Gressier Philippe — Mr Nouel
Dufrêne Pierre — Mr Kadionik

TEST A DISTANCE DE CIRCUITS PROGRAMMABLES

Présentation

Le projet proposé a pour but de réaliser une application permettant d'effectuer le test réel d'une logique implantée dans un circuit programmable. Ce test utilise un analyseur logique piloté par une station de travail, et doit pouvoir être effectué à partir de n'importe quel poste de travail connecté au réseau de l'ENSERB.

Le travail nécessaire comprend 2 parties distinctes :

- Réalisation d'une architecture client/serveur permettant d'utiliser l'analyseur depuis le réseau.
- Réalisation d'un programme qui prépare de façon la plus automatique possible le test, récupère et présente dans un simulateur logique classique les résultats du test.

Une des contraintes majeures réside dans la nécessité de concevoir un programme modulaire, à savoir un programme non dépendant au maximum du matériel et des logiciels utilisés.

Enfin, en ce qui concerne la réalisation pratique de ce projet, il a été choisi d'effectuer la majorité des développements en JAVA, afin de profiter de la souplesse de ce langage et des nombreuses fonctions réseaux déjà implémentées.

L'architecture client/serveur

Le choix de cette architecture a été fondamental dans la mesure où tout le projet est dépendant des choix effectués à ce niveau.

Après étude de diverses possibilités, la solution retenue est la suivante :

- Utilisation du client serveur classique en mode connecté avec des sockets.
- Ecriture d'un serveur minimaliste ayant pour rôle d'envoyer vers le port RS232 de l'analyseur les informations reçues du socket, et de renvoyer vers la socket les données reçues du port RS232.
- Gestion de la file d'attente au niveau du serveur.
- Ecriture d'un client ayant pour tache de générer les données nécessaires au test et récupérer les informations nécessaires à l'affichage du résultat.

Le serveur

Afin de pouvoir communiquer avec l'analyseur, il est nécessaire d'utiliser le port RS232. Cela a nécessité l'utilisation d'un package java supplémentaire (Commapi de Sun).

Ce package offre la possibilité de réaliser une programmation évènementielle, évitant ainsi la réalisation d'un pooling consommateur de ressources systèmes.

La gestion de la file d'attente est effectuée en utilisant un système de sémaphores liés a la ressource port RS232. Dès lors, les multiples threads correspondants aux différents clients attendent de pouvoir accéder à cette ressource chacune à leur tour.

Le serveur est donc relativement simple à mettre en œuvre, il fonctionne dès à présent correctement.

Le client

Il est beaucoup plus complexe à écrire du fait de sa richesse fonctionnelle.

La première étape a été la mise au point des routines de communications sur les sockets avec surtout l'utilisation d'évènements rattachés afin d'économiser les ressources de la machine.

Le traitement des informations reçues est quant à lui effectué au fur et à mesure de la réception afin de ne pas avoir à mémoriser l'ensemble des données reçues par souci d'efficacité.

Les données reçues sont spécifiques à un analyseur donné, c'est pourquoi ce module utilise une interface externe permettant de réaliser différents drivers pour différents analyseurs.

Les méthodes à implanter pour écrire le driver d'un analyseur sont peu nombreuses :

- Une méthode permettant de réaliser l'initialisation de l'analyseur.
- Une méthode réalisant le traitement des données reçues.

Ce travail peut être plus ou moins délicat suivant le type d'analyseur possédé.

Enfin, le dernier rôle du client est de préparer les données nécessaires à la réalisation de la simulation, à savoir la génération des signaux d'entrées.

Avancement du projet

Des premiers tests ont pu être effectués dernièrement : il est dors et déjà possible de récupérer les informations stockées sur l'analyseur et de les afficher.

Il reste encore à écrire les routines permettant de réaliser la génération de signaux.