



Mise en place d'un parc d'instrumentation sur le WEB

Rapport de Stage
3^e année ENSERB option TIC

Rougetet Frédéric
Thébaud Cyrille
ENSERB

responsable de projet:
Patrice Kadionik
Promotion 2000

Remerciements

Cette page est consacrée à toutes celles et ceux qui nous ont soutenus durant l'élaboration de ce projet et notamment à Patrice Kadionik, de nous avoir proposer ce sujet dans un premier temps, puis de nous avoir encadrer et dépanner dans les moments difficiles. Nous tenions également à associer messieurs Goudal, pour l'appareil photo numérique et Rodes pour la démonstration du HP8753D.

Sommaire

REMERCIEMENTS.....	1
SOMMAIRE	2
INTRODUCTION.....	3
I PRÉSENTATION DU PROJET RETWINE.....	4
I.1 CONCEPT DE BASE	4
I.2 ENJEU	4
I.3 LE CONCEPT D'INSTRUMENTATION VIRTUELLE	5
I.4 PRINCIPE DE FONCTIONNEMENT	5
I.5 CAHIER DES CHARGES.....	6
II MISE EN ŒUVRE TECHNOLOGIQUE.....	7
II.1 INSTALLATION DU SERVEUR ET GESTION DE LA SÉCURITÉ.....	7
II.1.1 <i>Installation du serveur</i>	7
II.1.2 <i>Gestion de la sécurité</i>	7
II.1.3 <i>Création de comptes</i>	9
II.2 L'INSTRUMENT HP 8753D.....	10
II.2.1 <i>Description de l'instrument</i>	10
II.2.2 <i>Le bus GPIB</i>	11
II.2.3 <i>Driver de l'instrument</i>	11
II.3 L'INTERFACE GRAPHIQUE.....	12
II.3.1 <i>La façade avant de l'instrument</i>	12
II.3.2 <i>Le choix de java</i>	12
II.3.3 <i>Le JDK et le parc d'instrument</i>	12
II.4 COMMUNICATION CLIENT/SERVEUR.....	14
II.4.1 <i>Architecture Client/Serveur</i>	14
II.4.2 <i>Scripts CGI</i>	15
III IMPLANTATION.....	16
III.1 INTERFACE GRAPHIQUE.....	16
III.1.1 <i>Fichier Main</i>	16
III.1.2 <i>Fichier Softkeys</i>	18
III.2 COMMUNICATION INTERFACE/HP8753D.....	22
III.2.1 <i>ComDataClients</i>	22
III.2.2 <i>Script CGI(Shell)</i>	22
III.2.3 <i>Driver (C)</i>	23
III.3 ORGANISATION DES FICHIERS SOURCES.....	24
IV UTILISATION.....	25
CONCLUSION.....	28
ABRÉVIATIONS.....	29
BIBLIOGRAPHIE.....	30
WEBGRAPHIE.....	30
ANNEXES.....	31

Introduction

Ce rapport présente le projet que nous avons effectué au sein des laboratoires de l'ENSERB (Ecole Nationale Supérieure d'Electronique et de Radioélectricité de Bordeaux) durant les mois de mars à juin.

Il s'inscrit dans le cadre du projet Retwine (**RE**mo**Te** **W**orldwide **I**nstrumentation **N**etwork), projet proposé par l'IXL avec le concours d'université étrangère telles l'universidad de Madrid (Espagne) et la Fachhochschule Münster (Allemagne).

Retwine propose à ses utilisateurs de se connecter à distance à des appareils de mesure souvent très coûteux, via le web. Notre travail a donc consisté par le biais de ce projet à implémenter un nouvel appareil, en l'occurrence l'analyseur de réseaux HP8753D. La nouveauté par rapport au parc d'instruments déjà existant est également le changement de serveur. On passe en effet du serveur de l'IXL à un serveur Apache.

Nous commencerons dans ce rapport par parler un peu plus en détail du projet Retwine et de l'extraordinaire outil pédagogique qu'il représente. Nous aborderons le travail déjà effectué ainsi que les nouveautés apportées.

Nous verrons ensuite comment s'est présenté le projet en abordant la partie technique avec le détail des programmes et le manuel d'utilisation.

I Présentation du projet RETWINE

I.1 Concept de base

L'idée de base a été de mettre en œuvre le concept d'application client/serveur du WWW : on a d'un côté un parc d'instruments de mesure pilotés par une application serveur accessible de n'importe où dans le monde par des applications clientes (ou "browser") via le réseau Internet. Un système d'identification est de plus mis en œuvre pour restreindre l'accès au parc d'instruments mis à disposition des personnes dûment enregistrées. L'application serveur propose en outre le panneau de face avant virtuel de chaque instrument qui pourra être ensuite piloté par le browser de la personne autorisée.

L'intérêt d'une telle approche est ainsi de mettre à disposition de la communauté universitaire un parc d'instruments de mesure performants et récents en vue d'apprendre à les utiliser mais aussi pour réaliser des mesures dans le cadre de recherches via le réseau Internet. On a donc ainsi mis en service sur le "Web" des instruments virtuels WWW.

I.2 Enjeu

Le test de composants et systèmes VLSI est fondamental pour le développement des procédés de fabrication, la caractérisation des composants et l'analyse de défaillance. Pour les tests électriques par exemple, on a à disposition une série d'équipements couramment utilisés : système de caractérisation VLSI, analyseur de paramètres électriques, analyseur de réseau, oscilloscope numérique, générateur de fonctions, analyseur de spectre...

La mise en place d'un cours universitaire pour des étudiants en microélectronique concernant l'utilisation et la maîtrise de ces équipements est nécessaire mais n'est pas toujours possible à cause généralement du coût élevé de tels instruments (500 KF pour un analyseur de réseau). Ce coût élevé est donc un frein majeur à la mise en place d'un cours universitaire pratique.

L'acquisition d'une telle instrumentation à des fins purement pédagogiques revient donc trop cher et de plus, les équipements seraient dépassés en quelques années. La durée de vie moyenne d'un cours universitaire pratique est d'environ 10-15 ans ; ce qui reste incompatible avec l'évolution rapide de la microélectronique et de son instrumentation.

L'approche originale présentée maintenant est basée sur l'utilisation des concepts multimédia du Web. Cette approche d'enseignement à distance ou téléenseignement est maintenant possible avec ce formidable moyen de communication qu'est le WWW ; ce qui va permettre aux étudiants dans le cadre d'une formation universitaire de maîtriser et d'avoir en plus une expertise dans l'utilisation d'instruments les plus complexes mais aussi les plus récents mis à leur disposition sur le Web.

1.3 Le concept d'instrumentation virtuelle

Dans chaque laboratoire, il existe toujours un parc d'instruments de mesure disponibles mais qui n'est pas toujours utilisé à plein temps. Généralement, chaque laboratoire possède ses propres spécificités dans un domaine d'activité particulier : télécommunications, conception d'ASICs, assemblage, conception de circuits VLSI... Le parc d'instruments est ainsi spécifique et donc très ciblé. Mais les étudiants devraient avoir accès à une large palette d'équipements pour posséder une connaissance la plus complète possible. Pour pallier ce problème, nous proposons de mettre en place le concept d'instrument de mesure virtuel accessible par le Web : les instruments et les mesures effectuées sont ainsi géographiquement distincts.

Nous avons d'un côté un parc d'instruments situés n'importe où dans le monde mis à la disposition de la communauté universitaire. Ce parc, composé de la contribution personnelle de chaque laboratoire est par nature des plus complets.

Les instruments de mesure doivent néanmoins posséder une interface IEEE-488 pour pouvoir être pilotés par logiciel (interface la plus commune). Pour un laboratoire donné, les instruments IEEE-488 sont pilotés par une application serveur lancée sur une station de travail connectée à Internet. Cette application interagira avec le serveur WWW développé sous Linux. Les principales fonctions de l'instrument et sa face avant "virtuelle" sont présentées dans une page Web écrite en langage HTML (" Hyper Text Markup Language ").

De l'autre côté, nous retrouvons les applications clientes ou browser Internet qui pourront interagir à distance avec cet instrument via la page Web. L'accès à ce nouveau service Internet d'instrument virtuel ne sera autorisé qu'aux personnes identifiées et reconnues. Le mécanisme d'identification est réalisé par le serveur WWW qui possède généralement un système d'identification. Après identification, il est alors possible d'utiliser et de faire des mesures avec l'instrument virtuel.

1.4 Principe de fonctionnement

Il est nécessaire d'avoir une station de travail (ou un PC) possédant une carte GPIB pour piloter les instruments IEEE-488. Une application serveur doit être développée à cet effet. Un serveur WWW doit être aussi mis en place et peut être installé sur la station de travail. Il convient ensuite de développer les pages HTML où l'on définira les principales fonctions et la face avant de chaque instrument. L'interaction entre l'application serveur et les requêtes valides reçues par le serveur WWW est réalisée à l'aide de requêtes HTML "GET" ou "POST"; ce qui oblige donc à écrire des scripts CGI (" Common Gateway Interface "). Ces pages deviennent alors accessibles par tout le monde par le Web. Un mécanisme d'identification sera enfin à mettre en place pour identifier par mot de passe l'accès au service d'instrument virtuel. La mise en place d'une base de données indépendante de celle du système d'exploitation est généralement offerte par le serveur WWW pour la gestion de mots de passe.

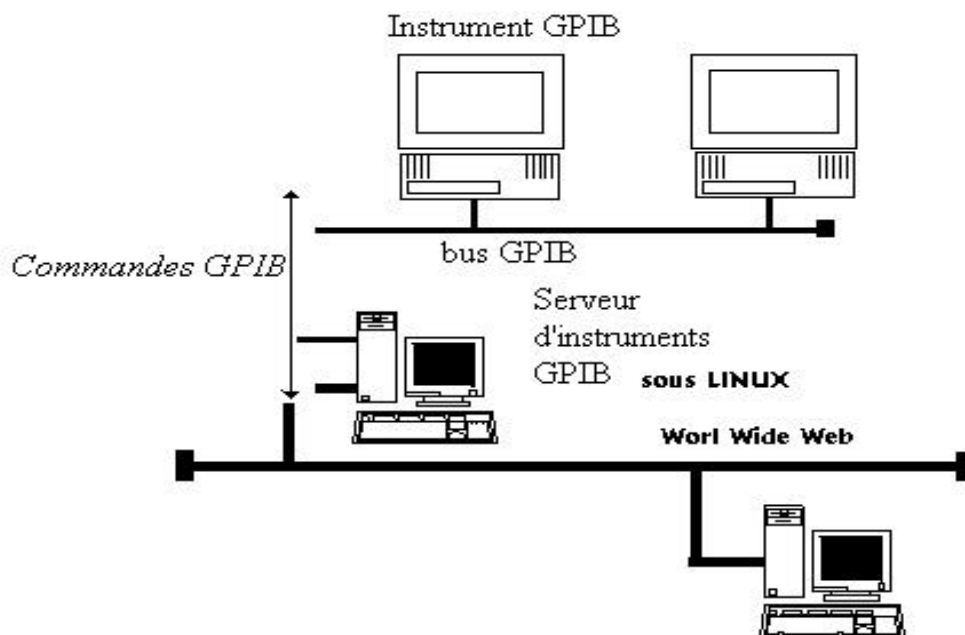


Figure 1: Architecture d'implémentation RETWINE

Par ailleurs, un système de chien de garde interdit la connexion d'un autre utilisateur désirant utiliser un appareil déjà sollicité.

1.5 Cahier des charges

L'objectif de notre projet est d'intégrer au parc d'instruments déjà existant l'analyseur de réseaux **Hewlett Packard 8753D**. Le dispositif de pilotage à distance repose sur une architecture client-serveur.

Une applet **Java** communique avec un serveur WWW. L'applet ouvre une socket sur le serveur pour y ordonner l'exécution d'un programme. Ce programme, le driver de l'instrument, commande via le **bus GPIB**, l'analyseur de réseaux HP 8753D et fournit en retour un résultat.

Le serveur WWW doit gérer les **scripts CGI** pour l'exécution de commandes locales sur l'appareil, et reconnaître les applets Java qui seront transmises au client pour l'interfaçage avec le serveur. Ainsi, il doit donc être connecté directement à l'appareil via le bus GPIB.

Par ailleurs, le serveur doit disposer de moyens de protection afin de contrôler l'accès et l'utilisation de l'appareil d'instrumentation. La solution la plus générique doit être adoptée pour faciliter l'implémentation ultérieure d'autres instruments.

II Mise en œuvre technologique

II.1 Installation du serveur et Gestion de la sécurité

La bonne marche du projet nécessitait un serveur de qualité. C'est donc le **serveur Apache** qui a été retenu. En effet le serveur Apache est un serveur http qui présente le double avantage d'être à la fois libre et très sécurisant.

II.1.1 Installation du serveur

Apache est un programme qui fonctionne sous un système d'exploitation multitâche approprié, Linux est par conséquent parfaitement adapté à ce type de serveur. Le répertoire d'un site contient typiquement trois sous-répertoires :

Conf

Contient le ou les fichier(s) de configuration, dont *http.conf* est le plus important.

htdocs

Contient les scripts HTML à servir aux clients du site. Ce répertoire et ceux qui se trouvent au-dessous de lui, l'espace web, sont accessibles à toute personne du web et présentent par conséquent un risque sévère pour la sécurité s'ils abritent autre chose que des données publiques.

logs

Contient les données d'historiques, aussi bien des accès que des erreurs.

Il est à noter que le répertoire *cgi-bin*, dans lequel résident nos scripts CGI ne doit absolument pas, pour des raisons de sécurité, se trouver dans l'espace web. Au repos, Apache ne fait qu'« écouter » les ports TCP des adresses IP indiqués dans son fichier *Config*. Quand une requête HTTP se présente sur un port valide, Apache la reçoit et en analyse les entêtes. Il applique alors les règles qu'il trouve dans le fichier *Config* et agit en conséquence.

Apache est un freeware, et l'utilisateur intéressé télécharge les sources depuis www.apache.org ou depuis un site miroir approprié, les compile, et exécute le résultat.

II.1.2 Gestion de la sécurité

Comme il a été dit précédemment, notre serveur nécessite un système de sécurité pour limiter l'accès aux instruments. Il faut compléter le fichier *http.conf*, puisque Apache possède une option pour limiter l'accès par utilisation d'un mot de passe.

Les scripts HTML sont contenus dans le répertoire `htdocs` et dans ces sous-répertoires, la demande de mot de passe doit s'effectuer lors de l'ouverture de la fenêtre pilote de chaque instruments, fenêtre qui propose comme nous le verrons plusieurs options : le manuel d'utilisation de l'appareil, le lancement de l'applet d'utilisation la sauvegarde et l'affichage des mesures. Les répertoires à protéger sont donc les `d_HP*****` correspondant au différents appareils commandés.

Voici maintenant les directives de protection des répertoires.

AuthType : indique le type de contrôle d'autorisation, le contrôle BASIC est employé étant plus standard pour les navigateurs.

AuthName : fournit le nom de royaume dans lequel les mots de passe et les noms d'utilisateurs sont valides.

AuthGroupFile : donne le nom d'un fichier qui contient les noms des groupes et des utilisateurs associés.(voire Require)

AuthUserFile : est le fichier de noms d'utilisateurs et de leurs mots de passe cryptés.

Limit : peut recevoir plusieurs arguments correspondant à des noms de méthode HTTP. Cette directive limite l'application aux scripts qui utilisent les méthodes spécifiées. Nous limiterons l'accès par GET et POST, en effet ce sont ces requêtes qui sont utilisées.

Require : est la directive qui lance la vérification des mots de passe. Plusieurs arguments peuvent lui être associés ; `user us1 us2 ...` , `group gr1 gr2 ...` , `valid-user`.

`Us1` et `us2` sont des noms d'utilisateurs, apache va alors directement chercher le mot de passe dans le fichier utilisateur pour l'autorisation, `gr1 gr2` sont des noms de groupe, apache passe d'abord par le fichier groupe, et enfin `valid-user` accepte n'importe quel utilisateur figurant dans le fichier mot de passe.

Voici donc la partie protection des instruments du `httpd.conf` :

```
<Directory /home/users/retwine/apache/htdocs/instruments/HP*****/d_HP*****>
AuthType Basic
AuthName retwine
AuthUserFile /home/users/retwine/utilisateur/fichier_utilisateur
AuthGroupFile /home/users/retwine/utilisateur/fichier_groupe

<Limit GET POST>
require group HP*****
</Limit>
</Directory>
```

Cette protection est la même pour les différents appareils configurés, à savoir entre autre HP4194A et HP8753D.

II.1.3 *Création de comptes*

Après avoir complété le fichier de configuration pour la mise en place de demande de mot de passe, il faut créer les fichiers de groupes et d'utilisateurs.

On commence par créer un répertoire « utilisateur » inaccessible par le Web afin de protéger les mots de passe. On crée alors un fichier « fichier_utilisateur » pour y stocker nom d'utilisateur-password correspondant et un fichier « fichier_groupe » contenant les groupes et leurs membres.

Le fichier groupe est composé d'un groupe par appareil, dont le nom correspond au nom de l'appareil, chaque nom de groupe est suivi du noms des utilisateurs autorisés, par exemple :

HP8753D : Paul Émile Victor
HP4194A : François Valéry

Le fichier utilisateur contient les noms d'utilisateurs et le mot de passe codé, ce fichier est rempli par la commande:

% htpasswd ../utilisateur/fichier_utilisateur username (1)

Ajout d'un nouvel utilisateur :

L'ajout d'un nouvel utilisateur est réalisé par la personne maintenant le serveur en place, pour cela il lui suffit :

- d'ajouter le nom du demandeur (ex Wally) dans la liste de l'appareil souhaité (ex HP8753D), on obtient alors :

HP8753D : Paul Emile Victor Wally

- de taper la ligne de commande (1) en remplaçant username par Wally
- d'entrer deux fois de suite le mot de passe de Wally.

11.2 L'instrument HP 8753D

11.2.1 Description de l'instrument

Le HP8753D (voir figure2) est un analyseur de réseaux offrant une combinaison de facilité d'utilisation, de vitesse et de performance. Sa gamme de fréquence d'utilisation va de 30KHz à 3GHz. Il offre une solution complète pour caractériser le comportement linéaire ou non-linéaire des réseaux passifs ou actifs.

Il possède quelques particularités très intéressantes :

- Il traite les signaux numériques associé à des commandes de microprocesseur.
- Sortie des données de mesure affichées, accompagnées d'une indication de date et d'heure, pour impression directe sur imprimante ou traceur compatible muni d'un port série, parallèle ou d'une interface HP-IB.
- Fonction de calcul automatique du temps de balayage établissant le temps de balayage minimal en fonction de la bande passante de fréquence intermédiaire, du nombre de points, du mode de moyennage, de la gamme de fréquence et du type de balayage spécifiés.
- Grande souplesse d'exploitation grâce à des fonctions mathématiques de trace et à diverses fonctions de moyennage des données, de lissage de trace, de retard électrique et d'amélioration de la précision.
- Ou encore plusieurs améliorations de la précision allant de la simple normalisation des données jusqu'à des corrections d'erreur vectorielle complètes sur un ou deux ports avec 1601 points de mesure.



Figure 2 : Face avant du HP8753D

11.2.2 *Le bus GPIB*

Le HP8753D peut être piloté grâce à son interface IEEE 488 (Institute of Electrical and Electronics Engineers). Le bus GPIB (General Purpose Instrumentation BUS) est donc utilisé pour le télécommander via un PC, ou une station, équipée d'une carte IEEE.

Le bus GPIB est une interface standard de communication parallèle, à 8 bits et dont la vitesse de transfert des données est supérieure à 1 M octets/s. Il permet la liaison entre un système de contrôle (PC ou station) et jusqu'à 15 appareils possédant une adresse GPIB distincte.

11.2.3 *Driver de l'instrument*

Afin de piloter l'instrument à partir du système de contrôle (PC ou station), un driver est nécessaire. Ce programme consiste à envoyer des données à l'appareil et à en recevoir via l'interface GPIB. La carte IEEE que possède notre PC, est fournie avec une bibliothèque de fichiers écrits en langage C. Ainsi, le driver sera développé en C, comme les drivers des appareils déjà implémentés au réseau RETWINE.

11.3 L'interface graphique

11.3.1 La façade avant de l'instrument

Pour un meilleur confort de l'utilisateur, nous avons décidé d'utiliser une photo de la face avant de l'appareil, dans le but d'être plus proche de la réalité. Pour cela une photo a été prise avec un appareil numérique, et nous l'avons ensuite modifié pour que celle-ci soit à la fois net et agréable (voir figure2).

11.3.2 Le choix de java

L'implantation de l'application représentant l'appareil étant faite dans une page Web, le choix de Java pour réaliser l'« Applet », se fait tout naturellement. Java est un outil adapté au multimédia et permettant la réalisation d'interface graphique.

11.3.3 Le JDK et le parc d'instrument

L'analyseur d'impédance et de gain/phase HP4194A

Le HP4194A est le premier appareil mis à la disposition d'utilisateur via Internet. Il a été développé avec le **JDK1.0**.

- L'interface graphique est constituée d'une photo de la face avant de l'instrument sur laquelle sont définies des zones qui génèrent des événements.
- La gestion événementielle n'est pas des plus pratiques.
- L'utilisation des softkeys dans une fenêtre séparées dont le concept sera repris par la suite.

L'analyseur de paramètres pour semi-conducteurs HP4155A

Le HP4155A est le second instrument du parc virtuelle RETWINE. Il a été développé avec le **JDK 1.2**. Le JDK1.2 est la version la plus récente qui a été utilisée pour le projet RETWINE. Ses points forts sont :

- Des composants graphiques tels que les *swing* permettant de réaliser facilement des interfaces de qualité.
- Une gestion événementielle remaniée (depuis le JDK 1.1)

Mais un problème non négligeable toutefois : actuellement, les machines virtuelles Java des navigateurs ne supportent pas tous le JDK1.2.

L'analyseur de réseau HP8510B

Pour éviter ce problème avec les navigateurs et permettre aux utilisateurs d'accéder sans difficultés à l'instrument implémenté, le HP8510B a été développé avec un JDK plus ancien mais qui est supporté par les navigateurs. Le choix du JDK 1.1.5 s'est trouvé être un bon compromis. Il permet de réaliser une interface graphique :

-
- Conviviale avec des composants graphiques posés sur la photo de l'instrument pour donner un peu plus de relief.
 - Une gestion événementielle plus efficace qu'avec un JDK1.0.
 - Une interface supportée par les navigateurs contrairement au JDK1.2 qui ne l'est pas encore.

L'analyseur de réseau HP8753D

Le développement du HP8753D s'est voulu être le prolongement du travail déjà effectué. Au niveau de l'interface graphique, elle est constituée d'une photo de la face avant de l'appareil comme pour tous les instruments du parc. Celle-ci se révélant d'une très bonne qualité, nous n'avons trouver utile de redessiner des boutons sur la photo. Celle-ci se retrouve donc constituée de zones qui génèrent des événements.

II.4 Communication Client/Serveur

II.4.1 Architecture Client/Serveur

L'architecture Client/Serveur est une technologie informatique très utilisée actuellement notamment dans le domaine des SGBD (Système de Gestion de Base de Données) ou de l'Internet. Deux acteurs entrent donc en jeu dans cette architecture, le client et le serveur, à travers un réseau le plus souvent.

Le serveur est un programme ou une machine programmée pour rendre toujours un même service suite à une requête qui lui est adressée. Le client est un programme ou une machine qui demande un service à un serveur en lui adressant une requête.

Le client/serveur est souvent utilisé pour accéder à de grandes bases de données qui sont stockées sur une machine, le serveur, et qui sont utilisées par plusieurs utilisateurs. Le serveur a pour mission de répondre aux requêtes que lui adressent les clients, en modifiant ou interrogeant la base de données qu'il gère et en envoyant ensuite la réponse attendue.

Dans l'exemple de l'Internet, les sites consultés constituent les serveurs, car ils rendent le même service au client qu'est le navigateur. Le service rendu par le serveur est le renvoi de la page du site demandé, et qui sera visualisée une fois téléchargée.

Cette architecture client/serveur permet de séparer complètement le serveur qui gère le stockage des données, du client qui cherche à y accéder. Un protocole commun à un client et au serveur est utilisé afin d'établir leur communication et ses normes. Le protocole le plus largement utilisé est le TCP (Transport Control Protocol) qui garantit que les données transmises d'un serveur arriveront sur le client ou vice-versa, et dans l'ordre où elles ont été émises.

Par ailleurs, les serveurs connectés sur Internet sont accessibles grâce à une mnémonique qui est le nom de la machine hôte (par exemple www.yahoo.com ou java.sun.com). Chaque mnémonique est associée à un identifiant numérique représentant la machine hôte qui héberge le serveur : c'est l'adresse IP (Internet Protocol) constituée de 32 bits, notée sous la forme de quatre nombres séparés par des points. Toute machine reliée à Internet possède son adresse propre et unique.

D'autre part, plusieurs programmes peuvent tourner en même temps sur une même machine grâce à leur système d'exploitation. Si plusieurs serveurs sont hébergés sur une même machine, ils vont devoir partager l'accès physique au réseau sur cette machine. Un deuxième niveau d'identification est donc nécessaire pour désigner le programme avec lequel on désire faire la connexion sur une machine donnée. Ainsi, à chaque programme désirent communiquer sur Internet est associé un numéro de port unique de 16 bits (de 0 à 65535). Les ports compris entre 0 et 1023 sont réservés à des services particuliers et ne doivent pas être utilisés inconsciemment.

Ainsi, dans le cadre du projet RETWINE, le numéro de port choisi est le 8080.

11.4.2 Scripts CGI

Pour faire appel au driver de mesure et le commander via le bus GPIB, le choix s'est porté sur les scripts CGI (Common Gateway Interface). Une autre possibilité était d'utiliser les servlets. Mais la solution des scripts CGI a précédemment été retenue pour la mise en place des instruments. Etant donné la faible durée de temps pour le développement du projet, il était préférable de reprendre l'emploi du script CGI en l'adaptant au HP8753D.

Les scripts CGI permettent à l'utilisateur de disposer de documents Web réalisés de façon dynamique par un programme du serveur sur requête du navigateur. Le document ainsi produit renferme aussi bien des données extraites du serveur que le résultat du traitement d'un formulaire complété et expédié par l'utilisateur. Le programme peut être écrit en Perl, C, Shell, etc : le choix du langage utilisé n'est pas restreint. Les scripts pour les appareils existants ont été développés en Shell, on gardera donc le même principe.

Les programmes CGI sont lancés par le serveur HTTP sous la demande du client, et s'exécutent sous formes de processus indépendants. Les avantages du CGI sont leur simplicité, leur relative sécurité et le coût moyen de leur exécution. Le programme peut avorter sans endommager le serveur, du moins sur un système d'exploitation en mémoire protégée et à gestion multitâche tel qu'UNIX. De plus l'accès au serveur reste limité pour un programme CGI. Par ailleurs, le coût moyen d'exécution du programme est comparable à celui exigé par l'extraction d'un fichier.

Voici un exemple :

(GET /POST) /cgi-bin/executable.sh ?parametre(s)

Cette ligne, appelée requête, est acheminée vers le serveur avec l'une ou l'autre méthode GET ou POST sous la forme d'un flux. Le serveur reconnaît au chemin **/cgi-bin** qu'il s'agit d'un programme CGI, et exécute `mon_executable.sh` avec les entrées à traiter.

De plus amples détails seront données plus tard au sujet des entêtes du flux que le navigateur transmet au serveur.

III Implantation

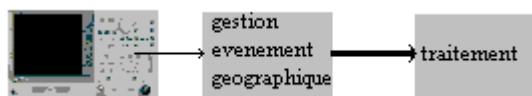


Figure 3 :Organisation des programmes

Deux parties distinctes composent l'organisation de nos programmes, la première génère l'interface graphique et le traitement des événements et la deuxième s'occupe du traitement des données pour piloter l'appareil depuis le web via le bus GPIB.

III.1 Interface graphique

On peut considérer que deux fichiers composent principalement l'interface graphique, fichiers qui sont HP8753D_main et HP8753D_Softkeys.

III.1.1 Fichier Main

Le corps de l'applet que nous avons développée, c'est en effet ce fichier qui est appelé lors du lancement de l'applet dans le document HP8753D.html. C'est lui qui gère l'environnement graphique choisi pour l'appareil, mais aussi les événements intervenant sur les touches de la face avant, hors touches interactives (voir III.1.2).

Voici comment est organisé le fichier :

- chargement des images nécessaires à l'applet et des Frames provenant du fichier de touches interactives, déclaration des variables,
- définition des fonctions de lancement et d'arrêt de l'applet, et de dessin du graphique,
- définition de la fonction principale, détection d'événements géographiques et action résultante. L'envoi des commandes sera détaillé dans la partie IV .

III.1.1.1 Chargement

Les images utilisées pour l'affichage de l'applet, sont la face avant du HP8753D, deux images de boutons d'appareil servant à redessiner les touches quand un clic de souris est détecté, l'image d'une led d'indication du canal utilisé.

On initialise ensuite un nouvel objet comme touches interactives :

```
SoftkeysWindow = new HP8753D_SoftkeysWindow()
```

ceci nous permet grâce aux fonctions, d'affichage, de redimensionnement entre autres de générer une nouvelle fenêtre pour les touches du fichier softkeys.

III.1.1.2 Détection d'événement

L'organisation de la fonction `handle_event` est la suivante, à chaque événement, on teste (fonction *if*) son type, ses coordonnées et suivant le cas on lance les actions appropriées :

Détection d'événement

```

{
  Type d'événement1
  {
    position géographique1 de l' événement
    {
      action coordonnée
    }
    position géographique2 de l' événement
    {
      .....
    }
  }
  Type d'événement2
  {
    position géographique de l' événement ...
  }
}

```

Trois événements sont pris en compte dans cette partie du programme (voir figure4), l'appuie sur un bouton de la souris (fct `MOUSE_DOWN`), le relâchement de ce même bouton (fct `MOUSE_UP`) et le déplacement du curseur de la souris (fct `MOUSE_ENTER`). Les deux derniers ne servant qu'à rendre, l'interface plus conviviale.

EVENEMENT	POSITION	ACTION
Bouton de souris appuyé	Bouton On /Off	Si appareil éteint teste : teste de disponibilité de l'appareil, et mise en marche Si appareil allumé, extinction
	Touche déroulement de menu (touche response et Menu)	Envoi du numéro de menu au softkeys, changement de l'apparence du bouton
	Touche stimulus hors Menu	Demande de valeur dans le menu des softkeys, changement de l'apparence du bouton
	Touche Preset	Reset de l'appareil, changement de l'apparence du bouton
	Touche de canal	Choix du canal, changement de l'apparence du bouton

EVENEMENT	POSITION	ACTION
Bouton de souris relâché	***	Apparence initiale du bouton
Déplacement du curseur	***	Envoie dans la fenêtre de status du nom de la touche

Figure 4 :Gestion des évènements.

III.1.2 Fichier Softkeys

Sur le HP8753D, il existe 8 touches interactives (appelées softkeys), celles-ci par l'intermédiaire des touches de menus, provoque une action, une demande d'entrer de valeurs ou de données, ou bien mènent tout simplement vers un autre menu. Les différents menus sont ainsi organisés en un arbre complexe (voir figure5).

Comme le fichier Main se fichier s'oriente sur trois partie :

- déclaration des variables et de la frame de communication
- une fonction permettant de choisir le menu à afficher
- et une fonction de détection d'évènement servant à se positionner dans l'arbre que représentent les menus.

III.1.2.1 Variables et Menus

Il faut premièrement définir les huit boutons, ainsi qu'un emplacement ou entrer des valeurs numériques pour les fonctions le demandant.

Pour cela on utilise la classe Button et la classe TextField (champs de texte) , dont on peut adapter les noms grâce aux fonctions setLabel ou SetText. Il suffit ensuite de définir des constantes « texte » et de les appeler au moment voulu. L'affichage du bon menu est le rôle de la fonction SetSK(Nb),c'est cette fonction qui nous place dans l'arbre.

En fait lors de la gestion des évènement, on fait appel à SetSK en lui indiquant un nombre dépendant de la position dans l'arbre et de l'évènement détecté (voir 3.1.2.2).

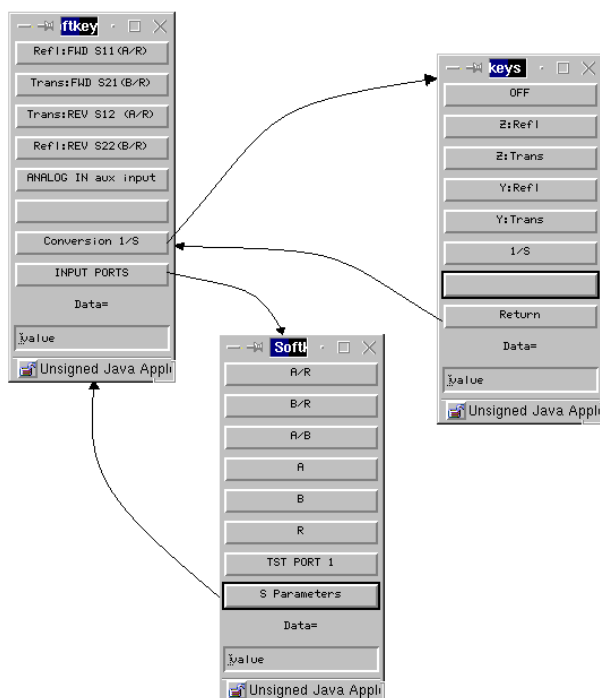


Figure 5 :Exemple de navigation(arbre) possible à partir de la touche MEAS

Voici le corps de la fonction SetSK :

```
public void SetSK (int Nb)
{
    .....
    else if (Nb == 17)
    {
        Softkey1.setLabel(SKS171);
        Softkey2.setLabel(SKS172);
        Softkey3.setLabel(SKS173);
        Softkey4.setLabel(SKS174);
        Softkey5.setLabel(SKS175);
        Softkey6.setLabel(SKS176);
        Softkey7.setLabel(SKS177);
        Softkey8.setLabel(SKS178);
    }
    .....
}
```

III.1.2.2 Fonction handleEvent

On fait appel ici à la fonction Action_Event, ces évènements sont les plus répandues, ils sont utilisés avec la plupart des composants d'interface utilisateur pour indiquer qu'un composant a été activé .Un évènement action est généré dans notre cas quand on appuie sur un bouton ou quand on appuie sur entrée dans un champs de texte.

En appuyant sur un bouton on peut rencontrer trois actions possibles simultanées ou pas, l'envoi d'une commande, l'envoi d'une donnée à saisir par l'utilisateur, un changement de menu.

Le changement de menu s'effectue par appel à la fonction SetSk (voir ci-dessus), l'envoi des commandes passe par la fonction Send (voir ci-dessous), et celui des données demande une intervention de l'utilisateur que nous allons détaillée. Voici pour l'instant l'exemple d'un appuie sur la softkey 6 en étant dans le menu numéro 4843:

```
else if (e.target == Softkey6 && N == 4843)
{
    ComTool.Send("COLOTEXT");
    SetSK(48431);
}
```

On remarque l'envoi de la commande "COLOTEXT", et le changement de menu "SetSK".

L'envoi de données est lui un peu plus complexe, on demande, avant l'envoi, à l'utilisateur de saisir dans le champ de texte la valeur qu'il désire faire prendre à la variable(voir figure**). Pour cela, après avoir appuyer sur un bouton (ci-dessous bouton5), on change le titre du Label précédent le champs de texte (3) par le nom de la variable à modifier, on vide le champs de texte de ce qu'il est susceptible de contenir(2), et on modifie la variable Data avec le nom de la variable.

```
else if (e.target == Softkey5 && N == 48)
{
    Data= "STAR";                (1)
    DataField.setText("");       (2)
    DataLabel.setText("START="); (3)
}
```

L'envoi définitif est effectué après l'appuie sur "Entrée", de la manière suivante:

On concatène la variable Data avec sa valeur numérique, la nouvelle valeur est appelée DataText (4), en on envoie cette donnée(5).

```
if (e.target == DataField)
{
    System.out.println(e.arg);
    DataText = (Data+e.arg + ""); (4)
    System.out.println(DataText);
    ComTool.Send(DataText);      (5)
    DataField.setText("Data:"+e.arg);
}
```

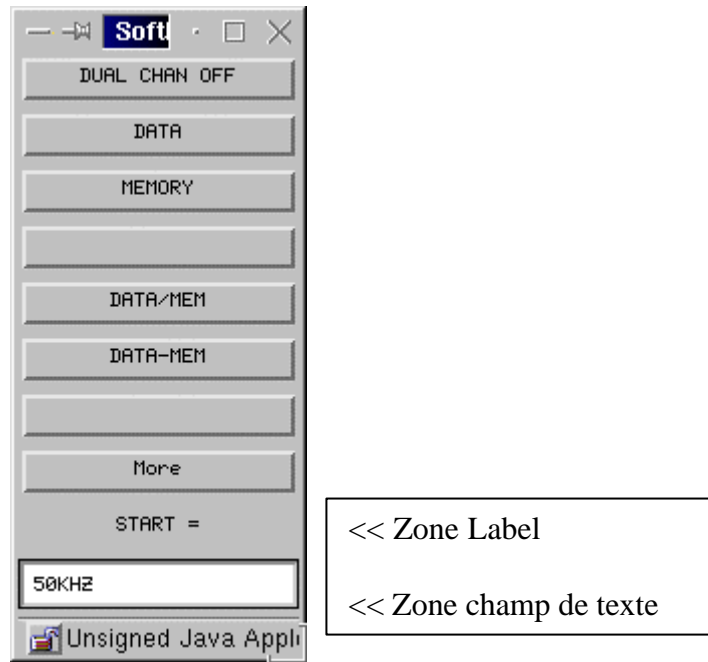


Figure 6 :Changement du Label et entrée de la valeur numérique.

III.2 Communication interface/HP8753D

III.2.1 ComDataClients

Le ComDataClient est un fichier qui a été conçu pour servir de passerelle entre les fichiers attachés à l'interface (Softkeys et Main) et le script CGI.

La fonction importante de ce fichier est la fonction **public void** Send (String Inst) qui gère les connexions aux serveurs pour transmettre les commandes et recevoir les données.

Elle ouvre une socket vers le serveur :

```
Sock = new Socket (RETWINESERVER, RETWINEPORT)
```

Elle crée des variables outStream et inStream qui sont les données émises et reçues par le programme.

Il est à noter que cette fonction détecte également les erreurs (telles que l'impossibilité de créer une Socket) et les gère.

La nouveauté avec la mise en place du HP8753D est la mise en place du serveur Apache. Il faut adapter l'envoi du HTTP header de la fonction à Linux et notamment au serveur Apache :

```
outStream.writeBytes("POST " +  
    cgiScript +  
    "?" +  
    Inst +  
    " HTTP/1.0\r\n");  
outStream.writeBytes("Content-type: " + ctype + "\r\n");  
outStream.writeBytes("Content-length: " + Inst.length() + "\r\n");  
outStream.writeBytes("\r\n");
```

III.2.2 Script CGI (Shell)

Lorsqu'une requête est effectuée pour appeler le script CGI, le serveur exécute celui-ci et lui passe par l'entrée standard les arguments du client. Un mécanisme de protection est alors mis en œuvre pour réserver les ressources matérielles du client. Celui-ci est identifié par l'adresse IP de sa machine. Pratiquement, un fichier est créé dont le nom comporte un préfixe déterminant l'appareil et un suffixe identifiant l'utilisateur. Il est alors simple de tester l'existence d'un fichier préfixé du nom de l'appareil si celui-ci est utilisé.

Un watchdog (chien de garde) est utilisé pour effacer régulièrement les verrous : un processus est lancé en tâche de fond, attend une période fixe déterminant le temps alloué au client avant libération de ressources pour finalement réinitialiser le verrou. Ainsi, si le client n'effectue pas de requête dans le temps réservé, l'appareil est libéré. Un message (USED ou NOT_USED) informe le client s'il peut accéder à l'appareil. Dans le cas affirmatif, i. e. dans le cas où les ressources sont libres, le script invoque par rsh l'exécution du driver de l'appareil en passant les arguments par l'entrée standard.

Les fichiers générés par le driver sont ensuite déplacés dans le répertoire adéquat afin que l'utilisateur puisse visualiser les résultats via son navigateur, sous forme de colonnes de mesures ou de courbes.

III.2.3 Driver (C)

Le driver utilise est une surcouche de celui de National Instruments Corporation: le NI-488.2M Driver pour des stations Sparc. Les fonctions pour communiquer avec l'appareil via le bus GPIB sont celles fournies par la bibliothèque.

La communication se fait tout d'abord par:

- Initialiser le bus GPIB
- Identifier la carte IEEE
- Identifier l'appareil en lui donnant une adresse GPIB: l'adresse 16 est attribuée au HP8753D.

Ces opérations ont été effectuées dans **int init()**, fonctions qui retourne le numéro identifiant l'appareil. Afin de debugger le pilotage du HP8753D, l'instruction "DEBUON;" est envoyée (fonction **void start(int ud)**): les commandes reçues par l'appareil sont alors affichées sur son écran.

Les commandes reçues sous formes de *char par le driver sont envoyées à l'instrument par la fonction GPIB:

void ibwrt (int identifiant_du_HP, char *commande, int longueur_du_char*)

Les réponses venant de l'appareil sont récupérés par:

void ibrd (int identifiant_du_HP, char *reponse, int longueur_du_char*)

Toutes les commandes sont envoyées directement au HP8753D, et des mesures ne sont effectuées qu'après "SING;" ou "REST;" grâce à la commande "FORM4;OUTFORM;" qui donne en retour les résultats sous forme ASCII. Les mesures sont ensuite stockées dans des fichiers.

Un message d'erreur est signalé pour tout problème rencontré dans le bus GPIB grâce aux variables `ibsta` et `ERR`.

La fonction **void data_tofile(int ud, char *answer)** génère trois fichiers de résultat contenant chacune une entête de description et les valeurs mesurées. Pourquoi trois?

Une applet de plotter déjà conçue pour traiter les mesures récupère deux colonnes de données (x,y) et construit la courbe correspondante $y=f(x)$. Or le HP8753D ne donne pas les valeurs de fréquences lui-même: il donne les valeurs de module (ou de phase, de réel ou autre) sans donner les fréquences correspondantes. Celles-ci sont donc calculées

par la fonction `data_tofile` et sont ensuite affichées dans le fichier qui sera utilisé par le `plotter` pour tracer la courbe.

Prenons l'exemple d'une mesure de module dans une bande de fréquence. Le HP8753D fournit les valeurs X =module et $Y=0$ qui vont être stockées dans trois fichiers différents. Ainsi le fichier `HP8753D_mesure.tab` contiendra deux colonnes de valeurs (fréquence F , module X), le fichier `HP8753D_mesureXY.tab` deux colonnes de valeurs (X,Y), le fichier `HP8753D_mesureFXY.tab` trois colonnes de valeurs (F,X,Y). Pour l'instant, seul le fichier `HP8753D_mesure.tab` est utilisé par le `plotter`.

Les fréquences, sont calculées grâce au nombre de points de mesure demandé par l'utilisateur. Par défaut, la fréquence de début de mesure est de 45MHz, celle de fin est de 26.8GHz, et le nombre de points est de 201 (d'après l'état d'initialisation du HP8753D).

III.3 Organisation des fichiers sources

Le serveur est structuré comme suit :

- Répertoire *Apache* où sont contenus les fichiers de configuration, les fichiers *.html* et les fichiers *.class*.
- Répertoire *cgi-bin* où sont contenus les fichiers *Shell script* et les exécutables.
- Répertoire *src* où sont contenus les fichiers *.java*, *.c*, et la documentation en *javadoc*. Les fichiers *.java* sont classés dans les répertoires *util*, *panel* et *menu* selon package.

IV Utilisation

Ce chapitre explique le fonctionnement de l'interface Java, de la connexion au serveur, du lancement de l'applet et de la configuration de l'appareil.

La connexion sur le serveur se réalise par l'adresse :
<http://poire2:8080>

Après avoir sélectionné dans la rubrique instruments et choisi le HP8753D une fenêtre proposée (voir ci dessous):



Figure 7 :Présentation du HP8753D

Nous avons juste mis en place « l'utilisation du pilote et les liens vers la webcam et Hewlett-Packard », lorsque l'on demande d'exécuter le pilote, il apparaît une fenêtre demandant la saisie du nom d'utilisateur suivie de son mot de passe.

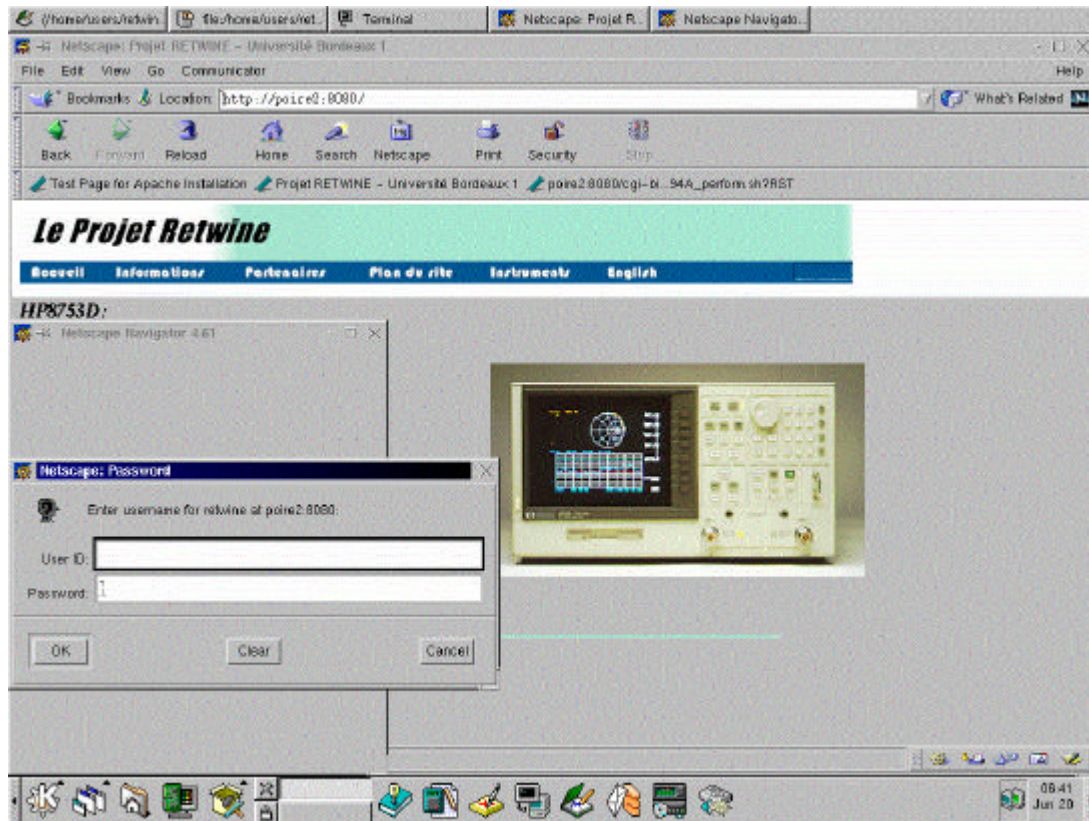


Figure 8 :Demande de mot de passe

Après la saisie du mot de passe valide, la fenêtre ouverte lors de la requête d'utilisation du pilote propose un manuel d'utilisation, l'exécution du pilote, la visualisation et la récupération des mesures par un plotter, ces mesures sont contenues dans un fichier txt.

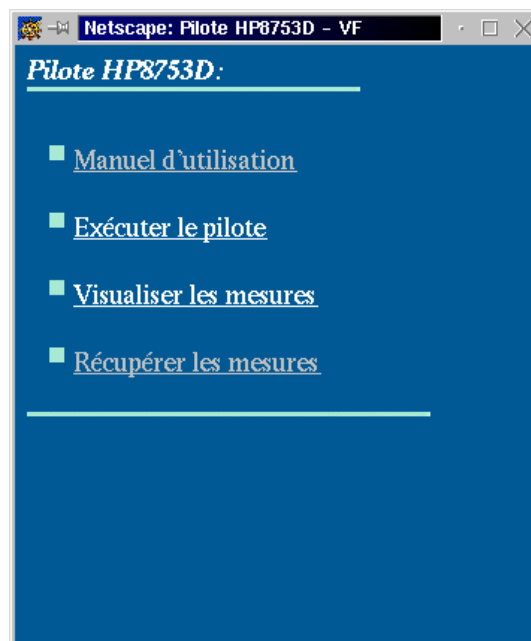


Figure 9 :Pilote HP8753D

Lors du lancement de l'utilisation du pilote, s'ouvrent comme nous l'avons vu précédemment trois fenêtres : l'applet, le menu des softkeys et une fenêtre de status rendant compte de l'état de l'appareil.

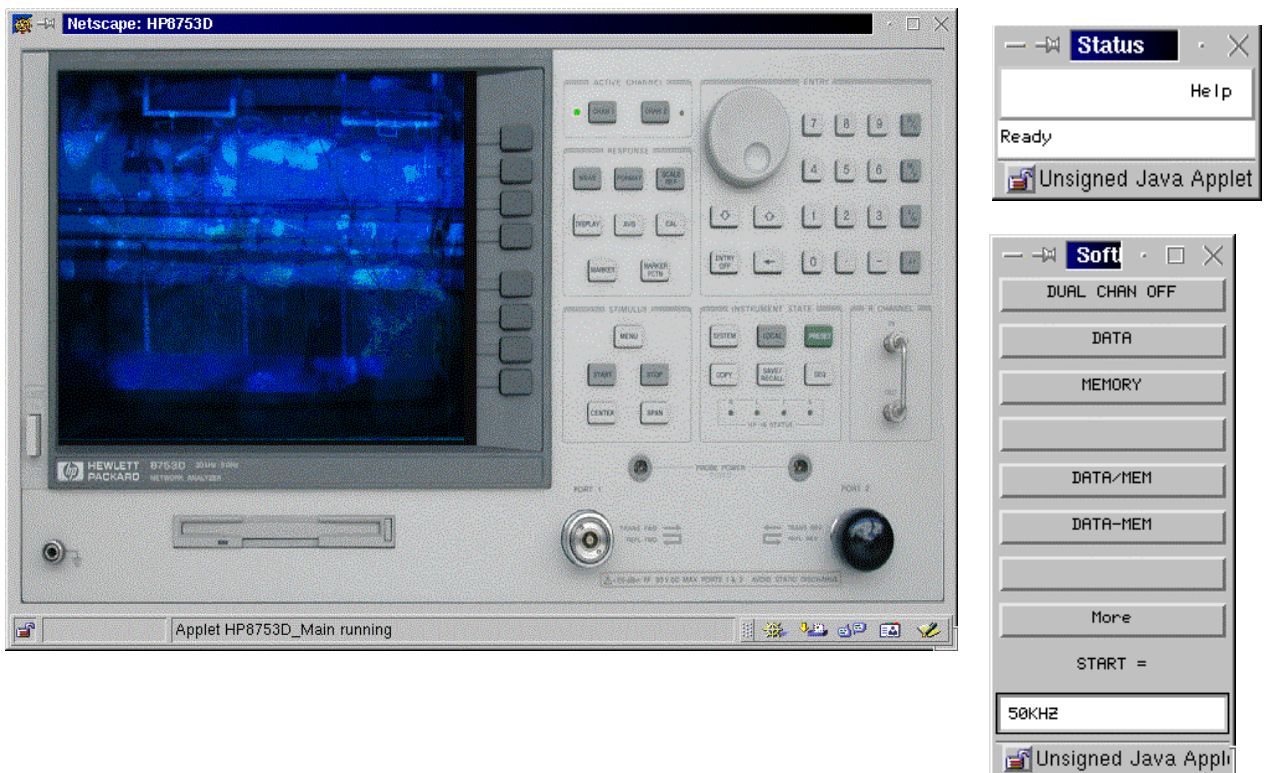


Figure 10 : Environnement d'utilisation

Conclusion

Un instrument supplémentaire est maintenant disponible sur le web à tous les internautes ayant accès au site de Retwine.

Il restera avant d'installer définitivement le HP8753D sur le parc d'instrument à lui ajouter sur le site une documentation technique ainsi qu'un manuel d'utilisation.

Ce projet s'est de toutes les manières révélé être d'un extrême enrichissement. En effet, il est très varié et aborde de nombreux outils informatiques aujourd'hui indispensable à un jeune ingénieur entrant dans la vie active. Ces outils vont des langages de programmation comme le C, le Java, le HTML ou le Shell, à la configuration du serveur Apache, en passant par l'architecture client/serveur ou encore l'apprentissage du Bus GPIB.

Il avait en outre l'avantage d'être un projet « visuel ». En effet, travaillant sur un site web, nous avons le plaisir de voir avancer le projet tout en ayant la possibilité corriger nos erreurs, sans être dépendant d'un débogueur.

Pour terminer, nous sommes conscients d'avoir travailler sur un projet ambitieux et ayant un avenir certain et dont le concept devrait avoir un impact sérieux dans le développement de nouvelles technologies grâce notamment au pilotage à distance.

Abréviations

CGI	Common Gateway Interface
GPIB	General Purpose Instrumentation Bus
HTML	HyperText Markup Language
HTTP	HyperText Markup Transfert Protocol
IP	Internet Protocol
JDK	Java Development Kit
MIME	Multipurpose Internet Mail Extensions
RETWINE	RemoTe Web Instrumentation Network
TCP	Transport Control Protocol

Bibliographie

JAVA 2

Rogers Cadenhead
Editions Le Tout en Poche

Les API de Java

James Gosling et Franck Yellin
Editions International Thomson Publishing

Le Programmeur JAVA 1.2

Laura Lemay et Rogers Cadenhead
Nouvelle Edition du Bestseller International

Apache, Installation et mise en œuvre

Ben Laurie et Peter Laurie
Editions O'REILLY

Webgraphie

Le site de Apache : <http://www.apache.org/>

Références sur Java : <http://java.sun.com/>

Le site de l'IXL : <http://www.ixl.u-bordeaux.fr>

Le site de Retwine : <http://www.apache.org/>

Le site de Hewlett Packard : <http://www.hp.com>

Annexes

Fichiers :

HP8753D_About.java

HP8753D_ComDataClient.java

HP8753D_SoftkeysWindow.java

HP8753D_Main.java

HP8753D_drive.c

HP8753D_perform.sh