LabVIEW est un environnement de développement spécialisé en informatique industrielle et scientifique. Sa particularité est qu'il s'appuie sur le langage G, créé par National Instruments, qui est entièrement graphique. Il permet de créer des logiciels complexes tout en facilitant la programmation et donc de diminuer les délais de développement. Grâce à ses librairies de fonctions dédiées à l'acquisition de données, l'instrumentation, à l'analyse mathématique des mesures, mais également grâce à la création rapide d'interfaces graphiques de qualité et le codage simplifié, l'ingénieur a plus de temps pour se concentrer sur les fonctions métiers de l'instrumentation et du traitement des mesures.

LabVIEW est particulièrement recommandé pour développer des systèmes de contrôle, supervision et les bancs de test et mesure.

LabVIEW pourquoi faire ?

Comme nous l'avons vu en introduction, LabVIEW est très approprié à l'informatique industrielle et scientifique. Vous pourrez donc l'utiliser pour le développement de :

- Logiciels pour Windows, UNIX/Linux ou Mac,
- Des librairies (DLL, Active X, .NET),
- Drivers d'instruments,
- Cibles embarqués,
- Temps réel,
- FPGA,
- Logiciels pour Pocket PC et Windows Mobile,
- Logiciels Palm OS.

Exemples de réalisations avec LabVIEW

Pour voir une idée des applications possibles de LabVIEW pour la réalisation et le développement de projets, cliquez sur les liens suivants :

- Inspection visuelle de blocs fusibles (Vision, industrie automobile)
- Armoire d'analyse de gaz (instrumentation, industrie pétrochimique),
- Mallette météo sous Windows Mobile (application militaire),
- Superviseur de contrôle de la qualité du sable (communication avec des PLC ,

matériaux de construction),

- Système de validation d'Electronic Control Module (ECM) (industrie automobile),
- Liste des projets réalisés avec LabVIEW auxquels j'ai participé (descriptions rapides).

LabVIEW et le matériel

LabVIEW permet de programmer sur bien des cibles différentes (voir paragraphe précédant). Il en va de même pour le matériel et surtout l'instrumentation. En effet, avec LabVIEW et grâce à ses nombreuses librairies, vous pourrez vous interfacer et commander les cartes et appareils de type suivants :

- VXI , PXI , Compact PCI,
- PCI, PCI express, PXI express,
- USB, FireWire,
- GPIB , CAN
- RS 232, 422, 485...
- TCP/IP,
- Bluetooth , WIFI,
- Automates programmables (PLC), modbus, profibus...

Le langage Graphique

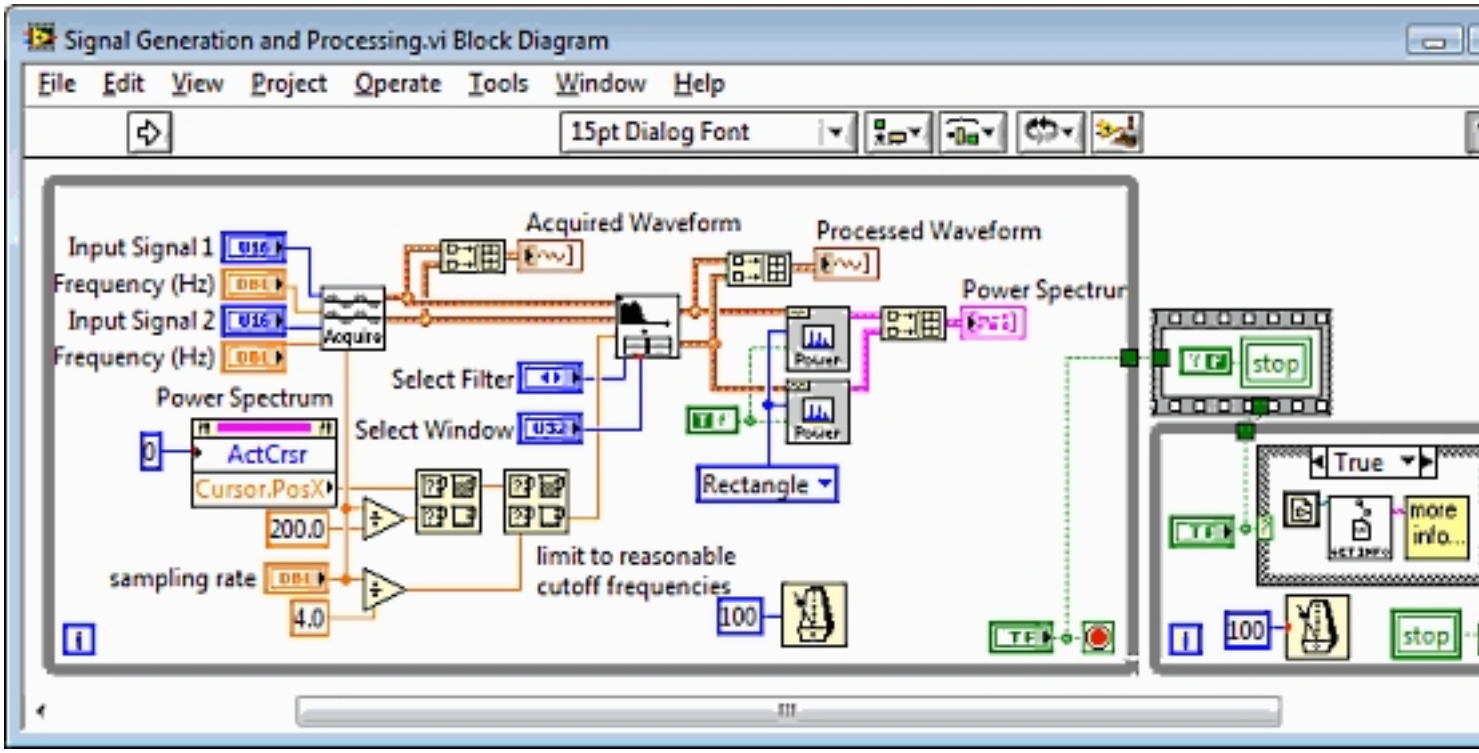
Avec LabVIEW , on ne programme pas en écrivant des lignes de code à la syntaxe complexe. La programmation est effectuée à l'aide d'icône, représentant des fonctions, reliés entre eux par des câbles qui représentent les flux de données (un peu à la manière d'une carte électronique avec ses composants et circuits intégrés).

Cette représentation très imagée du code source est proche de la conception telle qu'on peut la faire : avec des schémas; ceci, on le comprend, facilite beaucoup le travail que nécessite le codage du concept dans le programme. Cette abstraction du langage Graphique ne requiert pas d'être un expert en programmation pour développer des applications simples. De même, pour des applications plus complexe, le maître d'eouvre, "profane" en programmation mais expert métier, pourra lire et saisir l'idée et ainsi mieux guider le programmeur, expert technique.

Pour illustrer la simplicité du langage G, l'exemple ci-dessous présente un logiciel de génération de signal et de traitement de celui-ci. Regardez la vidéo puis le code source qui produit cette application.

{flowplayer}http://www.ajolly.fr/images/stories/ LabVIEW /signal-generation-and-processing.flv{/flowplayer}

Vidéo de l'application



Code source de l'application

Alternatives à LabVIEW

Pour bénéficier des mêmes fonctionnalités que LabVIEW pour d'autres langages, référez vous aux articles suivants :

- langage C, lisez l'article " LabWindows/CVI ",
- langage C++, C# ou Visual Basic sous Visual Studio, lisez l'article " Measurement Studio "

,

- Pour les banc de test, on utilise souvent LabVIEW avec un séquenceur de test comme TestStand

. Cette combinaison permet de développer les composants technique sous
LabVIEW
et de les intégrer dans le séquenceur
TestStand
. L'architecture reste alors la même, les "briques"
LabVIEW
permettent de spécialiser les séquences selon de produit à tester.