

The world of BAOS

Easy connectivity for KNX with Bus Access and Object Server

Overview and applications



WEINZIERL ENGINEERING GmbH Achatz 3 - 4 DE-84508 Burgkirchen / Alz Germany www.weinzierl.de



Content

1	Wha	t is BAOS?	3
	1.1	KNX communication	3
	1.2	Telegram interface	4
	1.3	BAOS communication	4
	1.4	Using KNX BAOS as application specific gateway	5
2	Thre	e interface types – one protocol	5
	2.1	KNX serial BAOS Modules	6
	2.1.1	KNX BAOS Starter Kit	7
	2.2	KNX USB BAOS	8
	2.3	KNX IP BAOS	9
	2.3.1	KNX IP BAOS 77x – highest performance via IP	9
	2.3.2	2 KNX IP BAOS with security	9
3	Usin	g KNX IP BAOS 777 as Residential Gateway1	0
	3.1.1	Using KNX IP BAOS 777 with a Web Browser1	1
4	BAO	S Server running on Linux: kTux1	2
5	Behi	nd the scene: The BAOS Protocol1	3
	5.1	Protocol variants1	3
	5.2	The Transfer Protocol1	3
	5.3	BAOS Protocol in Net'n Node1	4
	5.4	SDKs for BAOS Protocols1	5
	5.4.1	Demo Application for BAOS Serial Protocol1	5
	5.4.2	2 SDK for BAOS Binary Services1	6
	5.4.3	3 SDK for BAOS RESTful Web Services1	6
6	From	n BAOS to your Product1	6
	6.1	Product certification with BAOS1	6
	6.2	and what about the KNX Stack?1	6
	6.3	Any questions left?1	6



1 What is BAOS?

BAOS – short for "Bus Access and Object Server" – is a universal architecture to enable KNX connectivity for a great variety of applications and products. Within its range of KNX BAOS Solutions Weinzierl offers a scalable set of modules and powerful devices which rapidly enable the complete integration of applications into the KNX System.

Our BAOS Modules and Devices can be used as interfaces to connect to KNX both on the telegram and on the data-point level (the KNX Application Layer). Our BAOS Solutions allow an easy integration and implementation of KNX connectivity while releasing the application itself from the complexity of the KNX protocol, including the network management. Even the handling of KNX security can be passed to our BAOS implementation.



While our modules are designed to be fully integrated into your hardware, our KNX BAOS devices enable the connectivity of existing devices to KNX networks as external option but nevertheless with full integration in the ETS tool. Even a software variant of the BAOS server named kTux is available to be integrated into Linux based devices.

1.1 KNX communication

Devices that communicate via KNX require an implementation of the KNX protocol. The KNX protocol is specified according to the OSI (Open Systems Interconnection) reference model as a set of protocol layers.

The KNX system itself is a decentralized network and the runtime communication is based on group telegrams in multicast. To participate in the KNX runtime communication each device must be configured, e.g. group addresses must be assigned. Typically this is done via the ETS[®] (Engineering Tool Software) program, the official tool available from the KNX Association. The configuration can be loaded into the distributed devices via the KNX network during the device download. These management procedures are complex and are besides the runtime communication part of the KNX Standard.



In our KNX BAOS architecture all the KNX communication is handled within the BAOS Module or BAOS Device. A certified KNX Stack from Weinzierl ensures high performance runtime communication as well as complete compatibility with the ETS and all certified third-party KNX devices.



The BAOS architecture

1.2 Telegram interface

In addition to the BAOS protocol all Weinzierl KNX BAOS implementations allow to send and receive any KNX telegrams when setting the communication mode to data link layer. KNX telegrams are coded in cEMI (common External Message Interface) format according to the KNX specification. Today all BAOS Devices and Modules support KNX Long Frames.

The telegram interface based on cEMI is compatible with ETS software. So the USB BAOS as well as the IP BAOS devices can be used as programming interface for ETS. To integrate the telegram interface in applications running on Windows or Linux the cross-platform SDK kDrive is available from Weinzierl.

1.3 BAOS communication

The core of the BAOS implementation is the object server. The client can access the data of the group objects using the BAOS protocol. This protocol is specified by Weinzierl. It is designed to access KNX group objects and read ETS parameters in an efficient way. Receive events can trigger an indication to the client application to avoid the need for continuous polling.

Due to the data storage of the object server the communication to the object server is asynchronous and quite fast. Sending a value to KNX just requires to write an updated value to the object server and to trigger the send process. The KNX Stack of the BAOS implementation will carry out the send process autonomously. Also the receive process is handled by the object server. So no data is lost even if the client application is busy or not active.



1.4 Using KNX BAOS as application specific gateway

The KNX BAOS Modules and Devices can be used as application specific Gateways for non-KNX systems. They can be used for example to connect heating devices, audio application and many more to KNX. For a quick start in development a generic ETS product entry is available. The generic ETS product entry implements a flat list of group objects that can be configured from a set of standard data point types, for example

- DPT 1 1 Bit Switching
- DPT 3 4 Bit Dimming
- DPT 4 1 Byte Character
- DPT 5 1 Byte Scaling
- DPT 9 2 Byte Float Value
- DPT 10 3 Byte Time

Datapoint 1 to 10	1.		
Datapoint 11 to 20	Type of datapoint 1	DPT 01 - Binary - 1 bit	2
Datapoint 21 to 30	Description of datapoint 1	Switch	
Datapoint 31 to 40			
Datapoint 41 to 50	Type of datapoint 2	DPT 03 - Dim up/down - 4 bits	2
Datapoint 51 to 60			
Datapoint 61 to 70	Description of datapoint 2	Dim	
Datapoint 71 to 80	Type of datapoint 3	DPT 01 - Binary - 1 bit	
Datapoint 81 to 90	- Specific and a second second		
Datapoint 91 to 100	Description of datapoint 3	LED switch	
Datapoint 101 to 110	E	DOT 03 Dimension of the	
Datapoint 111 to 120	Type of datapoint 4	DP1 03 - Dim up/down - 4 bits	
Datapoint 121 to 130	Description of datapoint 4	LED dim rel	
Datapoint 131 to 140		and a second	
Datapoint 141 to 150	Type of datapoint 5	DPT 04 - Character - 1 byte	
Datapoint 151 to 160			
Datapoint 161 to 170	Description of datapoint 5	LED dim abs	
Datapoint 171 to 180	Type of datapoint 6	Disabled	
Datapoint 181 to 190		/	
Datapoint 191 to 200	Description of datapoint 6		
Datapoint 201 to 210		District	
Datapoint 211 to 220	Type of datapoint 7	Disabled	
Datapoint 221 to 230	Description of datapoint 7		
Datapoint 231 to 240			
Datapoint 241 to 250	Type of datapoint 8	Disabled	1
Parameter 1 to 10			
Parameter 11 to 20	Description of datapoint 8		
Parameter 21 to 30	Type of datapoint 9	Disabled	
Parameter 31 to 40	.,,	and the second s	
Parameter 41 to 50	Description of datapoint 9		
Parameter 51 to 60		Destad	
Parameter 61 to 70	 Type of datapoint 10 	Unsabled	

To use the communication object of the BAOS object server, the data points have to be configured. The KNX data point type has to be set and group addresses have to be assigned. To simplify this task for each KNX BAOS Module or Device a corresponding ETS product entry is available from the Weinzierl website.

The generic ETS entry supports either 250, 1000 or 2000 data points, and the setting of single parameter bytes which can be used by the client. The generic database targets product development and testing. For final products we recommend to create an individual database for your application to ensure an optimal representation of your

product within the ETS. For further information about development with KNX BAOS Modules and Devices please contact Weinzierl.

2 Three interface types – one protocol

The ObjectServer protocol has been defined to achieve the whole functionality on small embedded platforms and on data channels with limited bandwidth. As a consequence of this, the protocol is kept very slim and has no connection management, like connection establishment, user authorization, etc. Therefore, it is advisable and highly recommended to encapsulate the ObjectServer protocol into some existing transport protocol to get a useful solution for an easy access to the KNX datapoints and to the KNX bus. Depending on the interface type the BAOS protocol is encapsulated in:

- Serial: FT1.2 frames
- USB: HID reports
- IP: UDP or TCP frames

This section gives an overview about current solutions.



2.1 KNX serial BAOS Modules

Our BAOS Modules with serial interface offer a simple connection with a UART protocol based on FT1.2 frame format (IEC 870-5-1 and IEC 870-5-2 (DIN 19244)). With our KNX BAOS Modules we offer a quick and efficient solution to connect your devices to KNX. Typical applications are sensors, actuators or gateways to non-KNX systems

The compact PCBs include both a KNX transceiver and a microcontroller with a certified KNX stack. The modules can be integrated in devices and mounted via pin headers.

The KNX BAOS Module 830.1 is the first version supporting KNX security. To secure the host protocol a secure wrapper can be used as an option.

KNX BAOS Module 830	
	The KNX BAOS Module 830 is powered via the bus and provides galvanic isolation to the host. It supports 1000 group objects.
KNX BAOS Module 830.1 secure	
	The KNX BAOS Module 830.1 is the successor of the 830 module and supports KNX security as well as host security.
KNX BAOS Module 832	
	The KNX BAOS Module 832 corresponds to the version 830 but with power from the bus for the application and without the galvanic isolation.
KNX BAOS Module 838 kBerry	
	The KNX kBerry comes with the same schematic and firmware as KNX BAOS Module 830 but is mechanically designed for piggyback on popular Raspberry PI.
KNX BAOS Module 840 RF	
	The KNX BAOS Module 840 RF is the wireless alternative with up to 1000 data points. It implements KNX RF with full ETS support (ETS5 or higher).

A comprehensive user guide is available for our BAOS Modules on our website at www.weinzierl.de. A generic ETS database with up to 1000 group objects is available to get you up and running. Custom ETS databases can be created, as well.



2.1.1 KNX BAOS Starter Kit

To start your own project with our serial BAOS Modules a BAOS Development Kit is available which contains everything you need:



- BAOS Development board
- KNX BAOS Module 830
- KNX BAOS Module 832
- Tools and demo software
- USB cable

We provide a free demo software with source code to be programmed into the micro (ATSAMD20) on the development board. The software implements a simple KNX application.

For getting started the communication to the BAOS module can be redirected to the onboard USB connector. Via a virtual comport the BAOS protocol can be accessed by a PC application.

Our busmonitor and analyzer Net'n Node supports the BAOS protocol and helps for testing and understanding the BAOS protocol.

The development board can also be used for our KNX RF BAOS Module 840, which is not part of the kit.

More information about getting started you find in our user guide at www.weinzierl.de.



2.2 KNX USB BAOS

The USB connection is an alternative to serial interfaces for more complex devices running an operating system like Windows or Linux. The USB BAOS solution is available in these products:

KNX USB Interface 312.1	
	The KNX USB Interface 312.1 can be mounted on DIN rail with a width of only one unit (18 mm). It can be used as extension unit to connect to the KNX bus.
KNX USB Interface 322	
	The KNX USB Module 322 can be integrated in a PCB design using standard pin headers (2.54 mm)
KNX USB Interface 323 secure	
	The KNX USB Module 323 is the secure version of the 322 module and supports KNX security as well as host security.
KNX USB Interface 332	
	The KNX USB Stick 332 can be integrated in devices like touch panels but can be used as well as external device.
KNX USB Interface 333 secure	
	The KNX USB Stick 333 is the secure version of the 332 type and supports KNX security as well as host security.

All of them share the same functionality. They support the BAOS binary protocol V2 but can be used as programming interface for ETS as well.

While the type 312.1 is for DIN rail mounting the 322/323 and the 332/333 are more compact. The Module even can be integrated via pin headers in a PCB design. The Stick version 332 is also very small but comes as a complete device with an enclosure.

The most reasant implementations 323 and 333 support KNX security. To secure the host interface the same protocol extention has been implemented as for our serial modules.



2.3 KNX IP BAOS

With our range of KNX IP BAOS devices we bring the possibilities of our Bus Access and Object Server to LAN Networks. For a link via LAN or Wi-Fi our KNX IP BAOS devices allow interfacing complex devices. Even powerful visualization apps for mobile devices can be easily realized using the KNX IP BAOS architecture.

2.3.1 KNX IP BAOS 77x – highest performance via IP

The KNX IP BAOS series is a convienient solution to extend more complex devices with full-value KNX connectivity. Such devices typically include already a network interface and are based on an operating system like Linux. For the IP communication the BAOS protocol is encapsulated in an TCP IP frame.

The KNX IP BAOS devices offer a certified KNX Stack as a service to the application. So complex devices like a boiler, a ventilation unit or a PV inverter can be part of a KNX network without integrating a KNX Stack into the device. Instead a simple layer to adapt the BAOS protocol has to be added. The complexity of the KNX protocol including network configuation by ETS is handled in the KNX IP BAOS device.



Using the KNX IP BAOS 773, 774 or 774.1 the KNX bus is accessible from everywhere via Ethernet. Connection over WiFi (with an extra router) or the Internet is possible, too. The IP settings as well as the data points can be configured with ETS software. This device can be also be used as a programming interface for ETS.

All variants are based on a tailored hardware with low power consumption. The KNX IP BAOS 773 supports up to 250 communication objects, the KNX IP BAOS 774/774.1 up to 1000.

2.3.2 KNX IP BAOS with security

To be ready for future requirements the support of security is essential. Our latest KNX IP BAOS 774.1 *secure* supports of course KNX Data Security and KNX IP Security which protects the integrated KNX Programming interface.

But of course also the BAOS communication via IP might be a point of attack. To protect the BAOS IP protocol we are reusing the principles of KNX IP Security for Tunnelling interfaces. We use different credentials for the BAOS access. The additional key can be set via an ETS parameter.



3 Using KNX IP BAOS 777 as Residential Gateway

The KNX IP BAOS 777 is a powerful multifunctional device for KNX installation. It combines the following functionalities:

- Visualization and control
- Sending E-mail notifications
- Date & Time Server (NTP) synchronization
- Timer with astrotimer
- KNX IP Interface (KNXnet/IP)

The KNX IP BAOS 777 is an on-site solution that is very easy to integrate with the ETS. The device does not require a cloud connection or an app. No further costs are incurred during operation.



KNX configuration is done exclusively with the ETS software – no additional software is needed. Userrelated settings such as email or timer functions can be set by the user in the web frontend.

KNX IP BAOS as micro server builds the interface to the building with semantic information from the installation. Only functions are available which explicitly have been added to the configuration. This increases security as well as safety for users and for the building.

Best in Class: the KNX IP BAOS 777 is a powerful residential gateway for building control with smallest size and minimal power consumption. The KNX IP BAOS 777 requires either an external power supply or via Power-over-Ethernet (IEEE 802.3af) directly from the switch.

The KNX IP BAOS architecture not only allows access to the runtime data. In addition it retrieves the structure of the KNX installation. It encodes the rooms of a building as well as the available functions as a set of meta-data. Using the ETS commissioning tool the installer defines the rooms and which functions are available to the client. While a room is seen as a collection of functions, a function is a collection of datapoints representing a specific KNX interworking function. A simple example is a switching actuator with state which requires two datapoints. Functions are defined for, but not limited to:

- Switching Control
- Switching Control with State
- Dimming Control
- Dimming Control with Switching State
- Dimming Control with Value State
- Temperature with Set Point
- RGB Control



The benefit of user parameters is to enable the possibility of configuring the application via the ETS parameter dialog. It is even possible to configure a visualization program just via the ETS.

The main advantages are:

- Fast and easy configuration of the app in ETS
- No additional editor for visualization
- No training needed
- Separation of graphical design and building structure
- Separation of building installation and app handling
- Very low power consumption
- Object Server holds all actual values
- Simultaneous usage by multiple clients

3.1.1 Using KNX IP BAOS 777 with a Web Browser

The KNX IP BAOS 777 has an integrated web server which allows an easy and fast access to the building using a standard web browser. Using the ETS product entry with building structure the web server offers a visualization tool. Via the web interface also timers and history are available.



To access the web application, just enter the IP Address of the BAOS 777 into your browser. The IP address can be seen in the display of the device.

Once a connection has been established to the BAOS you should see the login page. The username and password are by default "admin" and can be set in the ETS database.



4 BAOS Server running on Linux: kTux

Besides our BAOS modules and device, a BAOS server can also be integrated into the host system. Nowadays, many devices for building automation use a Linux platform for control purposes. Typical examples are boilers, air conditions, ventilation, photovoltaic converters or charging station for electrical cars.



For this type of devices kTux has been introduced: our kNX Stack for Linux based devices. The purpose of this product is to connect a Linux device to the KNX network based on a certified KNX Stack. The result is a full valued KNX device which can be managed by the ETS® commissioning software and can be certified according to the KNX Standard.



kTux runs as a executable on Linux. The application can connect via IP local host using the well-tried BAOS protocol. kTux supports KNX Security and is avaiable for KNX Media TP, RF and IP. The medium access for TP and RF is realized via KNX USB Modules. kTux can also be extended with a KNXnet/IP server for Tunnelling. More information about kTux you find at www.weinzierl.de.



5 Behind the scene: The BAOS Protocol

The BAOS Protocol is a tailored solution to exchange data with an object server. It has a dynamic structure, not only for the length of individual data points but also for the number of data points encapsulated within one frame. That allows that in one request or indication multiple values can be transferred. This is essential for communication systems like IP which typically have a high bandwidth but sometimes quite high delays.

5.1 Protocol variants

The communication structure of the BAOS architecture is client-server. The KNX BAOS Modules or Devices act as a server. The application is the client. Most services are sent from the client to the server which always sends a response. Asynchronous indications can be sent by the server to inform the client about updates of values via the KNX network.

The BAOS Protocol is used in three different formats:

- KNX BAOS Binary
- KNX BAOS RESTful Web Services

The BAOS Binary Protocol uses arrays of bytes to code services. It is the best way to implement communication protocols directly in a microcontroller. The serial KNX BAOS Modules and Devices just use the binary protocol. The binary protocol is available as well on all KNX IP BAOS devices.

As a new generation of Web Services the KNX IP BAOS 777 supports all services in a RESTful API. The RESTful web services allow a semantic access to the structure of the installation. That means that the interpretation of the ETS parameters is now done in the KNX IP BAOS 777 device.

Currently the BAOS protocol is used in version V2. See the protocol specification for details.

5.2 The Transfer Protocol

The BAOS Protocol just contains application data. It does not define a transport protocol, so it can be used on different connection types.

For the modules and devices which support a serial link, the BAOS Protocol is encapsulated in a FT1.2 frame which defines the start and the end of the packet as well as a checksum and acknowledgement. FT1.2 is an international standard (IEC 870-5-1 and 870-5-2).

The KNX USB BAOS Devices and Modules use encapsulate the BAOS Protocol in HID reports as it is specified for KNX USB interface devices.



The KNX IP BAOS Devices use either IP UDP or TCP packets as frames. The headers and usage is very similar to the KNXnet/IP specification however they use the BAOS data structure as content. Although in principle UDP is a connectionless communication, there is some kind of transport connection on top of UDP which is required to send asynchronous indications to the client. The KNX IP BAOS Devices always act as a communication server.

A major benefit of the IP protocol is the possibility of multiple connections. Up to 10 clients can be connected to a single KNX IP BAOS device via the IP BAOS protocol.

5.3 BAOS Protocol in Net'n Node

The BAOS Binary Protocol is supported in the free version of Net'n Node 5. Net'n Node is a bus monitor and analyzer for KNX especially for development purpose. It supports all KNX media and uses standard KNX interfaces (IP, USB and Serial FT1.2).

The program allows opening one or more BAOS ports via serial, USB or IP even in addition to a KNX telegram port. The integrated BAOS view give access to all BAOS data like server item and data points. BAOS services can be sent via dedicated send masks. Sent and received services are displayed together with KNX telegrams on the bus in the telegram window.

		192.168.1.38 KN	K IP Baos 77	7 Ms	
Read				192.168.1.38 KNX IP Baos 7	77 Ms
erver I	tems Datapoints			Send Services	
Id	Description	Convertien		Telegram Type	
10	HardwareTune	00.00 C5.07.00.08		GetServerItem	•
2	Hardware/version	10	_		_
2	FirmwareVersion	0.8		Data	
4	ManufactureCodeDevice	107	_	Start Index 1	c 🔻
5	ManufactureCodeApp	197		Number Of 1	· •
6	ApplicationId	1804	_		
7	ApplicationVersion	11			
8	SerialNumber	00 C5 01 00 ED CE	_		
0	TimeSinceReset	10386	=		
10	RusConnected	Connected	_	Request Telegram	
11	MayBufferSize	4006 Bute		F0 01 00 01 00 01	
12	LengthOfDescriptionString	0 Bute	_		
13	Baudrateltem	Unknown		Send	
14	CurrentBufferSize	4096 Byte			
15	ProgrammingModeltem	Off (0)			
16	BinaryProtocolVersion	2.1			
17	IndicationSetting	Enabled (1)			
18	WebServicesProtocolVersion	2.1			
19	RestServiceProtocolVersion	2.1			
20	IndividualAddress	15.15.55			
21	MacAddress	00 24 6D 00 E1 33			
22	TunnellingEnabled	Enabled (1)			
23	BaosBinaryEnabled	Enabled (1)			
24	BaosWebEnabled	Enabled (1)			
25	BaosRestEnabled	Enabled (1)			
26	HttpFileEnabled	Enabled (1)			
27	SearchRequestEnabled	Enabled (1)			
28	IsStructured	Enabled (1)			
29	MaxManagementClients	1			

Net'n Node BAOS view showing server items



Uli Cett P Region Interface I Immunty Service Servide Servide Det Addr Mm 1 Telegrom 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 <	mmand	s Capture I	interfaces													
m telegram Interface T Timestram Served Served Served Det/Addr 24 10 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 </th <th>() Cle</th> <th>* <u>*</u>.[</th> <th>UN TUN TUN TUN TUN TUN TUN TUN</th> <th>TUN. TUN.</th> <th>US8 TP</th> <th>H_{P0}</th> <th></th> <th>P</th> <th>BAOS .</th> <th>P</th> <th>BAOS BAOS</th> <th>BAOS BAOS</th> <th></th> <th></th> <th></th>	() Cle	* <u>*</u> .[UN TUN TUN TUN TUN TUN TUN TUN	TUN. TUN.	US8 TP	H _{P0}		P	BAOS .	P	BAOS BAOS	BAOS BAOS				
25 P0 0 10 0 28 0 0 10 0 28 0 4 0 24 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 27 P1 0 10 0 2 0 0 10 0 20 0 4 1 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 28 P1 0 10 0 2 0 0 10 0 20 0 4 1 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 28 P1 0 10 0 2 0 0 10 0 28 0 4 10 1 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 29 P1 0 10 0 28 0 0 10 0 28 0 10 0 20 1 0 4 30 0 11 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 21 P1 0 10 0 27 0 0 11 0 0 27 0 4 56 D 6 0 10 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 21 P1 0 10 0 27 0 0 11 0 0 1 0 0 1 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 21 P1 0 1 0 0 3 0 0 0 1 0 0 3 1 0 0 1 122.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 21 P1 0 1 0 0 3 0 0 0 1 0 0 1 0 1 0 1 122.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 21 P1 0 1 0 0 0 0 0 0 1 0 0 0 9 4 0 0 0 29 6 8 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 21 P1 0 1 0 0 0 0 0 0 0 1 0 0 0 9 4 0 0 0 29 6 8 132.148.1.38. 1 2014-03-02 1106430. GetServerTeen.res 21 P1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 9 4 0 0 0 29 6 8 122.148.1.	Im	telegram		Inte	erface		Ŧ	Tim	estamp		Servic	e .	Src-Addr	Dest-Addr	Cont	
223 70 10 0.0 20 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0<	25	F0 81 00	2B 00 01 00 2B 04 C0 A8 01 26	192	2.168.1.	38	t	201	5-03-0	2 11	1:06:30. GetSe	rverItem.res				
27 0 10 0.2 0.2 0.2 0.0 0.0 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 0.2 0.4 </td <td>26</td> <td>F0 01 00</td> <td>2C 00 01</td> <td>192</td> <td>2.168.1.</td> <td>38</td> <td>±.</td> <td>201</td> <td>5-03-0</td> <td>2 11</td> <td>L:06:30. GetSe</td> <td>rverItem.req</td> <td></td> <td></td> <td></td>	26	F0 01 00	2C 00 01	192	2.168.1.	38	±.	201	5-03-0	2 11	L:06:30. GetSe	rverItem.req				
228 F0 01 00 2D 00 01 00 2D 00 01 00 2D 04 C0 AB 01 01 132.148.1.38. 2014-03-02 11:06:30. GetTerverTem.req 330 F0 01 00 2D 00 01 00 2D 04 C0 AB 01 01 132.148.1.38. 1 2014-03-02 11:06:30. GetTerverTem.req 331 F0 01 00 2D 00 01 00 2D 04 C0 AB 01 01 132.148.1.38. 1 2014-03-02 11:06:30. GetTerverTem.req 333 F0 01 00 2D 00 01 00 2D 04 C0 AB 0D 01 132.148.1.38. 1 2014-03-02 11:06:30. GetTerverTem.req 333 F0 01 00 2D 00 01 00 2D 04 C0 AD 00 10 132.148.1.38. 1 2014-03-02 11:06:30. GetTerverTem.req 335 F0 01 00 30 01 00 100 10 03 01 00 132.148.1.38. 1 2014-03-02 11:06:30. GetTerverTem.req 336 F0 01 00 09 00 01 100 01 00 31 00 11 132.148.1.38. 1 2014-03-02 11:06:30. GetTerverTem.req 337 F0 01 00 09 00 01 00 00 2D C4 132.148.1.38. 1 2014-03-02 11:06:30. GetTerverTem.req 338 F0 01 00 09 00 01 100 09 00 01 132.148.1.38. 1 2014-03-02 11:06:30. GetTerverTem.req 338 F0 01 00 09 00 01 132.148.1.38. 1 2014-03-02 11:06:30.	27	F0 81 00	2C 00 01 00 2C 04 FF FF FF 00	192	2.168.1.	38	1	2014	5-03-0	2 11	1:06:30 GetSe	rverItem.res				
223 F0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	28	F0 01 00	2D 00 01	192	2.168.1.	38	±.	201	5-03-0	2 11	L:06:30_ GetSe	rverItem.req				
303 T0 0 1 0 0 22 00 0 10 22 0 0 1 0 2 2 0 0 10 0 2 2 0 173 192.168.1.38. 1 2014-03-02 11:06:30. Getarvez1tem.req 313 T0 0 1 0 0 2 2 0 0 1 0 2 2 0 173 192.168.1.38. 1 2014-03-02 11:06:30. Getarvez1tem.req 313 T0 0 1 0 0 2 7 0 0 1 0 0 2 7 0 4 5 0 6 0 9 1C 192.168.1.38. 1 2014-03-02 11:06:30. Getarvez1tem.req 313 T0 0 1 0 0 3 0 0 1 0 0 3 0 1 0 0 192.168.1.38. 1 2014-03-02 11:06:30. Getarvez1tem.req 315 T0 0 1 0 0 3 0 0 1 0 1 0 3 0 1 0 0 192.168.1.38. 1 2014-03-02 11:06:30. Getarvez1tem.req 317 T0 0 1 0 0 3 1 0 0 1 0 3 1 0 1 0 192.168.1.38. 1 2014-03-02 11:00:30. Getarvez1tem.req 318 T0 0 1 0 0 9 0 0 1 0 0 9 0 4 0 0 0 2 9 8 8 192.168.1.38. 1 2014-03-02 11:00:30. Getarvez1tem.req 318 T0 0 1 0 0 9 0 0 1 0 0 9 0 4 0 0 0 2 9 C4 192.168.1.38. 1 2014-03-02 11:00:30. Getarvez1tem.req 318 T0 0 1 0 0 0 9 0 0 1 0 0 9 0 4 0 0 0 2 9 C4 192.168.1.38. 1 2014-03-02 11:00:30. Getarvez1tem.req 319 T0 0 1 0 0 0 9 0 0 1 10 0 0 9 0 0 1 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	29	F0 81 00	2D 00 01 00 2D 04 C0 A8 01 01	193	2.168.1.	38	1	201	5-03-0	2 11	1:06:30. GetSe	rverItem.res				
31 2 70 50 22 00 22 00 22 00 22 00 22 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 <td< td=""><td>30</td><td>F0 01 00</td><td>2E 00 01</td><td>193</td><td>2.168.1.</td><td>38</td><td>±.,</td><td>201</td><td>5-03-0</td><td>2 11</td><td>1:06:30_ GetSe</td><td>rverItem.req</td><td></td><td></td><td></td></td<>	30	F0 01 00	2E 00 01	193	2.168.1.	38	±.,	201	5-03-0	2 11	1:06:30_ GetSe	rverItem.req				
333 T0 01 00 27 00 01 0 27 0 01 00 27 00 01 0 27 0 01 00 27 0 00 01 192.168.1.38. 2016-03-02 11:06:30. GetErverItem.req 343 T0 01 00 30 00 01 00 27 0 45 65 65 65 01 192.168.1.38. 1 2016-03-02 11:06:30. GetErverItem.req 344 T0 01 00 31 00 01 00 31 01 01 192.168.1.38. 1 2016-03-02 11:06:30. GetErverItem.req 357 T0 01 00 31 00 01 00 31 01 01 192.168.1.38. 1 2016-03-02 11:06:30. GetErverItem.req 358 T0 01 00 09 00 01 00 00 40 00 02 98 192.168.1.38. 1 2016-03-02 11:06:30. GetErverItem.req 358 T0 01 00 09 00 01 00 09 04 00 00 29 88 192.168.1.38. 1 2016-03-02 11:00:30. GetErverItem.req 364 T0 01 00 09 00 01 00 09 04 00 00 29 C4 192.168.1.38. 1 2016-03-02 11:00:30. GetErverItem.req International Internation Internatinternational Internatinternational Internati	31	F0 81 00	2E 00 01 00 2E 01 73	192	2.168.1.	38	1	201	5-03-0	2 11	1:06:30. GetSe	rverItem.res				
333 T0 81 00 27 00 10 00 27 04 56 05 C9 1C 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.res 334 T0 01 00 30 00 01 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.res 335 T0 81 00 31 00 01 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.res 335 T0 81 00 31 00 01 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.res 336 T0 01 00 91 00 01 00 10 00 01 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.res 337 T0 01 00 09 00 01 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.res 338 T0 01 00 09 00 01 00 09 04 00 00 29 88 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.res 340 T0 01 00 09 00 01 00 09 04 00 00 29 C4 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.reg TO 81 00 09 00 01 00 09 04 00 00 29 C4 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.reg To 81 00 09 00 01 00 09 04 00 00 29 C4 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.reg TO 81 00 09 00 01 00 09 04 00 00 29 C4 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.reg TO 81 00 09 00 01 00 09 04 00 00 29 C4 192.168.1.38. ± 2016-03-02 11:06:30. GetErvertem.reg TO 81 00 09 00 01	32	F0 01 00	2F 00 01	192	2.168.1.	38	Ŧ.	201	5-03-0	2 11	1:06:30_ GetSe	rverItem.req				
34 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	33	F0 81 00	2F 00 01 00 2F 04 56 D6 C9 1C	192	2.168.1.	38	t	201	5-03-0	2 11	1:06:30_ GetSe	rverItem.res				
333 20 90 81.00 30.00 192.148.1.38. 1 2014-03-02 11:06:30. Getarwertem.ree 333 20 90 00 03.10 132.148.1.38. 1 2014-03-02 11:06:30. Getarwertem.ree 333 20 70 10 00 00 01 132.148.1.38. 1 2014-03-02 11:06:30. Getarwertem.ree 333 20 10 10 00 00 00 00 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 11:00:30. Getarwertem.ree 100 100 00 00 00 00 00 00 00 100 00 00 100 100 100 100 100 100 100 100 100 100 100 11:00 100 100 100 100 100 100 100<	34	F0 01 00	30 00 01	192	2.168.1.	38	±.	201	5-03-0	2 11	1:06:30_ GetSe	rverItem.req				
36 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	35	F0 81 00	30 00 01 00 30 01 00	193	2.168.1.	38	1	201	5-03-0	2 11	1:06:30_ GetSe	rverItem.res				
37 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	36	F0 01 00	31 00 01	193	2.168.1.	38	±.	201	5-03-0	2 11	1:06:30. GetSe	rverItem.req				
8 8 10 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 </td <td>37</td> <td>F0 81 00</td> <td>31 00 01 00 31 01 01</td> <td>192</td> <td>2.168.1.</td> <td>38</td> <td>1</td> <td>201</td> <td>5-03-0</td> <td>2 11</td> <td>1:06:30. GetSe</td> <td>rverItem.res</td> <td></td> <td></td> <td></td>	37	F0 81 00	31 00 01 00 31 01 01	192	2.168.1.	38	1	201	5-03-0	2 11	1:06:30. GetSe	rverItem.res				
9 0 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 <	38	F0 01 00	09 00 01	192	2.168.1.	38	1	201	5-03-0	2 11	1:07:30_ GetSe	rverItem.req				
40 192,168.1.38. 1016-03-02 11:00:30. GetServerItem.req 41 10 100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	39	F0 81 00	09 00 01 00 09 04 00 00 29 88	192	2.168.1.	38	1	201	5-03-0	2 11	1:07:30_ GetSe	rverItem.res				
41 TO 81 00 09 00 01 00 09 04 00 00 29 C4 192,168.1.38. 1 2016-03-02 11:08:30. GetServerItem.res Model Model ADS View Detroprison OPTIO - Binary DPTIO - Binary <td c<="" td=""><td>40</td><td>F0 01 00</td><td>09 00 01</td><td>192</td><td>2.168.1.</td><td>38</td><td>1</td><td>201</td><td>5-03-0</td><td>2 11</td><td>1:08:30 GetSe</td><td>rverItem.req</td><td></td><td></td><td></td></td>	<td>40</td> <td>F0 01 00</td> <td>09 00 01</td> <td>192</td> <td>2.168.1.</td> <td>38</td> <td>1</td> <td>201</td> <td>5-03-0</td> <td>2 11</td> <td>1:08:30 GetSe</td> <td>rverItem.req</td> <td></td> <td></td> <td></td>	40	F0 01 00	09 00 01	192	2.168.1.	38	1	201	5-03-0	2 11	1:08:30 GetSe	rverItem.req			
Image: Size Priority C R W T U I Rew Value(he) Interpreted Value DatapointType Size Priority C R W T U I Rew Value(he) Interpreted Value #Indicat DatapointType Size Priority C R W T U I Rew Value(he) Interpreted Value #Indicat DatapointType Size Priority C R W T U I Rew Value(he) Interpreted Value #Indicat DatapointType Size Priority C R W T U I Rew Value(he) Interpreted Value #Indicat DatapointType Size Priority C R W T U I Rew Value(he) Interpreted Value #Indicat DatapointType Size Priority C R W T U I I 000 False Interpreted Value #Indicat DPT00 - Cottet Float Value Size Priority C R W T U I I 000 False OD DPT01 - Binary 1 Bit() Low C C - W W - U I I 000 False O DPT01 - Binary 1 Bit() Low C C - W W - U I I 000 False O DPT01 - Binary 1 Bit() Low C C - W W - U I I 000 False O OPT01 - Binary 1 Bit() Low C C - W W - U I I 000 False <td>41</td> <td>FO 81 00</td> <td>09 00 01 00 09 04 00 00 29 C4</td> <td>192</td> <td>2.168.1.</td> <td>38</td> <td>1</td> <td>201</td> <td>5-03-0</td> <td>2 11</td> <td>1:08:30_ GetSe</td> <td>rverItem.res</td> <td></td> <td></td> <td></td>	41	FO 81 00	09 00 01 00 09 04 00 00 29 C4	192	2.168.1.	38	1	201	5-03-0	2 11	1:08:30_ GetSe	rverItem.res				
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	AOS Vi Read	ew							192.1	68.1	.38 KNX IP Bao	s 777 Ms				
Arr OF 102 e-Bet Unrighted Value 1 Bit (c) Cord C V I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I	AOS Vi Read	ew ems Datapoin	5						192.1	68.1	1.38 KNX IP Bao	s 777 Ms				
13 DP 10 - 2-Octet Float Value 1 Spetch Low C N I N 0 1 N 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Read erver It Id	ew ems Datapoin Description	5 DatapointType 105 fu Giopo	Size	Priority	c	R	w	192.1 r u	68.1	Aw Value(hex)	s 777 Ms		# Indicatio	ons f	
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	Read erver It Id 74	ew ems Datapoin Description	S DetapointType DPT01 - Binary DT756 - Bit Internet Value	Size 1 Bit(s)	Priority Low	c c c	R	W 1	192.1	68.1 1	Raw Value(hex)	s 777 Ms		# Indicatio	ons '	
bit Constraint Constraint <td>Read erver It Id 74 75</td> <td>ew ems Datapoin Description</td> <td>s DatapointType DPT 01 - Binary DPT 03 - 9 Bet Unigned Value DPT 04 - 9 Octo East Value</td> <td>Size 1 Bit(s) 1 Byte(s) 2 Byte(s)</td> <td>Priority Low Low</td> <td>с с с с с</td> <td>R - -</td> <td>W 1 - W</td> <td>192.1 T U T - U</td> <td>68.1</td> <td>Raw Value(hex)</td> <td>s 777 Ms Interpreted Value False 0 0 0 0</td> <td></td> <td>#Indicatio 0 0</td> <td>ons '</td>	Read erver It Id 74 75	ew ems Datapoin Description	s DatapointType DPT 01 - Binary DPT 03 - 9 Bet Unigned Value DPT 04 - 9 Octo East Value	Size 1 Bit(s) 1 Byte(s) 2 Byte(s)	Priority Low Low	с с с с с	R - -	W 1 - W	192.1 T U T - U	68.1	Raw Value(hex)	s 777 Ms Interpreted Value False 0 0 0 0		#Indicatio 0 0	ons '	
Def Gal Def Gal Binity Lowing C N C N C N C N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N <td>Read erver It Id 74 75 76</td> <td>ew ems Datapoin Description</td> <td>5 DatapointType DPT01 - Binny DPT03 - Bit Unkigned Value DPT09 - 2-Octer Float Value DPT09 - 2-Octer Float Value</td> <td>Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(c)</td> <td>Priority Low Low</td> <td>C C C C C C</td> <td>R - -</td> <td>w 1 • w</td> <td>192.1 T U T U - U</td> <td>68.1 1 - 1 1</td> <td>1.38 KNX IP Bao Raw Value(hex)</td> <td>5 777 Ms Interpreted Value Fable 0 0.000 Extense</td> <td></td> <td># Indication</td> <td>ons '</td>	Read erver It Id 74 75 76	ew ems Datapoin Description	5 DatapointType DPT01 - Binny DPT03 - Bit Unkigned Value DPT09 - 2-Octer Float Value DPT09 - 2-Octer Float Value	Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(c)	Priority Low Low	C C C C C C	R - -	w 1 • w	192.1 T U T U - U	68.1 1 - 1 1	1.38 KNX IP Bao Raw Value(hex)	5 777 Ms Interpreted Value Fable 0 0.000 Extense		# Indication	ons '	
DF 10.4 Binny D File Com C V V V I O False O 91 OPT0.4 Binny 1 Biti) Low C V V V I I O False O 91 OPT0.4 Binny 1 Biti) Low C V V V I I O False O 94 OPT0.4 Binny 1 Biti) Low C V V V I O False O 97 OPT0.9 -2-Octet Float Value 2 Byte() Low C V V V I O O 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00	Read erver It Id 75 76 79	ew ems Datapoin Description	s DatapointType DPT01 - Binary DPT03 - Beit Unsigned Value DPT03 - 2-Octer Float Value DPT01 - Binary DPT01 - Binary	Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(s)	Priority Low Low Low	с с с с с с с	R - -	w w w w	192.1 T - - U - U - U	68.1 1 1 1 1	Raw Value(hex)	s 777 Ms Interpreted Value False 0 0 0 0 False Exten		# Indicatio 0 0 0 0	ons	
Bit Derivation Derivation <td>Read erver It Id 74 75 76 79 82 85</td> <td>ew ems Datapoin Description</td> <td>5 Datapoint Type OPT 01 - Binary OPT 05 - 8-Bit Unsigned Value OPT 00 - 2-Octet Float Value OPT 01 - Binary OPT 01 - Binary</td> <td>Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s)</td> <td>Priority Low Low Low Low Low</td> <td>C C C C C C C C C C C C C C C C C C C</td> <td>R - -</td> <td>w 1 • w w w w</td> <td>192.1 T - U - U - U - U</td> <td>68.1 1 1 1 1 1 1 1</td> <td>Raw Value(hex)</td> <td>s 777 Ms Interpreted Value False 0 0.00 False Fa</td> <td></td> <td># Indicatio 0 0 0 0 0</td> <td>ons</td>	Read erver It Id 74 75 76 79 82 85	ew ems Datapoin Description	5 Datapoint Type OPT 01 - Binary OPT 05 - 8-Bit Unsigned Value OPT 00 - 2-Octet Float Value OPT 01 - Binary OPT 01 - Binary	Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s)	Priority Low Low Low Low Low	C C C C C C C C C C C C C C C C C C C	R - -	w 1 • w w w w	192.1 T - U - U - U - U	68.1 1 1 1 1 1 1 1	Raw Value(hex)	s 777 Ms Interpreted Value False 0 0.00 False Fa		# Indicatio 0 0 0 0 0	ons	
24 OP 103 - Binny 2 bit(s) Core C V 0 1 0 1 bit(s) 1 bit(s) <th1 bit(s)<="" th=""> 1 bit(s)<</th1>	Read erver It Id 74 75 76 79 82 85 88	ew ems Datapoin Description	5 DatapointType DP101 - Binary DP105 - Bit Unkingned Value DP109 - 2-Octet Float Value DP101 - Binary DP101 - Binary DP101 - Binary DP101 - Binary	Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s)	Priority Low Low Low Low Low Low	с с с с с с с с с с с	R - - -	W 1 • • • • W W W W W W	192.1 T U T U - U - U - U - U - U	68.1 1 1 1 1 1 1 1 1	Raw Value(hex)	s 777 Ms Interpreted Value False O False False False False False False		# Indication 0 0 0 0 0 0 0	ons	
OP 100 - 2-Octet Riost Value 2 Byte(t) Core C V I 00 0.000 0.000 0 99 OP 100 - 2-Octet Riost Value 2 Byte(t) Low C - W T U I 0.000 0.000 0 98 OP 100 - 2-Octet Riost Value 2 Byte(t) Low C - W T U I 0.000 0.000 0.000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 <td< td=""><td>Read erver It Id 74 75 76 79 82 85 88 91</td><td>ew Datapoin Description</td><td>5 DatapointType DPT 01 - Binary DPT 05 - Bit Unigned Value DPT 01 - Binary DPT 01 - Binary</td><td>Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s)</td><td>Priority Low Low Low Low Low Low Low</td><td>C C C C C C C C C C C C C C C C C C C</td><td>R - - -</td><td>W W W W W W W W W W W</td><td>192.1 T - U - U - U - U - U - U - U</td><td>68.1 1 1 1 1 1 1 1 1</td><td>Raw Value(hex)</td><td>s 777 Ms Interpreted Value Fabia 0 0.00 Fabia Fabia</td><td></td><td># Indication 0 0 0 0 0 0 0 0 0</td><td>ons</td></td<>	Read erver It Id 74 75 76 79 82 85 88 91	ew Datapoin Description	5 DatapointType DPT 01 - Binary DPT 05 - Bit Unigned Value DPT 01 - Binary DPT 01 - Binary	Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s)	Priority Low Low Low Low Low Low Low	C C C C C C C C C C C C C C C C C C C	R - - -	W W W W W W W W W W W	192.1 T - U - U - U - U - U - U - U	68.1 1 1 1 1 1 1 1 1	Raw Value(hex)	s 777 Ms Interpreted Value Fabia 0 0.00 Fabia		# Indication 0 0 0 0 0 0 0 0 0	ons	
OP 109 - 2-octet Rolar Mark 2 byte(s) Core W C I O 1 O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Read erver II Id 74 75 76 79 82 85 88 91 94	ew Datapoin Description	b DatapointType DP101 - Binny DP105 - 8-Bit Unkigned Value DP109 - 2-Octet Float Value DP101 - Binny DP101 - Binny DP101 - Binny DP101 - Binny DP101 - Binny DP101 - Binny	Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s)	Priority Low Low Low Low Low Low Low	C C C C C C C C C C C C C C C C C C C	R - - -	W W W W W W W W W W	192.1 T - U U U U U U U U U U U U U	68.1 1 1 1 1 1 1 1 1 1 1 1	Raw Value(hex) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	s 777 Hs Interpreted Value Fabs 0 0 0 0 0 0 Fabs Fabs Fabs Fabs Fabs Fabs Fabs Fabs		# Indication 0 0 0 0 0 0 0 0 0 0 0	ons	
DF 10 S - Control 1 βyte(t) Control C A T V Control Control <thcontrol< th=""></thcontrol<>	Read erver It Id 74 75 76 79 82 85 88 91 94 92 70	ew Datapoin Description	B DatapointType DPT01 - Binary DPT09 - 2-Bit Unsigned Value DPT09 - 2-Octor Float Value DPT01 - Binary DPT01 - Binary DPT01 - Binary DPT01 - Binary DPT01 - Binary DPT01 - Binary	Size 1 Bit(s) 1 Byte(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s)	Priority Low Low Low Low Low Low Low	C C C C C C C C C C C C C C C C C C C	R - - - - - - - -	W 1 W 2 W 2 W 2 W 2 W 2 W 2 W 2 W 2	192.1 192.1 1 - U - U - U - U - U - U - U - U	68.1 1 1 1 1 1 1 1 1 1 1 1	Raw Value(hex) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	s 777 Ms Interpreted Value Faire 0 0 0 0 0 0 Faire Fa		# Indicatie 0 0 0 0 0 0 0 0 0 0 0 0	ons	
Dirac Dira Dirac Dirac <th< td=""><td>Read erver II Id 74 75 76 79 82 85 88 91 94 97 08</td><td>ew Datapoin Description</td><td>5 Drtapoint Type OP101 - Binny OP105 - 8-Bit Unsigned Value OP109 - 2-Octet Float Value OP101 - Binny OP101 - Binny</td><td>Size 1 Bit(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s)</td><td>Priority Low Low Low Low Low Low Low Low</td><td></td><td>R - - - -</td><td>W W W W W W W W W W W W W W W W</td><td>192.1 F U T - U - U - U - U - U - U - U - U - U - U - U - U - U - U - U - - - - - - - - - - - - -</td><td>68.1 1 1 1 1 1 1 1 1 1 1 1 1</td><td>Raw Value(hex)</td><td>s 777 Hs Interpreted Value False Fa</td><td></td><td># Indicatie 0 0 0 0 0 0 0 0 0 0 0 0 0</td><td>ons</td></th<>	Read erver II Id 74 75 76 79 82 85 88 91 94 97 08	ew Datapoin Description	5 Drtapoint Type OP101 - Binny OP105 - 8-Bit Unsigned Value OP109 - 2-Octet Float Value OP101 - Binny OP101 - Binny	Size 1 Bit(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s)	Priority Low Low Low Low Low Low Low Low		R - - - -	W W W W W W W W W W W W W W W W	192.1 F U T - U - U - U - U - U - U - U - U - U - U - U - U - U - U - U - - - - - - - - - - - - -	68.1 1 1 1 1 1 1 1 1 1 1 1 1	Raw Value(hex)	s 777 Hs Interpreted Value False Fa		# Indicatie 0 0 0 0 0 0 0 0 0 0 0 0 0	ons	
Dif4 OPT222 - 3-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0	Read erver II Id 74 75 76 79 82 85 88 91 94 97 98	ew Datapoin Description	5 DatapointType DPT03 - Binary DPT03 - Selt Unsigned Value DPT03 - Selt Unsigned Value DPT01 - Binary DPT01 - Senary DPT01 - Senary DPT01 - Second Float Value DPT03 - 2-Octer Float Value DPT03 - 2-Octer Float Value	Size 1 Bit(s) 2 Byte(s) 2 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s) 2 Byte(s) 2 Byte(s)	Priority Low Low Low Low Low Low Low Low Low Low		R - - - - - - - - - - -	W V W W W W W W W W W W W W W W	192.1 T - - U - U - U - U - U - U - U -	68.1 1 - 1 1 1 1 1 1 1 1 1 1 1 1 1	Raw Value(hex) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	s 777 Ms Interpreted Value Fabie 0 0 0 0 fabie Fabie Fabie Fabie Fabie Fabie Fabie Fabie Cod 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		# Indicatie 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ons	
View C V C V O I 0000 RE0000 RE0000 RE0000 RE00000 RE00000 RE00000 RE000000 RE000000 RE000000 RE000000 RE000000 RE000000 RE000000 RE000000 RE0000000 RE0000000 RE0000000 RE0000000 RE0000000 RE0000000 RE0000000 RE00000000 RE00000000 RE00000000 RE000000000 RE000000000 RE00000000000000000000000 RE000000000000000000000000000000000000	Read erver II Id 74 75 76 79 82 85 88 91 94 97 96 100 103	ew Datapoin Description	b DetapointType DPT 01 - Binary DPT 05 - 8-Bit Unsigned Value DPT 00 - 2-Octet Float Value DPT 01 - Binary DPT 00 - 2-Octet Float Value DPT 03 - 2-Octet Float Value DPT 03 - 2-Octet Float Value DPT 32 - 3-Octet Float Value	Size 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s) 2 Byte(s) 1 Byte(s) 2 Byte(s) 1 Byte(s) 2	Priority Low Low Low Low Low Low Low Low Low Low		R - - - - - - - - - - - - - - - - - - -	W 1 W W W W W W W W W W W W W W W W	192.1 T - U U U U U U U U U U U U U	68.1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	.38 KNX IP Bao Raw Value(hex) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	S 777 MS Interpreted Value Fabie 0.00 0.00 Fabie Colo Colo Colo Colo Colo Colo Colo Colo Colo	0	# Indicatie 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ons	
130 DPT05 - 8-8t Unsigned Value 1 Byte(s) Low C - - - 0 0 0 0 133 DPT05 - 8-8t Unsigned Value 1 Byte(s) Low C - V I 0 0 0 0	Read erver II Id 74 75 76 79 82 85 88 89 91 94 99 99 100 103 104	ew Datapoin Description	s DatapointType DP101 - Binary DP105 - Bit Unsigned Value DP100 - Bit Unsigned Value DP101 - Binary DP101 - D- Octot Float Value DP109 - 2- Octot Float Value	Size 1 Bit(s) 2 Byte(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s) 2 Byte(s) 3 Byte(s)	Priority Low Low Low Low Low Low Low Low Low Low		R 	W 1 W W W W W W W W W W W W W W W W W W	192.1 T - U U U U U U U U U U U U U	68.1 1 1 1 1 1 1 1 1 1 1 1 1 1	.38 KNX IP Bao Raw Value(hex) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	s 777 Hs Interpreted Value Faise 0 0.00 Faise Faise Faise Faise Faise Faise Faise Faise Faise Control	0	# Indication 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ons í	
133 DPT 05 - 8-Bit Unsigned Value 1 Byte(s) Low C - T - 00 0 0	Read erver II Id 74 75 76 79 82 85 88 89 91 94 99 99 100 103 104 127	ew Datapoin Description	5 DatapointType DPT 01 - Binary DPT 02 - Binary DPT 03 - Binary DPT 03 - Binary DPT 01 - Binary DPT 02 - Cotter Float Value DPT 03 - Cotter Float Value DPT 03 - Cotter Float Value DPT 03 - Cotter Float Value DPT 04 - Cotter Float Value DPT 05 - 2- Octer Float Value DP	Size 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s) 2 Byte(s) 3	Priority Low Low Low Low Low Low Low Low Low Low		R 	W 1 W W W W W W W W W W W W W W W W W W	192.1 r U r U - U - U - U - U - U - U - U -	68.1 1 1 1 1 1 1 1 1 1 1 1 1 1		s 777 Ms Interpreted Value Faise 0 0.00 Faise F	0	# Indicati 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ons 1	
1.5 Criter of the state of the	Read erver II Id 74 75 76 79 82 85 88 91 94 97 98 100 103 104 127 120	ew Datapoin Description	Ditapoint Type DP101 - Binny DP102 - Binny DP103 - Bint DP109 - 2-Octet Float Value DP101 - Binny DP102 - Soctet Float Value DP103 - Binny DP104 - Binny DP105 - Octet Float Value DP105 - Soctet Float Value DP108 - Soctet Float Value DP122 - Soctet Float Value DP122 - Soctet Float Value DP122 - Soctet Float Value DP107 - Binny DP107 - Binny	Size 1 Bit(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s) 1 Byte(s) 3 Byte(s) 1	Priority Low Low Low Low Low Low Low Low Low Low	C C C C C C C C C C C C C C C C C C C	R 	W 1 W 2 W 2 W 2 W 2 W 2 W 2 W 2 W 2 W 2 W 2	192.1 r U r U - U - U - U - U - U - U - U -	68.1 1 1 1 1 1 1 1 1 1 1 1 1 1		s 777 Ms Traine Faise Coo O O O O O O O O O O O O O O O O O O	0	# Indication 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ons *	
124 DDT 05 - 8-Bit Unrigoned Value 1 Pate(r) Low C - W - II I 00 0	Read erver II Id 74 75 76 79 82 85 88 91 94 97 98 100 103 104 127 130	erer Datapoin	B DatapointType DPT01 - Binary DPT02 - Binary DPT02 - Binary DPT01 - Binary DPT02 - Societ Float Value DPT03 - Societ Float Value - Societ Float Value - Societ Float Val	Size 1 Bit(s) 2 Byte(c) 2 Byte(c) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s) 2 Byte(s) 3 Byte(s) 3 Byte(s) 1 Byte(s)	Priority Low Low Low Low Low Low Low Low Low Low		R 	W 1 W 2 W 2 W 2 W 2 W 2 W 2 W 2 W 2	192.1 T - U U U U U U U U U U U U U	68.1 1 - 1 1 1 1 1 1 1 1 1 1 1 1 1	.38 KIX IP Bao Raw Value(hea) 00 00 00 00 00 00 00 00 00 0	s 777 Ms Interpreted Value Faire 0 0.00 Faire Good Bao O O	0	# Indicatie 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ons *	
Arr 0 07 00 2 00 00 00 00 00 0 0 0 0 0 0 0	Read erver II Id 74 75 76 82 88 88 91 94 97 98 100 103 104 127 130 133	ew Datapoin Description	B DetapointType OPT 01 - Sinary OPT 05 - 8-Bit Unsigned Value OPT 05 - 8-Bit Unsigned Value OPT 01 - Sinary OPT 02 - Societ Float Value OPT 03 - Societ RGS Value OPT 03 - Societ RGS Value OPT 05 - Societ RGS Value OPT	Size 1 Bit(s) 2 Byte(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 1 Bit(s) 2 Byte(s) 2 Byte(s) 3 Byte(s) 3 Byte(s) 1 Byte(s)	Priority Low Low Low Low Low Low Low Low Low Low		R 	W 1 W 2 W 2 W 2 W 2 W 2 W 2 W 2 W 2	192.1 T - U U U U U U U U U U U U U	68.1 1 - 1 1 1 1 1 1 1 1 1 1 1 1 - - - 1 - - - - - - - - - - - - -		777 Ms Interpreted Value Failue Catla Activate (0), Scenee R=0 G=0 B=0 R=0 G=0 0 0 0 0 0 0	0	# Indicatis 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ons f	

Net'n Node telegram view with BAOS data points

5.4 SDKs for BAOS Protocols

For easy integration of the KNX BAOS protocol into different environments free SDKs are available in source code also for download on our web page:

- SDK for BAOS Serial Protocol
- SDK for BAOS Binary Services
- SDK for BAOS RESTful Web Services

5.4.1 Demo Application for BAOS Serial Protocol

A free software framework is available for download on our web page which integrates the FT1.2 protocol as well as the BAOS protocol. A simple application shows the usage of the communication drivers for embedded micros.

The sample is prepared for Atmel/Microchip SAM D20 microcontroller and can be used with the Atmel Studio free compiler and debugger. The project is prepared to run directly on the development board of our starter kit for KNX BAOS 83x Modules.



5.4.2 SDK for BAOS Binary Services

The Software Development Kit SDK for BAOS Binary Services is a cross-platform C++ implementation of the KNX BAOS Binary Protocol. It can be used for a rapid development of native applications for KNX control which use the KNX BAOS Interfaces and Modules. The SDK is prepared but not limited to Linux and Windows.

5.4.3 SDK for BAOS RESTful Web Services

The SDK for BAOS RESTful Web Services is a Java Script implementation of the KNX BAOS RESTful Web Services Protocol. It can be used for the rapid development of web applications for KNX control which are using the KNX BAOS IP 777 Interfaces.

6 From BAOS to your Product

6.1 Product certification with BAOS

For decades, its cross-manufacturer compatibility of various applications and products is the main pillar of the KNX system. This is achieved by the advanced certification system of the KNX Association. All devices with a KNX logo must be tested by a test laboratory accredited by the KNX Association for compatibility.

The KNX BAOS modules and devices build a perfect basis for applications which can be certified because a certified KNX Stack is implemented. If you are using a BAOS component the Stack is not required to be tested again. Therefore only the application specific tests, i.e. interworking and functionality, are required.

Weinzierl has its own accredited test lab for KNX certification. If you have any questions regarding product qualification please refer to our website or contact us directly.

6.2 ...and what about the KNX Stack?

A KNX Stack is a software solution as a basis for KNX device development. Also in all KNX BAOS Modules or Devices a Weinzierl KNX Stack is running. The usage of a stack gives a maximum on flexibility and can be integrated with the application on a single micro.

The application development on a KNX Stack typically requires more effort and system knowledge compared to the usage of a BAOS solution. It is mainly used for devices with a higher volume. For more information, please visit our web page at weinzierl.de

6.3 Any questions left?

The choice of the best architecture for an application is not trivial but always necessary at the beginning. If you have any questions how to start KNX development or about our solutions please do not hesitate to contact us.