

Table of contents

1 Overview of the Integration	3
1.1 Introduction.....	3
1.2 Which Bosch products can be used for this integration.....	3
2 How to set up the IP Horn Loudspeaker for this integration	4
2.1 Programming the IP Horn Speaker Web page settings	4
3 The integration of the IP Horn Speaker with Bosch Camera – an autonomous stand-alone system	7
3.1 Creating an Alarm Task Script in the Bosch camera.....	7
3.2 Testing the autonomous behavior and functionalities	8
4 The integration of the IP Horn Loudspeaker in a BVMS System	9
4.1 How to call a IP Horn Loudspeaker by a BVMS script ?.....	9
4.2 A basic BVMS Client or Server Scriptlet to trigger the IP Horn Loudspeaker.....	10
4.3 An alternative, more sophisticated BVMS Client or Server Scriptlet:	13
5 History	14
6 Disclaimer	14

1 Overview of the Integration

1.1 Introduction

In many Vertical Markets where people are on the move, central monitoring video surveillance systems expanded with automated additional Audio Assistance is a very welcome feature.

Examples are:

- A BVMS Operator spots undesired behavior and manually sends a specific recorded message to the area.
- A person crosses a virtual line in a video camera scene and triggers the start of a warning message.
- etc etc.

Before the IP Horn Speaker can be used over IP in order to broadcast pre-recorded messages, the installation should be programmed on both IP Horn Loudspeaker side as well as the Management side (BVMS) or in an individual camera that should trigger the message call via its internal camera events (like IVA) in an autonomous surveillance system.

1.2 Which Bosch products can be used for this integration.

1.2.1 IP Horn Loudspeaker and Amplifier types

The supported Types are:

- LHN-UC15L-SIP
- LHN-UC15W-SIP
- AMN-P15-SIP



Minimum Firmware required:

Bosch_LHN15SIP_Firmware_V1_0_233 (=Release Version)

1.2.2 Which Bosch cameras and BVMS versions are supported for audio integration?

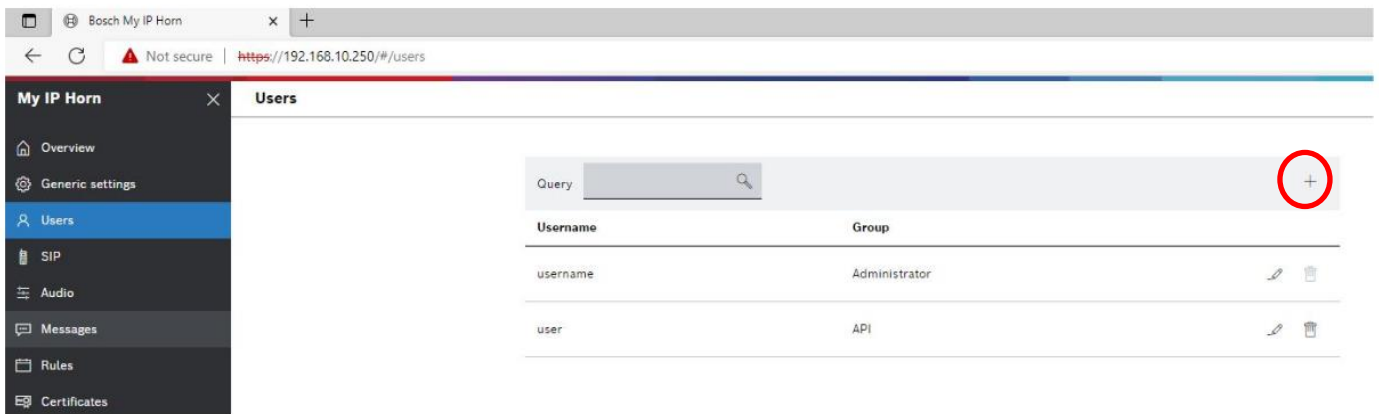
- In principle all Bosch IP cameras with FW 7.8x Versions and later supporting internal Alarm Task Script Language.
- BVMS version 9 and later.

2 How to set up the IP Horn Loudspeaker for this integration

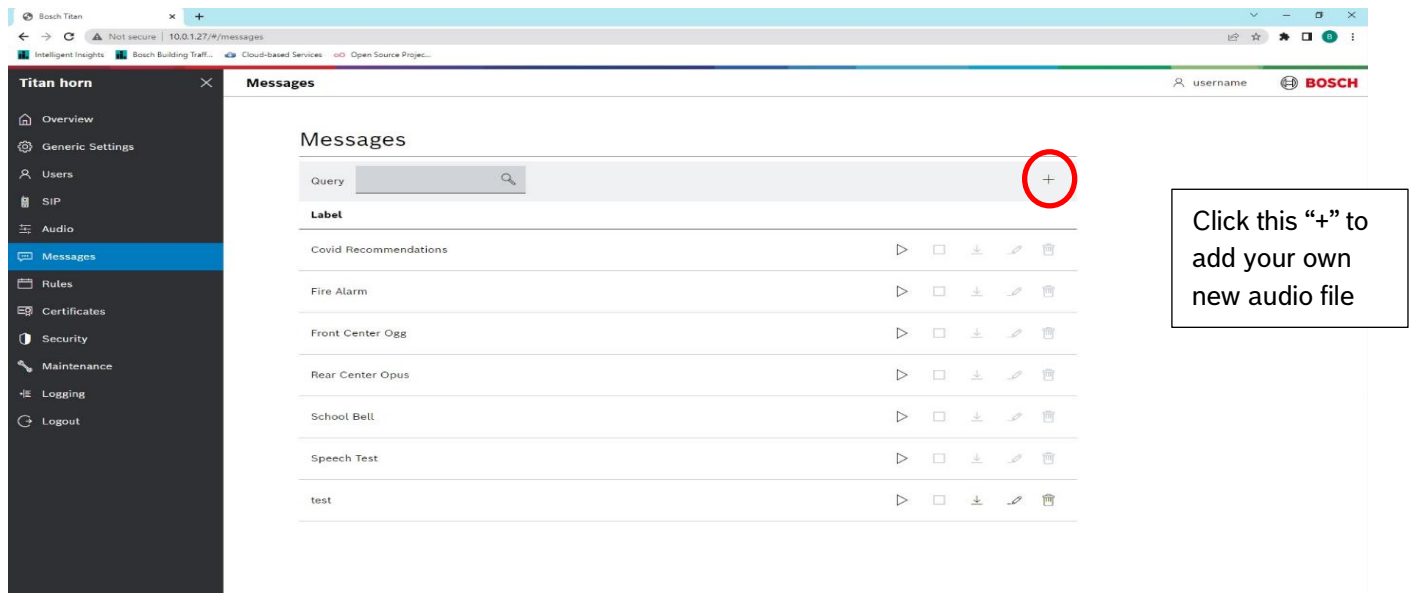
The IP Horn Speaker can be activated using an external generated HTTP(S) call that triggers the internal Virtual General Purpose Input (VGPI). An input triggered can then be predefined to activate for instance a predefined message. The following settings have to be programmed in the IP Horn Speaker to achieve this.

2.1 Programming the IP Horn Speaker Web page settings

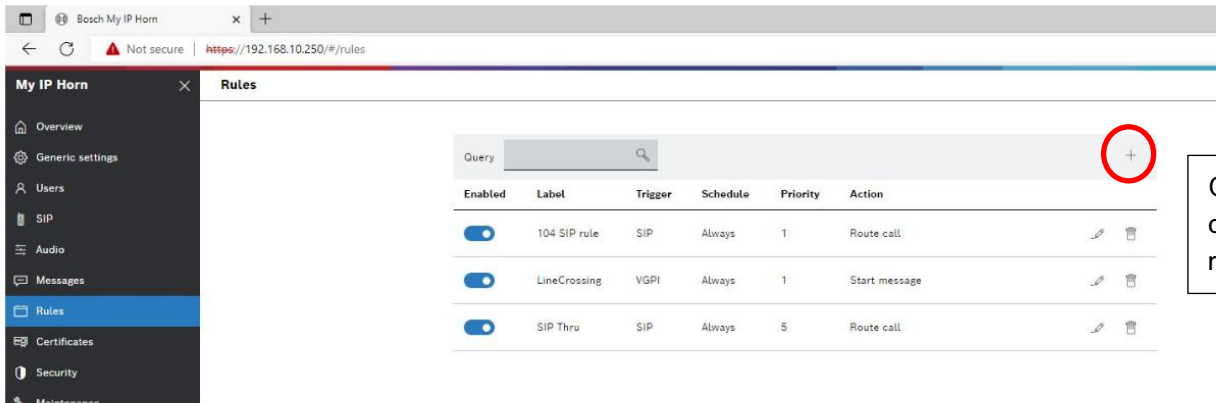
2.1.1 Open the Settings in the IP Horn Loudspeaker and create a new user with access to group API only



2.1.2 Import your customized audio file



2.1.3 Create a new Rule



The Rule menu looks like this:

<
Edit rule

Label
Line Crossing

Enabled

Priority
1

– +

Trigger settings

Trigger type
VGPI

Trigger starts but does not stop the action

Virtual GPI index
1

– +

Schedule type
Always

Action settings

Action type
Start message

– +

Covid message

Repeat count (0 is indefinitely)
1

Gain (dB)
20.0

GPO action
NONE

– +

Save

Cancel

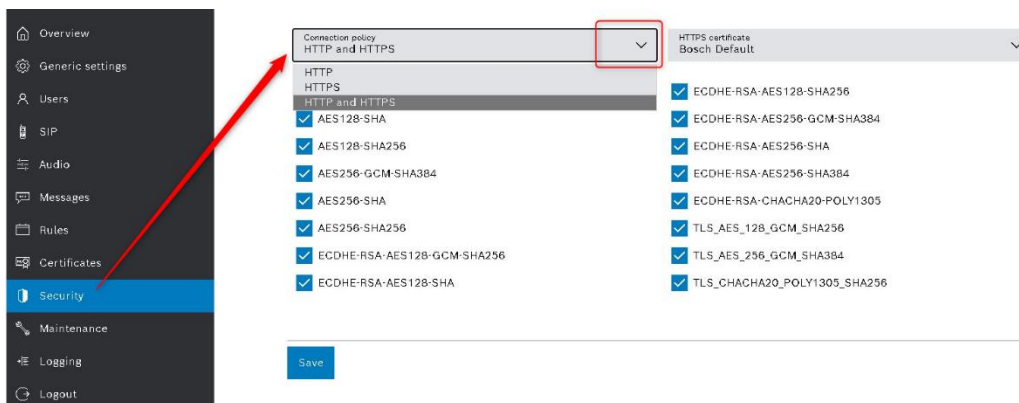
2.1.4 Hints on setting a new Rule.

Take note of the following :

- Priority is up to you to define
- Trigger Type should be VGPI (Virtual General Purpose Input)
- Virtual GPI Index – there are a total of 16 Virtual GPI available (1-16)
If you need to trigger multiple type of audio message, use a different Virtual GPI for each message.
- Action Settings :
 - Type should be Start_Message
 - “Trigger starts but does not stop the message” needs to be selected, avoiding a message cut off whenever a stop command is received. By selecting this option, the audio message will completely play ignoring any stop command entering before the audio message playback finishes. See also the timers comment in the script source codes below to match the playback time and virtual input toggle time needed to avoid message repeats or cut-offs.
- GPO action can be set to activate physical (relay) as well as virtual output (Rest API), in the IP Horn speaker.

2.1.5 Hints on using HTTP or HTTPS

The IP Horn Speaker Webbrowser config software lets you define in the Security Menu what type of communication policies the IP Horn Speaker must use. You can select HTTP or HTTPS or both.



Please make sure this setting also matches the security policy as used by the Scripts below both for ATSL scripts as also for BVMS scripts.

This completes the basic settings of the IP Horn Speaker.

You can now proceed to integrate the IP Horn Speaker in either a standalone system with Bosch cameras or integration into BVMS or other applications capable to launch the IP Horn Speaker messages.

3 The integration of the IP Horn Speaker with Bosch Camera – an autonomous stand-alone system

Note: Make sure you have programmed the IP Horn Loudspeaker as illustrated above first.

The IP Horn Speaker can produce messages automatically invoked by an Alarm Task script in the camera itself without using an external application. This allows the design of a fully autonomously surveillance system using the cameras Intelligent Video Analytics (IVA) ability to detect suspicious events and the IP Horn Loudspeaker to produce instructive audio messages.

3.1 Creating an Alarm Task Script in the Bosch camera

First you have to set an IVA trigger Rule in the Camera for the desired IVA event (i.e Crossed line) that should trigger the IP Horn Loudspeaker message. This can be done via the Config Manager software.

In the camera Alarm Task script editor, each IVA Rule defined has its own identification syntax like this:

VCARule(1,x)

The x should correspond with the IVA Rule used.

In the example Alarm Task script below, IVA rule 1 (first rule) is used.

Add the following example script to the Alarm Task script editor space (use your local IP number and password).

```

HttpCommand sendHttp_on :=
{Command("api/ext/v1/vgpis/1")SSL(false)Port(80)IP("192.168.10.250")Password("12345678")UserName("User"
)Method(POST)ContentType("application/json")Payload("true")ForceBasicAuth(true)};
HttpCommand sendHttp_off :=
{Command("api/ext/v1/vgpis/1")SSL(false)Port(80)IP("192.168.10.250")Password("12345678")UserName("User"
)Method(POST)ContentType("application/json")Payload("false") ForceBasicAuth(true)};
// Set TempState to monostable mode, 3 seconds
OperationMode monostable := { High(30) };
TempState(1) := monostable;
if(VCARule(1,1)) then TempState(1) := true;
if(TempState(1)) then sendHttp_on else sendHttp_off;

```

Note:

Make sure that this script compiles with no error. If compiling generate still errors and you are sure that the above syntax was exactly copied into your camera script editor area, then please manually retype with your local keyboard all quotation marks (") and (") in the script in order to make sure the compiler compiles to the correct local ASCII code.

Hints:

- The above script example has the following settings:
 - IP Horn Speaker IP address : 192.168.10.250. (**Please adapt to your local IP used**)
 - Username of IP Horn Speaker created in step 1: User (**Please adapt to your user**)
 - Password of user created in step 1 : 12345678 (**Please adapt to your password**)
 - In the IP Horn Speaker, VGPI "1" should be pre-defined to launch a message.
- The Alarm Task script will send an HTTP "On" command to the IP Horn Speaker when IVA rule 1 triggers, followed by a 3 seconds time delay, and an HTTP "Off" command to the IP speaker .
- The "OFF" command will reset the Virtual input in the IP Horn Speaker The message playback behaviour (like repetition, schedules etc) can adapted in the Action Settings of the Rule above.
- If the call should be send via an HTTPS connection (secured) then adapt following script elements in the scripts above: **SSL(false)Port(80) should be changed to SSL(true)Port(443)**. Also see chapter 2.1.5 for IP Horn Speaker to communicate in HTTPS
- The IP Horn Speaker does not (yet) support the Digest Authentication method, so be sure to always implement the **ForceBasicAuth(true)** parameter in your HTTP(s) command definitions. If you forget to do so it might still work however a message trigger will have a delay due to an automatic Digest to Basic fallback delay of about 1-5 seconds.

6. If you get compiler errors regarding **ForceBasicAuth(true)** then your camera Firmware is too old. See chapter 1.1.2
7. The user name as defined in the IP Horn Loudspeaker should be defined with quotation marks (") as follows **UserName("<name as defined in IP Horn Speaker>")**. If you get a compiler error but no error when not using quotation marks then your camera Firmware is too old. See chapter 1.1.2

3.2 Testing the autonomous behavior and functionalities

You can now test the application by triggering the IVA Rule. The message assigned should sound. Check the individual settings in both camera and IP Horn Speaker first if your system does not work. The camera internal IVA Rule trigger messages should be visible in the IP Horn Speaker config Manager whilst the message can be manually invoked by the VirtualInput simulation in the IP Horn Speaker Webbrowser Maintenance menu.

4 The integration of the IP Horn Loudspeaker in a BVMS System

Note: Make sure you have programmed the IP Horn Loudspeaker as illustrated in chapter 2 first.

The IP Horn Speaker can produce messages automatically invoked by events in a BVMS system using a Server Script or simply manually by a BVMS Operator using a Client Script in his Logical Tree or on a Map. Such a message call can be the result of a programmed Server or a Client Script added to the internal BVMS software using the BVMS Config Client Tools menu called the Command Script Editor.

4.1 How to call a IP Horn Loudspeaker by a BVMS script ?

BVMS embeds a C#/VB editor and compiler to create scripts. Please consult the BVMS documentation how to create C# or Visual Basic.Net language based scripts. There are 2 types of scripts possible in BVMS namely a Server script and a Client Script. You should decide upfront which type you need to create.

A server script can be activated by an event in BVMS while a Client script can only be invoked via an Operator click on a script icon in his logical tree or on a MAP.

This scriptlet example below uses the so called WebRequest Method in C#/VB to produce an HTTP(S) call to the IP Horn Loudspeaker. A pre-recorded message should be linked first to a Virtual General Purpose Input (VGPI) of the IP Horn Speaker (see chapter 2). This virtual input could be triggered for instance by an IVA Crossed Line event in BVMS via a server script or by the Operator via a client script.

This below C# Client script will enable a BVMS Operator to manually trigger the IP Horn Loudspeaker Virtual Input called VGPI 1 via a script “launch message” Icon in his Logical Tree.

Of course you can define a call of various different messages in below code example defined in the IP Horn Loudspeaker by taking a unique virtual input (VGPI 2 – VGPI=16 etc)

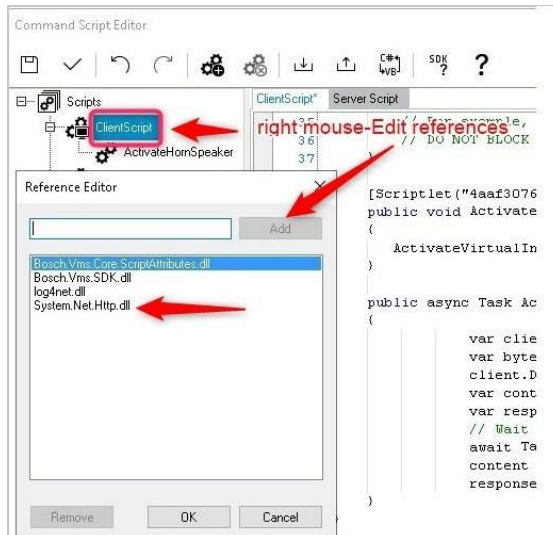
4.2 A basic BVMS Client or Server Scriptlet to trigger the IP Horn Loudspeaker

Here are the step by step instructions for a Client script how to accomplish such a test case.

The following References and C# code must be added to the Command Script Editor.

-Open the to the Client Script code editor space:

-Add the name System.Net.Http.dll to the References (right-mouse on ClientScript in the left screen – edit References)



-In the Code editor space on the right, press the Client script editor TAB, add the following **red lines** to the very top just below the first line using System;

```
using System.Net;
using System.IO;
using System.Text;
using System.Net.Http;
using System.Net.Http.Headers;
```

Now “right mouse click” on the text ClientScript in the left window (which let you add a New Scriptlet) and find your first new empty scriptlet that automatically will be added to the right window code edit area at the bottom, it looks like this

```
[Scriptlet(".....");
public void Clientscriptlet()
{
    // insert code here
}
```

Now replace the “`//insert code here`” in the above block and rename the name to create the first scriptlet called `activateMessage1` it should look like this:

```
[Scriptlet(".....");           // do not change this line
public void activateMessage1 () //change the default name to this name
{
    try
    {
        ActivateVirtualInput1("true"); // activate Virtual Input in IP Horn Speaker
        // create a delay, see chapter 2.14 to avoid that message cut off or repetitions occur.
        System.Threading.Thread.Sleep(1000);
        ActivateVirtualInput1("false"); // de-activate Virtual Input in IP Horn Speaker
    }
    catch (WebException we)
    {
        //write here your code to report the exception if needed.
    }
}
}

// The below code is a Public routine to be used by scriptlets to activate messages, like "activateMessage1" and can be pasted at this code location too.
private string username = "<user>"; //please adapt to local values
private string password = "<password>"; //please adapt to local values
private string IPHornIp = "xxx.xxx.xxx.xxx"; //please adapt to local values
public void ActivateVirtualInput1(string strGotoStatus)
{
    //ServicePointManager.ServerCertificateValidationCallback = (sender, certificate, chain, errors) => true; //See Note below
    WebRequest request = WebRequest.Create("http://" + IPHornIp + "/api/ext/v1/vgpis/1"); // See Note below for HTTPS
    string svcCredentials = Convert.ToBase64String(ASCIIEncoding.ASCII.GetBytes(username + ":" + password));
    request.Headers.Add("Authorization", "Basic " + svcCredentials);
    request.Method = "POST";
    request.ContentType = "application/json";
    request.ContentLength = strGotoStatus.Length;
    using (var stream = request.GetRequestStream())
    {
        byte[] data = Encoding.UTF8.GetBytes(strGotoStatus);
        stream.Write(data, 0, data.Length);
    }
    WebResponse response = request.GetResponse();
    using (var stream = response.GetResponseStream())
    {
        using (var reader = new StreamReader(stream))
        {
            // Api.ApplicationManager.ShowMessage(reader.ReadToEnd()); //In case you need to debug and to see response
        }
    }
}
}
```

4.2.1 Note regarding the use of HTTPS as opposed to HTTP between BVMS and IP Horn Speaker:

If the IP Horn Speaker defined connection is HTTPS then you must adapt the 2 code lines above to insert proper encryption/certificate validation. Please consult the internet help facilities.

You can however test HTTPS script by temporarily removing the comment brackets (//) of the `//ServicePointM....` code line which now when enabled, will always validate a returned certificate check. Again, after testing you must add your own code lines to be safe.

Also the `WebRequest request = WebRequest.Create("http://.....` must then be set to `WebRequest request = WebRequest.Create("https://.....`

Now save this script (left top, floppy) which compiles the code. Make sure there are no errors reported at the bottom status line by the compiler.

The Client Script can now be added to the Operator Logical Tree or MAP to visualise this script via a script icon.

To test, open the Operator Client and double click the script icon in the logical Tree to test the launch of the message.

A server script to be activated automatically by BVMS events , can be added to any BVMS Event via the Event TAB in the Configuration Client to automatically call a Message.

4.3 An alternative, more sophisticated BVMS Client or Server Scriptlet:

You could also use the following Client or Server Scriptlet in BVMS to obtain a dynamic Virtual Input selection and its timing for any of the available 16 Virtual Inputs in the IP Horn Loudspeaker to address other messages via other VGPIs inputs but using all the time the same general code to send the request to the IP Horn Speaker. In this way also multiple BVMS events scriptlets can be used to address a relevant unique VGPI and Messages combination in the IP Horn Speaker.

```
// ScriptType: ClientScript
// ScriptLanguage: CS
// Please add the name System.Net.Http.dll to the references in the left screen (right-mouse on ClientScript – edit References ) see picture on the right.....
```

```
using System;
using System.Diagnostics;
using System.Collections.Generic;
using Bosch.Vms.Core;
using Bosch.Vms.SDK;
using System.IO;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Net.Http;
using System.Net.Http.Headers;
```

```
[BvmsScriptClass()]
```

```
public class ClientScript : IDisposable
```

```
{
    private readonly IClientApi Api;
    private readonly ILog Logger;

    public ClientScript(IClientApi api)
    {
        this.Logger = LogManager.GetLogger("ClientScript");
        this.Api = api;
    }
}
```

```
public void Dispose()
{
```

```
    // Use this method to clean up 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.
}
```

```
[Scriptlet("4aaf3076-6b6a-41e3-8b99-2c2e9e6af18c")]
```

```
public void ActivateHornSpeakerAwait() //this is the generic call syntax that can be used in BVMS in any BVMS local scriptlet
{
```

```
    // select any of the 16 VGPIs and call the same activation process with adapted timing
    ActivateVirtualInput(1,1000); //activate for 1000 mSec ,see chapter 2.14 to avoid that message cut off or repetitions occur.
}
```

```
public async Task ActivateVirtualInput(int VGPI,int activationTime)
{
```

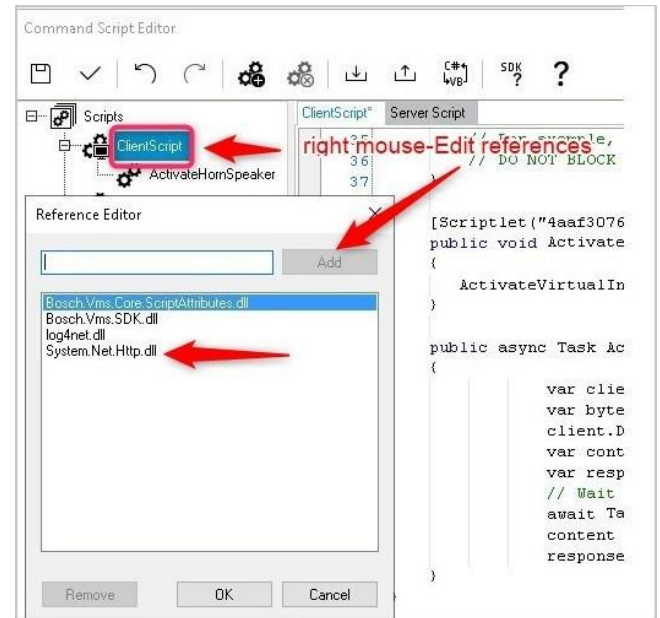
```
    var client = new HttpClient();
    var byteArray = Encoding.ASCII.GetBytes("user:12345678"); //user your local defined account in the IP Horn Speaker
    client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Basic", Convert.ToBase64String(byteArray));
    var content = new StringContent("true", Encoding.UTF8, "application/json");
    var response = await client.PostAsync("http://<Horn Speaker IP>/api/ext/v1/vgpis/"+VGPI, content);
    // Wait for "activationTime" mSeconds
    await Task.Delay(activationTime);
    content = new StringContent("false", Encoding.UTF8, "application/json");
    response = await client.PostAsync("http://<Horn Speaker IP>/api/ext/v1/vgpis/"+ VGPI, content);
}
```

4.3.1 Note on using IP Horn Speaker Hostname :

- You can also use the IP Horn Speaker Hostname for the parameter <Horn Speaker IP> in order to be IP independent i.e "LHN15SIP-11A95D" provided the DNS is set or DNS DHCP is enabled in the IP Horn Speaker.

The above hex value 11A95D in the Hostname name are the last 4 octets of the IP Horn Speaker MAC address

The DNS way will cause a slight extra start delay to launch the message.



5 History

VERSION	DATE	AUTHOR	DESCRIPTION
V 1.0	April 2023	Jan Noten (BT-VS/MKR-EU) Eindhoven The Netherlands	First version for distribution
V 1.1	June 2023	Jan Noten (BT-VS/MKR-EU) Eindhoven The Netherlands	Renewed simple script, added HTTPS notes, hints
V 1.2	July 2023	Jan Noten (BT-VS/MKR-EU) Eindhoven The Netherlands	Adapted "User" in ATSL script, added notes about Camera FW and IP Horn Speaker FW incompatibility issues

6 Disclaimer

Bosch cannot not accept any liability on the implementation or use of scripts mentioned in this document. Your activity in developing products that interface with Bosch products is at your own risk and responsibility regarding fitness for use, completeness, faultlessness, or any claims of third parties which may arise based on such further development.

Bosch Security Systems B.V.

P.O. Box 80002 | 5600 JB Eindhoven

THE NETHERLANDS | www.boschsecurity.com