



## Interface *SignalRanger* DDCI

La pièce maîtresse de l'architecture *SignalRanger* est son interface DDCI. DDCI (*Development to Deployment Code Instrumentation*) permet à une application de contrôle exécutée sur un PC de communiquer avec un périphérique embarqué basé sur une carte *SignalRanger* et de le contrôler.

L'interface en vigueur offre une visibilité et un contrôle en temps réel du code s'exécutant dans le dispositif intégré. Son utilité se situe à deux étapes du cycle de vie de l'application :

- Au cours du développement, il est utilisé pour fournir un débogage en temps réel à un niveau d'application qui n'est généralement pas réalisable à l'aide des techniques de débogage et d'émulation standard. En particulier, les fonctions de lecture, d'écriture et de contrôle de code ne nécessitent pas d'arrêt du processeur. L'interface permet d'instrumenter le code en temps réel et dans les conditions réelles de fonctionnement.
- Après le déploiement de l'application, lorsque la même interface est utilisée pour prendre en charge le contrôle de l'utilisateur et les communications avec le dispositif intégré via une application PC spécifique à l'application développée à cet effet.

Les trois caractéristiques essentielles de l'interface DDCI sont les suivantes :

- La même interface USB physique et les mêmes bibliothèques et fonctions hôtes sont utilisées pour prendre en charge l'interface aux deux étapes du cycle de vie de l'application.
- Le fonctionnement de l'interface DDCI ne nécessite aucun ajout ou adaptation de code DSP, et ne nécessite qu'un temps processeur ou une surcharge de mémoire minimales.
- L'interface USB physique peut être connectée et déconnectée en cours de fonctionnement, sans aucune interruption du code DSP fonctionnant sur la plate-forme *SignalRanger*

Le principal résultat de l'utilisation de l'interface est de condenser les deux étapes du cycle de vie de l'application en une seule étape plus courte. En effet, l'application est généralement déployée « en l'état » juste après la phase de débogage.

Un autre avantage important de l'utilisation de l'interface est qu'elle fournit, avec très peu d'efforts, une visibilité en temps réel beaucoup plus grande sur le fonctionnement du code embarqué. Cette plus grande visibilité pendant le développement se traduit directement par un code embarqué plus fiable.

Dans de nombreuses applications où il est nécessaire d'exécuter des simulations de l'algorithme de traitement du signal pour valider son fonctionnement, l'instrumentation temps réel fournie par DDCI permet d'analyser le comportement de haut niveau du code de traitement du signal avec facilité, et avec des conditions et des données réelles. Souvent, l'étape de simulation peut être complètement contournée, avec de meilleurs résultats basés sur des données réelles.

Lors de l'utilisation de l'interface LabVIEW, par opposition à l'interface C/C++, un troisième avantage de l'interface est que les bibliothèques LabVIEW étendues sont disponibles pour ajouter de puissantes capacités de traitement du signal, d'analyse et d'affichage en temps réel, à la fois lors des phases de débogage et de déploiement de l'application.

Les fonctions de l'interface peuvent être exercées pendant l'exécution du code intégré. Toutes ces fonctions prennent en charge l'accès symbolique, où le nom des variables et des fonctions DSP peut être utilisé pour y accéder, au lieu de leurs adresses absolues. L'avantage de cette fonctionnalité est que le code DSP intégré peut être modifié et relié sans avoir à mettre à jour l'application d'accès utilisateur associée exécutée sur un PC. Tant que les noms de variable et de fonction restent identiques, l'accès reste opérationnel malgré les mises à jour du code DSP.

Les fonctions en temps réel suivantes sont prises en charge directement par l'interface :

- Lecture et écriture de la RAM
- Lecture et écriture Flash
- Lecture et écriture de périphériques
- Forcer l'exécution du code
- Réinitialisation du processeur
- Gestion de la mise à niveau du firmware en service
- Reconnaissance et gestion automatiques de l'appareil cible.

L'interface est composée de plusieurs parties :

- **Pilote signé pour Windows XP, Windows Vista et Win 7 (plates-formes x86 et x64)** : Ce pilote permet de connecter n'importe quel nombre de cartes au PC. Le pilote est installé dans le cadre de la procédure d'installation de *SignalRanger*.
- **Noyau de communication** : ce noyau réside dans la mémoire DSP, avec le code spécifique à l'application de l'utilisateur. Il améliore la communication entre PC et DSP.
- **Bibliothèques LabVIEW** : ces bibliothèques prennent en charge un large éventail de fonctions de communication, de programmation et de contrôle avec le noyau résident. Ils peuvent être utilisés pour construire un exécutable LabVIEW spécifique à une application afin de contrôler le périphérique embarqué *SignalRanger*.
- **Bibliothèques C/C++** : pour les développeurs qui préfèrent travailler dans un environnement C/C++, nous fournissons des bibliothèques sous forme de DLL. Ces bibliothèques ont des fonctionnalités similaires à celles des bibliothèques LabVIEW.
- **Mini-débogueur** : le mini-débogueur est une application d'interface à usage général qui prend en charge la programmation et le débogage symbolique en temps réel du code intégré DSP générique. Il comprend des fonctionnalités telles que le traçage graphique des données en temps réel, l'accès symbolique en lecture/écriture aux variables, l'exécution dynamique, la programmation Flash... etc. À la base, le mini-débogueur utilise les mêmes bibliothèques d'interface qu'un développeur utilise pour concevoir une application DSP autonome. Cela garantit une transition transparente de l'environnement de développement/débogage à l'application déployée.