Integration solution
IP Horn Loudspeaker - Bosch Video Systems
An automated Audio intervention Solution

Table of contents

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 triggering 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 an individual camera that should trigger the message call via IP.

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?

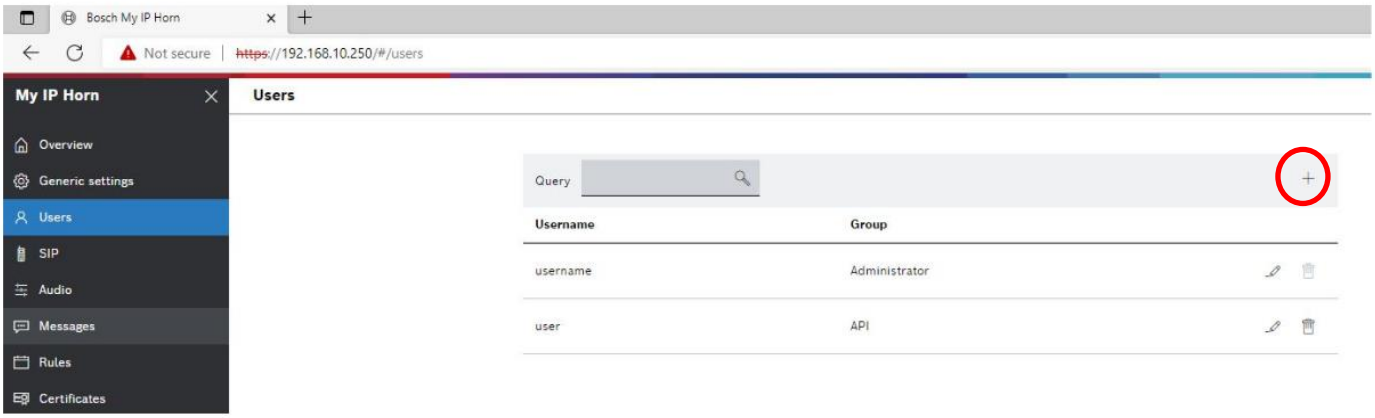
- In principle all Bosch IP cameras with FW6x, 7x or 8x Versions supporting internal Alarm Task Script Language.
- BVMS version 9 and later.

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

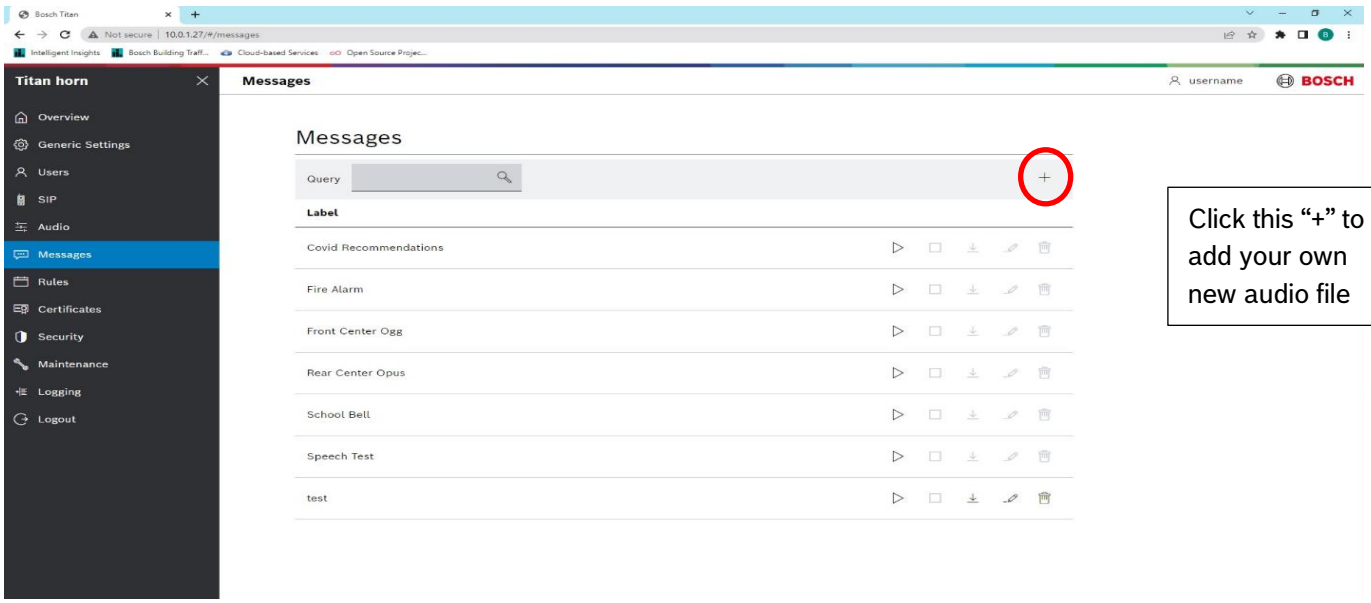
The IP Horn Speaker can be activated using an external generated HTTP(S) call that triggers the internal Virtual General Purpose Input (VGPI). An input triggered can then be predefined to activate for instance a predefined message.
The following settings have to be programmed in the IP Horn Speaker to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

2.1 Programming the IP Horn Speaker Web page settings

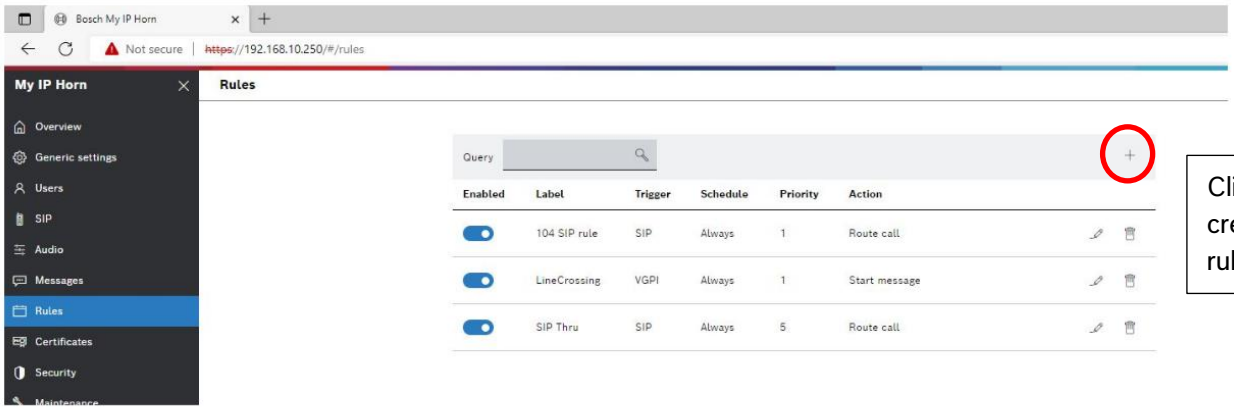
2.1.1 Open the Settings in the IP Horn Loudspeaker and create a new user with access to group API only



2.1.2 Import your customized audio file

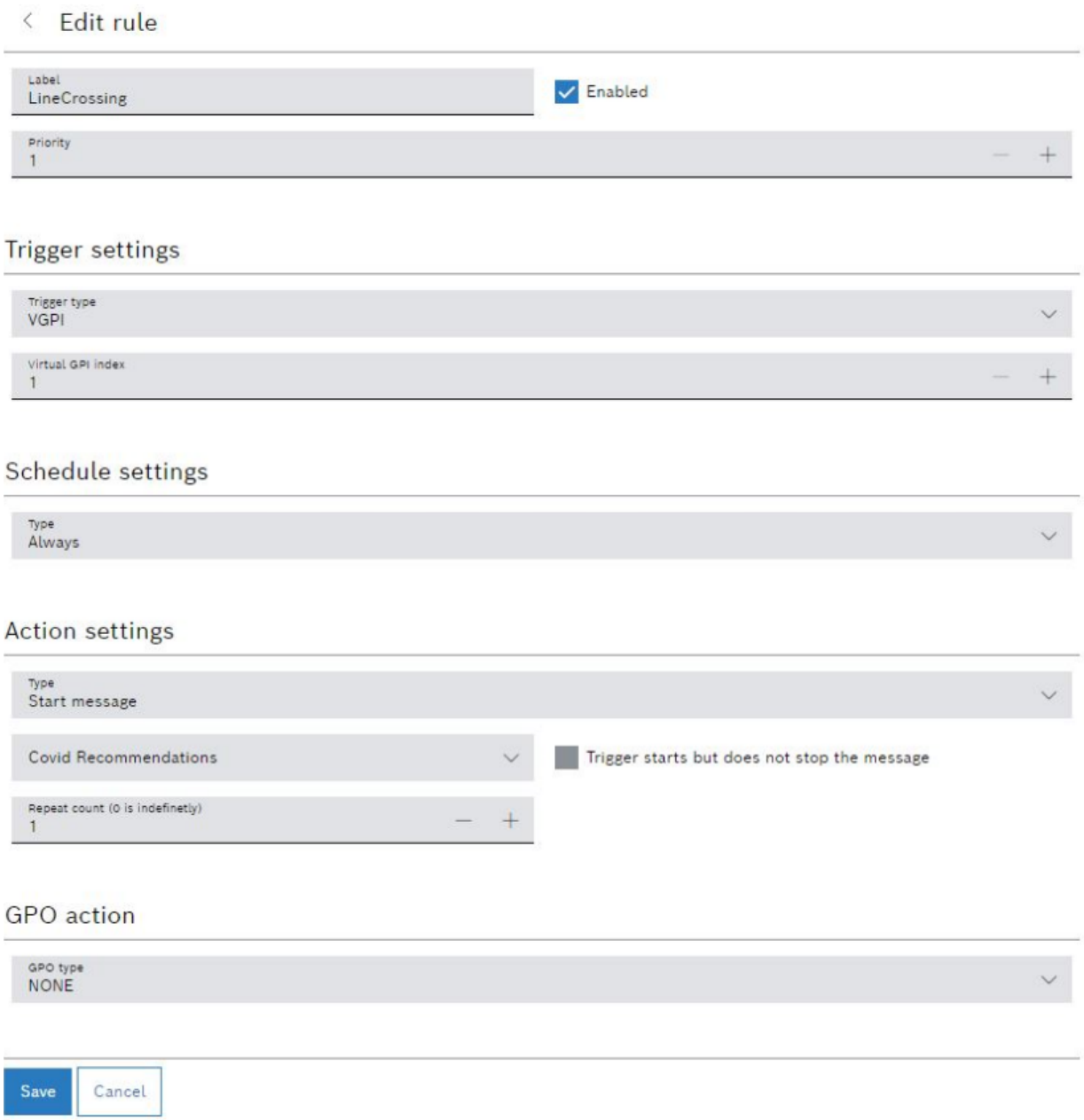


2.1.3 Create a new Rule



Click this “+” to create a new rule

The Rule menu looks like this:



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 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” need to be selected, else the audio will be immediately cut off when a stop command is been received. By selecting this option, the audio message will complete playing it full message even though the stop command comes in before the audio message is played completely.
- GPO action need not check it check box, as there will not be any triggering of the output on the speaker.

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 a so called ATSL 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 ATSL script, 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 of the camera (please adapt the IP number , user and password to the local situation).

```
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 complier compiles the correct local ASCII code.

Hints:

1. The above script example has the following settings:
   a. IP Horn Speaker IP address : 192.168.10.250. (Please adapt to your local IP used)
   b. Username of IP Horn Speaker created in step 1: User (Please adapt to your user)
   c. Password of user created in step 1 : 12345678 (Please adapt to your password)
   d. In the IP Horn Speaker, VGPIS “1” should be pre-defined to launch a message.
2. 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.
3. Not sending the “OFF” command will result in the IP Horn Speaker continue to play the audio file in loop. This behaviour can adapted in the Action Settings of the Rule above.
4. If the call should be send via an HTTPS connection (secured) then you have to change to following script elements in the scripts above:
   SSL(false)Port(80) should be changed to **SSL(true)Port(443)**.
   Note: If this does not work then make sure that you enabled the IP Horn Speaker to communicate in HTTPS as well (see in the config software the Security Menu)

5. The IP Horn Speaker does not (yet) support the Digest Authentication method so be sure you 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 of about 1-5 seconds.

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 you system does not work. The IVA Rule trigger should be visible in the config Manager whist the message can be invoke 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 above 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 script ?

BVMS embeds a C#/VB editor and compiler to create scripts. Please consult the BVMS documentation how to create C# language 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. 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.

Find below an example script in C# for the IP Horn Loudspeaker configuration as described in chapter 2.

As a result of this script, a BVMS Operator will then be able to 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 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 C# code must then be added to the Client Script code editor space:

```
//add the following red lines to the very top of the client script editor just below the line
using System;
using System.Net;
using System.IO;
....
[BvmsScriptClass()]
public class ClientScript :IDisposable
{
// add the following 2 lines here
private WebRequest   wrActivateMessage;
private Stream objStream;
....
....
....
```

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

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

Change the lines of the above block to create the first scriptlet called **activateMessage1**
it should look like this:

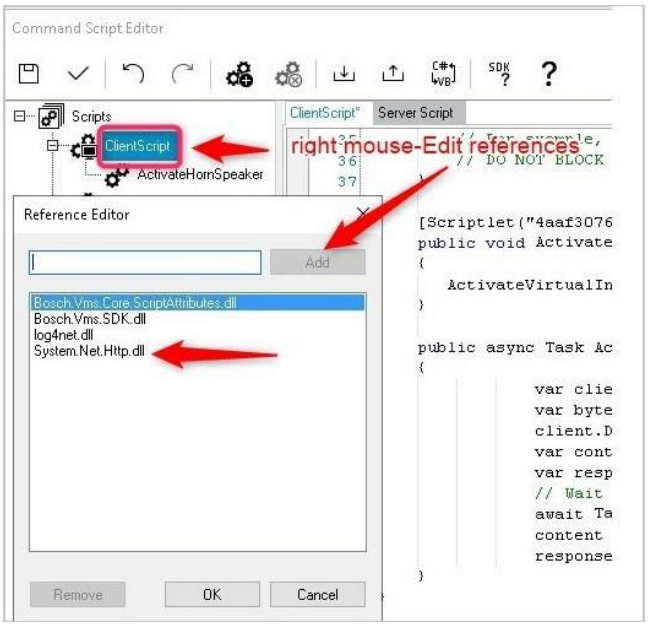
```
[Scriptlet(“……..);                  // do not change this line
public void activateMessage1 ()     //change the default name to this name
{
 try
   {
          wrActivateMessage = WebRequest.Create(https://<Horn Speaker IP>/api/ext/v1/vgpis/1);
           // Use above your local IP Horn Speaker IP as the correct Virtual Input X in IP Horn Speaker. The trailing /X is the Virtual input number X as used in the IP Horn Speaker
          wrActivateMessage.Credentials = new NetworkCredential("Your IP horn Speaker defined user","its localpassword");
          // set above your account values to access your IP Horn Loudspeaker
          wrActivateMessage.PreAuthenticate = true;
          wrActivateMessage.Method= “POST”;
          wrActivateMessage.postData= “true”;
          objStream = wrActivateMessage.GetResponse().GetResponseStream();
          objStream.Close();
          System.Threading.Thread.Sleep(2000); // Wait 2 seconds
          wrActivateMessage.postData= “false”;  // Reset the IP Horn Speaker Virtual Input
          objStream = wrActivateMessage.GetResponse().GetResponseStream();
          objStream.Close();
   }
 catch (WebException we)
   {
      Logger.Info (we);
   }
}
```

Now save this script (left top, floppy) which compiles the code. Make sure there are no errors reported at the bottom status line.
A Client Script can be added to the Operator Logical Tree or MAP to visualise this script via a script icon.
Double click the script icon in the logical Tree to test the launch of the message.
Likewise, a Server Script can be added to any Event as available in 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 timing length 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 code for sending the request to the IP Horn Speaker. In this way also multiple BVMS events scriptlets can be used to address a relevant unique VPGI 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  log4net;
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()
      {
            // select any of the 16 VGPIs and call the same activation process with adapted timing
            ActivateVirtualInput(1,2000); //activate 2000 mSec Virtual General Purpose Input 1
      }

      public async Task ActivateVirtualInput(int VGPI,int activationTime)
      {
            var client = new HttpClient();
            var byteArray = Encoding.ASCII.GetBytes("user:12345678");        //make sure this account  is defined in 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

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