Getting Started with ThingWorx on a Conel V3 Router using the ThingWorx C SDK and the Conel V3 C SDK

Contents

[1. Introduction 3](#_Toc412626363)

[Working with our Example 3](#_Toc412626364)

[2. Setting up the Router 4](#_Toc412626365)

[3. Installing and Configuring the ThingWorx Application 5](#_Toc412626366)

[4. Importing the Project into ThingWorx Server 6](#_Toc412626367)

[5. Application Explanation 7](#_Toc412626368)

[6. To Augment this Example 8](#_Toc412626369)

# Introduction

ThingWorx is a software platform designed to build and run the applications of the connected world. Conel V3 routers are wireless cellular routers that enable user to connect to end devices wirelessly. More information about ThingWorx can be found here (<http://www.thingworx.com/learn-more/>) and Conel routers here (<http://www.conel.com/spectre-v3-lte-platform/>).

ThingWorx provides a C SDK to allow devices to connect to a ThingWorx Platform Server, Conel provides a C SDK to enable users to build User Modules (applications) that can run on the router.

Using a combination of these SDKs, we can demonstrate how to represent the data available at a remote sensor site to a Cloud (or Hosted) platform.

At the remote sensor site, we will use the B+B Smartworx / Conel V3 router; at the hosted platform end, we will use a Thingworx server instance.

Before we get started here’s a list of what you’ll need:

* A B+B Smartworx / Conel V3 Router and power supply (you can order from here: <http://www.bb-elec.com/Products/Wireless-Cellular/Cellular-Routers.aspx>)
* An Ethernet Cable (you will receive this with your router)
* A PC running Linux (you only need to run Linux if you want to modify our example code, and compile your own)
* A ThingWorx Platform Server Installed and set up
  + Tutorial can be found here (<https://twc.thingworx.com/Thingworx/FileRepositoryDownloader?download-repository=ThingWorxPublicRepository&download-path=%2FDocumentation%2F5+0+Install+and+Configuration.pdf> ).
  + You will need a ThingWorx User Community login (<https://twc.thingworx.com/Thingworx/twc_welcome.html>)

### Working with our Example

To begin, download the example (from here: <https://github.com/bb-elec/thingworx/releases/download/v1.0.0/thingworx-v3.tgz>)

This is a Tarball archive, which is organised as follows:

|  |  |
| --- | --- |
| router/application | Contains the .tgz User Module that goes on the router |
| router/src | Contains the source of the V3 User Module |
| doc | Contains this document |
| server | Contains a V3 project for the ThingWorx Platform Server that can be imported and used with the V3 User Module |
|  |  |

# Setting up the Router

If you need to, you can find the Start Guide for V3 routers here: <http://www.conel.com/spectre-v3-lte-platform/>

* Go to the Downloads tab. You will need to create a free account and sign-in.

Follow the instructions in the Start Guide.

When you are finished, select User Modules under the Customizations section on the left hand side of the screen.

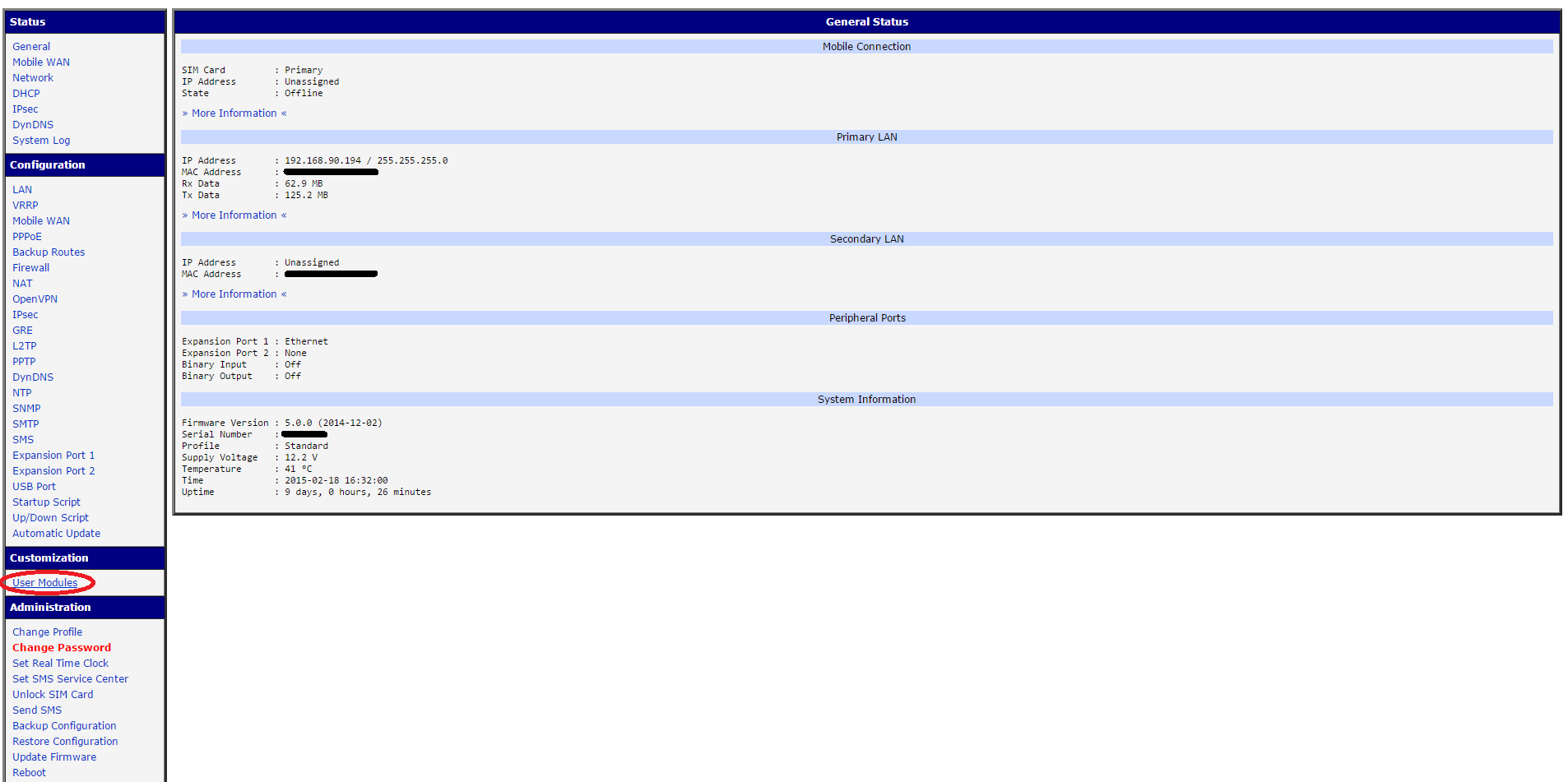


Figure When finished setting up, select User Modules

# Installing and Configuring the ThingWorx Application

We have provided a ThingWorx application example that sends statistics about the router to a specified ThingWorx Platform Server

* We have also supplied a ThingWorx project that can be imported onto your Thingworx Platform Server. We will speak more about that in section 4.

To load this example onto your router:

On the User Modules page click Choose File, then navigate to where you downloaded and extracted thingworx-v3.tgz, in the router/application directory select thingworx.tgz and hit enter.

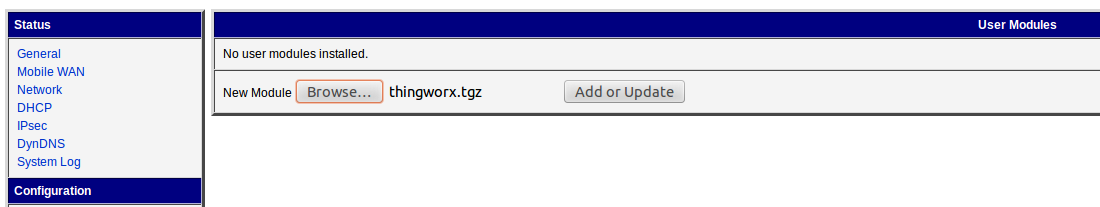


Figure Uploading User Module

Now press the Add or Update button and the User Module will be put on the router.

Next, you can configure the example application.

To configure the application, click its name.

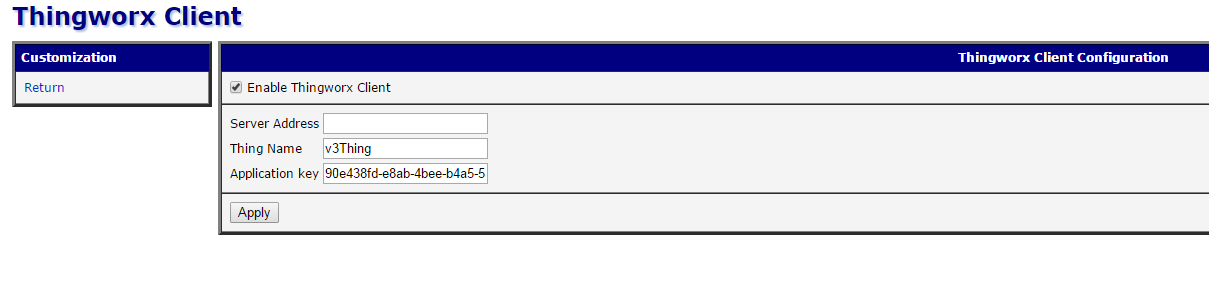


Figure Configuring the Application

Enter the address of your ThingWorx Platform Server.

If you want to use the server project we have provided in this example, enter “v3Thing” in the Thing Name field (in this example, this is the name of your device) and “90e438fd-e8ab-4bee-b4a5-5aec440e8086” in the Application Key field.

# Importing the Project into ThingWorx Server

We have provided an example ThingWorx Project that you can import and use as a starting point for development or as a test to see if your device is connecting as expected.

To import go to the ThingWorx Composer on the ThingWorx Server you have set up.

In the top right hand corner select Import/Export then Import from File.

In the pop-up select Choose File and navigate to the server directory and select the v3thing.xml file then press import.

Three new items should appear in the list on the front page, v3Thing, v3Key and v3 Mashup.

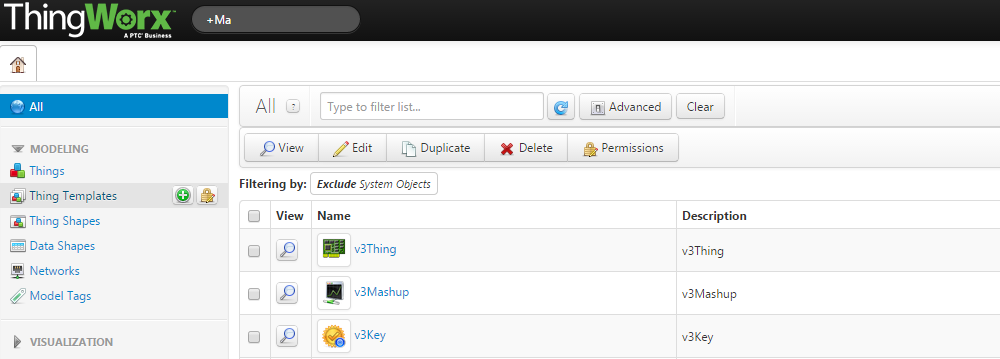


Figure New Items

From here select v3Mashup, and then make sure Info is selected on the top bar and Mashup Preview is selected on left menu.

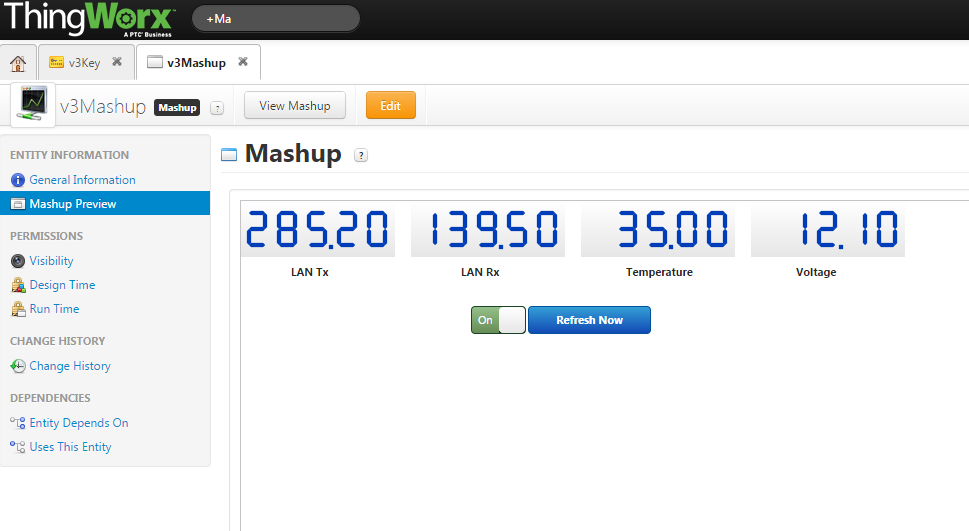


Figure A Mashup showing data retrieved from the router

You should now be able to see data from the router (you may need to press Refresh Now).

# Application Explanation

The User Module uses the Conel C SDK (you can get this from here: <https://github.com/bb-elec/v3-api-public/releases/download/v0.8.0/SDK-v3.tgz>) and some in-built functions on the router, to get data from the router.

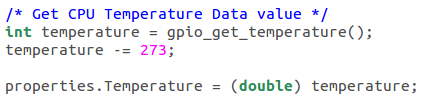
These data values provided in our example are:

* Data transferred over the LAN connection
* CPU temperature
* Supply voltage.

The following code snippets retrieve the data values from the router, using the Router’s C API.

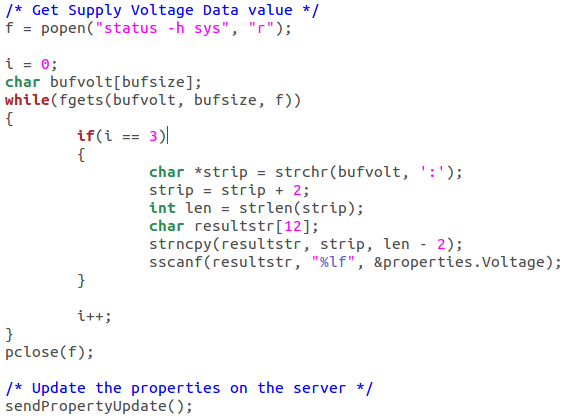
The ThingWorx C SDK handles connecting to the server and sending the data (refer to the ThingWorx C SDK documentation for a better explanation of how it connects and sends data). The ThingWorx SDK has a function called dataCollectionTask, this is where you get the values to send to the server.

To get the temperature of the CPU we use the V3 GPIO API call:



The temperature is given in degrees kelvin so first you convert to Celsius then set the properties temperature.

The next example makes use of the status command which is a system-command on the router:



This captures the output of the status command then parses it to get just the value.

# To Augment this Example

The first thing you need to do is setup your environment.

Router SDK:

* Download the Router SDK
* Extract the Router SDK to the ROOT directory  
  $ tar -zxf SDK-v3.tgz -C /

Get the Router C SDK from here: (<https://github.com/bb-elec/v3-api-public/releases/download/v0.8.0/SDK-v3.tgz>)

Thingworx SDK:

* Download the ThingWorx C SDK
* In your home directory create a directory called Thingworx  
  $ mkdir ~/ThingWorx
* Extract the ThingWorx C SDK to this directory  
  $ unzip C-SDK-1.0.0.27.zip -d ~/ThingWorx/

Get the Thingworx C SDK from here: (<https://twc.thingworx.com/Thingworx/FileRepositoryDownloader?download-repository=PlatformDownloads&download-path=%2F5.1%2FMicroServer%2FC-SDK-1.0.0.27.zip>)

ThingWorx application example:

* Download the ThingWorx application example
* Extract the example to the ThingWorx directory you already created

$ tar -zxf thingworx-v3.tgz -C ~/ThingWorx/

Get the example from here: (<https://github.com/bb-elec/thingworx/releases/download/v1.0.0/thingworx-v3.tgz>)

The following steps describe how to build the supplied example:

* At this stage you should have a directory in HOME called ThingWorx with tow directories in it: thingworx-v3 and tw-c-sdk
* You should also have a directory called toolchain in /opt
* To compile go into thingworx-v3 then router then src and run make:  
  $ cd ~/ThingWorx/thingworx-v3/router/src  
  $ make
* When make is finished there should be an archive called thingworx-v3.tgz, this is what goes on the router.