INTRODUCTION

What is the protocol?

Micasaverde's Vera is a z-wave home automation box.

What is great about it?

It is a standalone box with very low energy consumption.

Not very easy to program but very powerful thanks to its plugin market.

You can have a look at http://www.micasaverde.com/
## Step 1 — How To - MiCasaVerde Vera

### Hardware

- **What hardware is out there?**
  Micasaverde's Vera is an home automation standalone box. It comes with it's own UI but you can interface it with OpenRemote to make it more powerful and user friendly.

- **What do you need to keep in mind when choosing a product?**

- **Which brands and products are tested and work well with OpenRemote?**
  Micasaverde's Vera fully works with OpenRemote. A more user friendly implementation of Vera in OpenRemote is on the way, but all functionalities are already available.

- You can find below a list of all devices that OpenRemote can control through Vera. But as the power of Vera resides in its plugins, you will also find a list of OpenRemote supported plugins (to do).

- The functionalities of the following products are supported by OpenRemote. Certified products are extensively tested in combination with Professional Designer and supported by OpenRemote certified Integrators.

<table>
<thead>
<tr>
<th>Brand</th>
<th>Product</th>
<th>Support</th>
<th>Certified Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Everspring</td>
<td>SM103</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Everspring</td>
<td>SP814</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Everspring</td>
<td>AN1576</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Everspring</td>
<td>AD142-6</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Fibaro</td>
<td>FG5-211</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Fibaro</td>
<td>FG5-221</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Fibaro</td>
<td>FGD-211</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Fibaro</td>
<td>FGBS-001</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Oregon Scientific</td>
<td>THGN132ES</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>Oregon Scientific</td>
<td>THGR132NX</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>OWL</td>
<td>CM160</td>
<td>Certified</td>
<td>All</td>
</tr>
<tr>
<td>RFXCOM</td>
<td>RFXtrx433 USB</td>
<td>Certified</td>
<td>All</td>
</tr>
</tbody>
</table>
Step 2

- **Configuration - Native integration**
  - First, you will have to install OR controller. Have a look to the tutorials to install it on a NAS, computer, Raspberry or get an eBox! OpenRemote's team has decided to integrate Vera as a native OpenRemote protocol. This way, adding devices is really easier and the UI becomes really responsive!

- **Configure your Vera unit in the Designer**
  - Please go to the Config for Controller tab and click Vera. Just complete the line with your Vera IP address. *(See image 1)*

- **Add new devices using the wizard**
  - Not available for now. Work in progress... Thank you Marcus!
Step 3

- Add new devices manually
  
  - You can also add new devices manually. You know how OpenRemote works: you will have to create http commands and sensors/ sliders/ switches for each device. The team did a great job on making it easier by proposing standard commands for standard devices.

  - As usual, create a new device. Fill its name, vendor and model and then create a new command.
    
    - Name it
    
    - Select Vera protocol
    
    - Fill the Deviceld (#xx, the number of your device in the advanced tab of your device in Vera's UI)
    
    - Choose a command
    
    - Submit it and create all the commands, sensors, switches or sliders you need for your device. *(See image 1)*
Step 4

- Now you know the process.
- **Commands list** *(See image 1)*

- Those commands may help you to control most of your devices. The two last are really interesting because they allow to command or get any status of any device, or plugin!

- **GENERIC_ACTION**

- To create a custom action you will need additional informations you could find in the advanced tab of your device:
  - serviceld: this is the line starting with urn...blablabla
  - action
  - variable
Step 5

- Value
- See this line to understand: (to be completed)

**GENERIC_STATUS**

- The vera protocol window needs only a few lines in order to get the status from any vera device. This works by accessing a status xml file generated by vera. In order to fill this window in properly, you will need to know what vera thinks the Id of your device is, and the unique identifier of the information you wish to report.

- Navigate here in order to gather the information you need concerning vera(replace "veraIpAdress" with the IP of your vera:
  
  [http://veraIpAdress:3480/data_request?id=status&output_format=xml](http://veraIpAdress:3480/data_request?id=status&output_format=xml) You will need the device id and the variable value of the information you are interested in.

- You may also need to know what sort of information the command will be returning in the value attribute.
This explanation will be made by way of example. In this example, we will retrieve the device target.

Step 6

- An example of some vera xml can be seen below: *(See image 1)*
- In order to retrieve information from this device, you will need a command. Create a new command. Choose the "vera" protocol.
- In order for us to retrieve information from the "target" attribute, we need the following command configuration:
  - Name - Any
  - Protocol - Vera
  - Device Id - 10
  - Command - GENERIC_STATUS
  - Status attribute - target
Step 7

- **NOTE** The value entered into the "Status attribute" field is all lower case, even though in the vera xml one of the letters is upper case ("Target"). Make sure you enter your values in lowercase.

- The above configuration would pull the value "1" from the "Status" attribute if OpenRemote encountered xml like the example above. In some cases, you will see information which is not binary, and sometimes not even an integer.

- In these cases you will need to [Designer 2.0 - Create Custom Sensor](#) in order to interpret the information that the command is returning.
Step 8

- **Configuration - PHP or RULES solution**

  - These are the old ways to integrate Vera to OpenRemote. They will stay online as long as Vera protocol is not available for every controller. If Vera protocol is available on your controller, you should avoid PHP or Rules solutions.

- **PHP solution**

  - First, you will have to install OR controller. Have a look to the tutorials to install it on a NAS, computer, Raspberry or get an eBox! In fact, Vera is returning its status to a poll loop and can be remote controlled by HTTP requests. Status request:

    - `http://VERAIP:3480/data_request?id=user_data&output_format=xml`

  - To use this solution you will need to set up a PHP server. Here is the scheme: an OpenRemote sensor runs a PHP script each second. This script copies the XML status file that Vera returns into the file vera.xml.
Then, each OpenRemote sensor can poll vera.xml each 500ms (or less often). This solution avoids Vera flooding in case of multiple requests of multiple sensors.

**Step 9**

- **Create a PHP script named getVERAxml.php and host it on your server** (check Vera IP adress and vera.xml location). *(See image 1)*

- **Create a HTTP command that runs the PHP script and poll it each second** (check server adress and file location). *(See image 2)*
  - http://SERVERIP/getVERAxml.php

- **Create a sensor using this command.** *(See image 3)*

- **Place this sensor on your panel in the Designer.**
  - Now your Vera controller launches the PHP script each second and copies the status file on your server into vera.xml.
Step 10

**Rules solution**

- First, you will have to install OR controller. Have a look to the tutorials to install it on a NAS, computer, Raspberry or get an eBox! This method uses OpenRemote rules to do the PHP job. Not stable today on eBox: the box freezes after some hours.

- Go to the designer in the Config for Controller > Rules section. *(See image 1)*

- Add this script to the rules and click submit. Check Vera IP adress and vera.xml location. *(See image 2)*

- Now your sensors can poll the vera.xml here (check file location and server adress):
  - http://SERVERIP:8688/controller/vera.xml
Step 11

- **Add new devices**
  - Creating a Vera device in OpenRemote means 3 things
    - 1. Creating one or more HTTP commands
    - 2. Creating a sensor or/ and a switch/ slider
    - 3. Creating an object in the Designer
  - To send a HTTP command (switch a light ON for example) you have to follow this syntax:
    - `http://VERAIP:3480/data_request?id=lu_action&output_format=xml&DeviceNum=DEVICEID&serviceId=urn:upnp-org:serviceId:SwitchPower1&action=SetTarget&newTargetValue=1`
    - Needs VERAIP, service, name, variable.
Step 12

For devices which need statuses, you have to poll the vera.xml and get the Xpath to the value. Have a look to your vera.xml to understand the path. Regular Xpath expression for a light status (for example) can use Vera device id or Vera device name:

- //devices/device[@id='DEVICEID']/@status
- //devices/device[@name='DEVICENAME']/@status

Have a look to your vera.xml file and you can find all the informations you need to write your http commands and Xpath.

Example: Temperature sensor

Here is the example to create a temperature sensor.
Step 13

- **Create a HTTP command to get the temperature** by using these lines. Don't forget to set a polling interval.
  - http://SERVERIP:8688/controller/vera.xml
  - //devices/device[@id='DEVICEID']/@temperature
    - *(See image 1)*

- **Create a sensor using this HTTP command.**
  - *(See image 2)*

- **Create a label in the Designer using this sensor.**
  - *(See image 3)*
Step 14

- Example: Switch

- Here is the example to create a switch.

- **Create a HTTP command to turn the switch ON**

  ```
  http://VERAIP:3480/data_request?id=lu_action&output_format=xml&DeviceNum=DEVICEID&serviceName=urn:upnp-org:serviceId:SwitchPower1&action=SetTarget&newTargetValue=1
  ```

- **Create a HTTP command to turn the switch OFF**

  ```
  http://VERAIP:3480/data_request?id=lu_action&output_format=xml&DeviceNum=DEVICEID&serviceName=urn:upnp-org:serviceId:SwitchPower1&action=SetTarget&newTargetValue=0
  ```
Step 15

- **Create a HTTP command to get the switch status.** Don't forget to fill the polling interval (500 for 500 milliseconds for example).
  - http://SERVERIP:8688/controller/vera.xml
  - //devices/device[@id='DEVICEID']/@status

- **Create a sensor using the switch status command.** Choose custom type to have custom state items. *(See image 1)*

- **Create a switch using ON and OFF commands and the switch sensor.** *(See image 2)*

- **Create a switch in the Designer using the created device.** *(See image 3)*

- You can choose different images for ON and OFF states.
  - **Example:** Slider
Here is the example to create a slider.

**Create a HTTP command to DIMM.** Param is the dimming level returned by the slider.

- `http://VERAIP:3480/data_request?id=lu_action&output_format=json&DeviceNum=DEVICEID&serviceId=urn:upnp-org:serviceId:Dimming1&action=SetLoadLevelTarget&newLoadLevelTarget=${param}`

**Create a HTTP command to get the slider status.** Don't forget to fill the polling interval (500 for 500 milliseconds for example).

- `http://SERVERIP:8688/controller/vera.xml`
- `//devices/device[@id='DEVICE ID']/@level`

**Create a sensor using the slider status command.** Choose range type from 0 to 100 (or adjust) to have custom states items.

(See image 1)
Step 17

- Create a slider using the DIMM command and the slider sensor.
  - *(See image 1)*

- Create a slider in the Designer using the created device.
  - *(See image 2)*

- You can choose different images to customize your slider.
Step 18

REFERENCE CASES

- **Reference Cases**
  - [http://www.milillicuti.com](http://www.milillicuti.com) My own website as an advertising freelance but you can contact me about OpenRemote or home automation.
  - Always check the OpenRemote forum for the latest developments.