

# MAP (MANIPULATION D'AUTOMATE PROGRAMMABLE)

**Philippe Gandit<sup>1</sup>, Alain Benoît<sup>1</sup>, Paul Lotrus<sup>2</sup>**

<sup>1</sup>*Institut Néel – CNRS/UJF  
25 avenue des Martyrs, BP 166,  
38042 Grenoble Cedex 9*

*philippe.gandit@grenoble.cnrs.fr*

<sup>2</sup>*CEA SACLAY / DSM/IRFU/SIS*

*Laboratoire de Développement en Informatique Industrielle  
91191 Gif sur Yvette Cedex 3*

**Résumé :** Le programme MAP pour “ Manipulation d’Automate Programmable “, entièrement écrit en langage de programmation graphique LabVIEW, permet de réaliser des ensembles contrôle/commande sans aucune programmation. Les automates utilisés sont des modules FieldPoint ou CompactFieldPoint de la société National Instruments. Ce programme se décompose en deux modules. Le premier, appelé Serveur, est un programme qui s’exécute au démarrage du contrôleur de l’automate. Le second, appelé Client, permet de réaliser la supervision ; c’est soit un instrument virtuel, soit un exécutable.

**Mots clés :** Fieldpoint, Compact FieldPoint, Client, Serveur, TCP/IP

## 1 CONTEXTE

EDELWEISS II (<http://edelweiss2.in2p3.fr/index.php>) est une collaboration internationale de cosmologie. Le but de cette expérience est de mettre en évidence l’existence des WIMPs (Weakly Interacting Massive Particule). Ces particules expliqueraient une partie de l’énergie manquante dans l’univers. Ces particules interagissant faiblement avec les détecteurs, il est très difficile de les mettre en évidence. L’énergie mise en jeu lors de la collision d’une particule avec un détecteur est si faible qu’il est nécessaire de le refroidir au voisinage du zéro absolu (-273.15 °C), afin que le signal ne soit pas noyé dans le bruit de fond thermique. Lors d’une collision, le détecteur (cristal de germanium de 300 grammes) doit s’échauffer de quelques microdegrés et s’ioniser. La détection est donc double, les événements doivent être concomitants. Afin d’augmenter la probabilité de détection, l’expérience à terme contiendra une centaine de détecteurs soit 30 kg de Germanium. Par ailleurs, afin de s’affranchir au maximum des rayonnements parasites (rayons cosmiques, neutrons, radioactivité naturelle), l’expérience a été installée dans le Laboratoire Souterrain de Modane (LSM) qui se situe au milieu du tunnel du Fréjus.

Les physiciens, étant surtout préoccupés par la collecte et l’analyse des données, désiraient un système de contrôle-commande du cryostat simple, robuste et très facilement configurable ou modifiable à volonté sans l’intervention d’informaticiens spécialisés. Compte tenu d’une première expérience à plus petite échelle (EDELWEISS I), les automates programmables Fieldpoint de National Instruments ont été retenus, mais à condition que soit écrit un programme spécifique qui permette une utilisation simple et fiable. C’est dans cette optique

que le programme MAP a été développé. Il permet de mettre en œuvre les automates, sans être un spécialiste des automates et du langage de programmation. De plus, l'utilisateur peut à sa guise dessiner son écran de supervision.

## 2 CAHIER DES CHARGES

Pour contrôler l'expérience (hors acquisition des données) trois automates programmables (Fieldpoint et CompactFieldpoint) avec chacun quatre modules entrées/sorties (soit environ 200 paramètres) sont utilisés. Le premier contrôle le refroidissement jusqu'à 1 K (-272 °C). Le second pilote le réfrigérateur à dilution d'hélium 3 dans l'hélium 4 qui permet de refroidir les détecteurs. Enfin, le troisième surveille l'environnement du cryostat (température et humidité du hall, niveau des fluides cryogéniques, climatisation, refroidissement des pompes). Le programme doit :

- assurer la mesure de tous les capteurs et effectuer les conversions associées,
- permettre la commande des vannes, pompes et de tous les dispositifs,
- transmettre les données et recevoir des commandes des écrans de supervision,
- assurer un fonctionnement autonome même sans supervision,
- savoir prendre des décisions en cas d'incident,
- permettre aux utilisateurs de créer les écrans de contrôle sans programmation,
- permettre une intervention sur les entrées/sorties (ajout) sans arrêt du programme,
- rendre possible l'utilisation de plusieurs supervisions sur un même ordinateur.

Pour assurer la robustesse du système, il a été décidé de scinder le programme en deux parties. La première est un serveur qui est un exécutable, écrit en LabVIEW résidant dans le module Fieldpoint et qui se lance automatiquement à la mise sous tension. La seconde est un client, également entièrement écrit avec LabVIEW, qui permet sur un ordinateur de contrôler de superviser l'expérience. La communication entre le serveur et les clients se fait au travers du protocole TCP/IP avec des trames ASCII, ce qui permet éventuellement à l'utilisateur de créer son propre client. Le serveur possède une machine d'état qui permet à l'automate de se mettre dans un état prédéfini (valeur de certaines variables). A partir de tests, il est possible de changer automatiquement d'état.

## 3 LE SERVEUR

Ce programme, après avoir recensé les modules Fieldpoint présents et lu deux fichiers "texte" de configurations (config.txt et etat.txt), tourne en boucle (voir *figure 1*). Les objets définis par l'utilisateur, par exemple une entrée analogique ou un PID, sont rassemblés dans un tableau de cluster. Ce tableau est passé en paramètre à tous les sous-programmes (Vi) du diagramme. Ce cluster contient en particulier, le nom de la variable, son index, sa valeur ainsi qu'un drapeau qui caractérise son état. A chaque itération :

- tous les drapeaux sont remis à zéro,
- tous les objets de type "entrée" sont lus,
- les objets nécessitant un calcul sont analysés (sécurité, PID, compteur, ...),
- la machine d'état est exécutée,
- les ordres reçus des clients sont analysés,
- tous les objets de type "sortie" sont écrits,

- la valeur de tous les objets est envoyée aux clients, les nouveaux clients sont abonnés,
- enfin, une LED sur le contrôleur est activée ou désactivée pour visualiser l'activité et une autre signale la présence d'une erreur (drapeau négatif).

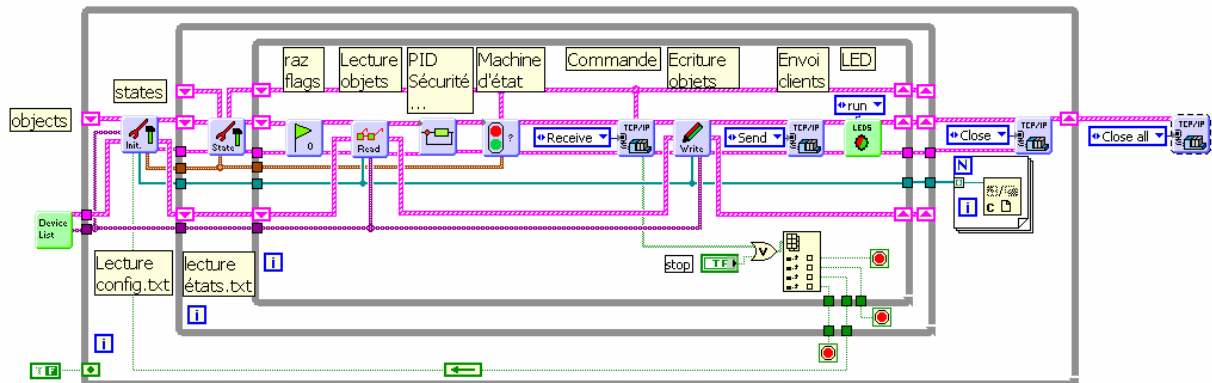


Figure 1 : diagramme principal du serveur.

### 3.1 Le fichier de configuration

C'est un fichier texte qui contient des paramètres nécessaires au programme comme le numéro du port de communication, le nom de l'automate, la liste des modules présents (générée automatiquement par le programme). Il est possible dans ce fichier de commenter des lignes ou des ensembles de lignes ainsi que de définir des raccourcis mnémoniques. Seize types d'objets ont été prédéfinis :

- entrée/sortie analogique avec leur échelle,
- entrée/sortie numérique avec un ou deux retours possibles,
- PID, sécurité, hystérésis,
- constante, variable, compteur, valeur de la machine d'état,
- temps, temps dans l'état.

Les objets sont décrits par une ligne dont le nombre de paramètres dépend du type d'objet :

<Nom>,<type>,<valeur par défaut>,<para1>,<para2>,<para3>,.....

Les paramètres peuvent être des objets ou des expressions mathématiques. Par exemple : pour un PID, le gain peut être un objet de type constante donc modifiable ; l'échelle d'un capteur peut être définie par une loi ce qui permet une lecture directe de la grandeur physique associée. De plus il est possible de définir un objet comme étant le résultat d'un Vi externe au programme. Il suffit pour cela qu'il soit simple (sans sous-Vi) et possède un connecteur prédéfini. Il est ainsi possible par exemple de lire et de commander un appareil relié sur le port série du contrôleur. Un automate peut également lire et écrire des objets sur un autre automate en s'abonnant comme le ferait un client (dans ce cas, seuls les objets concernés sont transmis à chaque itération du serveur).

### 3.2 Le fichier d'état

Ce fichier texte décrit l'ensemble des états. Chaque état possède un nom et un ensemble d'attribution de valeurs de sortie. Exemple :

\$état1  
output1=1  
output2=1

```
$état2
output1=0
output3=if (input1 < var1) then var3 else var4
```

```
if input3>10 then goto état2          goto manual
```

Il est ainsi possible sur un test de changer la valeur des objets mais aussi de changer d'état. Un état "manual" permet de désactiver la machine d'état.

### 3.3 Communication

Les clients communiquent avec le serveur au travers d'une liaison TCP/IP. Pour s'abonner, le client doit simplement établir une connexion avec le serveur sur le port spécifié dans le fichier de configuration. A chaque itération, le client reçoit l'ensemble des valeurs et drapeaux des objets. Il est possible d'envoyer une requête à l'automate comme : la version du programme, la liste des objets, la liste des états, le nom de l'automate, le nom des clients connectés, la relecture des fichiers de configuration et d'états et, bien entendu, la modification de la valeur d'un objet.

Les trames TCP/IP sont constituées d'une entête qui identifie la requête et d'un nombre qui détermine la longueur de la chaîne transmise, pour éviter les erreurs.

## 4 LE CLIENT

Le client permet de récolter la valeur de tous les objets, de les éditer et d'enregistrer un historique. De plus, l'utilisateur doit pouvoir éditer son panneau de supervision sans aucune programmation. Au lancement du client, un dialogue permet de choisir l'automate sur lequel on veut se connecter. Lors de la première connexion à un automate, un dossier nommé par l'utilisateur, est créé. Il contient le panneau vierge de ce nouvel automate, ainsi que des fichiers de configurations. Pour s'abonner à un automate, il suffit d'établir une connexion TCP/IP avec l'automate, de demander la liste des objets et la liste des états. Une fois abonné, le client reçoit à chaque itération du serveur l'ensemble des valeurs des objets.

### 4.1 Ecran de contrôle

Pour éditer facilement le panneau de contrôle, nous avons utilisé la possibilité offerte par LabVIEW d'utiliser une face avant secondaire (voir *figure 2*). L'écran de contrôle (panel.vi) contient une face avant secondaire constitué de la barre d'outil qui est en fait le Vi (client.vi) qui est le programme principal. Le diagramme de panel.vi est constitué essentiellement d'une boucle "while" qui ne fait rien. Lorsque client.vi est exécuté, il lance l'exécution de panel.vi qui à son tour incorpore client.vi sur sa face avant. Pour éditer la face avant de panel.vi, il suffit d'appuyer sur le bouton "Edit". La face avant secondaire est retirée et panel.vi est arrêté. L'utilisateur peut alors créer sa face avant en utilisant tous les indicateurs qu'il souhaite **sans les câbler dans le diagramme**. La relâche du bouton "Edit" relance panel.vi qui incorpore à nouveau la barre d'outil (client.vi). Les indicateurs n'étant pas connectés, pour agir dessus, le programme principal ayant fait la liste de tous les indicateurs de panel.vi est capable de générer un événement suite à un clic de souris sur un indicateur. Si c'est un clic droit, un dialogue propose de l'associer à un objet. Si c'est un clic gauche, un autre dialogue dépendant du type d'objet associé (valeur analogique, booléenne, état) s'affiche et permet de changer la valeur de l'objet. Ce dialogue est également un élément de sécurité, afin d'éviter le changement d'état d'un booléen sur un simple clic.

Suivant la valeur du drapeau, l'indicateur associé à l'objet est grisé (objet contrôlé par la machine d'état, un PID, une sécurité...) ou clignote dans le cas d'une erreur.

### 4.2 Barre d'outils

L'écran de supervision est donc surmonté par un bandeau (voir *figure 2*) qui permet de contrôler l'état de la connexion (chenillard), l'enregistrement des points, la présence d'un objet avec une erreur. De plus il est muni de boutons dont les actions sont décrites ci-après.



Figure 2 : extrait d'un panneau de contrôle

#### 4.2.1 Graphiques

Le bouton “Graph” permet d’ouvrir une fenêtre (voir figure 3) contenant un graphique représentant l’évolution temporelle d’un ou plusieurs objets (choix à partir d’une fenêtre d’options). On peut représenter les données issues directement des mesures ou à partir des fichiers de sauvegarde. Dans ce cas, il est possible de mettre bout à bout plusieurs fichiers.

Il est possible d’ouvrir plusieurs graphes simultanément. Pour cela, le Vi générant ces graphes est en fait dupliqué autant de fois que nécessaire. Lorsque le programme client est quitté, tous les paramètres (voies, styles, dimensions et position des fenêtres) sont sauvegardés. Ainsi, lorsque le programme est relancé tous les graphes définis s’ouvrent automatiquement.

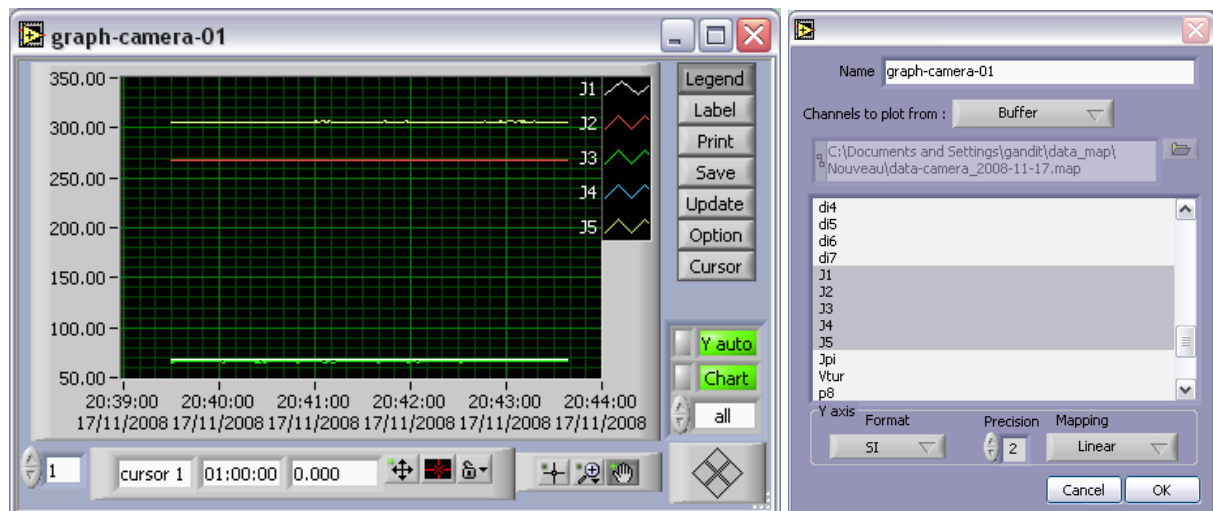


Figure 3 : la fenêtre graphique et ses options

#### 4.2.2 Outils

Le bouton “Tools” permet de définir certaines préférences du panneau de contrôle comme le format (texte ou binaire), le chemin et la périodicité de la sauvegarde de l’historique. Il permet également de changer d’automate, de l’arrêter ou de le rebooter. Il est possible d’obtenir la version du programme qui s’exécute sur l’automate et la liste des clients connectés.

#### 4.2.3 Valeurs

Le bouton “values” permet d’ouvrir une fenêtre qui liste l’ensemble des objets avec leur nom, drapeau et valeur. En cliquant sur la ligne correspondant à un objet, il est possible de modifier sa valeur via un champ qui dépend de la nature de l’objet (état, valeur analogique ou

numérique). Cette fenêtre permet de contrôler les objets qui ne seraient pas représentés sur l'écran de supervision.

#### 4.2.4 Configurations

Enfin le bouton "Config" permet d'éditer les fichiers de configuration et d'état. Des aides sont disponibles pour faciliter la saisie de ces fichiers. Le programme récupère et envoie les fichiers au travers d'un protocole FTP. Les fichiers sont également sauvegardés sur le disque dur de l'ordinateur. Une vérification de la syntaxe est opérée avant l'envoi des fichiers.

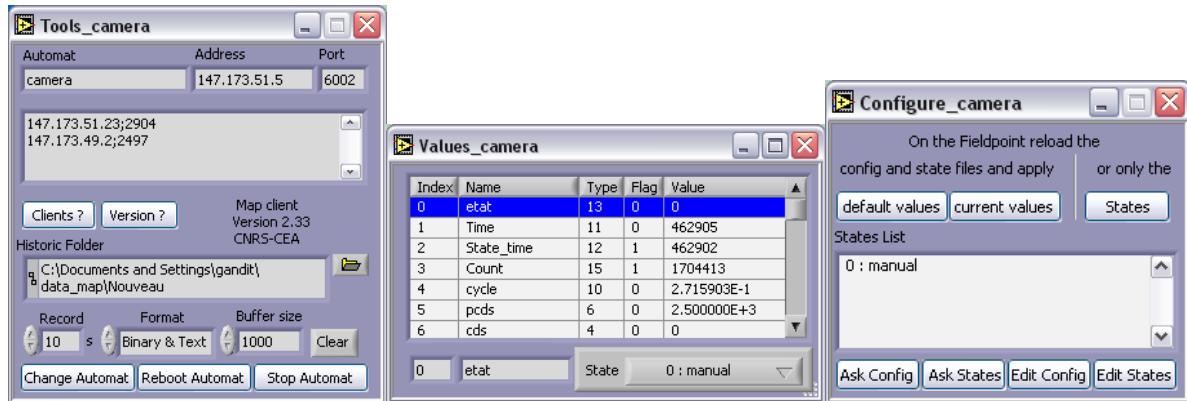


Figure 4 : les trois fenêtres Outils, Valeurs et Configurations

### 4.3 Exécutable

Pour pouvoir à partir d'un même client dialoguer avec plusieurs automates, nous avons décidé de créer un exécutable par automate. Pour chaque automate, un exécutable est dupliqué lors de la création du dossier. En exécutant ces programmes qui ouvrent automatiquement les panneaux de contrôle situés dans les dossiers correspondants, il est ainsi possible de piloter sur le même ordinateur, autant d'automates que de dossiers. Dans ce cas, la seule différence est l'absence du bouton "Edit" qui permet de modifier la face avant.

## 5 CONCLUSION

À ce jour, le programme est totalement opérationnel, il assure sans souci le contrôle de l'expérience EDELWEISS II. Si un physicien décide d'ajouter une entrée ou une sortie sur l'automate, il la câble physiquement, puis depuis son client, il édite et modifie le fichier de configuration et demande au serveur de relire le fichier de configuration. Il ne lui reste qu'à ajouter un objet sur son écran de supervision, l'associer à cette nouvelle entrée pour que son interface soit opérationnelle. Sa simplicité et sa robustesse sont très appréciées des utilisateurs qui peuvent surveiller l'expérience de n'importe quel endroit pour autant qu'une liaison TCP/IP soit possible. Il permet en quelques heures de configurer complètement un ensemble Fieldpoint (ou Compact), il fonctionne avec les modules entrées ou sorties numériques et analogiques ainsi que les modules relais.

L'évolution des matériels étant très rapide, il est possible d'envisager maintenant la sauvegarde d'un historique des mesures sur le module contrôleur directement. De même l'aspect sécurité n'a pas été traité. En effet, ce programme devant fonctionner dans des laboratoires de recherche derrière des tunnels VPN, il n'a pas été demandé d'assurer la confidentialité des données. Cependant, il est parfaitement envisageable de crypter les données lors des transmissions.