

# Application Note:

## How to integrate PAVIRO Public Address and Voice Alarm System and Bosch Video Management System

Author:  
Jan Noten BT-SC/MKR-EU/MKI-Ein

Version 1.07 Aug 2019



## Table of Contents

<b>1.</b>	<b>Overview of the Integration</b>	<b>3</b>
1.1	Application introduction and benefits	3
1.2	Application examples for BVMS-PAVIRO integration solutions	3
1.3	BVMS- PAVIRO Integration Features	3
1.4	Application software and hardware needed to build this integration	4
<b>2.</b>	<b>Technical details and Setup of the integration</b>	<b>4</b>
2.1	Preparing the PAVIRO for interfacing to BVMS	4
2.2	Preparing the Logical Trigger and TaskEngine setting in PAVIRO	5
2.3	How to setup the BVMS side to make a BVMS invoked call.	7
<b>3.</b>	<b>Testing and tuning the integration solution</b>	<b>10</b>
<b>4.</b>	<b>Appendix A THE PAVIRO SCRIPT</b>	<b>11</b>
<b>5.</b>	<b>Appendix B Debugging Server Script runtime actions</b>	<b>17</b>
5.1	How to print debug information in the Operator Client GUI	17
<b>6.</b>	<b>Appendix C Simulation Tools for Integrators</b>	<b>19</b>
6.1	Simulation of the PAVIRO Controller Interface connection and its functions	19
6.2	Simulation of 3 <sup>rd</sup> Party Applications to invoke a PAVIRO Call	21
6.3	Understanding the PAVIRO – BVMS protocol syntax, debugging developments	22
<b>7.</b>	<b>References and Documentation</b>	<b>24</b>

# 1. Overview of the Integration

## 1.1 Application introduction and benefits

For many security applications, the ability to make automatic announcements, invoked by Bosch VMS, DIP3000 or DIP 7000 located in a central control room, is a highly desirable extra layer of automation to warn, inform or instruct persons about possible danger in areas or buildings. At the same time, this functionality can be used to make general announcements.

## 1.2 Application examples for BVMS-PAVIRO integration solutions

There are many applications in various verticals and sub verticals, where audio and video combined could strengthen the total security solution.

Some use case examples:

- Airports:  
Produce an automatic security warning when a bag is left alone in an airport area.
- Traffic:  
Produce a warning to leave the area in a car park when a vehicle parks in an emergency area monitored via Bosch CCTV cameras.
- Transportation:  
Evacuation messages or friendly messages on Ferries or Cruise ships triggered by video analysis.
- Retail/Sport/Industrial Sites/Harbors:  
Various public announcements in stores, stations , stadiums, harbor sides, industrial sides activated from CCTV control rooms.
- Etc.

With this BVMS – PAVIRO integration, voice and audio announcements can be linked to BVMS events and automatically be activated by **Alarm Contacts triggers, Virtual Inputs status changes, Intelligent Video Analysis triggers in cameras, or even initiated from a BVMS connected external 3<sup>rd</sup> party subsystem.**

Apart from system events in BVMS, also Operator requested announcements could be set-up. To simplify and improve ease of use the installer could create BVMS Operator controls on Maps or synoptic panels or boards to activate via double-click-icons any announcement to any PAVIRO Zone.

## 1.3 BVMS- PAVIRO Integration Features

The integration between BVMS and PAVIRO can be made as rich as desired by using the correct Open Interface command syntax as available for this interface. Please refer the "**ASCII Control Protocol**" section of the IRIS-Net manual for the syntax of the Open Interface command set.

To set up and demonstrate integration functionality, this Application Note will show the basic installation and communication format needed between BVMS Server and the PAVIRO Controller which basically represents any future command expansion to be added in a similar way.

To make the integration easy to understand as well as create maximum functionality for the PAVIRO system, you can use Logical Triggers (Logical Values) and the Task Engine blocks in IRIS-NET software to program virtually any follow-up action (like Calls and Chime in this application note).

By programming Script activation in BVMS, PAVIRO Logical Triggers can be activated to trigger these Task Engine automated actions.

For Example:

A BVMS camera detected an IVA alarm when an intruder crossed a preprogrammed trip wire line in the camera's scene, this event invokes a preprogrammed BVMS Script that connects to PAVIRO and toggles the PAVIRO Logical Trigger.

The result is that this intruder will be immediately warned via the loudspeaker(s) to promptly leave the area.

## 1.4 Application software and hardware needed to build this integration

To set up this integration you need on BVMS and on PAVIRO side some initial setup and configuration. There is no need for additional external software.

The interfacing described in this Application Note was developed and tested with the following software/Firmware:

1. BVMS Server, BVMS Operator Client, BVMS Config Client - Software Version 9.0
2. PAVIRO Controller (PVA-4CR12) firmware: 1.20.0, Call station (PVA-15CST) firmware 2.4.0 and Amplifier (PVA-2P500) firmware 1.1.0
3. PAVIRO Open Interface Syntax and IRIS-Net version 4.0.0 build 8664

In order to test the interface, some preprogrammed Business Messages and Chimes were used. Please refer to the [PAVIRO documentation](#) for additional information.

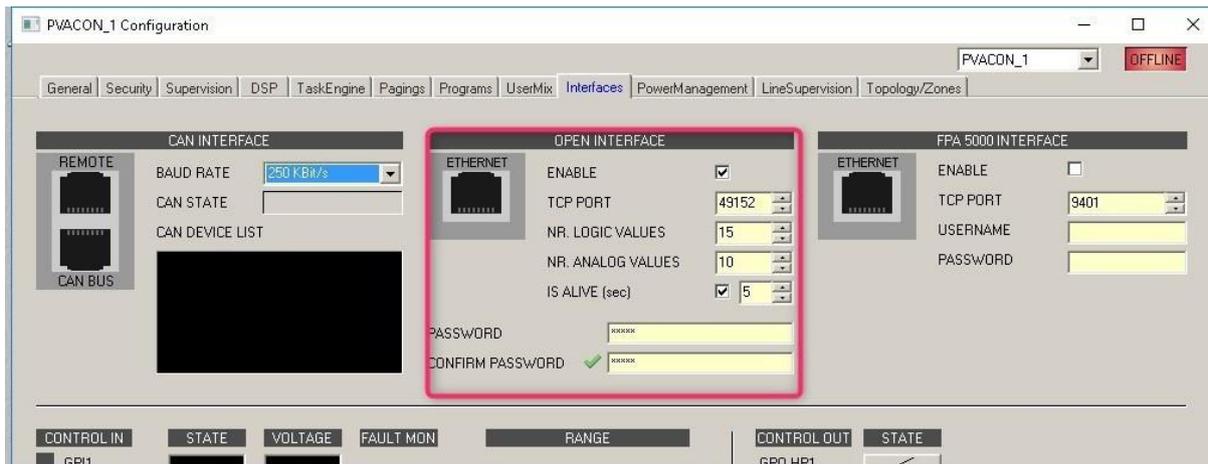
## 2. Technical details and Setup of the integration

### 2.1 Preparing the PAVIRO for interfacing to BVMS

Before you can use the PAVIRO interface you should set first the PAVIRO Open Interface parameters via the Intelligent Remote Integrated Supervision software. This IRIS-Net program allows you to enable the OI and set the parameters.

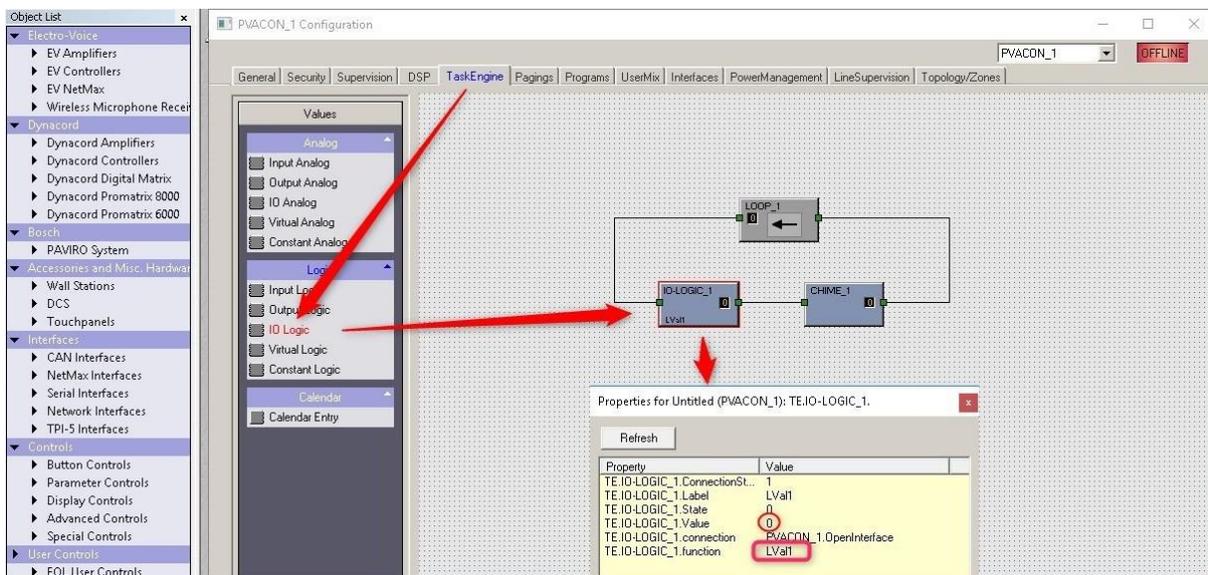
Parameter to set for instance are the IP Open Interface Account login details, number of Logic and Analog Values and the heartbeat repetition time for IP connection monitoring purpose.

Please see red box below with the test settings used.



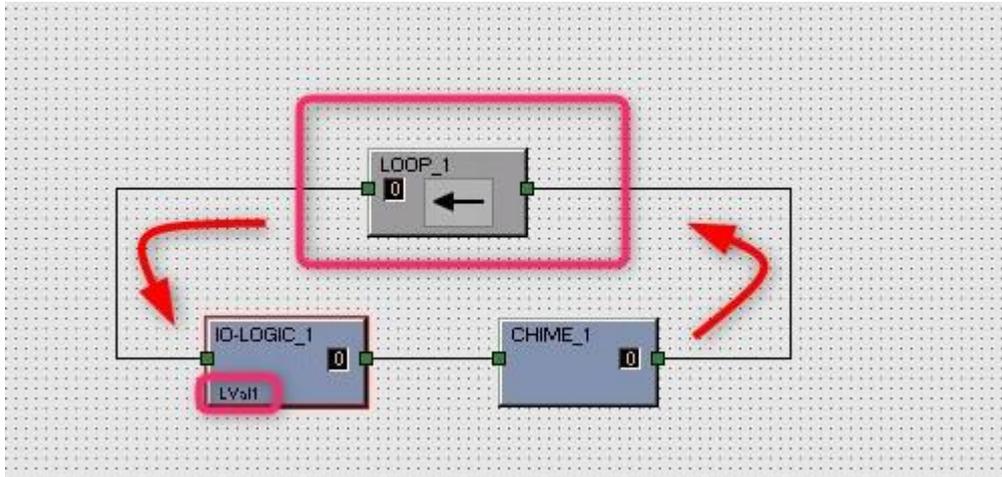
## 2.2 Preparing the Logical Trigger and TaskEngine setting in PAVIRO

In order to trigger Logical Triggers in PAVIRO via BVMS and use these to invoke further PAVIRO processing, you can define the Logical Value in PAVIRO and use the Task Engine TAB in IRIS-NET. The default way to use LVAL is via the pagings tab. However when triggering via the pagings Tab, the LVAL will remain 1, once it is activated. You have to manually reset it to zero for the next BVMS trigger. That is why we use the task engine to automatically reset the LVAL to 0. See below the principle set up of LVAL1 to invoke a call.

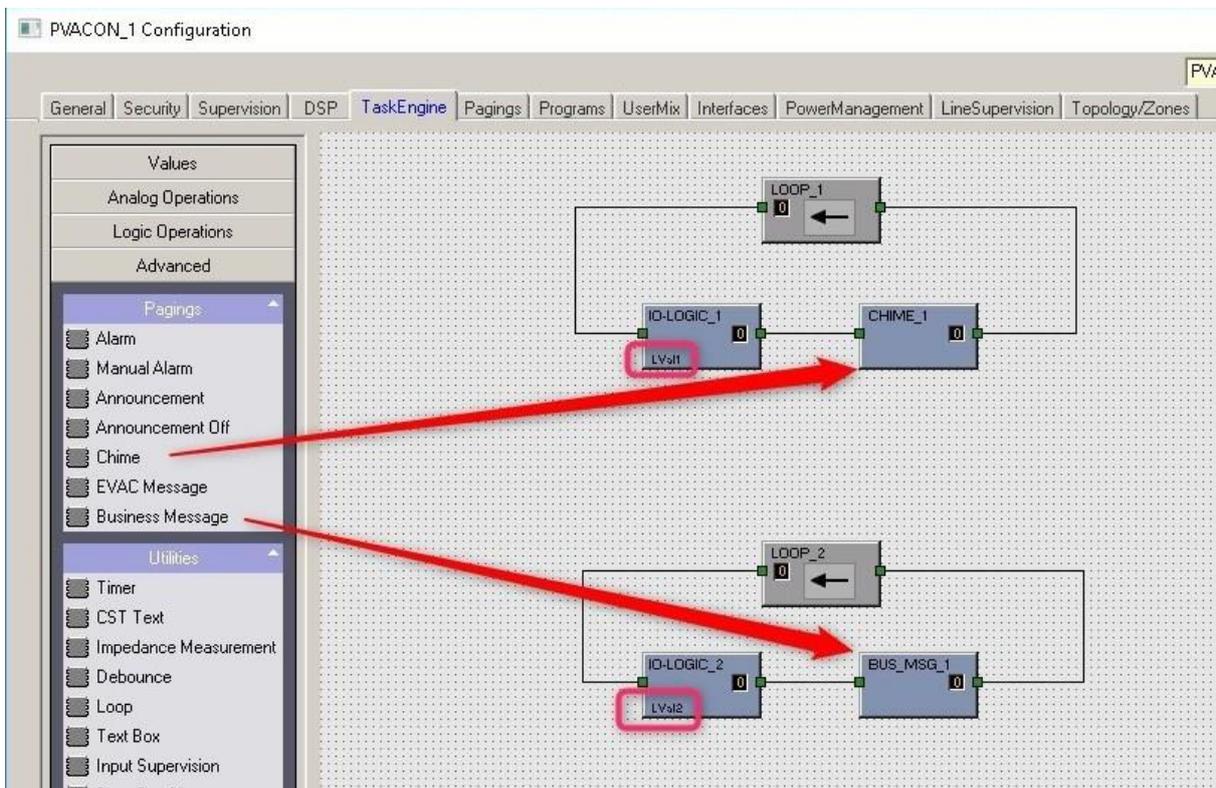


Because Calls and Messages have different length, BVMS is set up to only activate the Logical Trigger and initiate the call via the Open Interface. All further processing is done in PAVIRO. This includes also resetting the Logical trigger in order to be ready for the next BVMS event. For that purpose we use a LOOP setting in the Task Engine.

In the below example the LVAL1 set to ON status will activate the Chime\_1.  
 When the Chime has finished, the LOOP Object will reset the LVAL1 to OFF.  
 In this way the Chime is always played till the end and LVAL1 is ready for the next BVMS trigger again.



This is virtually all that has to be done on PAVIRO side. Of course you could make it more interesting defining also similar set up for a Business Message like this using LVAL2



### 2.3 How to setup the BVMS side to make a BVMS invoked call.

In principle the PAVIRO call is generated by a Server script that connects to the PAVIRO at an initial call request and after being connected it sends the logical triggers (LVAL1 1 or LVAL2 1 etc ) to PAVIRO.

At this point you should decide if the call should be generated by a BVMS Server event (like input events, cameras events i.e motion detected etc ) or by a BVMS Operator clicking an Icon in his Operator Client Workstation User Interface (either an icon on a MAP or an icon in the Logical Tree of his Operator User Interface).

In case of Server events, you should deploy a Server script and in case of Operator clicks, deploy a Client Script.

As an example, this explanation will guide you for a Server event when the operator activates a so called User Event 1 in his GUI. (Fig 1.)

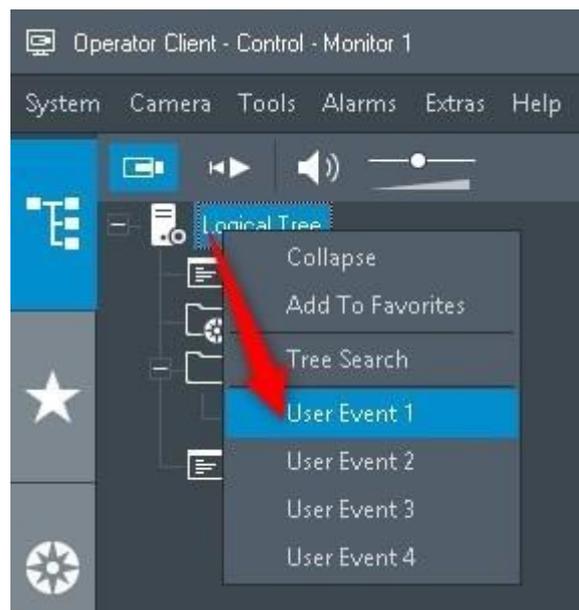


Fig 1.

For this event to generate a PAVIRO call we must add a Server Script.

To achieve this, open the script area in the Configuration Client (Tools-Command Script Editor menu)

1. Next step is to add a new ServerScript C# (or VB) by right mouse clicking on ServerScript in the left panel and choose New Scriptlet (Fig. 2.) This automatically creates in the right panel an empty script template
2. Next replace the ENTIRE existing code by the **red code** in Appendix A as new code.

**Note:** If you had already scriptlets defined then you should merge these to this **red code** in order to not loose these scriptlets.

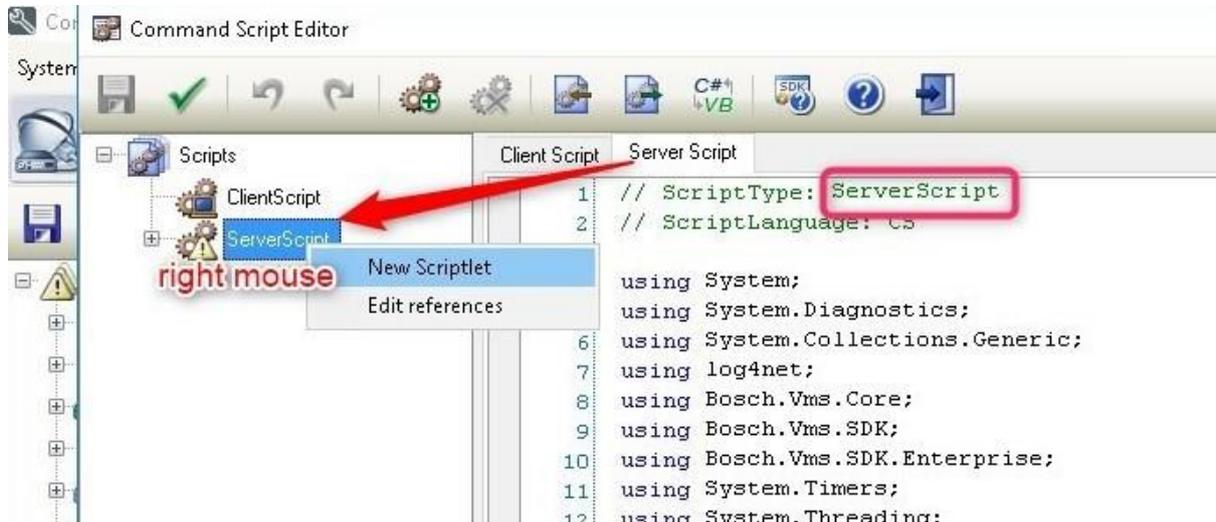


Fig 2

Please adapt the following code lines in Appendix A to your own PAVIRO variables:

```

public string pvrSocketIP="192.168.10.202"; ///your PAVIRO IP number
public int pvrSocketPort=49152; ///your PAVIRO socket
string Password="jan" + "\n"; ///"<your PAVIRO Password>" + "\n";

```

**Note:**

1. Your PAVIRO Password (as set and used with IRIS-NET) must be **correct** in the script to avoid BVMS server crashes at a call activation !!!
2. Also make sure that there is NO other connection to PAVIRO via the OpenInterface IP socket active whenever BVMS is connected. Failing to do so will also crash the BVMS server.
3. In general, whenever a connection fails due to above reasons, PAVIRO will reject a connection the hard way. As a result the BVMS Server will crash and has to be restarted by either a reboot or via the Windows **service.msc** option.

Next step is to save the new settings (Floppy symbol) and restart the Server with the new settings (Green button icon). If you get Compiler Errors then first fix these errors. See also Appendix B for debugging help guide lines.

After compiling, some server scriptlets will be added automatically to the ServerScript list and should look like Fig 3 :

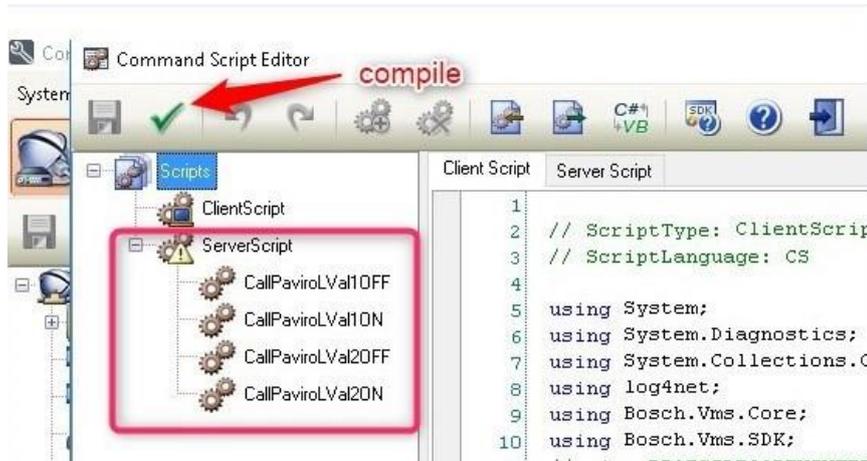


Fig 3.

To initiate this script action via a BVMS system Event, you must select the correct Event in BVMS (A) - add the script name (B) to that event - and set the desired schedule (C) for it.

See Fig 4 below from the Event Tab in the BVMS Config Client software where the User Event 1 is used to launch this call with Schedule set to “Always”.

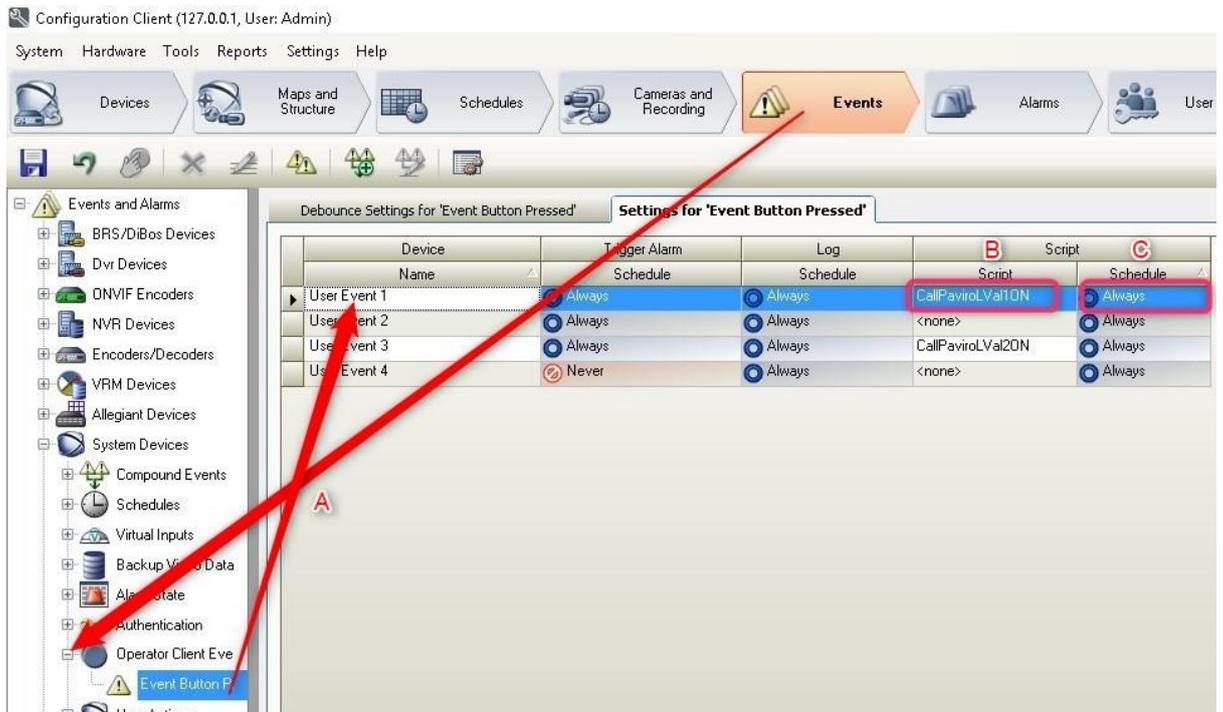


Fig 4.

This is basically all it takes to program calls in BVMS. You can use the same principle for LVAL2 ON scripts binding them to other events etc. All PAVIRO logical trigger actions should be programmed in the PAVIRO Controller via IRIS-NET for flexibility in equal way as described for LVAL1 On above.

To create synoptic call panels in BVMS for the Operator to address announcements, you can drop script icon on maps or empty forms (panels) as many as desired and populate these with Client Script icons that calls the correct announcement and destination. This is however beyond the scope of this AppNote.

Note:

For further details, please consult the BVMS documents or contact Bosch Integration Partner Program (IPP) support MKI department in Eindhoven the Netherlands.

### **3. Testing and tuning the integration solution**

This above example script integration is a basic integration and opens a connection to PAVIRO at first call. In principle the PAVIRO will submit heart beats to monitor the connection health. The script is neither dealing with this at the moment nor is this integration designed to receive and handle PAVIRO events (callbacks). Of course and according to the required project features set, this basic integration can serve as a concept to enhance the interface functionality. In that case and with added requirements like callbacks etc, a constant open and monitored connection to PAVIRO is preferred.

For further details, please consult the BVMS and PAVIRO documents or contact the Bosch MKI department in Eindhoven.

In order to provide a jump start to enhance the interfacing method to a more rugged and functional interface, please find an enhanced concept of communication/interfacing between PAVIRO and BVMS as simulators in Appendix C below.

## 4. Appendix A THE PAVIRO SCRIPT

```
// ScriptType: ServerScript
// ScriptLanguage: CS
using System;
using System.Diagnostics;
using System.Collections.Generic;
using log4net;
using Bosch.Vms.Core;
using Bosch.Vms.SDK;
//using Bosch.Vms.SDK.Enterprise; // debugging tool to show diagnostic info in Operator Client GUI
// For instructions how to debug the script runtime , please consult the AppNote Appendix B
using System.Timers;
using System.Threading;
using System.Net;
using System.Net.Sockets; //For Paviro Socket to port 49152

[BvmsScriptClass()]
public class ServerScript : IDisposable
{
    private readonly IServerApi Api;
    private readonly ILog Logger;
    //RemoteClientApiEnterprise RCE; // enable/disable for debugging
    public ServerScript(IServerApi api)
    {
        this.Logger = LogManager.GetLogger("ServerScript");
        this.Api = api;
    }

    public void Dispose()
    {
        // Use this method to cleanup any resources here (consider fully implementing the Dispose pattern).
        // For example, stop and dispose any started timers. Ensure that all threads that were started are stopped here.
        // DO NOT BLOCK in this method for a very long time, as this may block the applications current activity.
        if ( pvrSocketClient != null )
        {
            pvrSocketClient.Shutdown (SocketShutdown.Send);
        }
    }
}
```

```

        pvrSocketClient.Close ();
        pvrSocketClient = null;
    }
}
public IAsyncResult m_asynResult;
public AsyncCallback pvrCallBack ;
public Socket pvrSocketClient;
public string pvrSocketIP="192.168.10.202";
public int pvrSocketPort=49152;
string Password="jan" + "\n";
byte[] PasswordBytes;
string ConnectionStatus="Disconnected";
System.Text.ASCIIEncoding encoding= new System.Text.ASCIIEncoding();
public void Connect_To_Paviro()
{
    try
    {
        if (ConnectionStatus=="Disconnected")
        {
            //create the socket instance...
            pvrSocketClient = new Socket (AddressFami-
ly.InterNetwork,SocketType.Stream ,ProtocolType.Tcp );
            // get the remote Paviro IP address...
            IPAddress ip = IPAddress.Parse(pvrSocketIP);
            //create the end point
            IPEndPoint ipEnd = new IPEndPoint(ip.Address,pvrSocketPort);
            //connect to the remote host...
            pvrSocketClient.Connect (ipEnd );
            pvrSocketClient.Send(encoding.GetBytes(Password));
            ConnectionStatus="Connected";
            //watch for data ( asynchronously )...
        }
        WaitForData();
    }
    catch(SocketException se)
    {
        ConnectionStatus="Disconnected";
        //RCE=new RemoteClientApiEnterprise("127.0.0.1:5394","Admin", "");
        //RCE.ApplicationManager.ShowMessage("Exeption in CallPaviro: "+ se.Message );
    }
}
}

```

```
[Scriptlet("2d90b2e3-8f81-4fd7-a9a9-c861c31514c6")]
public void CallPaviroLVal1ON(EventData e)
{
    Connect_To_Paviro();
    try
    {
        pvrSocketClient.Send(encoding.GetBytes("LVal1 1"+ "\n"));
    }
    catch(SocketException se)
    {
        //RCE=new RemoteClientApiEnterprise("127.0.0.1:5394","Admin",
""");
        //RCE.ApplicationManager.ShowMessage("Exeption in CallPaviro: "+
se.Message );
    }
}
}
```

```
[Scriptlet("b4650ca1-cd66-4e17-ab69-c9dfb8949e6d")]
public void CallPaviroLVal1OFF(EventData e)
{
    Connect_To_Paviro();
    try
    {
        pvrSocketClient.Send(encoding.GetBytes("LVal1 0" +"\n"));
    }
    catch(SocketException se)
    {
        //RCE=new RemoteClientApiEnterprise("127.0.0.1:5394","Admin", "");
        //RCE.ApplicationManager.ShowMessage("Exeption in CallPaviro: "+
se.Message );
    }
}
}
```

```
public void WaitForData()
{
    try
    {
        if ( pvrCallBack == null )
        {
            pvrCallBack = new AsyncCallback(OnDataReceived);
        }
    }
}
```

```

    }
    CSocketPacket pvrSocPkt = new CSocketPacket();
    pvrSocPkt.thisSocket = pvrSocketClient;
    // now start to listen for any data...
    m_asynResult = pvrSocketClient.BeginReceive (pvrSocPkt.dataBuffer
,0,pvrSocPkt.dataBuffer.Length ,SocketFlags.None,pvrCallBack,pvrSocPkt);
    pvrSocPkt=null;
    }
    catch(SocketException se)
    {
        //RCE=new RemoteClientApiEnterprise("127.0.0.1:5394","Admin", "");
        //RCE.ApplicationManager.ShowMessage("Exeption in WaitForData: " + se.Message
);
    }
}

public void OnDataReceived(IAsyncResult asyn)
{
    try
    {
        CSocketPacket pvrSockId = (CSocketPacket)asyn.AsyncState ;
        //end receive...
        int iRx = 0 ;
        iRx = pvrSockId.thisSocket.EndReceive (asyn);
        char[] chars = new char[iRx + 1];
        System.Text.Decoder d = System.Text.Encoding.UTF8.GetDecoder();
        int charLen = d.GetChars(pvrSockId.dataBuffer, 0, iRx, chars, 0);
        System.String szData = new System.String(chars);
        if (!szData.Contains("OpenInt_IsAlive")) //ignore heart beats
        {
            //RCE=new RemoteClientApiEnterprise("127.0.0.1:5394","Admin", "");
            //RCE.ApplicationManager.ShowMessage("Received from Paviro:" + szData
);
        }
        WaitForData();
    }
    catch (ObjectDisposedException )
    {
        System.Diagnostics.Debugger.Log(0,"1","\nOnDataReceived: Socket has been
closed\n");
    }
}

```

```

        catch(SocketException se)
        {
            //RCE=new RemoteClientApiEnterprise("127.0.0.1:5394","Admin", "");
            //RCE.ApplicationManager.ShowMessage("Exception in OnDataReceived: " +
se.Message );
        }
    }

```

```

public class CSocketPacket
{
    public System.Net.Sockets.Socket thisSocket;
    public byte[] dataBuffer = new byte[50];
}

```

```

[Scriptlet("cbf4bf45-95ff-438f-bcee-0ff238a4c3f9")]
public void CallPaviroLVal2ON(EventData e)
{
    Connect_To_Paviro();
    try
    {
        pvrSocketClient.Send(encoding.GetBytes("LVal2 1" + "\n"));
    }
    catch(SocketException se)
    {
        //RCE=new RemoteClientApiEnterprise("127.0.0.1:5394","Admin", "");
        //RCE.ApplicationManager.ShowMessage("Exception in CallPaviro: "+ se.Message );
    }
}

```

```

[Scriptlet("8e56c284-4949-4d31-92f7-bf0acd340d59")]
public void CallPaviroLVal2OFF(EventData e)
{
    Connect_To_Paviro();
    try
    {
        pvrSocketClient.Send(encoding.GetBytes("LVal2 0" + "\n"));
    }
    catch(SocketException se)

```

```
{  
    //RCE=new RemoteClientApiEnterprise("127.0.0.1:5394","Admin", "");  
    //RCE.ApplicationManager.ShowMessage("Exception in CallPaviro: "+ se.Message );  
}  
}
```

## 5. Appendix B Debugging Server Script runtime actions

### 5.1 How to print debug information in the Operator Client GUI

Apart from diagnosing logging information in the BVMS server, some extra script code can also help you to debug script actions.

Due to the fact that BVMS server scripts lack displaying capabilities on the BVMS server (no GUI) you can use the Operator Client GUI to temporarily show debugging information of a server script for diagnostic purpose.

To enable a server script to print debugging info on the Operator Client GUI, the compiler needs a reference to the `Bosch.SDK.Enterprise.dll`.

How to add this is shown below.



Now you have added the DLL for the compiler, you also have to enable the corresponding code line in the editor space on the right by removing the 2 slashes in the beginning of the script (line 9) here :

```
using Bosch.Vms.Core;
using Bosch.Vms.SDK;
//using Bosch.Vms.SDK.Enterprise; // debugging tool to show diagnostic info in Operator Client GUI
// For instructions how to debug the script runtime , please consult the AppNote Appendix B
using System.Timers;
```

And also here :

```

[BMSScriptClass]
public class ServerScript : IDisposable
{
    private readonly IServerApi Api;
    private readonly ILog Logger;
    // RemoteClientApiEnterprise RCE; // enable/disable for debugging
    public ServerScript(IServerApi api)
    {
        this.Logger = LogManager.GetLogger("ServerScript");
        this.Api = api;
    }
}

```

For debugging, the PAVIRO server script of Appendix A contains already commented debug printing lines that in normal situations will not be executed. These code lines pairs look like this...

```

//RCE=new....etc
//RCE.Appl....etc

```

To execute these code lines for printing you should remove the proceeding 2 slashes (//) likewise. Next save and recompile (floppy icon left top and green V) and restart the Server (green light in Config Client).

**NOTE:**

Please DO NOT forget to disable printing debug code lines after finishing debugging (re-insert the removed comment slashes and compile/restart)

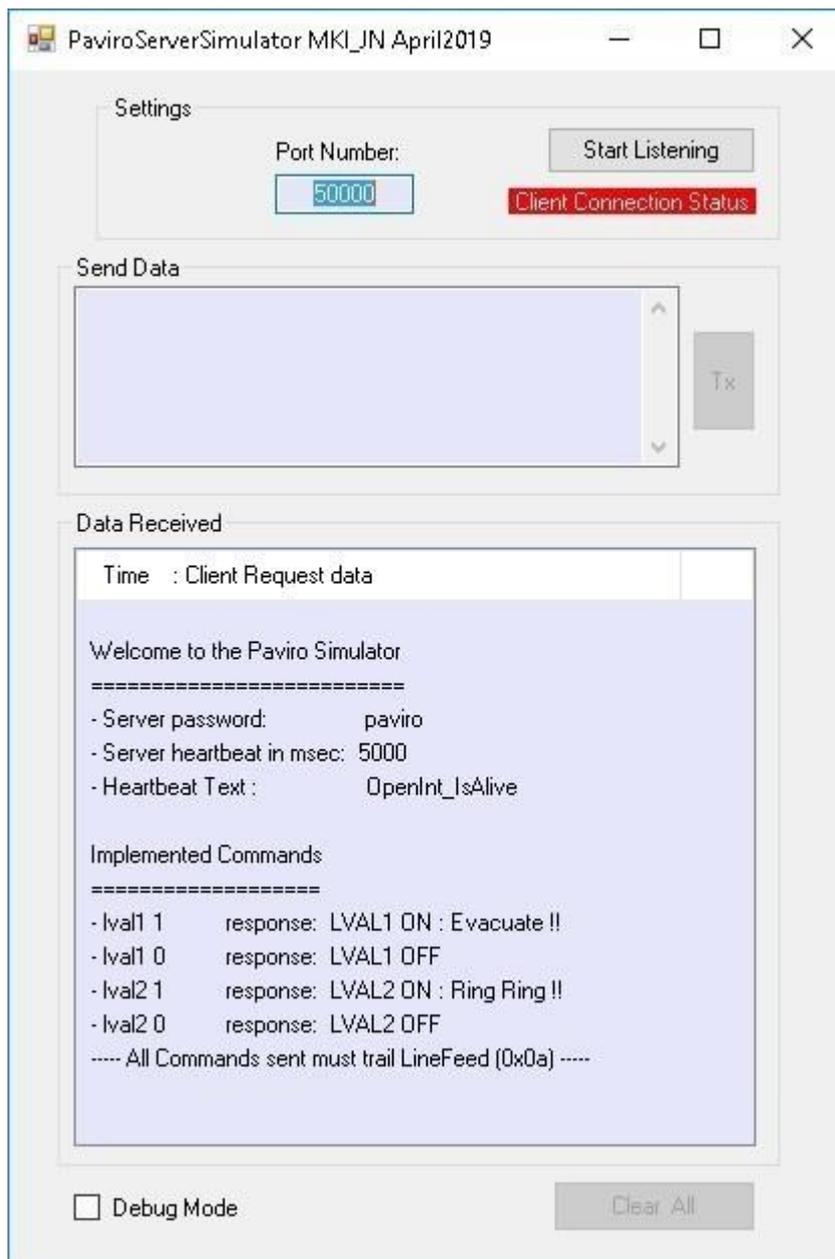
## 6. Appendix C Simulation Tools for Integrators

### 6.1 Simulation of the PAVIRO Controller Interface connection and its functions

In order to test the BVMS Script without the availability of the actual PAVIRO hardware, PAVIRO Simulator Software can be used.

This standalone PAVIRO (Server) Simulator software is a ZIP file compressed executable and simulates some initial but crucial PAVIRO commands and responses when you connect to it and fire calls at it.

The zip file must be unzipped to a directory on a PC (or even the BVMS server) location. The simulator can be started by running the file *PAVIROServerSimulator.exe* and looks like this:



The features and properties of the Simulator are shown in the DataReceived window at start-up. The Server simulates the PAVIRO password service as well as a Heartbeat broadcast to the Client. To test the BVMS script, you can add the Simulator IP, Port and Password to the script as indicated earlier. Of course the response from the Simulator to BVMS can be either visualized via a communication logging in the script to a file or added c# functionality that shows the response to an Operator Client Workstation.

Note:

1. For further details, please consult the BVMS documents or contact Bosch Integration Partner Program (IPP) support MKI department in Eindhoven the Netherlands.

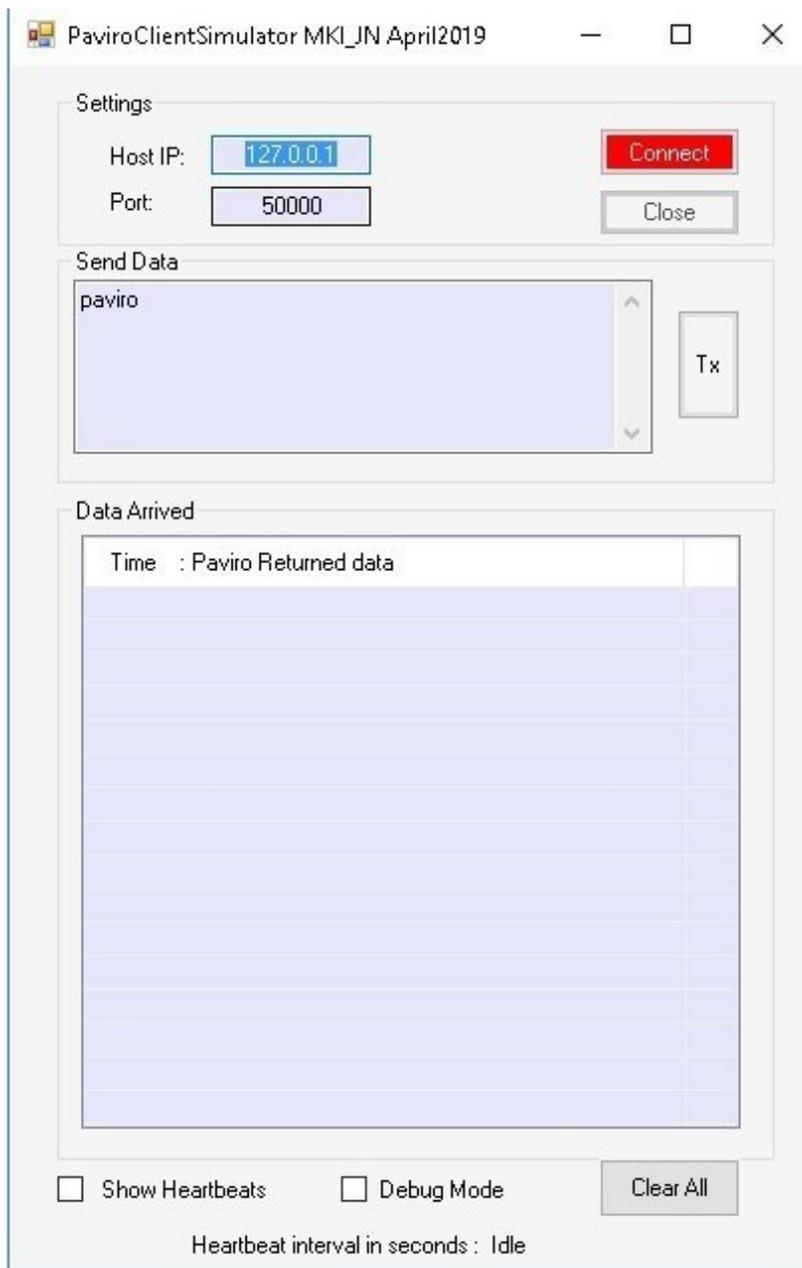
2. Both the PAVIROServerSimulator and PAVIROClientSimulator have an option to start with customized values like IP number , port used etc. To change these values to your own devices, the configuration files called **PAVIROServerSimulator.exe.conf** and **PAVIROClientSimulator.exe.conf** can be edited to match your requirements. These files can be found in your simulator install directory. The new values will be adopted at next start of the simulator. Please find the default values below:

```
PaviroServerSimulator.exe.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="ServerPassword" value="paviro" />
    <add key="HeartBeatIntervalInmsec" value="5000" />
    <add key="ListenPort" value="50000" />
  </appSettings>
</configuration>
```

```
PaviroClientSimulator.exe.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="ServerIP" value="127.0.0.1" />
    <add key="ServerPort" value="50000" />
    <add key="ServerPassword" value="paviro" />
  </appSettings>
</configuration>
```

## 6.2 Simulation of 3<sup>rd</sup> Party Applications to invoke a PAVIRO Call

Preprogrammed PAVIRO announcements can also be invoked from 3<sup>rd</sup> Party Applications ( Access Control systems, Fire systems or other dedicated applications. If these systems are not available yet, or you want to test the PAVIRO configuration programmed then these systems can be simulated with the PAVIRO Client Simulator which looks like this:

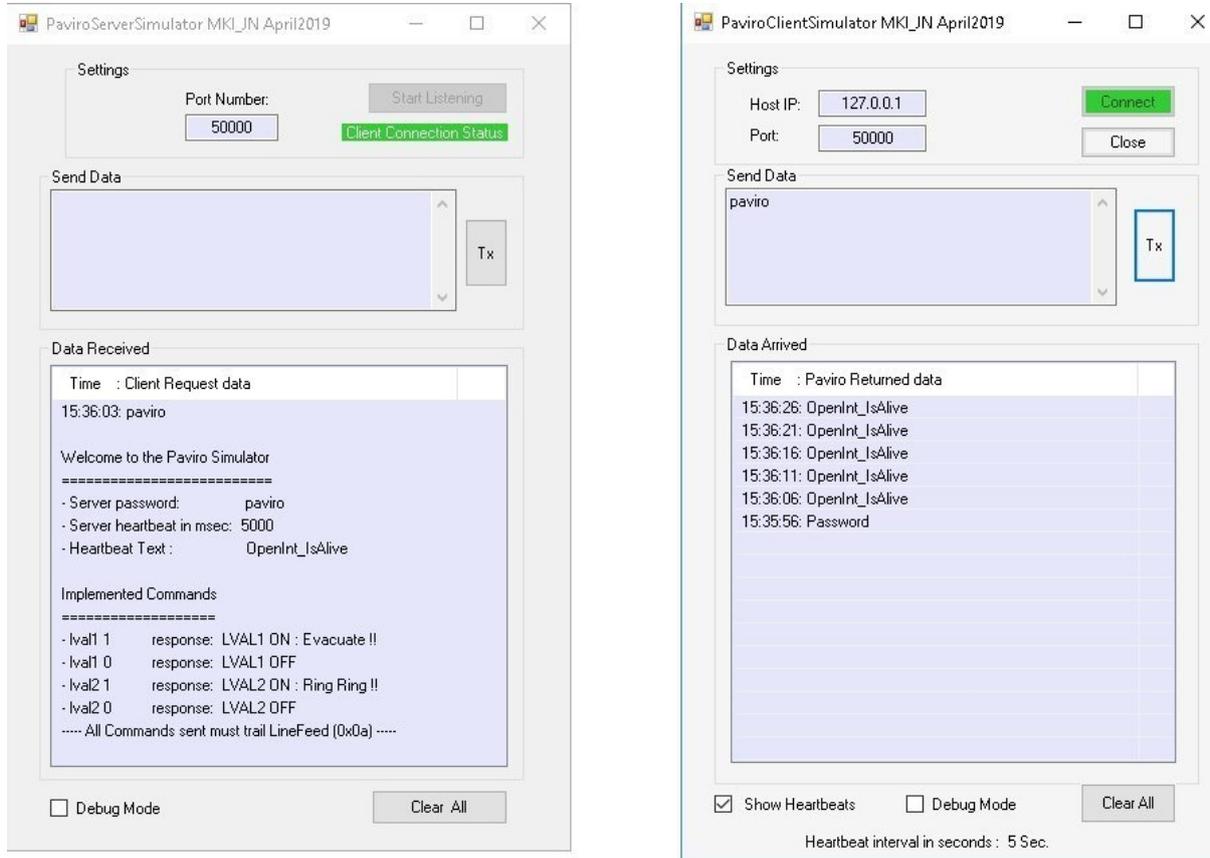


The Simulator Client can also be used to test any PAVIRO Interface command and syntax by simply typing the command syntax into the **SendData** Text Box and fire this to PAVIRO. You will also receive PAVIRO's response immediately in the **DataArrived** Text Box

### 6.3 Understanding the PAVIRO – BVMS protocol syntax, debugging developments

To understand the Protocol syntax both Server Simulator and Client Simulator can be used at the same time and connect to one another, either on the same PC or a network PC. The **SendData/ Data Arrived** Text Boxes in each simulator can then be used to communicate between both simulators testing the supported PAVIRO Commands and Callbacks.

In order to debug the communication protocol between the Server and Clients, the simulators can show the basic Hexadecimal values of received information. For this diagnostic behavior both simulator can be set into Debug Mode (tick box). This can be handy to diagnose existence of non-printable characters like Carriage Return/Line Feed and Esc characters.



**Note:**

To get a download of the simulators , please contact Bosch Integration Partner Program (IPP) support MKI department in Eindhoven the Netherlands.

## 7. References and Documentation

For Bosch Integration Partner Program (IPP) :

- <http://ipp.boschsecurity.com/en/>

For Bosch DIP 3000 and BVMS:

- [https://emea.boschsecurity.com/en/products\\_3/videosystems\\_17/recording\\_17/iprecording\\_17/divarip3000\\_18/divarip3000\\_18\\_21177](https://emea.boschsecurity.com/en/products_3/videosystems_17/recording_17/iprecording_17/divarip3000_18/divarip3000_18_21177)

For Bosch DIP 7000 and BVMS:

- [https://emea.boschsecurity.com/en/products\\_3/videosystems\\_17/recording\\_17/iprecording\\_17/divarip70002u\\_32/divarip70002u\\_32\\_34825](https://emea.boschsecurity.com/en/products_3/videosystems_17/recording_17/iprecording_17/divarip70002u_32/divarip70002u_32_34825)

For general Bosch Video Systems:

- [https://emea.boschsecurity.com/en/products\\_3/videosystems\\_17/videosoftware\\_17/videomanagement-systems\\_17/boschvideomanagementsyste\\_144/boschvideomanagementsyste\\_144\\_36910](https://emea.boschsecurity.com/en/products_3/videosystems_17/videosoftware_17/videomanagement-systems_17/boschvideomanagementsyste_144/boschvideomanagementsyste_144_36910)

For Bosch PAVIRO Systems:

- [https://emea.boschsecurity.com/en/products\\_3/videosystems\\_17/videosoftware\\_17/videomanagement-systems\\_17/boschvideomanagementsyste\\_144/boschvideomanagementsyste\\_144\\_36910](https://emea.boschsecurity.com/en/products_3/videosystems_17/videosoftware_17/videomanagement-systems_17/boschvideomanagementsyste_144/boschvideomanagementsyste_144_36910)

For Bosch PAVIRO information:

IRIS-Net Software and PAVIRO Documents:

[https://emea.boschsecurity.com/en/products\\_3/publicaddressandvoicealar\\_16/pacommercialsoundandemerg\\_16/PAVIRO\\_21/PAVIRO\\_21\\_products\\_62989](https://emea.boschsecurity.com/en/products_3/publicaddressandvoicealar_16/pacommercialsoundandemerg_16/PAVIRO_21/PAVIRO_21_products_62989)