

Rendez vos objets communicants et interactifs avec des Phidgets SBC 107x

1 Phidget SBC 1072 (SBC2) et 1073 (SBC3)

Vous êtes en possession maintenant d'une plate-forme Phidgets SBC 1072 ou 1073 qui dispose d'un processeur ARM. Nous allons configurer et déployer sur cette plate-forme (qui fonctionne sous Linux) les paquetages nécessaires pour faire fonctionner l'environnement .NET 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://www.phidgets.com/downloads/libraries/Phidget-x86.exe>

ou

<http://www.phidgets.com/downloads/libraries/Phidget-x64.exe>

Vous devez aussi disposer de Bonjour sur votre PC pour détecter la connexion de la plate-forme Phidget (sélectionnez *Services d'Impression Bonjour pour Windows* en haut à droite :

<https://www.apple.com/fr/support/bonjour/>

Une fois ces applications installées, vous pouvez connecter votre plate-forme Phidget 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).

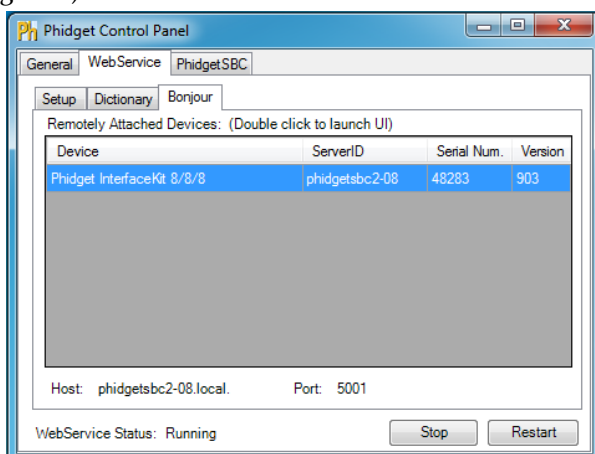


Figure 1: Web Service accès Phidget IK

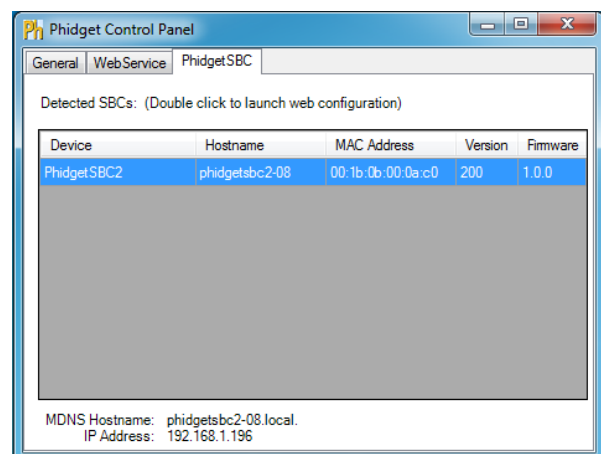


Figure 2: Phidget SBC accès plateforme

1.2 Configuration et installation des outils de la plate-forme

1.2.1 Réinitialisation de la plate-forme Phidget SBC 107x

Sauf si la plate-forme n'a jamais été utilisée, réinitialisez la en vous référant aux indications à l'adresse suivante :

http://www.phidgets.com/docs/OS_-_Phidget_SBC#Factory_Reset

Avant de pouvoir utiliser la plate-forme 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.

Lors de la première utilisation de la plate-forme, vous arrivez 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 plate-forme prête à être configurée.

1.2.2 Configuration système et réseau de la plate-forme Phidget SBC 107x

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 via la connexion http :

Rendez vos objets communicants et interactifs avec des Phidgets SBC 107x

- *Network / Settings*, SSH server *Enabled* puis *Save Changes* (cela prend un peu de temps, soyez patients)

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, vous devez configurer l'interface réseau :

- *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.

Avant de débrancher votre câble Ethernet, il faut mettre à jour le système pour corriger un bug dans le paquetage `phidgetsbcwebif`.

1.2.3 Installation et mise à jour des paquetages pour Phidget SBC 107x

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 préférons les commandes via la console qui pose globalement moins de problèmes.

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*.

Connectez-vous via l'interface Web, et faites les actions suivantes :

- Dans *System / Packages*, activez l'option *Include Full Debian Repository* puis *Save Changes*
- Mettez à nouveau à jour votre système. Préférez la mise à jour à l'aide des commandes (plutôt que l'interface web) car c'est une grosse mise à jour et elle se déroulera plus rapidement et avec moins de problème que par l'interface Web. Soyez patient après avoir lancé les commandes suivantes :

```
apt-get update ; apt-get dist-upgrade.
```

Vous pouvez aussi installer quelques outils complémentaires et utiles grâce à la commande suivante :

```
apt-get install apt-utils dialog deborphan wget unzip
```

Vous disposez maintenant d'un système à jour. Vous pouvez faire un peu de place en supprimant tous les paquetages téléchargés grâce à la commande :

```
apt-get clean  
apt-get purge $(deborphan)
```

1.2.4 Installation de Mono

Pour disposer de l'environnement .NET sous GNU/Linux, il est nécessaire d'installer Mono.

1.2.4.1 Mono sur la plateforme Phidget SBC 1072

Pour bénéficier de Mono sur la plateforme, il nous faut tout d'abord installer l'environnement .NET. Rien de plus simple, la distribution du paquetage standard fonctionne parfaitement.

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

Vous disposez maintenant sur votre plate-forme Phidget SBC 1072 de l'environnement .NET grâce à Mono.

1.2.4.2 Mono sur la plateforme Phidget SBC 1073

La distribution Mono (2.10.8.1) fournie dans le repository stable (Debian Wheezy) pose problème sur la plate-forme Phidget. C'est une version de mono très ancienne et qui présente plusieurs bugs, ceux-ci ayant un impact sur le fonctionnement de WComp. Il n'existe pas de paquetage plus récent « backporté » sur la distribution Wheezy. Nous sommes donc obligés de compiler ou d'installer une version de Mono plus récente. Nous allons utiliser la version plus récente de la distribution testing (Debian Jessie).

Rendez vos objets communicants et interactifs avec des Phidgets SBC 107x

Pour cela, nous allons devoir ajouter les sources des paquetages de la distribution testing (et donc potentiellement pouvoir installer un paquet de la version stable ou testing). Ajoutez les fichiers suivants à votre système :

```
/etc/apt/sources.list.d/testing :
```

```
deb http://http.us.debian.org/debian testing main contrib
```

Pour conserver la priorité aux paquetages de la version stable, nous allons créer les fichiers suivants :

```
/etc/apt/preferences.d/stable.pref :
```

```
Package: *
Pin: release a=stable
Pin-Priority: 995
```

```
/etc/apt/preferences.d/testing.pref :
```

```
Package: *
Pin: release a=testing
Pin-Priority: 750
```

Faites maintenant une mise à jour de la liste des paquetages disponibles :

```
apt-get update
```

Il est maintenant possible d'installer la version la plus récente de mono provenant de testing grâce à la commande suivante :

```
apt-get clean
apt-get install mono-4.0-gac/testing mono-gac/testing mono-runtime-common/testing
mono-runtime-sgen/testing mono-runtime/testing mono-mcs/testing mono-gmcs/testing
libmono-system-web2.0-cil/testing libc6/testing libc6-dev/testing libmono-
corlib2.0-cil/testing libmono-sqlite2.0-cil/testing libmono-system-data-linq2.0-
cil/testing libmono-system-data2.0-cil/testing libmono-system-runtime2.0-
cil/testing libmono-wcf3.0a-cil/testing libmono2.0-cil/testing libmono-system-
core4.0-cil/testing libmono-microsoft-csharp4.0-cil/testing
```

Si lors de l'installation vous rencontrez un problème de place mémoire lors de l'installation des paquetages (cela m'est arrivé dans certains cas), et que vous avez bien téléchargé et résolu les dépendances, vous pouvez lancer l'installation « à la main » des paquetages, ce qui résoudra le problème de manque de mémoire :

```
dpkg -i /var/cache/apt/archives/*.deb
```

Vous disposez maintenant sur votre plate-forme Phidget SBC 1073 de l'environnement .NET grâce à Mono.

1.2.5 Configuration de la librairie Phidget pour Mono

Il est nécessaire d'installer les bibliothèques de développement sur la plate-forme. Voici la procédure pour chacune des versions de Phidget SBC.

1.2.5.1 Installation de la librairie de développement pour Phidget SBC 1072

```
apt-get install libphidget21-dev
```

1.2.5.2 Installation de la librairie de développement pour Phidget SBC 1073

```
apt-get clean
apt-get install libphidget21-dev libusb-1.0-0-dev
```

Idem que précédemment, si vous rencontrez un problème à l'installation, il faut installer les paquetages à la main grâce à la commande :

```
dpkg -i /var/cache/apt/archives/*.deb
```

1.2.6 Configuration de la librairie Phidget pour Mono pour Phidget SBC 107x

Nous allons ajouter la librairie Phidget dans le gestionnaire d'assemblies (gac) pour donner accès aux fonctionnalités Phidget à l'environnement .NET.

Vous avez déjà du installer wget et unzip pour récupérer et extraire les bibliothèques.

```
wget http://www.phidgets.com/downloads/libraries/Phidget21-windev1_2.1.8.20140924.zip
unzip Phidget21-windev1_2.1.8.20140924.zip
```

Puis installez ces bibliothèques dans le gestionnaire d'assemblies (gac).

Rendez vos objets communicants et interactifs avec des Phidgets SBC 107x

```
gacutil -i phidget21-windevel/Phidget21.NET.dll  
ln -s /usr/lib/mono/gac/Phidget21.NET/*/* /usr/lib/mono/2.0/Phidget21.NET.dll
```

1.2.7 Installation de WComp pour Phidget SBC 107x

Enfin, il ne vous reste plus qu'à déployer l'archive contenant WComp à l'endroit qui vous conviendra le mieux. Vous récupérez celui-ci à l'adresse suivante (avec les instructions de votre encadrant) :

<http://trolen.polytech.unice.fr/cours/oc/td7/private/>

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 107x 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 copiez les fichiers nécessaires :

- votre assemblage `.wcc` que vous venez de réaliser ou les fichiers `SensorToConsole.wcc` ou `SensorToLedsSLCA.wcc` fournis en exemple
- 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 vos dll, correspondant à vos composants).

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 :

```
cd wcomp/SharpWCompContainer  
mono Container.exe -r ../Beans -l fichier.wcc -n nomDuDispositifUPnP -p port
```

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.

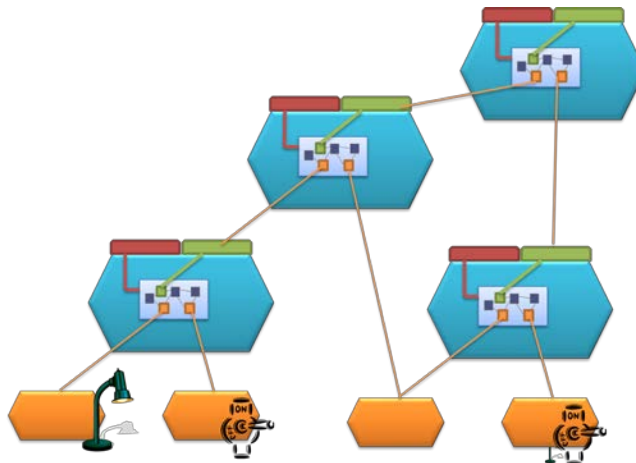
- Fichier / Nouveau / Fichier ...
- WComp.NET / UPnP Device WebService Proxy
- Sélectionner le dispositif correspondant à votre objet communicant dans la liste, ainsi que toutes les méthodes et variables d'état auxquelles vous souhaitez avoir accès via ce composant proxy. Cliquer sur suivant, sélectionner 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.

Rendez vos objets communicants et interactifs avec des Phidgets SBC 107x

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.



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

Rendez vos objets communicants et interactifs avec des Phidgets SBC 107x

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/>