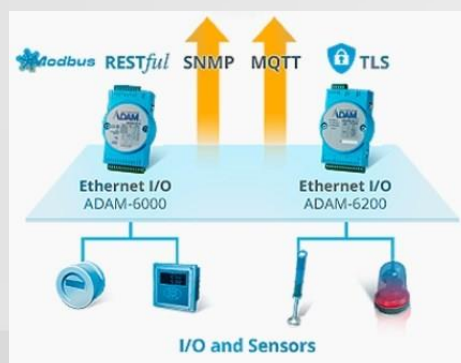
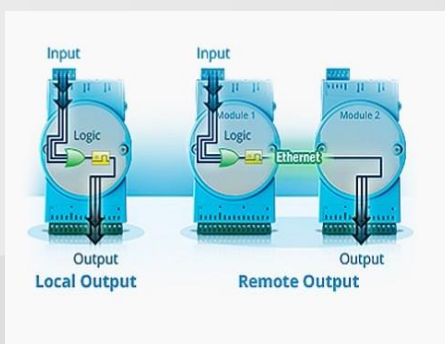


## Bosch IVA managed network I/O Automation

Advantech I/O Adam Modules autonomously controlled by Bosch “on the edge” IVA technology



# Table of Contents

- 1. Overview of the Integration 3
  - 1.1 Introduction .....3
  - 1.2 Principle and benefit of the integration .....3
  - 1.3 Advantech products for this integration .....4
  - 1.4 Bosch products for this integration .....4
- 2. Technical details and setup of the integration 4
  - 2.1 Advantech ADAM I/O device preparation .....4
  - 2.2 Bosch Camera device preparation .....5
  - 2.3 Alarm Task Script preparation .....6
- 3. Test and see the integration at work! 6
  - 3.1 Test Set up .....6
  - 3.2 Expanding IVA Rule transparency to other systems via multiple I/O Adam outputs.....8
- 4. History 9
- 5. Disclaimer 9

# 1. Overview of the Integration

## 1.1 Introduction

In many camera systems installed for observation and or security purposes, the camera is used as an information gathering source, providing video streams and I/O alarm facilities as part of the total system. Information, such as scene details in the provided camera's stream, can be sent to monitoring centers where operators can interpret the relevance of the information. According to his or her human involvement in this case, the follow-up actions for this information for the end-user are manually invoked.

In systems where no operators are available, the same camera information can be diagnosed internally; programmed and processed via smart algorithms in the camera and, as a result, IP connections are invoked to follow-up automation by other smart devices on the network. In this AppNote, you will find such an automation process between **Bosch Cameras** and **Advantech Adam I/O devices**.

In various Verticals, this Bosch - Advantech combined solution can fully automate adequate follow-up actions for the end-user without the cost of a video management system and human intervention.

## 1.2 Principle and benefit of the integration

In principle, a Bosch Camera embeds Intelligent Video Analytic algorithms that can be utilized internally as source for its own Alarm Task engine. This Alarm Task Engine has a dedicated Script editor (a sort of Macro editor and compiler) that allows an installer to combine internal events, I/O triggers as well as the output of IVA rules for smart follow up automation. These powerful macros are called ATSL Scripts.

These installer programmed ATSL scripts are the smart engine for controlling the Advantech Adam devices and to propagate the processed camera intelligence to end-user devices as automation.

The concept can be expanded as required and is not limited to a single camera, single Adam device. On the contrary, one camera as "smart detector" can control multiple I/O devices in your network.

### Some examples:

1. A camera observing an overnight storage room controls lights by controlling various I/O devices in the network.
2. A camera IVA rule violation automatically locks doors, closes a barrier and sends an e-mail to security guards.
3. A camera event activates contacts in AMAX control panels on a cruise ship
4. A camera sends emergency related relay closures to multiple Adam units over various networks.
5. A carpark camera detects loitering incidents and triggers an automatic deterrent audio message

The script integration, once programmed, is autonomous and works independently.

### 1.3 Advantech products for this integration

All latest Advantech Adam I/O D-Series units with REST-API interfacing will work. (min. Firmware Version v6.11 B20)

These units are :

ADAM 6050 – D  
ADAM 6051 – D  
ADAM 6052 – D  
ADAM 6060 – D  
ADAM 6066 – D  
ADAM 6224 – AE  
ADAM 6250 – B  
ADAM 6256 – B  
ADAM 6260 – B  
ADAM 6266 – B



Remember only outputs are controlled via the Camera Script. Not Inputs.

Please find references to the ADAM-6000 series below:

[https://www.advantech.com/products/ethernet-i-o-modules-adam-6000/sub\\_a67f7853-013a-4b50-9b20-01798c56b090](https://www.advantech.com/products/ethernet-i-o-modules-adam-6000/sub_a67f7853-013a-4b50-9b20-01798c56b090)

and the ADAM-6200 series:

[https://www.advantech.com/products/ethernet-i-o-modules-with-daisy-chain-adam-6200/sub\\_7447e150-338d-402d-b5a1-c9ce6d98816e](https://www.advantech.com/products/ethernet-i-o-modules-with-daisy-chain-adam-6200/sub_7447e150-338d-402d-b5a1-c9ce6d98816e)

### 1.4 Bosch products for this integration

For this integration, basically all Bosch Camera types will work provided that:

1. The camera Firmware Version is min. 7.51 or newer
2. The camera supports Adam's Basic Authentication protocol (which all Bosch cameras do)
3. The camera supports ATSL scripting (which all Bosch cameras do).

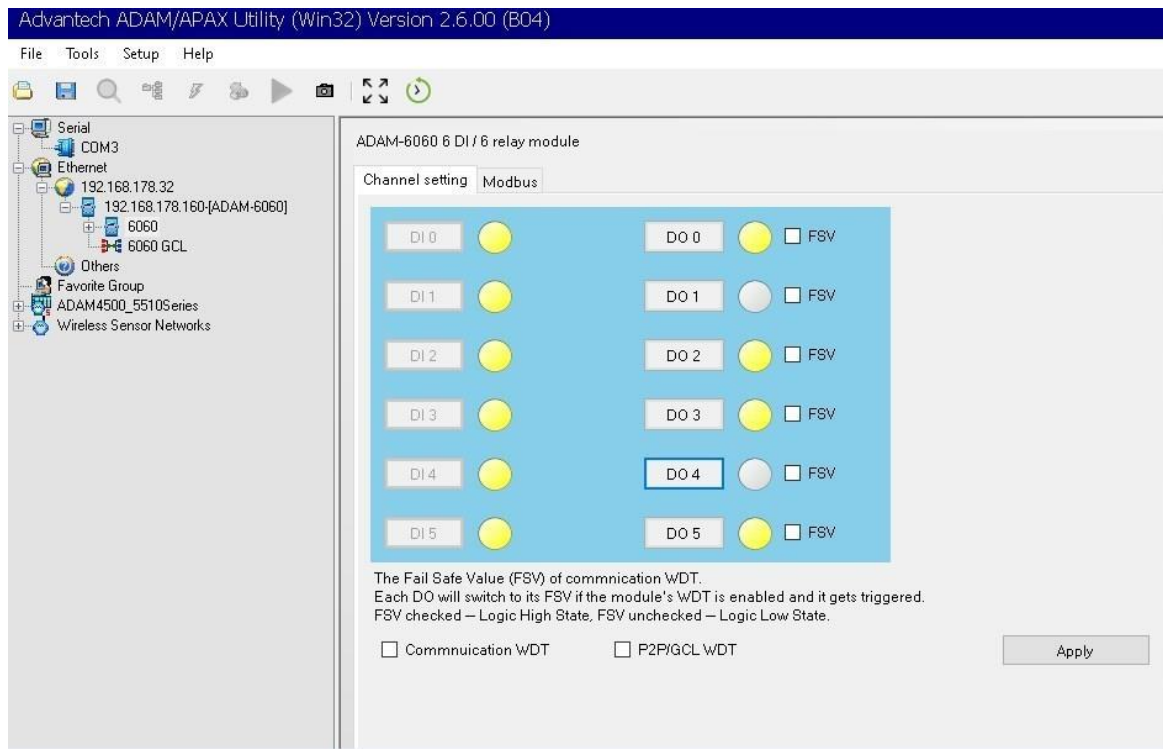
## 2. Technical details and setup of the integration

### 2.1 Advantech ADAM I/O device preparation

There is not much preparation for the ADAM units as long as the unit complies with the chapter 1.3 requirements. Best practise is to start the ADAM unit in a factory default setup and set the device IP address.

You can use the **Advantech AdamApax .Net Utility tool** for this.

This tool was also used in the test screenshots of this document to show the ADAM relay status while testing.



**ADAM 6060 Relay outputs as shown in Advantech AdamApax .Net Utility**

Please download **AdamApax .Net Utility** here:

[https://support.advantech.com/support/DownloadSRDetail\\_New.aspx?SR\\_ID=1-2AKUDB&Doc\\_Source=Download](https://support.advantech.com/support/DownloadSRDetail_New.aspx?SR_ID=1-2AKUDB&Doc_Source=Download)

or download directly the installer pack here:

[https://downloadt.advantech.com/download/downloads.aspx?File\\_Id=1-1Z05YL7](https://downloadt.advantech.com/download/downloads.aspx?File_Id=1-1Z05YL7)

## 2.2 Bosch Camera device preparation

To autonomously control the Adam units by the camera's Intelligent Video Analysis (IVA) or Essential Video Analysis (EVA), the camera must be set up to detect incidents or local site events that need adequate automatic follow up i.e. **line crossing, idle object, entering/leaving field, loitering** etc.

The tool to set up these functions is provided by the Bosch Configuration Manager and can be downloaded here:

<https://downloadstore.boschsecurity.com/index.php?type=CM>

There are also various instruction videos published on YouTube, for example:

IVA configuration:

<https://www.youtube.com/watch?v=Ue3Vmqbvem8&list=PLz97rFi-OzLdPrzs-SR-g1oo4RkV9mvf5>

IVA calibration:

<https://www.youtube.com/playlist?list=PLz97rFi-OzLf12F4sP-ocX3BhX0rtJvgt>

## 2.3 Alarm Task Script preparation

The Bosch Camera Alarm Task engine is the camera's on-board smart interface between the cameras IVA detection events as input and the resulting controlled Adam device as output.

The autonomous operation of this solution is based on the smartness of the Alarm Task Script that manages the above process flow as a pre-programmed follow-up of site activities.

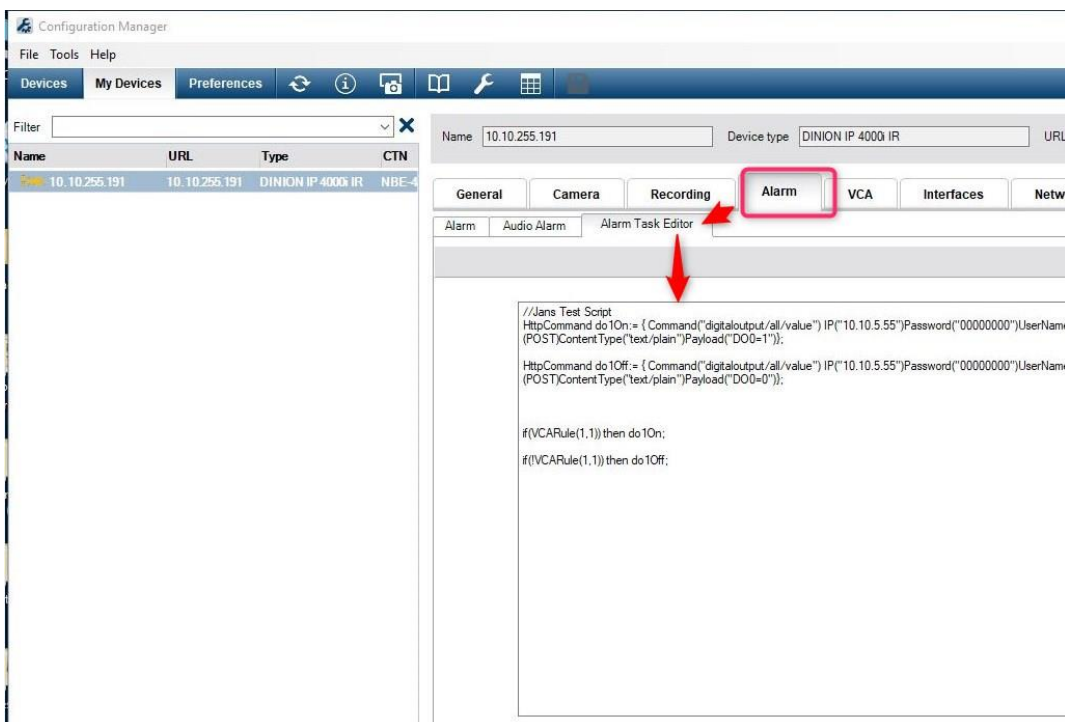
Example: A door will be locked for a person crossing a line allowing only traffic in the other direction.

The Script can be added, edited and compiled all via the Bosch Configuration Manager tool.

<https://downloadstore.boschsecurity.com/index.php?type=CM>

By mapping IVA inputs to Adam outputs for various preprogrammed scenarios and using smart Alarm Task Script filters, timers, logical booleans combined with internal camera logical and virtual inputs, the automation can be smart, tailor-made and endless.

This is what an example Alarm Task Script looks like in the Configuration Manager:



## 3. Test and see the integration at work!

### 3.1 Test Set up

The Test described below shows the basic concept of forwarding an IVA Rule trigger in the camera to an IP connected ADAM I/O device.

Once you have set up the ADAM IP address and line trigger for your application in the camera (using the Configuration Manager or WebBrowser), the smart interfacing via the Alarm Task script can be programmed.

In this test, we will use an IVA rule alarm (Rule 1) to control the ADAM Data output 0(=DO0) transparently.

Please adapt the Adam device properties such as IP and password if needed.  
The UserName must be "root"

The script to do this automation looks like this:

//VCA Profile 1 Rule 1 to Adam Relay DO0

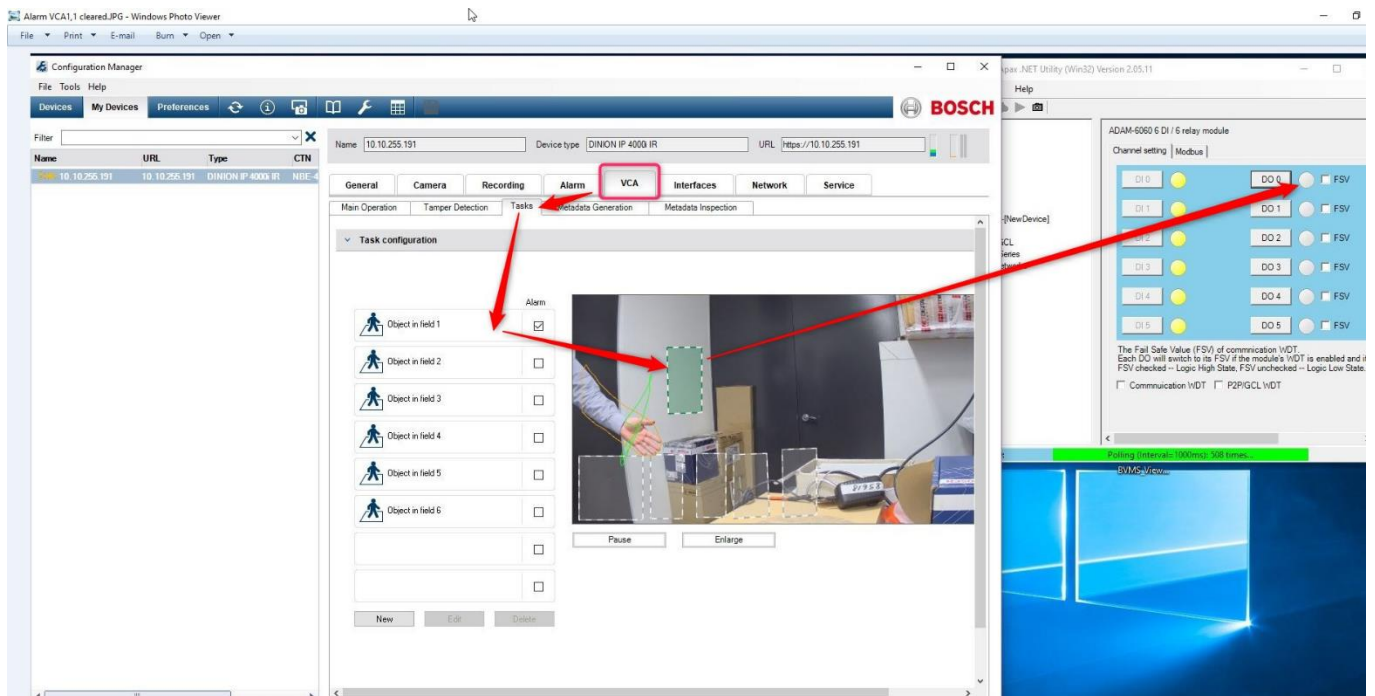
```
HttpCommand dataOutput0on:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")  
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO0=1")};  
HttpCommand dataOutput0off:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")  
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO0=0")};
```

if (VCARule(1,1)) then dataOutput0on;

if (!VCARule(1,1)) then dataOutput0off;

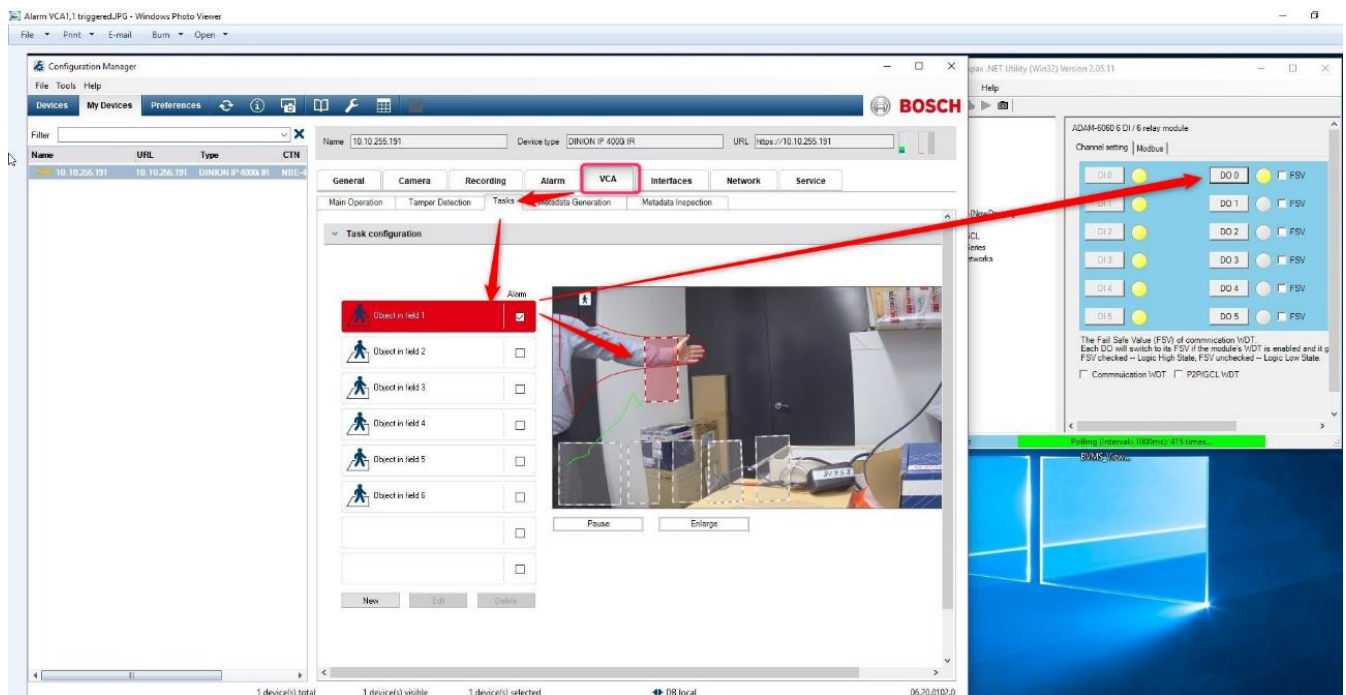
After compiling with 0 Errors, you can check the result in the GUI in the Advantech AdamApax .Net Utility as shown below.

First check if the Adam relay is idle when there is no IVA Rule Alarm (green)





Then create a IVA rule Alarm (red) and check that the Relay status turns to ON.



### 3.2 Expanding IVA Rule transparency to other systems via multiple I/O Adam outputs.

As indicated, the Alarm Task Scripting editor can be used to implement tailor-made IVA controlled routing to multiple Adam Outputs.

This flexible routing principle makes it possible to signal IVA events generated from one location to remote locations as desired.

Another advantage is that in this autonomous camera controlled system, the IVA Alarms can be interfaced to any other low or high level system.

**Example: Alarm Task Script of 6 IVA Rule Alarms routed to 6 individual outputs:**

//VCA Profile 1 Rule 1 Adam Relay DO0

```
HttpCommand do0On:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO0=1")};
HttpCommand do0Off:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO0=0")};
if (VCARule(1,1)) then do0On;
if (!VCARule(1,1)) then do0Off;
```

//VCA Profile 1 Rule 2 Adam Relay DO1

```
HttpCommand do1On:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO1=1")};
HttpCommand do1Off:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO1=0")};
if (VCARule(1,2)) then do1On;
if (!VCARule(1,2)) then do1Off;
```



//VCA Profile 1 Rule 3 Adam Relay DO2

```
HttpCommand do2On:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO2=1");
HttpCommand do2Off:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO2=0");
if (VCARule(1,3)) then do2On;
if (!VCARule(1,3)) then do2Off;
```

//VCA Profile 1 Rule 4 Adam Relay DO3

```
HttpCommand do3On:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO3=1");
HttpCommand do3Off:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)vUserName("root")Method(POST)ContentType("text/plain")Payload("DO3=0");
if (VCARule(1,4)) then do3On;
if (!VCARule(1,4)) then do3Off;
```

//VCA Profile 1 Rule 5 Adam Relay DO4

```
HttpCommand do4On:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO4=1");
HttpCommand do4Off:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO4=0");
if (VCARule(1,5)) then do4On;
if (!VCARule(1,5)) then do4Off;
```

//VCA Profile 1 Rule 6 Adam Relay DO5

```
HttpCommand do5On:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO5=1");
HttpCommand do5Off:= { Command("digitaloutput/all/value") IP("10.10.5.55")Password("00000000")
ForceBasicAuth(true)UserName("root")Method(POST)ContentType("text/plain")Payload("DO5=0");
if (VCARule(1,6)) then do5On;
if (!VCARule(1,6)) then do5Off;
```

## 4. History

Version	Date	Author	Description
V1.0	June 2020	Jan Noten (BT-SC/MKI) Eindhoven The Netherlands	First version for distribution
V1.1	July 2020	Jan Noten (BT-SC/MKI) Eindhoven The Netherlands	Changed script: <i>dataOutput0on</i> and <i>ForceBasicAuth(true)</i> in http command
V1.2	Jan. 2021	Jan Noten (BT-SC/MKI) Eindhoven The Netherlands	Bugfix FW Versions Advantech adapted

## 5. Disclaimer

Bosch cannot not accept any liability on implementing or using 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.



**Robert Bosch GmbH**  
Bosch Security Systems B.V  
P.O. Box 80002  
5600 JB Eindhoven  
The Netherlands