peut fonctionner sur une grande partie des terminaux minitel.



Figure 1 - Minitel 1 Bistandard Alcatel Telic

- Plateforme microcontrôleur : **Arduino Uno avec Ethernet shield**

Nous avons choisi la plateforme microcontrôleur Arduino Uno qui comporte un microcontrôleur Atmega 328P pour communiquer au minitel via liaison série les caractères à afficher à l'écran et récupérer les caractères tapés au clavier. Les niveaux électriques de la prise péri-informatique du minitel sont compatibles avec le niveau TTL, collecteur ouvert. Par conséquent, le minitel et l'Arduino peuvent communiquer directement via leurs broches RX et TX sans circuit d'adaptation des niveaux de tension. De plus, l'Arduino peut être alimenté directement par le minitel via sa broche de sortie 8,5 Volts – 1 Ampère car la broche d'alimentation Vin de l'Arduino dispose d'un régulateur de tension pouvant supporter jusqu'à 12 Volts. C'est aussi une plateforme que nous avons utilisée dans d'autres projets et TP, et sur laquelle nous avons déjà acquis pas mal de connaissances.

L'Arduino Uno ne dispose pas de contrôleur ethernet intégré, mais cela est compensé par l'ajout de la carte d'extension Ethernet Shield équipée du contrôleur ethernet Wiznet 5100 qui permet de se connecter à internet afin de récupérer les pages de Facebook.com.

- **Connecteur DIN 5 broches (180°)**

Nous avons besoin d'un connecteur DIN 5 broches mâle pour connecter proprement l'Arduino sur la prise péri-informatique du minitel qui est en fait un connecteur DIN 5 broches femelle. Ce type de connecteur n'est plus utilisé sur les appareils électroniques

actuels comme liaison série à cause de sa grande taille et du développement de l'USB mais il est encore possible d'en trouver sur internet ou dans des magasins spécialisés.



Figure 2 - Connecteur DIN 5 broches mâle

3. Calendrier prévisionnel et organisation

Pour le bon déroulement du projet nous avons décomposé notre travail en plusieurs étapes principales, suivant le digramme de gant qui suit.

Semaine	1	2	3	4	5	6	7	8	9	10	11	12
Phase 1												
<i>Etude du matériel</i>												
Phase 2												
<i>Communication avec le minitel</i>												
<i>Gestion de l'affichage</i>												
Phase 3												
<i>programmation de l'Ethernet shield</i>												
Phase 4												
<i>Implémentation du proxy</i>												
Phase 5												
<i>Traitement des pages</i>												
Phase 6												
<i>Réalisation et implémentation finale</i>												

Figure 3 - Diagramme de gant

II – Déroulement du projet

1. Schéma de principe et organisation modulaire du code

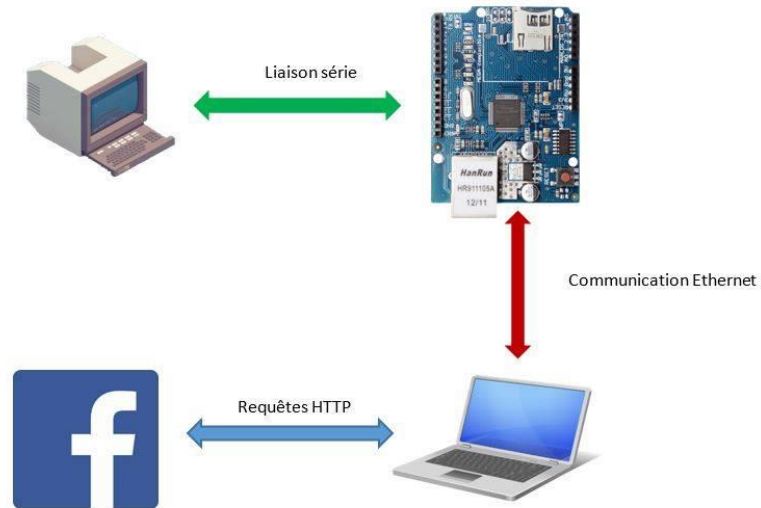


Figure 4 - Schéma de principe du projet

Notre programme arduino en C comporte 4 fichiers .c et les .h correspondant ainsi que le fichier main.c et un Makefile.

- **arduino.c** contient les fonctions de configuration et d'utilisation de la liaison série de l'arduino
- **minitel.c** contient les fonctions de configuration du minitel
- **affichage.c** contient les fonctions gérant l'affichage à l'écran de l'interface graphique du 3615 facebook
- **connexion.c** contient les fonctions relatives à la configuration et à l'utilisation de l'éthernet shield
- **main.c** contient la fonction main gérant l'exécution du programme d'affichage

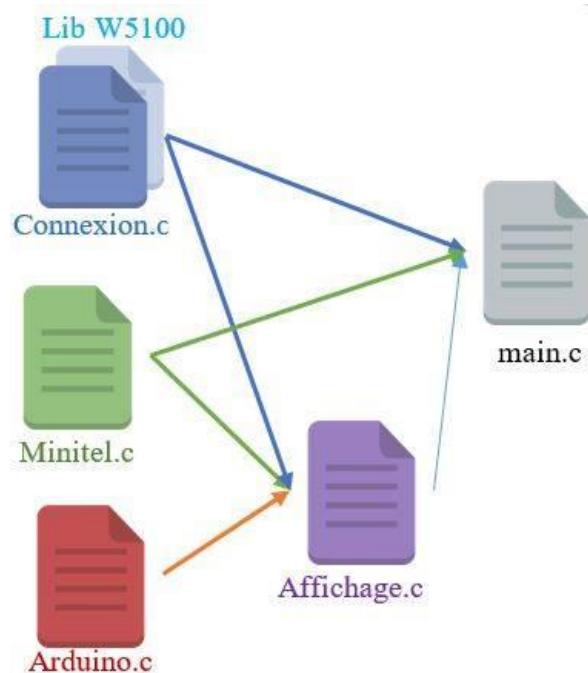


Figure 5 - Organisation modulaire du code

2. Gestion de l'affichage sur le Minitel avec l'Arduino

2.1 Prise en main du Minitel

Le Minitel étant un appareil plutôt ancien (plus de 30 ans) et que nous ne connaissons pas très bien du point de vue technique, nous avons dû consacrer les premières semaines du projet à la documentation et aux recherches sur internet. Grâce à des documents techniques de l'époque de l'appareil destinés aux professionnels trouvés sur internet nous avons pu nous renseigner de façon détaillée sur le fonctionnement de celui-ci.

Le minitel que nous avons à notre disposition est bistandard, c'est à dire qu'il dispose de deux « standards » de fonctionnement :

- Un standard Télétel comprenant deux modes :
 - Un mode Vidéotex qui est le mode actif à la mise sous tension du minitel
 - Un mode mixte
- Un standard téléinformatique

Nous allons utiliser le mode mixte car il permet d'utiliser l'écran en mode 80 colonnes de caractères (au lieu de 40 en mode Vidéotex) et des échanges bidirectionnels simultanés quand le minitel est relié à un autre appareil via sa prise péri-informatique à la vitesse de 4800 bauds maximum. Des commandes dites « protocole » tapées au clavier ou envoyés directement sur la broche RX du terminal via des codes de plusieurs octets successifs permettent de configurer le standard d'utilisation du minitel, la vitesse de transmission série et d'autres fonctions de configuration du terminal. Les fonctions utiles pour notre projet ainsi que les commandes correspondantes sont données en figure 6 ci-dessous. Le Minitel ne dispose pas de mémoire intégrée, à l'extinction de l'appareil les paramètres sont perdus, la configuration par défaut : standard Télétel, mode Vidéotex, 1200 bauds est active à chaque allumage. Nous devons donc envoyer les codes permettant le passage en mode mixte et la modification de la vitesse de la prise à 4800 bauds à chaque fois que l'on allume le minitel avec l'Arduino.

Tâche	Commande clavier	Commande par la prise (Hexadécimal)
Passage au standard téléinformatique	Fnct + T puis F	1B 3A 31 7D
Passage du mode Vidéotex au mode mixte	Indisponible	1B 3A 32 7D
Passage de l'écran en mode page/rouleau	Fnct + E puis P / Fnct + E puis R	1B 3A 6A 43 / 1B 3A 69 43
Modifier la vitesse de la prise (1200 bauds par défaut)	Fnct + P puis 3 pour 300 bauds Fnct + P puis 1 pour 1200 bauds Fnct + P puis 4 pour 4800 bauds	1B 3A 6B 52 1B 3A 6B 64 1B 3A 6B 76

Figure 6 - Commandes de configuration du Minitel

2.2 Fabrication du câble pour connecter le minitel et l'Arduino

Nous avons fabriqué le câble série pour connecter le Minitel à l'Arduino à l'aide du connecteur DIN 5 broches que nous avons acheté et d'un vieux câble ethernet. Nous avons soudé 5 fils du câble au broches du connecteur et soudé des pins à l'autre extrémité du câble pour pouvoir connecter facilement les connecter facilement à l'Arduino. Le schéma des contacts de la prise du minitel, la photo de soudure des fils sur les broches de la prise ainsi que la correspondance fils/broches sont données en figures 7, 8 et 9 ci-dessous. L'entrée « PT-périphérique prêt à travailler » ne nous sera pas utile pour le projet.

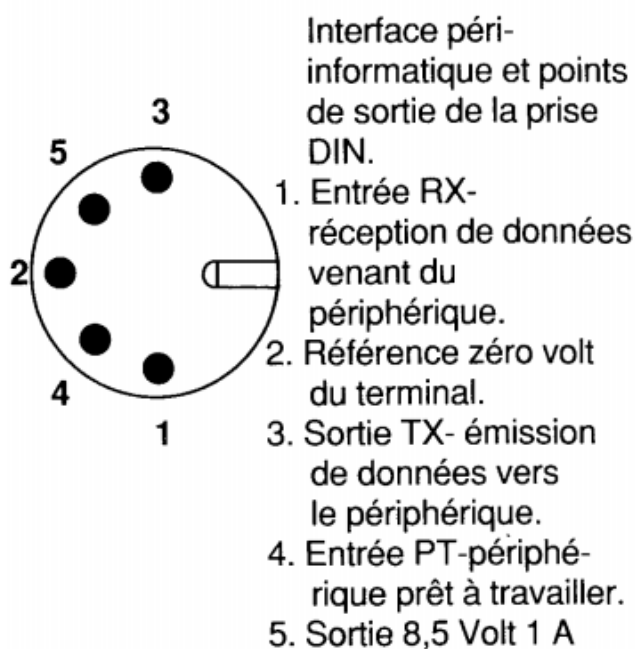


Figure 7 - Broches de la prise péri-informatique du Minitel

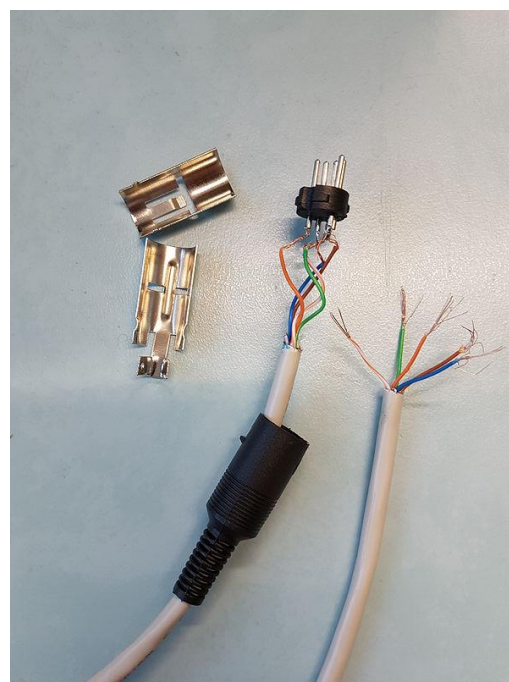


Figure 8 - Soudage des fils sur le connecteur DIN

Numéro de la broche	Fonction	Couleur du fil
1	Entrée RX	Bleu
2	0V	Vert
3	Sortie TX	Orange
4	Entrée PT	Marron
5	Sortie 8,5V 1A	Blanc

Figure 9 - Correspondance fils/broches

2.3 Communication série entre le Minitel et l'Arduino

Ayant réalisé le câble permettant de connecter la prise péri-informatique du minitel à l'Arduino, nous avons essayé d'envoyer des caractères au minitel via la liaison série et de les visualiser à l'écran correctement. En effet, lorsque que le minitel reçoit un caractère, il est par défaut directement affiché à l'écran.

D'après le manuel "Minitel 1 B Spécifications techniques d'utilisation", les niveaux électriques de la prise du minitel sont compatibles avec le niveau TTL, collecteur ouvert :

- Un niveau de tension supérieur ou égal à 2,5 V présenté sur l'entrée RX sera interprété comme un "1" logique
- Un niveau de tension inférieur ou égal à 0,4 V présenté sur l'entrée RX sera interprété comme un "0" logique.

Le format des signaux RX et TX de réception et d'émission du minitel est fixe avec 7 bits de données plus un bit de parité paire.

Nous avons donc codé en C un programme envoyant en continu le caractère « a » sur le port série afin de s'assurer que le minitel reçoit et affiche les caractères. On utilise la bibliothèque `avr/io.h` de façon à écrire dans les registres de l'Atmega 328 P de l'Arduino. Les valeurs des registres à programmer sont données dans la datasheet de l'Atmega 328 P.

2.4 Fonctions d'affichage et de mise en page de l'écran du minitel

Le minitel permet de choisir différents attributs de visualisation pour les caractères reçus à l'aide de codes envoyés via la liaison série. Dans le mode mixte sur lequel nous travaillons, peu d'attributs de visualisation des caractères sont disponibles, mais cela est suffisant pour rendre l'affichage en niveaux de gris plutôt agréable. Pour appliquer un attribut de visualisation à une suite de caractères, on envoie d'abord le code correspondant à l'attribut puis la suite de caractères. Il faut renvoyer le code correspondant à « aucun attribut » ou le code qui met fin à l'attribut choisi pour revenir à l'affichage par défaut. La fonction « `void attribut(char a, int i)` » permet selon le caractère et la valeur entière passés en argument d'envoyer au Minitel le code correspondant à l'attribut de visualisation que l'on souhaite utiliser. Les attributs de visualisation disponibles, leur code et l'appel à la fonction attribut correspondant sont donnés en figure 10 ci-dessous.

Attribut	Code (Hexadécimal)	Appel fonction
Aucun	1B 5B 30 30 6D	attribut('a',1) ;
Surintensité / Intensité normale	1B 5B 30 31 6D / 1B 5B 32 32 6D	attribut('i',1) ; / attribut('i',0) ;
Souligné / Non souligné	1B 5B 30 34 6D / 1B 5B 32 34 6D	attribut('s',1) ; / attribut('s',0) ;
Clignotant / Non clignotant	1B 5B 30 35 6D / 1B 5B 32 35 6D	attribut('c',1) ; /attribut('c',0) ;
Fond inversé / Fond normal	1B 5B 30 37 6D / 1B 5B 32 37 6D	attribut('f',1) ; / attribut('f',0) ;

Figure 10 - Attributs de visualisation des caractères du Minitel

Le Minitel affiche les caractères les uns à la suite des autres à l'écran sur la position du curseur. Celui-ci se déplace à chaque caractère reçu. Quand le curseur arrive en fin de ligne, il faut envoyer les caractères « saut de ligne » (\n ou 0x0A) et « retour chariot » (\r ou 0x0D) pour positionner le curseur sur la première case de la ligne suivante. Le fonction « void entree(int n) » dans le fichier minitel.c permet de sauter n lignes et de positionner le curseur en début de ligne. Cependant, pour pouvoir placer du texte à des endroits précis de l'écran, et ainsi soigner la mise en page, il est nécessaire du pouvoir déplacer le curseur à tout moment sur l'écran. On utilise pour cela les commandes relatives au déplacement du curseur qui sont envoyés comme pour les attributs de visualisation à l'aide de codes via liaison série. Il est aussi parfois nécessaire d'effacer le texte à l'écran avant de réafficher une nouvelle page ou de bloquer l'affichage à l'écran pour quand l'utilisateur tape un mot de passe par exemple. Les commandes de mise en page, leur code et les appels fonction correspondants sont donnés en figure 11 ci-dessous.

Fonction	Code (Hexadécimal)	Appel fonction
Déplacement du curseur de n colonnes vers la gauche	08 (à envoyer n fois)	curseur_gauche(n) ;
Déplacement du curseur de 8 fois n colonnes vers la droite	09 (à envoyer n fois)	curseur_droite(n) ;
Déplacement du curseur de n lignes vers le bas et retour du curseur en position 1 de la nième ligne	0A 0D (à envoyer n fois)	entree(n) ;
Effacer l'écran et placer le curseur en première ligne, première colonne	1B 5B 32 4A 1B 5B 00 3B 00 48	effacer_ecran() ;

Placer le curseur ligne l, colonne c	1B 5B 00 3B l c	curseur_position(l, c) ;
Bloquer l'affichage sur l'écran / Débloquer	1B 3B 60 58 50 / 1B 3B 61 58 50	ecran(1); / ecran(0) ;

Figure 11 - Fonctions de mise en page du Minitel

2.5 Création de l'interface graphique du 3615 Facebook

Le minitel ne comportant ni souris, ni écran tactile, il est nécessaire de créer une interface qui se commande uniquement au clavier. L'ergonomie est donc un critère capital sur un terminal de ce type ayant, de plus, un affichage plutôt limité. Pour concevoir l'interface, nous nous sommes inspirés des pages minitel de l'époque (voir figure 12 ci-dessous) qui utilisaient les touches de fonction du clavier (Sommaire, Annulation, Retour, Guide, Envoi ...) pour naviguer dans les différents menus.

Nous avons fait les choix suivants pour l'interface :

- Une touche du clavier sera représentée à l'écran en noir sur fond blanc à côté du nom du menu correspondant.
- Une barre de menu sera en permanence affichée en haut de l'écran avec le nom des pages dont on peut demander l'affichage ainsi que la touche de fonction du clavier correspondante.
- Les champs seront représentés par des "underscore" (Recherche : _____).
- La touche « correction » du minitel est utilisée pour effacer le dernier caractère tapé lorsque l'utilisateur remplit un champ.
- La touche entrée permet de valider une fois le champ rempli.

Le prototype de la page « fil d'actualité » qui est la page qui apparaît après que l'utilisateur se soit connecté est donné en figure 13 ci-dessous. L'ensemble des menus, touches et pages affichables à l'écran sont codés dans des fonctions du fichier affichage.c. Le fichier main.c fait appel à la fonction d'affichage de chacune des pages en fonction de la touche pressée par l'utilisateur.



Figure 12 - Page minitel de l'annuaire électronique

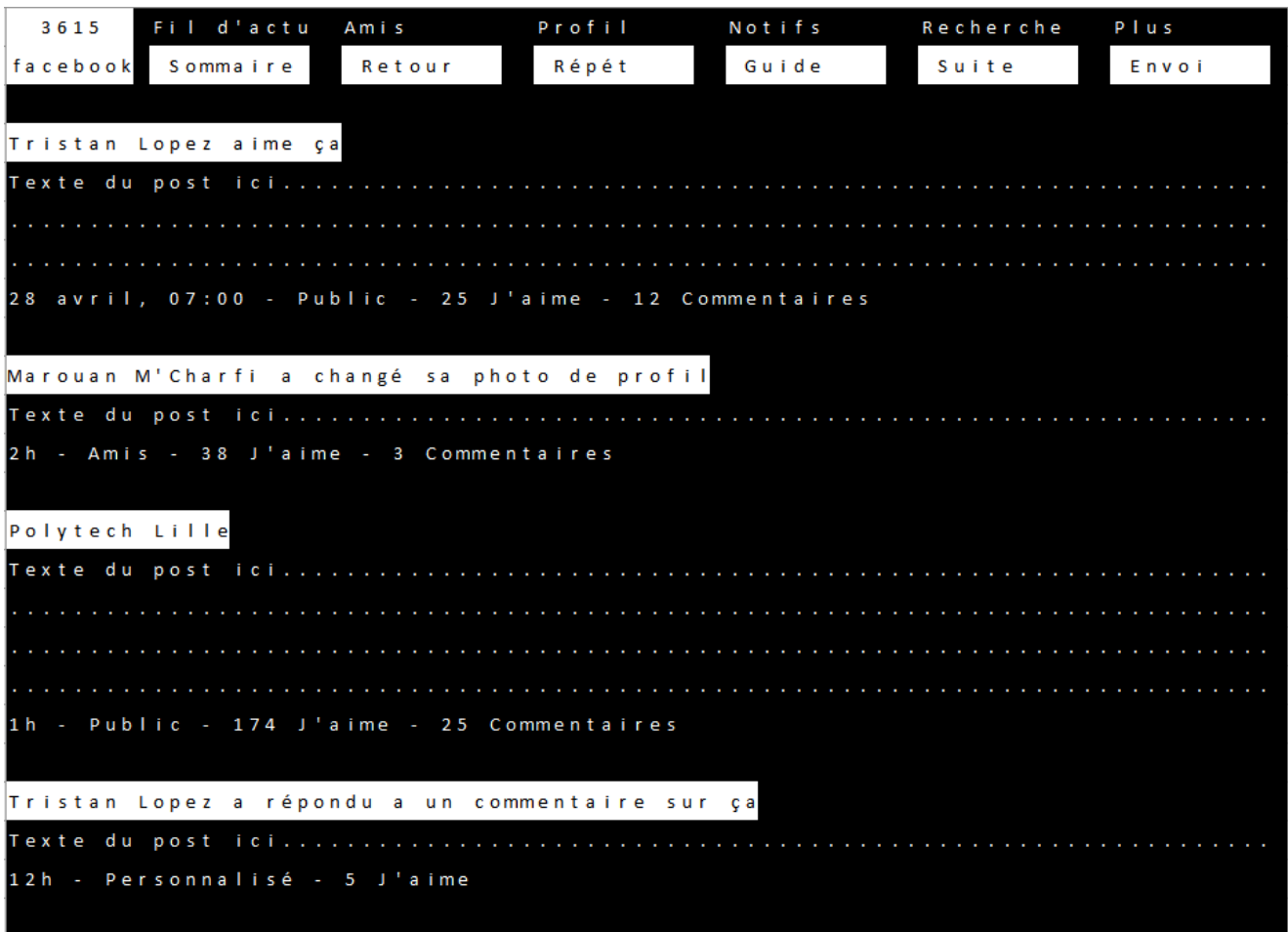


Figure 13 - Prototype de la page "fil d'actualité de Facebook"

3. Création du proxy

L'Arduino Uno ne comportant que très peu de mémoire vive (2 ko), il est difficile de se connecter, de récupérer et de stocker directement les pages Facebook avec l'ethernet shield installé sur celui-ci. En effet, les pages web de Facebook comportent beaucoup trop de caractères dont la grande majorité sert à l'esthétique du site. Par exemple, la page de connexion pour appareils mobiles <https://m.facebook.com/> comporte plus de 20 000 caractères, et cela est pire sur la version pour ordinateur qui comporte en plus du JavaScript. Nous n'avons de plus besoin que du texte brut des pages pour afficher sur notre minitel, d'où la nécessité de le filtrer.

3.1 Navigation sur Facebook.com en mode texte

Pour réfléchir à comment adapter les pages Facebook sur Minitel, nous avons installé le navigateur web Links sur Linux qui est un navigateur n'affichant que le texte des pages web sur la console. La version pour ordinateur de Facebook nécessite JavaScript, elle ne peut donc pas fonctionner correctement sur ce type de navigateur. On utilise par conséquent la version mobile de Facebook (<https://m.facebook.com>) plus simple et donc plus adaptée pour un affichage rudimentaire.

3.2 Installation d'un serveur Apache 2

Nous avons installé Apache 2 pour héberger les pages de notre proxy. Nous avons aussi installé PHP 5.6 ainsi que son extension cURL qui permet de récupérer le texte des pages web et de le stocker dans une chaîne de caractères.

3.3 Connexion à un compte Facebook depuis le serveur avec les fonctions cURL

Pour pouvoir entrer l'adresse e-mail ainsi que le mot de passe de connexion à un compte Facebook directement depuis le minitel, il faut trouver une méthode permettant de remplir automatiquement les champs correspondants du formulaire PHP de la page "connexion" de Facebook. Nous avons tout d'abord essayé de nous connecter en créant une page avec un formulaire comportant les champs e-mail et mot de passe et qui envoie ces données à la page de connexion à facebook via la méthode POST. Cependant, Facebook détecte les tentatives de connexion venant d'une page extérieure au site et empêche la connexion. Nous avons donc dû trouver un moyen de contourner la sécurité de facebook. Pour cela, nous avons utilisé les fonctions cURL de PHP.

3.4 Filtrage du texte des pages Facebook à afficher sur le minitel

Une fois qu'on a récupéré la page d'accueil de facebook connectée sur un compte avec cURL dans une variable `$p`, on découpe le texte utile de la page avec les fonctions de traitement des chaînes de caractères de PHP. Pour cela, on commence par repérer les balises `<p></p>`, `<a>`, ou encore les balises `<h3</h3>`, `<h5</h5>` qui contiennent du texte. On utilise la commande `strip_tags` qui permet de supprimer des balises html.

```
$p=strip_tags($p, '<a>'); supprime toutes les balises html sauf <a</a>
```

Ainsi, cela permet de se débarrasser des nombreuses balises qui ne contiennent pas de texte à afficher telles que `<div>`, ``... On se sert ensuite des balises restantes pour placer des caractères de séparation qui permettent de délimiter les différentes lignes de texte, les différents posts... On utilise la commande `strtr` qui permet de remplacer toutes les occurrences d'une chaîne de caractères dans la page.

```
$trans=array("<p>" => "<p>|");  
$p=strtr($p, $trans); ici on remplace <p> par <p>| ce qui revient à ajouter le caractère | après chaque balise <p>
```

Une fois qu'on a placé tous les caractères de séparation dans le texte, on enlève toutes les balises qui restent dans le texte :

```
$p=strip_tags($p);
```

Ainsi, il ne reste que les caractères de séparation et le texte des balises. Les caractères de séparation sont interprétés par le minitel de différentes manières et ne seront pas affichés sur son écran. Ils permettent de gérer la mise en page du texte récupéré de façon à l'aérer et qu'il ne soit pas en un seul bloc. Les caractères de séparation introduits dans le texte récupéré par le proxy et leur signification pour le minitel sont donnés dans le tableau ci-dessous :

Caractère de séparation	Signification pour le minitel
@	Début de texte
	Retour à la ligne
%	Saut de ligne (double retour à la ligne)
{	Inversion du fond (texte noir sur fond blanc) jusqu'au prochain retour à la ligne
}	Fin de texte

Figure 14 - Caractères de séparation

On utilise aussi d'autres fonctions telles que `strpos` qui permet de trouver la position de la première occurrence d'une chaîne de caractères dans un texte ou `strstr` qui fait de même mais retourne le texte à partir de la première occurrence de la chaîne au lieu de retourner sa position. `substr` retourne un segment de chaîne et `substr_replace` remplace un segment dans une chaîne. Enfin `echo` affiche une chaîne de caractères.

`$p=strpos($p,"Accueil");` retourne la position de la première occurrence de la chaîne "Accueil" dans \$p

`$p=strstr($p,"Accueil");` retourne le texte à partir de la première occurrence de la chaîne "Accueil" dans \$p

`$p=substr($p,5,25);` retourne la chaîne \$p entre les caractères 5 et 25

`$p=substr_replace($p,"@",5);` remplace le caractère 5 par '@' dans \$p

`echo $p;` affiche la chaîne \$p

4. Programmation de l'éthernet shield

4.1 Fonctionnement du module

L'extension Arduino ethernet shield permet à notre application d'accéder à un réseau. Elle interface une prise RJ45 et un circuit intégré Wiznet W5100 qui permet la gestion de paquets IP (TCP/UDP).

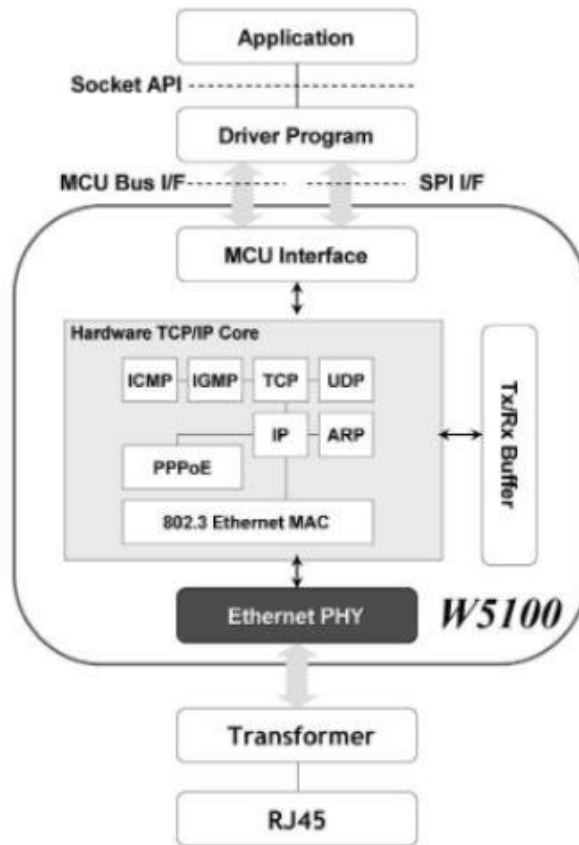


Figure 15 – Schéma du W5100

La puce Wiznet W5100 communique avec notre microcontrôleur via un bus SPI. Il se comporte comme un esclave, le maître vient lire et écrire dans les différents registres de l'esclave.

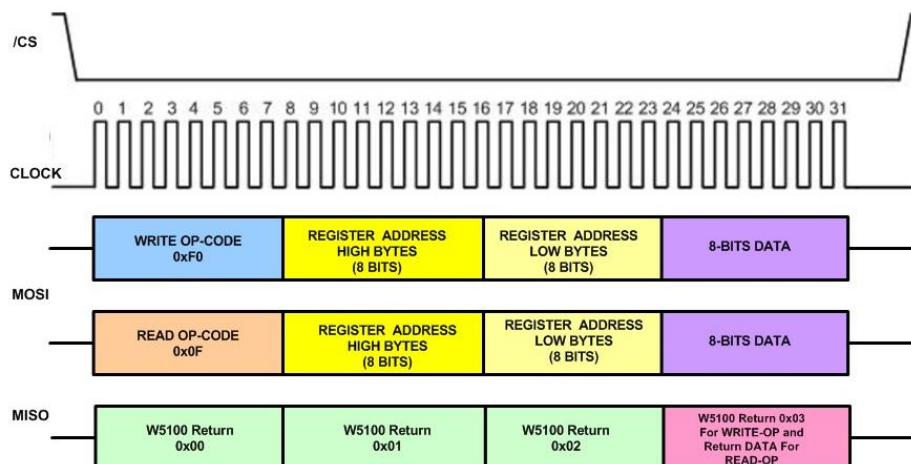


Figure 16 – Communication SPI du Wiznet W5100

Pour pouvoir configurer une connexion, nous devons initialiser le W5100. Pour cela il nous faut écrire dans les registres suivants :

- **MR** (Mode Register) : permet de placer le W5100 dans un mode choisi.
- **SUBR** (Subnet mask Register) : permet de lui assigner un masque de sous réseau.
- **SAR** (Source Hardware Register) : permet de lui assigner une adresse MAC.
- **SIPR** (Source IP Register) : permet de lui assigner une adresse IP.
- **RMSR** (Receive Memory Size Register) et **TMSR** (Transmit Memory Size Register) : permettent d'assigner une taille de buffer de réception et d'émission au W5100.

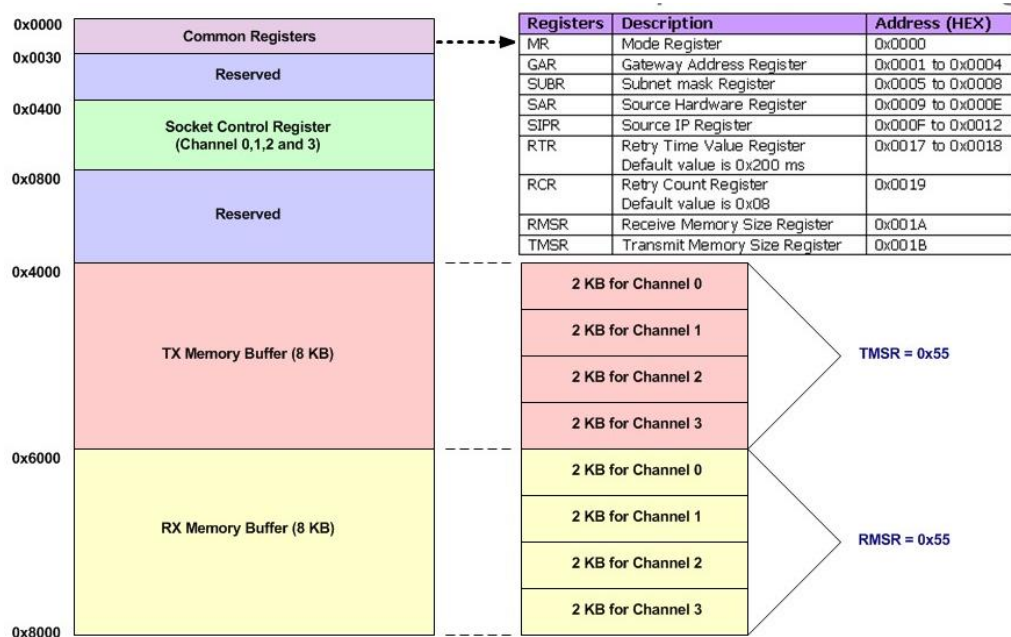


Figure 17 - Plan mémoire du W5100

Pour récupérer les différentes données reçues mais aussi en envoyer, notre microcontrôleur lit et écrit aux adresses spécifiées par TX Memory Buffer et RX memory Buffer (figure 17). Le Wiznet W5100 peut effectuer quatre communications simultanées en utilisant quatre sockets (figure 17).

4.2 Configuration réseau

Comme décrit dans la partie 1, l'ethernet shield communique en ethernet avec notre serveur Apache qui accède à Facebook. Pour que la communication soit possible, il faut que l'Arduino soit connecté sur le même réseau que notre serveur.

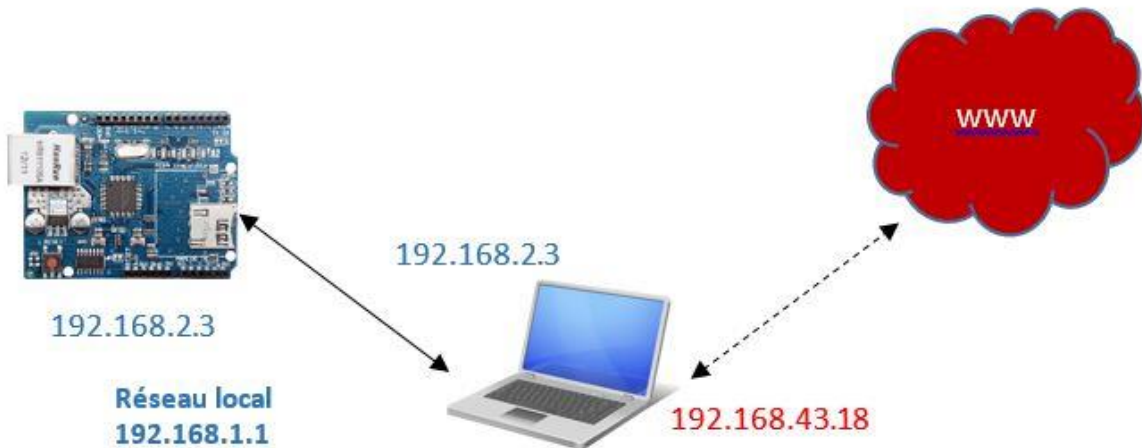


Figure 18 - schéma connexion

C'est pourquoi nous avons tout d'abord configuré une interface ethernet sur la machine hébergeant le serveur.

```

eno1    Link encap:Ethernet  HWaddr 10:60:4b:47:e5:e9
        inet adr:192.168.2.3  Bcast:192.168.2.255  Masque:255.255.255.0
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1000
        Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)

lo      Link encap:Boucle locale
        inet adr:127.0.0.1  Masque:255.0.0.0
        adr inet6: ::1/128  Scope:Hôte
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        Packets reçus:1404 erreurs:0 :0 overruns:0 frame:0
        TX packets:1404 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1
        Octets reçus:106207 (106.2 KB) Octets transmis:106207 (106.2 KB)

vlo1   Link encap:Ethernet  HWaddr 20:68:9d:3d:bd:51
        inet adr:192.168.43.18  Bcast:192.168.43.255  Masque:255.255.255.0
        adr inet6: fe80::a605:772f:19ad:7dce/64  Scope:Lien
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        Packets reçus:28 erreurs:0 :0 overruns:0 frame:0
        TX packets:122 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1000
        Octets reçus:4104 (4.1 KB) Octets transmis:18368 (18.3 KB)

```

Figure 19 - Interface créée

Il faut aussi que les paquets envoyés sur notre interface du réseau local soient transmis sur l'interface connectée à internet. On peut observer la configuration du routage dans la figure ci-dessous.

Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
0.0.0.0	192.168.43.1	0.0.0.0	UG	600	0	0	wlo1
169.254.0.0	0.0.0.0	255.255.0.0	U	1000	0	0	wlo1
192.168.2.0	0.0.0.0	255.255.255.0	U	0	0	0	eno1
192.168.43.0	0.0.0.0	255.255.255.0	U	600	0	0	wlo1

Figure 20 - Table de routage

Ensuite nous assignons une adresse IP à notre Arduino se trouvant dans la plage du réseau local de notre interface.

4.3 Fonctionnement en tant que client TCP

L'ethernet shield fonctionne en tant que client TCP. Il va tout d'abord initier une demande de connexion au serveur. Une fois la connexion établie, le client et le serveur vont pouvoir communiquer.

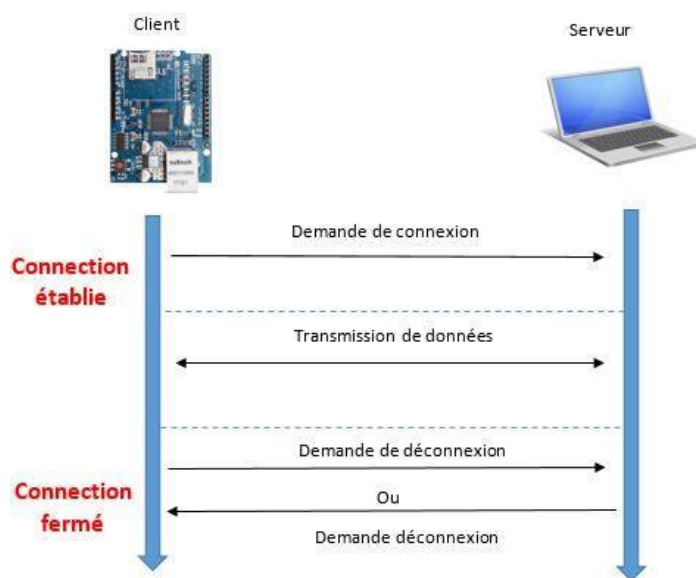


Figure 21 – Communication client / serveur

Notre application envoie des requêtes en utilisant le protocole http. Nous utilisons principalement la méthode GET qui permet de récupérer les différentes pages de notre proxy. Une requête http possède la forme suivante :

```
GET /nom_page.php HTTP/1.1\nHost :192.168.2.3\n\n
```

Exemple de requête envoyée :

```
GET /recherche.php ?email=mon_email@gmail.com&password=*****&rech=nom
+prénom HTTP/1.1\nHost :192.168.2.3\n\n
```

4.4 Programme

Pour programmer la partie réseau nous avons utilisés différentes bibliothèques gérant la configuration du Wiznet W5100 au travers de différentes fonctions :

- W5100.h : décrit les différentes valeurs de registres du W5100 et les fonctions associées.
- Ethernet.h : déclare les paramètres et contient les fonctions d'initialisation de l'éthernet shield.
- Socket.h : contient les différentes fonctions relatives à la gestion des sockets.
- Spi.h : déclare les différentes valeurs relatives à la communication SPI.

4.4.1 Fonction d'initialisation

Cette fonction permet d'initialiser la plateforme Ethernet et retourne un socket `s`. Tout d'abord elle assigne au Wiznet une adresse IP, adresse mac, la passerelle, et le masque de sous réseau via la fonction `ethernet_init()` (`ethernet.h`). Ensuite elle initialise un socket via la fonction `socket()`, en lui passant en paramètres le socket à initialiser, le protocole de communication souhaité : ici `Sn_MR_TCP` pour une communication TCP, le port utilisé, et le mode. Elle teste si le socket est correctement initialisé et retourne ce dernier si c'est le cas.

```
SOCKET init_connexion(){
    uint8_t mac[MAC_SIZE]={0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
    uint8_t ip[IPV4_SIZE]={192,168,2,9}; //Ip arduino 192.168.2.9
    uint8_t gateway[IPV4_SIZE]={192, 168, 2, 1}; //Ip passerelle: 192.168.0.1
    uint8_t mask[IPV4_SIZE]={255, 255, 255, 0}; //Masque: 255.255.255.0

    ethernet_init(mac,ip,gateway,mask);

    //on initialise un socket
    SOCKET s=0;
    if(socket(s, Sn_MR_TCP,PORT_SRC ,0)==0){
        return -1;
    } else return s;
}
```

Figure 22 - Fonction `init_connexion`

4.4.2 Connexion au serveur

Cette fonction est utilisée afin de se connecter au serveur, elle prend en paramètre un socket et renvoie 0 si la connexion est établie et 1 sinon.

```

int serveur_connexion(SOCKET s){
    uint8_t server[IPV4_SIZE]={192, 168, 2, 3}; //Ip server: A choisir
    uint16_t port=PORT_HTTP;

    int i=0;

    connect(s, server, port);

    while(IINCHIP_READ(Sn_SR(s))!=SOCK_ESTABLISHED){
        i++;
        if (i==1000){
            if(IINCHIP_READ(Sn_SR(s))!=SOCK_ESTABLISHED){
                return 1;
            }
        }
    }
    return 0;
}

```

Figure 23 - Fonction serveur_connexion

4.4.3 Envoi de requête

Cette fonction permet l'envoi et la réception de paquets http. Elle prend en paramètre les buffers d'envoi et de réception et le socket de dialogue. Elle retourne la taille des paquets reçus.

```

int envoi_requete(char * requettehttp, SOCKET s, char * reponsehttp){
    memset(reponsehttp, '\0', RXBUF_SIZE);
    if(send(s, (unsigned char*)requettehttp, strlen((char*)requettehttp))){

        while (getSn_RX_RSR(s)==0);

        unsigned int recvSize = recv(s, (unsigned char*)reponsehttp, RXBUF_SIZE-1);

        return recvSize;
    }
    else return 0;
}

```

Figure 24 - Fonction envoi_requete

4.4.4 Test de la connexion

Cette fonction permet de vérifier si la connexion est active ou non, elle prend en paramètre le socket de dialogue et retourne 0 lorsque la connexion est fermée.

```

int connection_active(SOCKET s){
    if((IINCHIP_READ(Sn_SR(s))!=SOCK_ESTABLISHED) && (IINCHIP_READ(Sn_SR(s))==SOCK_CLOSED)){
        return 0;
    }
    else return 1;
}

```


Figure 25 - Fonction connection_active

5. Résultats

Afin de montrer les étapes d'adaptation des pages Facebook sur Minitel, nous vous proposons ci-dessous des captures d'écran d'une page Facebook sur un navigateur, puis sa version adaptée sur le serveur Apache et enfin la page affichée sur le Minitel.

Bacem Hagui a commenté ça.


Fal Fs, 😄 amusé, à **Bordeaux**.

 Ajouter

C'est parti pour 4 mois !

 **Bordeaux**
Ville

Hier, à 14:11 · Bordeaux, Aquitaine · Public
👍❤️ 36 · J'aime · Réagir · 5 commentaires · Partager · Actualité intégrale · Plus

 **Bacem Hagui** Nice! Je passerai te voir alors 😄

Votre commentaire...

Cheikh Soilih Said Ahmed aime **The Erasmus Bible**.

 **The Erasmus Bible**
The Erasmus Bible is the new web resource for Erasmus students.
Find:
+ City Guides
+ Accommodation...

Hier, à 11:33 · Public
J'aime · Réagir · Commenter · J'aime cette Page · Actualité intégrale · Plus

Connaissez-vous...


 **Alexis Dorian**
14 amis en commun

Figure 26 - Une page Facebook

/minitel.polytech.5@%{Bacem Hagui a commenté ça.|Fal Fs, amusé, à Bordeaux.Ajouter|C'est parti pour 4 mois !|Bordeaux|Ville|Hier, à 14:11 - Bordeaux, Aquitaine - Public - 36 J'aime - 5 commentaires Bacem Hagui Nice! Je passerai te voir alors :p%{Cheikh Soilih Said Ahmed aime The Erasmus Bible.|The Erasmus Bible|The Erasmus Bible is the new web resource for Erasmus students. Find: + City Guides + Accommodation...|Hier, à 11:33 - Public - J'aime %}{Connaissez-vous...|Alexis Dorian|14 amis en communAjouterSupprimer|Mehdi Bekhadda|8 amis en communAjouterSupprimer|Thomas Drelonfle Stievenard|12 amis en communAjouterSupprimerAfficher plus|Cheikh Soilih Said Ahmed a réagi à ça.|Qu'en pensez-vous Arsène?|#Oui.|1 h - Public - 494 J'aime - 26 commentaires %}{Cheikh Soilih Said Ahmed aime ça.|En tout cas on te le souhaite|c'est un pote Steve|4 h - Public - 1 519 J'aime - 303 commentaires %}{Cheikh Soilih Said Ahmed aime ça.|So Foot|La coupe d'Europe sur SFR, la lLiga sur beIN, la L1 sur Canal, la Coupe de France sur France 3 Régions... Putain, bientôt il faudra une télécommande pour chaque match de Ligue 2 !|Tous les matchs de Ligue des champions et de Ligue Europa seront sur SFR Sport !sofoot.com|5 h - Public - 728 J'aime - 150 commentaires %}{Minitel Polytech|Pret pour la soutenance !!|4 h - Confidentialité : Amis|6 J'aime - Ajouter des photos %}{Connaissez-vous...|Cedric Montuelle|5 amis en communAjouterSupprimer|Tom Vnt5 amis en communAjouterSupprimerAfficher plus|Cheikh Soilih Said Ahmed aime ça.|Un type qui cogite|Hier, à 15:00 - Public - 772 J'aime - 83 commentaires }

Figure 27 - La même page adaptée sur le proxy

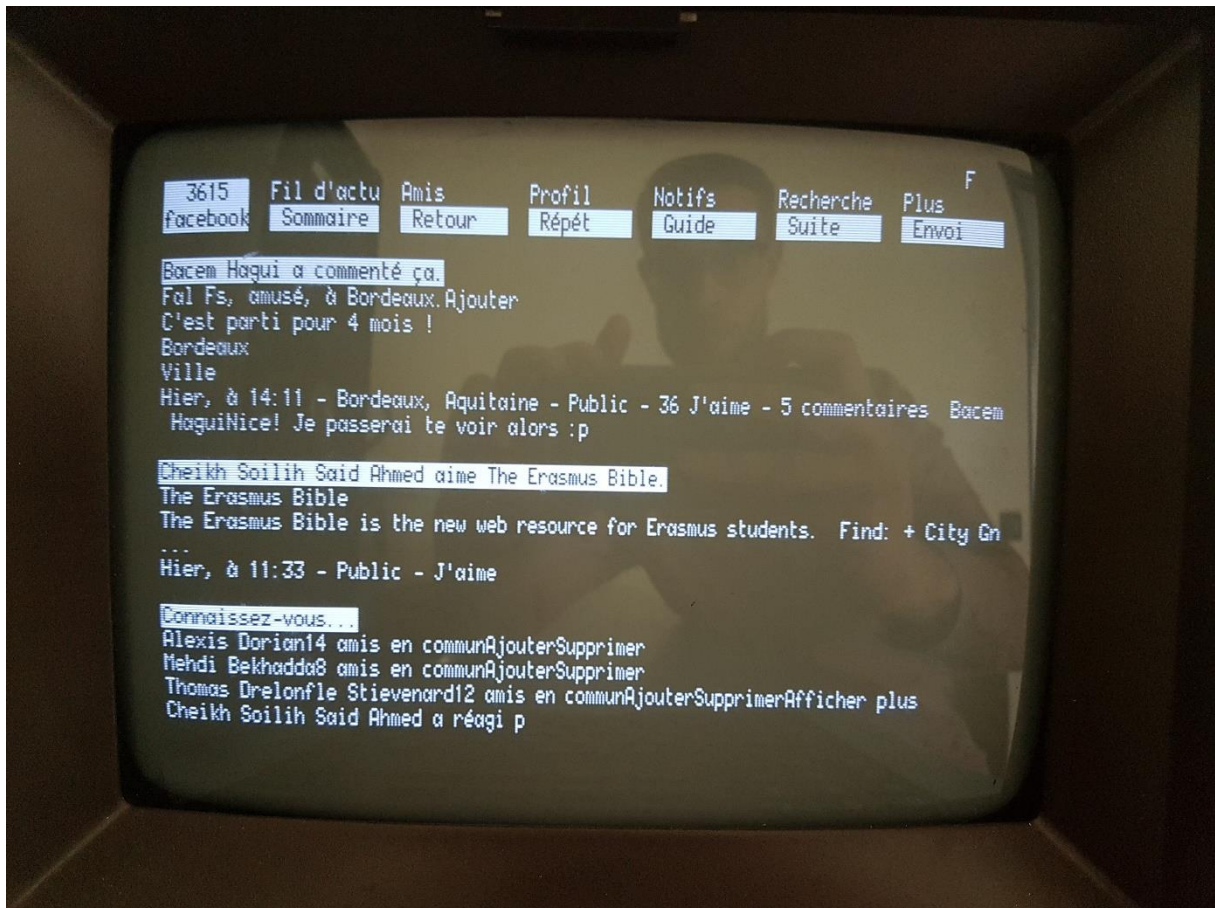


Figure 28 - Le résultat final sur l'écran du Minitel

III – Difficultés rencontrées et améliorations possibles

1. RAM limitée sur l'Arduino Uno

Lors du codage de l'application sur l'Arduino, nous avons constaté un problème avec les chaînes de caractères. En effet, le programme compile correctement, mais les dernières chaînes de caractères de notre programme ne s'affichent pas correctement sur le minitel. Cela est dû au fait que les chaînes de caractères sont stockées dans la mémoire SRAM de l'Arduino et que cette mémoire ne peut contenir que 2048 octets. Nous avons cependant besoin de stocker beaucoup de chaînes de caractères dans l'Arduino. Une solution existe à ce problème, il est possible de stocker les chaînes de caractères dans la mémoire flash de l'Arduino (mémoire de programme) qui peut contenir 32 Ko.

Pour stocker une chaîne de caractères dans la mémoire flash, on utilise l'attribut `PROGMEM` de la bibliothèque `avr/pgmspace.h` :

```
static const char chaine[] PROGMEM = "Chaine de caracteres";
```

Pour lire les chaînes de caractères stockées dans la mémoire flash et les envoyer sur le port série pour l'affichage sur l'écran du minitel, on utilise la fonction suivante :

```
void afficher_flash(const char chaine[])  
{  
    int k;  
    char myChar;  
    int len = strlen_P(chaine);  
    for (k = 0; k < len; k++)  
    {  
        myChar = pgm_read_byte_near(chaine + k);  
        send_serial(myChar);  
    }  
}
```

Cependant, seules les chaînes de caractères statiques telles que les titres, nom de touches et textes des différents menus peuvent être stockés dans la mémoire flash qui est une mémoire de programme en lecture seule. Les chaînes de caractères qui stockent l'e-mail, le mot de passe, les différentes requêtes http ou encore le texte des pages récupérées par l'éthernet shield doivent être déclarées comme chaînes de caractères normales car elles sont non statiques. On ne peut donc pas stocker dans une chaîne de caractères l'intégralité d'une page du proxy qui peut contenir jusqu'à 2000 caractères environ à cause de la RAM

trop limitée. La récupération d'une page doit donc se faire en plusieurs fois : on remplit le buffer, on affiche, on remplit le buffer, on affiche, etc...

2. Améliorations possibles

Notre version de Facebook pour Minitel est plutôt limitée en termes de fonctionnalités. On ne peut pas aimer une publication ou la commenter, ni rédiger de post. Pour pouvoir ajouter ces fonctionnalités, il faudrait utiliser une carte microcontrôleur avec plus de RAM, l'Arduino Uno en ayant vraiment trop peu pour pouvoir manipuler beaucoup de chaînes de caractères. En effet, pour pouvoir aimer une publication directement depuis le minitel, il faudrait récupérer, pour chaque publication du fil d'actualité, le lien permettant d'aimer cette publication et de le stocker dans une chaîne de caractère de grande taille car les liens sont assez longs. L'Arduino Mega comporte 8 Ko de RAM, soit 4 fois plus que l'Arduino Uno, nous pourrions donc l'utiliser pour ajouter quelques fonctionnalités au 3615 Facebook sans changer la base de notre programme en C vu que l'ethernet shield est compatible avec l'Arduino Mega. Cependant, pour pouvoir proposer un Facebook le plus complet possible, on remplacera l'Arduino Uno par un Raspberry Pi 2 ou 3, beaucoup plus performant qu'une plateforme microcontrôleur. Le programme serait alors à revoir intégralement, mais moins difficile et long à coder que sur Arduino car le Raspberry Pi dispose d'une version complète de linux, ce qui nous facilite notamment le travail au niveau de la programmation réseau.

Conclusion

En conclusion, nous pouvons dire que nous avons rempli notre objectif de projet IMA 4 qui était d'adapter Facebook sur Minitel. Les fonctionnalités du 3615 Facebook sont au final assez limitées, mais cela est dû en grande partie aux limitations matérielles imposées par l'Arduino. Nous proposons tout de même une interface ergonomique adaptée à l'appareil sans écran tactile ni souris, la connexion à un compte directement depuis le Minitel en tapant l'e-mail et le mot de passe au clavier, l'affichage de quatre pages différentes de Facebook.com : le fil d'actualités, les notifications, les amis et le profil. Il est aussi possible de rechercher une personne ou une page et d'afficher le résultat de la recherche en tapant son nom dans le champ « recherche » du menu « recherche » du Minitel.

Nous nous sommes pleinement investis dans ce projet que nous avons choisi et qui nous a beaucoup intéressés. Il nous a permis de mettre en pratique les compétences acquises au cours de réseau (client web sur Ethernet shield, proxy) et système (communication série, programmation d'un microcontrôleur Atmega 328 P) des semestres sept et huit, et même de les approfondir. De plus, nous avons travaillé la plupart du temps en autonomie sur ce projet, en gérant son organisation et avancement au fil des semaines.

Enfin, nous avons pu nous rendre compte que réutiliser un appareil électronique vieux de plus de 30 ans n'était pas forcément une chose très compliquée car le Minitel a été conçu en respectant les standards de son époque en terme logiciel et matériel (caractères ASCII, liaison série, niveaux électriques compatibles TTL, etc ...). Ces standards existent toujours aujourd'hui, cela permet de faire communiquer le Minitel avec les microcontrôleurs actuels sans aucun circuit d'adaptation. Réutiliser plutôt que jeter les appareils obsolètes est de plus une démarche intéressante dans le cadre de la protection de l'environnement.

Bibliographie

Documents

- Minitel 1B, Spécifications Techniques d'Utilisation
Ministère des Postes et des Télécommunications, 1986
- Minitel 1 Bistandard, Notice à usage professionnel, Modèle Telic
France Télécom, 1988
- Datasheet du Wiznet W5100
http://www.wiznet.io/wp-content/uploads/wiznethome/Chip/W5100/Document/W5100_Datasheet_v1.2.7.pdf
- Datasheet de l'ATMEGA 328 P
http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

Sites web

- Integrating Wiznet W5100, WIZ811MJ network module with Atmel AVR Microcontroller :
<http://www.ermicro.com/blog/?p=1773>