

2012

Pratique : Rendez vos objets  
communicants et interactifs  
avec des Phidgets



Partie 2

S. Lavirotte, O. Carrillo

Université de Nice – Sophia Antipolis

03/01/2012

# Rendez vos objets communicants et interactifs avec des Phidgets

## 1 Phidget 1072 (SBC2)

Vous êtes en possession maintenant d'une plate-forme Phidgets 1072 qui dispose d'un processeur embarqué de type ARM. Nous allons configurer et déployer sur cette plate-forme (qui fonctionne sous Linux) les paquetages nécessaires pour faire fonctionner l'environnement .NET (mono) nécessaire à WComp.

### 1.1 Installation des applications Phidget sur votre PC

Pour avoir accès à la plate-forme Phidget, vous devez disposer sur votre ordinateur des drivers :

[http://kistren.polytech.unice.fr/cours/oc/td7/Phidget-x86\\_2.1.8.20111220.exe](http://kistren.polytech.unice.fr/cours/oc/td7/Phidget-x86_2.1.8.20111220.exe)

ou

[http://kistren.polytech.unice.fr/cours/oc/td7/Phidget-x64\\_2.1.8.20111220.exe](http://kistren.polytech.unice.fr/cours/oc/td7/Phidget-x64_2.1.8.20111220.exe)

Une fois ces application installées, vous pouvez connecter votre plate-forme Phidget 1072 via le câble Ethernet fourni. Au bout de quelques secondes, l'application détectera votre plate-forme (visible dans l'onglet Phidget SBC, voir Figure 2).

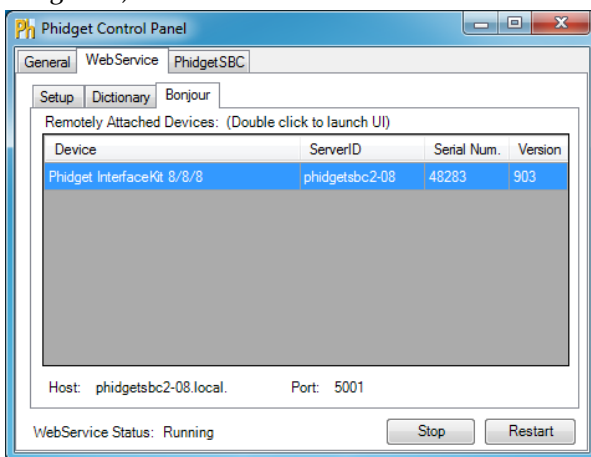


Figure 1: Web Service accès Phidget IK

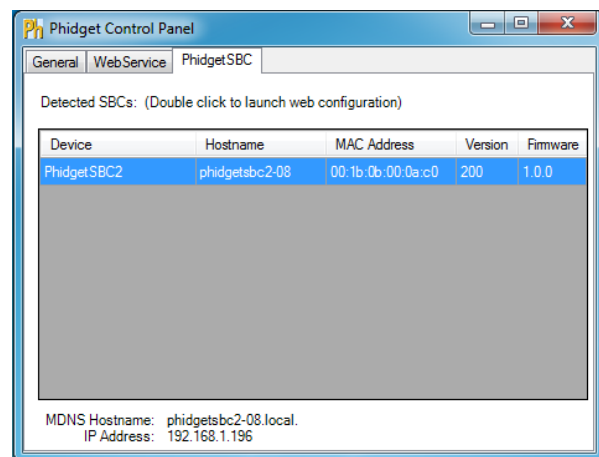


Figure 2: Phidget SBC accès plateforme

## 1.2 Configuration et installation des outils de la plate-forme

### 1.2.1 Configuration système et réseau de la plateforme Phidget 1072

Avant de pouvoir utiliser la plateforme en Wifi, nous devons la configurer via le câble Ethernet. En double cliquant sur la ligne correspondant à votre équipement dans l'onglet Phidget SBC du Phidget Control Panel, vous vous connecterez à l'interface http de la plate-forme à l'aide de votre navigateur préféré.

Dans le cas d'une première utilisation de la plateforme, vous arriverez sur une page vous demandant un nouveau mot de passe (l'utilisateur étant admin). Donner `admin` comme mot de passe.

Les identifiants de connexion doivent être `admin / admin` (**ne changez pas ce mot de passe par défaut !**). Vous êtes maintenant connectés à la plateforme prêt à la configurer.

Vous devez configurer le système en choisissant le nom de l'hôte et la timezone

- *System / General*, remplir le *Hostname* et choisir la *Timezone* puis *Save Changes*

Puis, nous allons activer le serveur ssh pour éviter d'utiliser uniquement l'interface graphique :

- *Network / Settings*, *SSH server Enabled* puis *Save Changes*

Enfin, nous allons activer le(s) réseaux sans fil pour autoriser la plateforme à se connecter en Wifi. Après avoir branché la clé USB WiFi à la plateforme (et éventuellement redémarré pour sa prise en compte), vous devez configurer l'interface réseau :

## Rendez vos objets communicants et interactifs avec des Phidgets

- *Network / Wireless*, choisir le SSID du réseau auquel se connecter dans *Add a Wireless Network*, entrez le *Password* si nécessaire puis *Add This Network*. Une fois le(s) réseau(x) ajouté(s), dans la section *Saved Networks*, choisissez celui auquel vous souhaitez vous connecter puis activez *Join The Network*.

Voilà maintenant la configuration réseau sans fil de votre plate-forme effectuée. Il ne vous reste plus qu'à débrancher le câble Ethernet et à redémarrer votre plateforme. Celle doit être maintenant détectée avec une adresse sur le réseau WiFi.

Dans le cas plus que probable où avec le Phidget SBC (Figure 2) vous détectiez plus d'une plateforme, vous pourrez reconnaître la votre à son adresse MAC (dont le numéro est inscrit au dos de la carte). Ceci vous permettra de faire les réglage sur votre plateforme et non celle du voisin !

**Attention : l'adresse MAC est l'adresse de l'interface filaire et non celle de l'interface Wifi de votre carte.** Mais peu importe dans notre cas, c'est juste pour la différencier.

### 1.2.2 Installation et mise à jour de paquetages

Vous devez être connecté à Internet pour la mise à jour des paquetages. Vous pouvez gérer les paquetages soit via l'interface Web ou bien via la console ; nous allons faire les deux.

Connectez vous sur la plateforme via ssh avec votre client favori (PuTTY par exemple). La connexion doit être faite avec les identifiants suivants : root / admin.

Faites la mise à jour du système : `apt-get update ; apt-get upgrade` (cette mise à jour peut aussi être faite via l'interface Web : *Refresh Available Packages* puis *Upgrade All Packages*)

Reconnecter vous déjà via l'interface Web, et configurez de la manière suivante :

- Dans *System / Packages*, commencez par activer l'option *Include Full Debian Repository*
- Vérifiez que les paquetages *Java Support* et *C/C++ Development Tools/Headers* sont bien *Installed*

### 1.2.3 Ajout de WComp sur la plateforme

Pour bénéficier de WComp sur la plateforme, il nous faut tout d'abord installer l'environnement .NET.

```
apt-get install mono-runtime mono-gmcs libmono-system-web2.0-cil
```

Puis, nous allons ajouter la librairie Phidget dans le gestionnaire d'assembly.

```
apt-get install wget unzip
wget http://www.phidgets.com/downloads/libraries/Phidget21-windevel_2.1.8.20111220.zip
unzip -p Phidget21-windevel_2.1.8.20111220.zip phidget21-windevel/Phidget21.NET.dll >
./Phidget21.NET.dll
gacutil -i Phidget21.NET.dll
ln -s /usr/lib/mono/gac/Phidget21.NET/*/* /usr/lib/mono/2.0/Phidget21.NET.dll
```

Enfin, il ne vous reste plus qu'à déployer le package contenant WComp à l'endroit qui vous conviendra le mieux.

## 2 Embarquez votre assemblage dans votre objet communicant

### 2.1 Portage de votre prototypage sur la plate-forme Phidgets embarquée

Maintenant que votre prototypage a été réalisé lors du TD précédent sur phidget et que votre plate-forme Phidget 1072 est configurée, vous pouvez faire fonctionner votre travail d'objet communicant dans une version autonome.

A l'aide de l'application Phidget (zone de notification) vous pouvez trouver l'adresse IP de votre plate-forme (onglet Phidget SBC). Puis avec WinSCP configuré avec l'adresse IP trouvée, vous copierez les fichiers nécessaires :

## Rendez vos objets communicants et interactifs avec des Phidgets

- votre assemblage .wcc que vous venez de réaliser ou le fichier `SensorToLedsSLCA.wcc` fournit en exemple, à copier dans le dossier `wcomp`
- les dll nécessaires à votre assemblage dans le dossier `wcomp/beans` (les dll de bases sont déjà copiés vous ne devriez donc avoir qu'à copier la dll correspondant au composant Blink que vous avez réalisé dans la partie précédente par exemple).

Après vous être connecté via putty (accessible depuis WinSCP), vous vous rendez dans le dossier `/root/wcomp` afin de lancer la commande suivante pour exécuter le container `WComp` :

```
mono Container.exe -r beans -l fichier.wcc -n nomDuDispositifUPnP
```

### 3 Utiliser votre objet communicant depuis votre PC

Maintenant que nous avons un objet communicant autonome, nous pouvons utiliser celui-ci depuis notre PC pour construire une chaîne de service lié à celui-ci. Nous allons tout d'abord

#### 3.1 Génération d'un composant proxy pour un service UPnP

Nous souhaitons pouvoir accéder dans `WComp` aux dispositifs de type UPnP. Pour cela nous allons devoir générer un composant proxy pour le service UPnP de votre objet communicant.

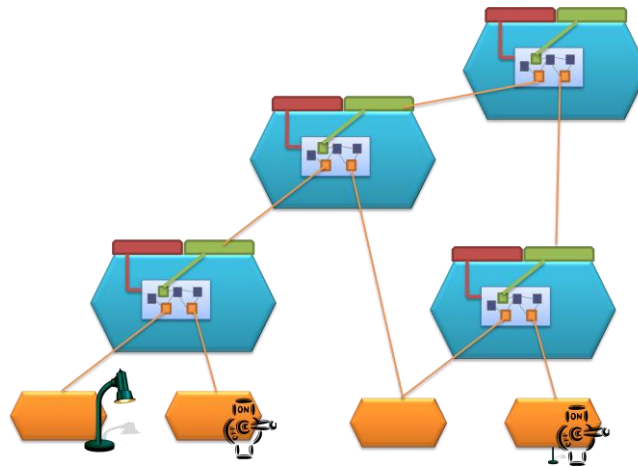
- Fichier / Nouveau / Fichier ...
- `WComp.NET / UPnP Device WebService Proxy`
- Sélectionner le dispositif correspondant à votre objet communicant dans la liste, ainsi que toutes les méthodes et variables d'état auxquelles vous souhaitez avoir accès via ce composant proxy. Cliquer sur suivant, sélectionner le remplacement de l'adresse IP par localhost puis cliquez sur Terminer. Vous venez de générer un composant proxy pour ce service UPnP.
- Recharger les composants pour avoir le nouveau composant généré (Menu `WComp.NET / Reload Beans...`)
- Rechercher le composant dans la catégorie Beans : UPnP Device (onglet Outils) et l'instancier dans le container.

Pour tester ce composant et vérifier qu'il fonctionne bien, nous allons utiliser les valeurs du capteur qui sont récupérées par le service UPnP.

#### 3.2 Réutilisation d'un service composite

Avec les étapes précédentes, vous venez de rendre votre assemblage réutilisable sous la forme d'un service composite. Nous sommes donc partis d'un service lié à l'objet communicant afin de le réutiliser pour remplir d'autres fonctionnalités en l'interconnectant à d'autres services de la maison. Et grâce aux deux types de sondes que nous venons d'étudier, il est donc maintenant possible d'utiliser à nouveau ce service composite dans un nouvel assemblage. C'est ce que nous allons faire pour pouvoir commander l'application Winamp à l'aide des capteurs connectés à notre objet communicant.

## Rendez vos objets communicants et interactifs avec des Phidgets



Pour réaliser cela, vous devrez réaliser les étapes suivantes :

- Créer un nouveau container pour réaliser l'assemblage qui réutilisera votre service objet communicant
- Générer le composant proxy vers le service UPnP lié à l'application Winamp qui vous a été distribuée
- Réaliser l'assemblage qui met en place la logique de contrôle de l'application Winamp à l'aide d'un capteur de votre objet communicant. Le but sera de contrôler le volume sonore de Winamp à l'aide de votre objet communicant.

### 4 Designers : outils pour la manipulation de Containers

Comme nous avons pu le voir dans la partie précédente, il est possible d'utiliser le modèle SLCA de WComp, et en particulier l'interface de contrôle, pour agir sur l'assemblage de composants en dehors de l'utilisation de l'interface graphique de SharpDevelop.

#### 4.1 Designer UPnP

Le Designer UPnP va permettre d'automatiquement générer le composant proxy et l'instancier dans le container auquel il est associé à chaque notification d'apparition d'un nouveau dispositif UPnP.

Pour le tester, après avoir lancé WComp et le Designer UPnP, activez la connexion « Bind to UPnP Device » dans le menu WComp. Vous vérifierez dans le Designer UPnP que vous êtes bien connecté sur votre Container. Lancez un nouveau dispositif UPnP (le service UPnP de l'objet communicant par exemple). Automatiquement, le code du composant proxy est généré (si celui-ci n'est pas déjà présent dans les Beans) et un composant de ce type est instancié et configuré avec les bonnes propriétés (adresse du serveur UPnP, etc.). Vous obtenez ainsi la possibilité de rendre votre container réactif à l'arrivée et à la disparition des services UPnP sur votre réseau local.

Nous avons vu comment réaliser des assemblages de composants à l'aide de WComp. Grâce à l'outil UPnP Designer que nous venons de présenter, nous constatons que nous pouvons avoir de la dynamique dans la découverte des services de l'infrastructure. Faute de temps, nous ne pouvons maintenant voir comment adapter les applications dynamiquement à ces changements d'infrastructure matérialisés par l'apparition et la disparition de composants proxy sur des services pour dispositifs. Pour plus d'infos, consultez le site <http://www.wcomp.fr/>