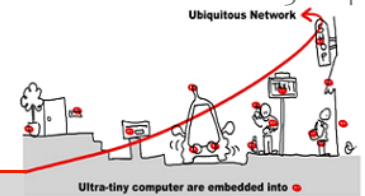


Tutorial 1: Creating a Web Service for Device: UPnP



The aim of this tutorial is to implement a new UPnP device that will be used later in a larger composite service. The interface of our UPnP Device should be the same as the one defined below.

1 INTEL Tools for UPnP Technology

This tutorial is inspired by the Intel video you can find at the following address:

<http://kistren.polytech.unice.fr/cours/oc/td1/videos/>

1. First of all, you should install the software for **UPnP Technologies** (latest is vo.o.68). You can download it for the following addresses:

<http://opentools.homeip.net/dev-tools-for-upnp>

or

<http://kistren.polytech.unice.fr/cours/oc/td1/DeveloperToolsForUPnPTechnologies.msi>
(v0.0.44)

2. Start the “*Device Spy*”, which is a “universal” Control Point. This tool allows you to search, discover and control existing instances of UPnP Devices on the local network. You can invoke methods, subscribe to events for any device you want. First of all, you should start the UPnP device NetworkLight and verify that you can interact with it with the “*Device Spy*”.

2 Developing your own UPnP Web Service for Device

First of all you should have a look to the “Full Tutorial” video available at the following URL:

http://kistren.polytech.unice.fr/cours/oc/td1/videos/Full_Tutorial.wmv

2.1 Your First Device: “Virtual” Switch

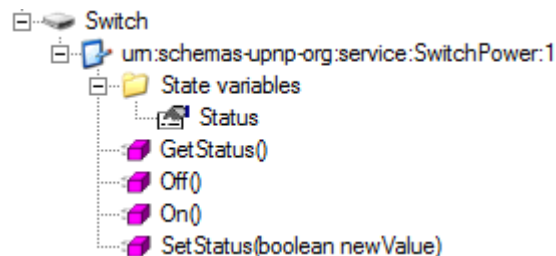


Figure 1: UPnP Interface of the Virtual Switch

2.1.1 Implementing SwitchPower Service

You should first start by creating the SwitchPower service (see Figure 1) with the **Service Author** tool. You should pay attention to define the variable “Status” as evented one and to define it before defining the method “SetStatus” that use this variable. Of course, the method “GetStatus” returns a Boolean that is the value of variable “Status”. After saving your defined service, you could have a look to the generated XML files.

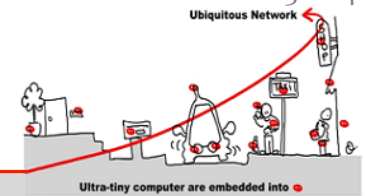
2.1.2 Composing SwitchService into a UPnP Device “Switch”

Once you have created your UPnP service, you can assemble it inside a Switch Device using the **Device Builder**.

2.1.3 Generating C# code for your device

The **Device Builder** allows you to generate the source code skeleton corresponding to the description of your UPnP Device. You should pay attention to select code generation for C# Visual Studio and to modify the Namespace to “Switch”.

Tutorial 1: Creating a Web Service for Device: UPnP



2.1.4 Implementing your device

With your favorite C# Integrated Development Environment, you should now add the code in methods to give the correct behavior to your device.

2.2 Second device: Traffic Light Device

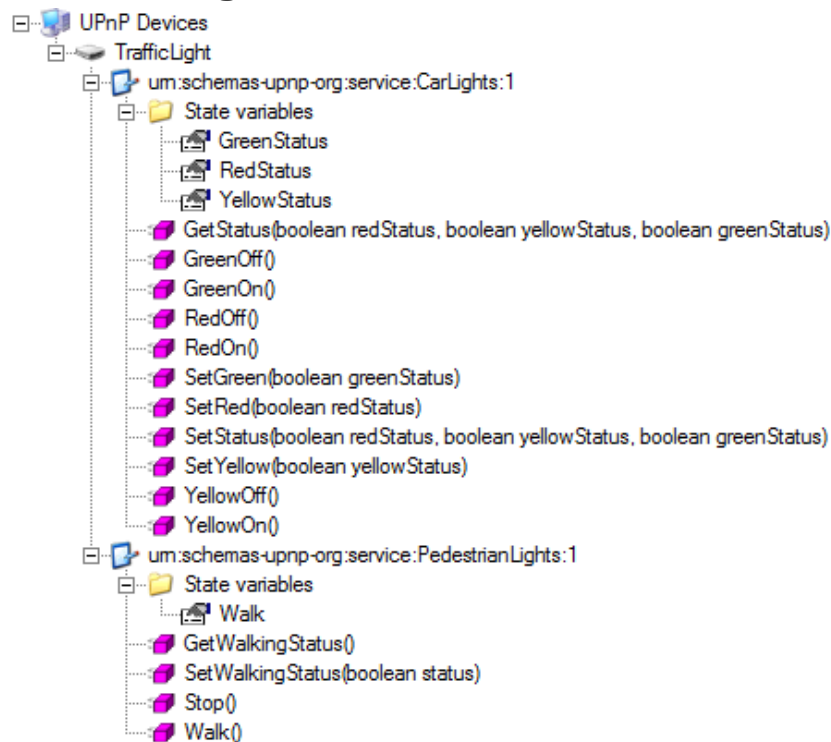


Figure 2: UPnP Interface of the Traffic Light Device

2.2.1 Implementing Traffic Light Services

You should first start by creating the CarLights and PedestrianLights service (see Figure 2) with the **Service Author** tool (please respect the given names for you to be able to link you generated device to a graphical representation of the device). You should pay attention to define all the variables as evented ones and to define them before defining the methods that use these variables. After saving your defined services, you could have a look to the generated XML files.

2.2.2 Creating the Traffic Light UPnP Device

Once you have created your UPnP services, you can assemble them inside a Traffic Light Device using the **Device Builder**. Then, **generate** the source code skeleton corresponding to the description of your Traffic Light Device (using Namespace “TrafficLight”) and implement the correct behavior to your device.

2.2.3 Graphical representation of your Traffic Light

You can create your own Graphical representation of your Traffic Light, including your generated code inside a WinForms application with your IDE. But to avoid this step, which does not represent any interest for this tutorial, you should get the ready to use Graphical Traffic Light.

<http://kistren.polytech.unice.fr/cours/muc/tutorialla/TrafficLight.msi>