SDK kdriveBAOS Web Services

Getting Started

WEINZIERL ENGINEERING GmbH
Achatz 3
84508 Burgkirchen / Alz
Tel.: 08677 / 91 636 0
Fax: 08677 / 91 636 19
E-Mail: info@weinzierl.de
Web: www.weinzierl.de

Index

History

Description	Date	Author
Creation	2011-02-03	F. Häusl

1. Introduction

The KNX IP BAOS 771 is used as interface to connect to KNX/EIB both on telegram level (KNXnet/IP Tunneling) and on data-point level (KNX Application Layer). BAOS stands for "Bus Access and Object Server". This device can be used with ETS as a programming interface. As Object Server the device supports up to 250 data points.

The KNX IP BAOS 771 offers two separate client access protocols:

- Binary Protocol V2
- Web Services based on JSON (Java Script Object Notation)

The KNX BAOS Binary Protocol typically precludes the development of client applications that run in a Web Browser. For this reason access to the Object Server is now possible via the new KNX BAOS Web Services, based on HTTP and Java Script Object Notation (JSON). This means it is now possible to embed KNX IP BAOS 771 directly in your own Web applications.

The Web Services offer the same feature set as KNX BAOS binary protocol, however use a familiar text-based syntax that is sent over HTTP (port 80). The Web Services do not implement a graphical interface. This must be done separately, typically in HTML and Java Script, and can be stored, for example, in client memory, or wrapped directly into a stand-alone application using Webkit.

To simplify the start using the web services, we provide the software development kit (SDK) “kdriveBAOS Web Services”. This SDK contains an application programming interface (API) and two demos how to use it. The first one is called “Push Button Demo”. It represents a two channel push button including the notification of the feedback of the corresponding actuator. The other one is called “KNX Datapoint Types Demo”, it shows the usage of the different available datapoint types in a web application.

Please find the description of the API in the document KNX_IP_BAOS_WebServices.pdf, which is available on our web page.

2. Push Button Demo

a. ETS Configuration

The Push Button Demo uses four datapoints provided by the KNX IP BAOS. See Figure 1:

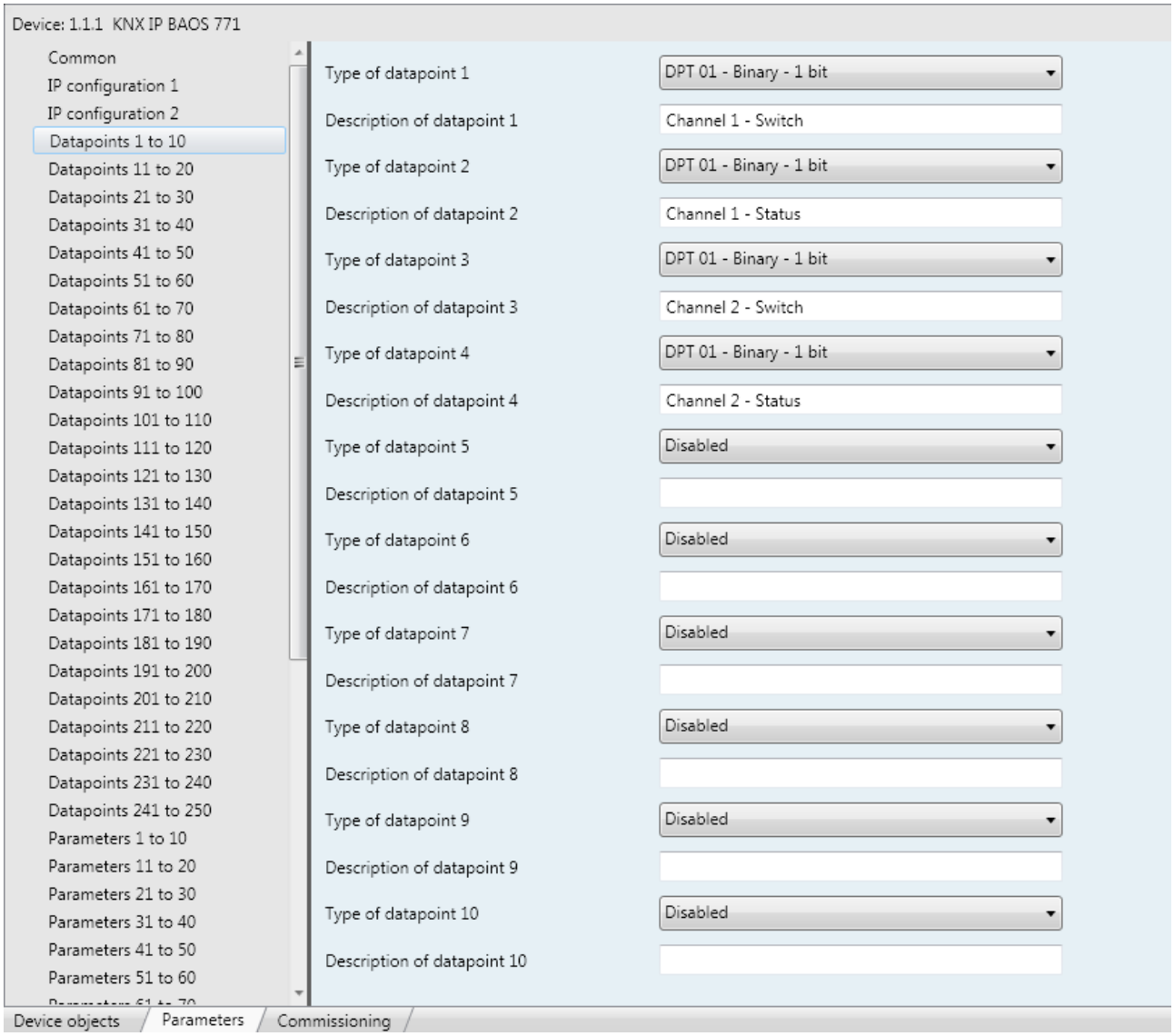


Figure 1: Push Button Demo: Parameter dialogue

Figure 2 shows the Group Objects including the assigned Group Address:

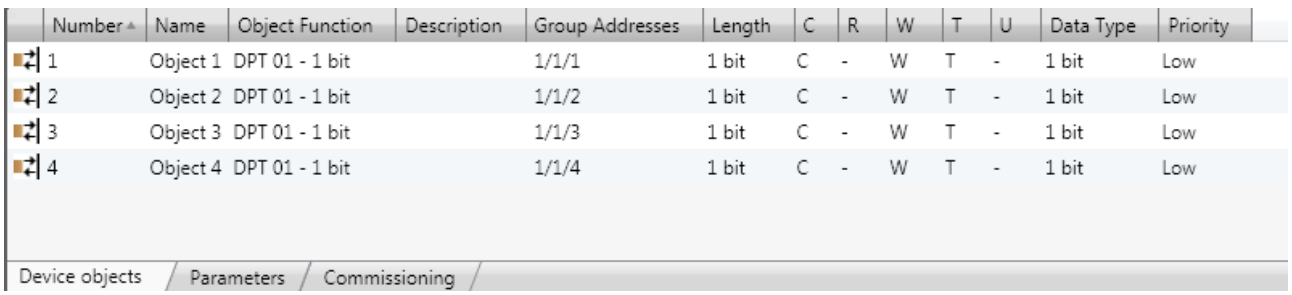


Figure 2: Push Button Demo: Group Objects dialogue

b. Description of the Demo

The Push Button Demo can be opened with any standard browser (tested with Firefox, Internet Explorer, Chrome and Safari).

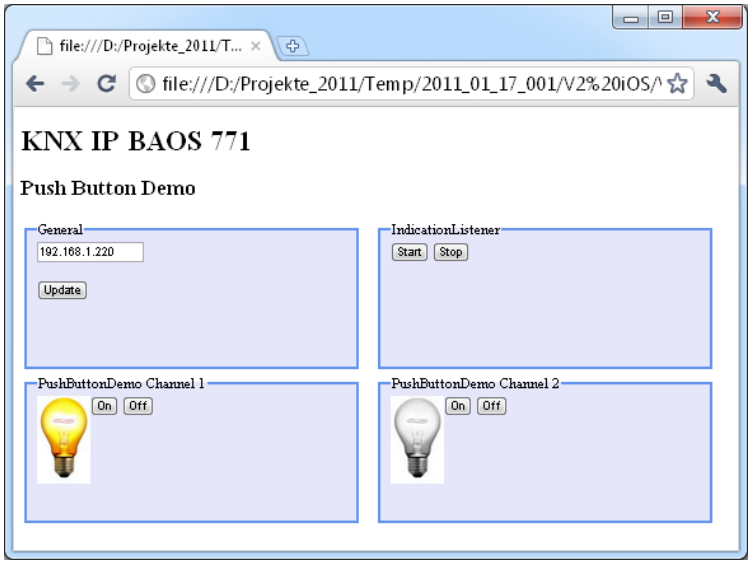


Figure 3: Push Button Demo opened with Chrome

Figure 3 shows the Push Button Demo. It is divided into four sections: General, Indication Listener, Channel 1 and Channel 2.

General:
The IP address of the KNX IP BAOS has to be set up. The button “Update” refreshes the IP address information in the API, calling the API “API_SetIpAddress”.

Indication Listener:
To receive indications (e.g. a feedback), you have to start the Indication Listener. It calls the API “API_StartIndicationListener”, which internally starts an indication session and opens a long-poll “GetIndication”. This long-poll “GetIndication” returns if an indication is received or if a timeout occurs. A new long-poll “GetIndication” is opened by the API, and so on.
The Indication Listener can be stopped calling “API_StopIndicationListener”. It stops the open “GetIndication” and stops the indication session.

Source code for starting / stopping the Indication Listener:

```
function OnBtnStartIndicationListener()
{
   baos.API_StartIndicationListener("Raw");                    // Start indication listener
}

function OnBtnStopIndicationListener()
{
   baos.API_StopIndicationListener();                          // Stop indication listener
}
```

Channel 1:
The button “On” switches channel 1 on, the button “off” switches channel 1 off, calling the API “API_SetDatapointValue”. The bulb represents the feedback of a switching actuator. The notification of the feedback requires an active Indication Listener. A callback handler “_API_Callback_IndicationUpdate” is called in case of an indication update.

Source code for switching channel 1:

```
function OnBtnPbDemo_SwitchChn1(value)
{
   if(value == 0)                                                    // If we want to switch off
   {
      baos.API_SetDatapointValue(1, "RAW", "SetSendVal", "1 Bit", "0x00");// Set communication object 1 to 0x00
   }
   else                                                              // Else: We want to switch on
   {
      baos.API_SetDatapointValue(1, "RAW", "SetSendVal", "1 Bit", "0x01");// Set communication object 1 to 0x01
   }
}
```

Source code for receiving an indication:

```
function _API_Callback_IndicationUpdate(jsonData)
{
   var nIndex;                                                       // Counting variable
   var lightbulb;                                                    // Reference to imgage "lightbulb"

   if((jsonData.Result == true) &&
            (jsonData.Service == "GetIndication"))                   // If successfully received
                                                                     // indications
   {
      for(nIndex = 0; nIndex < jsonData.Data.length; nIndex ++)      // Loop to handle every datapoint
      {
         if(jsonData.Data[nIndex].Datapoint == 2)                    // If we have communication object 2
                                                                     // (status response of switching
                                                                     // actuator)
         {
            lightbulb = document.getElementById("lightbulb1");       // Get element "lightbulb1"

            SwitchLightBulb(lightbulb, jsonData.Data[nIndex].Value);   // Switch lightbulb
         }

         if(jsonData.Data[nIndex].Datapoint == 4)                    // If we have communication object 4
                                                                     // (status response of switching
                                                                     // actuator)
         {
            lightbulb = document.getElementById("lightbulb2");       // Get element "lightbulb1"

            SwitchLightBulb(lightbulb, jsonData.Data[nIndex].Value);   // Switch lightbulb
         }
      }
   }
}
```

Channel 2:

See channel 1. The only difference is that we are now handling channel 2. The functionality is identical.

c. How to start

Setting up the KNX IP BAOS:
You have to set up the KNX IP BAOS device using the ETS. Please find attached a ETS project file (BaosSdkWebService.pr5), which includes the sample project. Open your ETS and import the project-file. The configuration of the device with address 1.1.1 shall be used for the Push Button Demo.
Connect the KNX IP BAOS and download the configuration. It is recommended to use an additional interface to the KNX bus for this purpose.

Using the demo within a browser:
Just start your browser and open the html file provided by the Push Button Demo (Client_PushButtonDemo.html).

3. Datapoint Type Demo

a. ETS Configuration

The Datapoint Type Demo uses 18 datapoints provided by the KNX IP BAOS. See Figure 4:

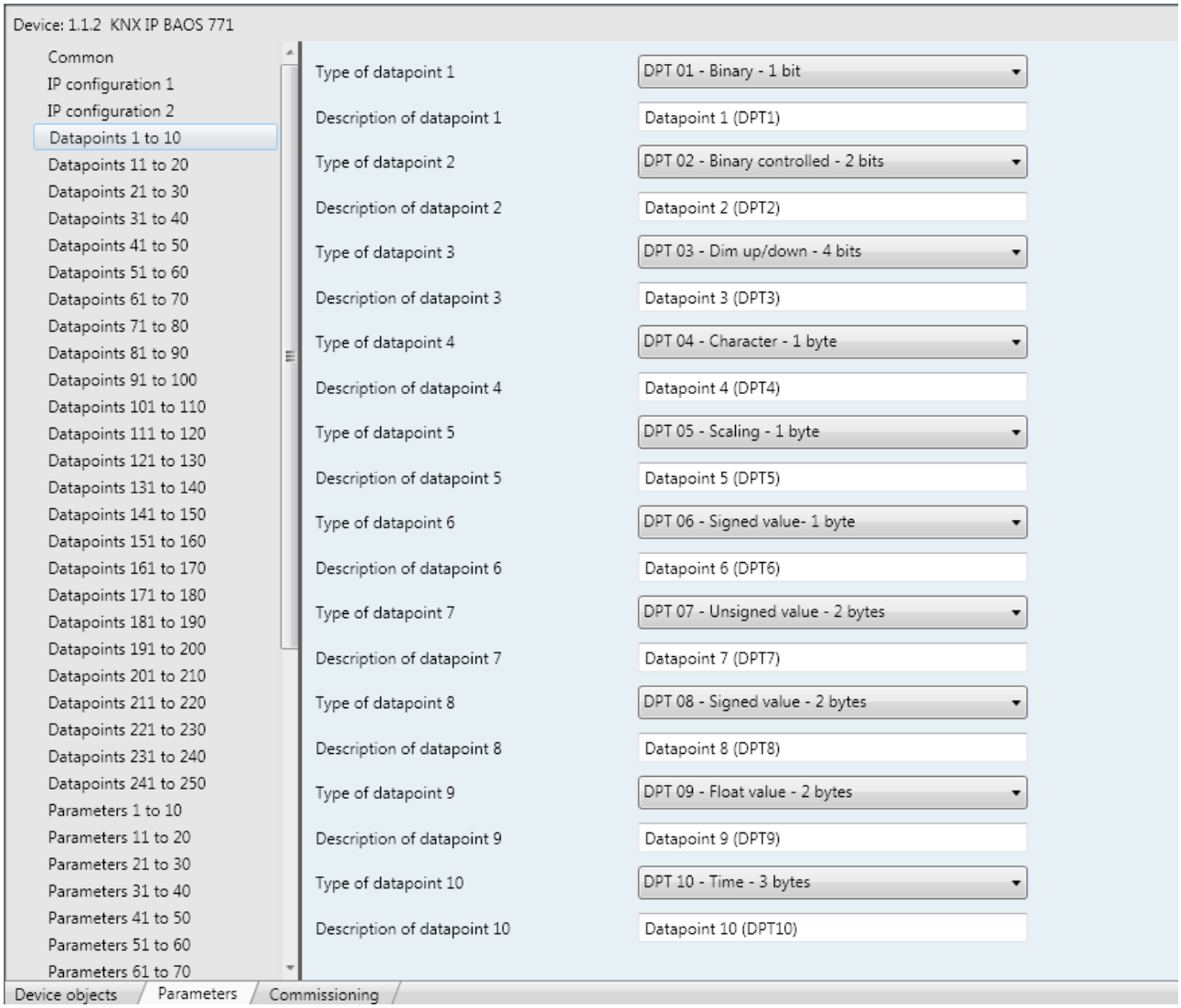


Figure 4: Push Button Demo: Parameter dialogue

Figure 5 shows the Group Objects including the assigned Group Address:

Number	Name	Object Function	Description	Group Addresses	Length	C	R	W	T	U	Data Type	Priority
1	Object 1	DPT 01 - 1 bit		1/0/1	1 bit	C	-	W	T	-	1 bit	Low
2	Object 2	DPT 02 - 2 bits		1/0/2	2 bit	C	-	W	T	-	1 bit controlled	Low
3	Object 3	DPT 03 - 4 bits		1/0/3	4 bit	C	-	W	T	-	3 bit controlled	Low
4	Object 4	DPT 04 - 1 byte		1/0/4	1 Byte	C	-	W	T	-	Character	Low
5	Object 5	DPT 05 - 1 byte		1/0/5	1 Byte	C	-	W	T	-	8 bit unsigned value	Low
6	Object 6	DPT 06 - 1 byte		1/0/6	1 Byte	C	-	W	T	-	8 bit signed value	Low
7	Object 7	DPT 07 - 2 bytes		1/0/7	2 Byte	C	-	W	T	-	2 byte unsigned value	Low
8	Object 8	DPT 08 - 2 bytes		1/0/8	2 Byte	C	-	W	T	-	2 byte signed value	Low
9	Object 9	DPT 09 - 2 bytes		1/0/9	2 Byte	C	-	W	T	-	2 byte float value	Low
10	Object 10	DPT 10 - 3 bytes		1/0/10	3 Byte	C	-	W	T	-	Time	Low
11	Object 11	DPT 11 - 3 bytes		1/0/11	3 Byte	C	-	W	T	-	Date	Low
12	Object 12	DPT 12 - 4 bytes		1/0/12	4 Byte	C	-	W	T	-	4 byte unsigned value	Low
13	Object 13	DPT 13 - 4 bytes		1/0/13	4 Byte	C	-	W	T	-	4 byte signed value	Low
14	Object 14	DPT 14 - 4 bytes		1/0/14	4 Byte	C	-	W	T	-	4 byte float value	Low
15	Object 15	DPT 15 - 4 bytes		1/0/15	4 Byte	C	-	W	T	-	Entrance access	Low
16	Object 16	DPT 16 - 14 bytes		1/0/16	14 Byte	C	-	W	T	-	Character string	Low
17	Object 17	DPT 17 - 1 byte		1/0/17	1 Byte	C	-	W	T	-	1 bit	Low
18	Object 18	DPT 18 - 1 byte		1/0/18	1 Byte	C	-	W	T	-	1 bit	Low

Device objects | Parameters | Commissioning

Figure 5: Push Button Demo: Group Objects dialogue

b. Description of the Demo

The Datapoint Type Demo can be opened with any standard browser (tested with Firefox, Internet Explorer, Chrome and Safari).

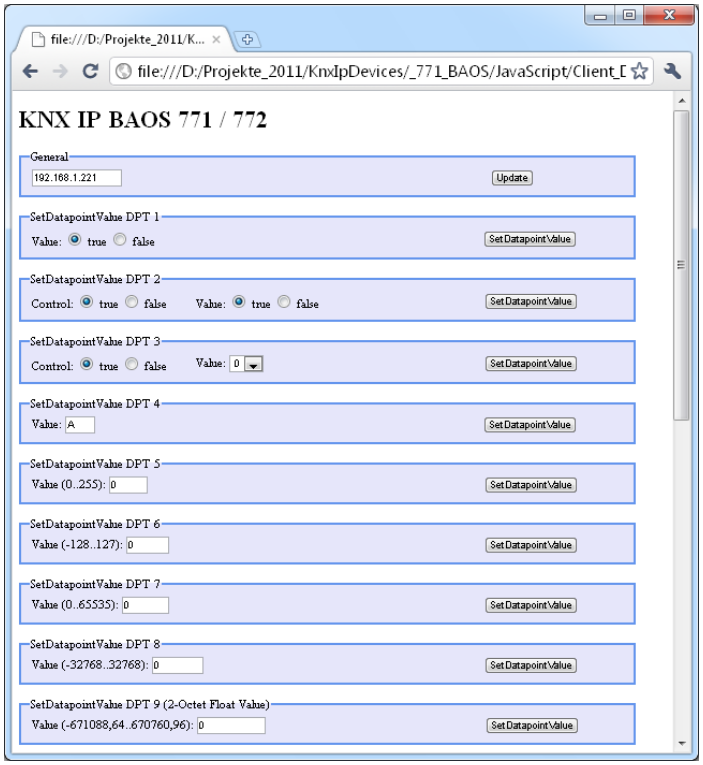


Figure 6: Datapoint Type Demo opened with Chrome (part 1)

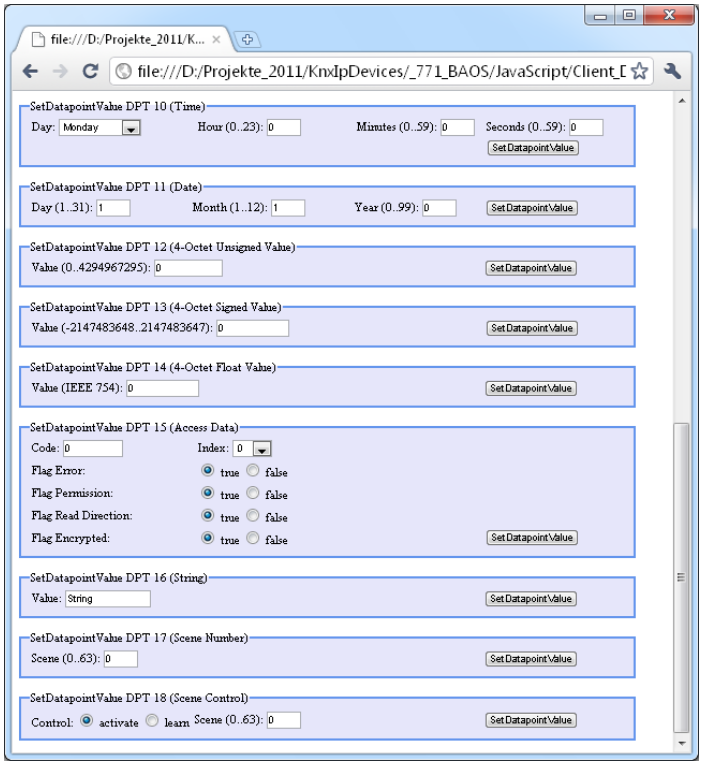


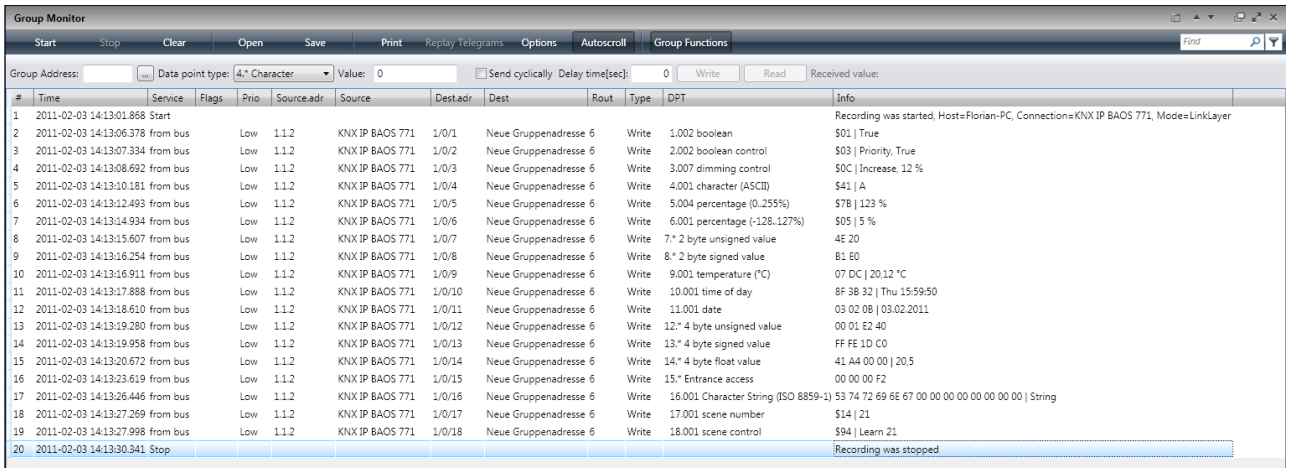
Figure 7: Datapoint Type Demo opened with Chrome (part 2)

Figure 6 and Figure 7 show the Datapoint Type Demo. It is divided into different sections:

General:
The IP address of the KNX IP BAOS has to be set up. The button “Update” refreshes the IP address information in the API, calling the API “API_SetIpAddress”.

SetDatapointValue DPT:
In each of these sections, you can enter data associated to a certain datapoint type. For example, if you are using DPT1, you can select “true” or “false”. Clicking the button “SetDatapointValue” sets the corresponding datapoint and writes the value to the bus, calling the API “API_SetDatapointValue”.

To verify the sent data, the group monitor of the ETS can be used in parallel to the object server. See Figure 8.



#	Time	Service	Flags	Prio	Source.adr	Source	Dest.adr	Dest	Rout	Type	DPT	Info
1	2011-02-03 14:13:01.868	Start										Recording was started, Host=Florian-PC, Connection=KNX IP BAOS 771, Mode=LinkLayer
2	2011-02-03 14:13:06.378	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/1	Neue Gruppenadresse 6		Write	1.002 boolean	$01 | True
3	2011-02-03 14:13:07.334	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/2	Neue Gruppenadresse 6		Write	2.002 boolean control	$03 | Priority, True
4	2011-02-03 14:13:08.692	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/3	Neue Gruppenadresse 6		Write	3.007 dimming control	$0C | Increase, 12 %
5	2011-02-03 14:13:10.181	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/4	Neue Gruppenadresse 6		Write	4.001 character (ASCII)	$41 | A
6	2011-02-03 14:13:12.493	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/5	Neue Gruppenadresse 6		Write	5.004 percentage (0..255%)	$7B | 123 %
7	2011-02-03 14:13:14.934	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/6	Neue Gruppenadresse 6		Write	6.001 percentage (-128..127%)	$05 | 5 %
8	2011-02-03 14:13:15.607	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/7	Neue Gruppenadresse 6		Write	7.* 2 byte unsigned value	4E 20
9	2011-02-03 14:13:16.254	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/8	Neue Gruppenadresse 6		Write	8.* 2 byte signed value	B1 E0
10	2011-02-03 14:13:16.911	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/9	Neue Gruppenadresse 6		Write	9.001 temperature (°C)	07 DC | 20,12 °C
11	2011-02-03 14:13:17.888	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/10	Neue Gruppenadresse 6		Write	10.001 time of day	8F 3B 32 | Thu 15:59:50
12	2011-02-03 14:13:18.610	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/11	Neue Gruppenadresse 6		Write	11.001 date	03 02 0B | 03.02.2011
13	2011-02-03 14:13:19.280	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/12	Neue Gruppenadresse 6		Write	12.* 4 byte unsigned value	00 01 E2 40
14	2011-02-03 14:13:19.958	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/13	Neue Gruppenadresse 6		Write	13.* 4 byte signed value	FF FE 1D C0
15	2011-02-03 14:13:20.672	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/14	Neue Gruppenadresse 6		Write	14.* 4 byte float value	41 A4 00 00 | 20,5
16	2011-02-03 14:13:23.619	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/15	Neue Gruppenadresse 6		Write	15.* Entrance access	00 00 00 F2
17	2011-02-03 14:13:26.446	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/16	Neue Gruppenadresse 6		Write	16.001 Character String (ISO 8859-1)	53 74 72 69 6E 67 00 00 00 00 00 00 00 00 | String
18	2011-02-03 14:13:27.269	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/17	Neue Gruppenadresse 6		Write	17.001 scene number	$14 | 21
19	2011-02-03 14:13:27.998	from bus		Low	1.1.2	KNX IP BAOS 771	1/0/18	Neue Gruppenadresse 6		Write	18.001 scene control	$94 | Learn 21
20	2011-02-03 14:13:30.341	Stop										Recording was stopped

Figure 8: ETS 4 Group Monitor