REAL SMART HOME GmbH

APPMODULE
ASCII Tool App Documentation

Version: 2.0.1
Type: Application
Article No.: BAB-0007

Documentation version I
Actual state 05/2019
Date: 17. September 2019

EN

REAL SMART HOME GmbH

Hörder Burgstraße
D-44139 Dortmund

Email: info@realsmarthome.de

Tel.: +49 (0) 231 – 586974 -00
Fax.: +49 (0) 231 – 586974 -15
www.realsmarthome.de

TABLE OF CONTENTS

1 INTRODUCTION

Thank you for your trust, and the purchase of the **ASCII Tool** -app for the BAB **APP**MODULE. With the **ASCII Tool** - app, which is based on user-defined bidirectional TCP and UDP protocol commands, you get one of the most comprehensive integration solutions. This documentation will help you get started with the app and aims to improve your setup experience.

REAL SMART HOME GmbH

IMPORTANT INFORMATION ON THE OPERATING INSTRUCTIONS

We reserve the right continually improves the product. This entails the possibility that parts of this documentation might be out-of-date. You will find the latest information at:

www.bab-appmarket.de

2 ASCII TOOL – FUNCTIONAL OVERVIEW

This app allows for bi-directional UDP or TCP/IP ASCII. Create a protocol from some 32 freely definable commands and responses including unique EOL marker per each instance. Up to 10 instances can be created.

2.1 HIGHLIGHTS

Per command:

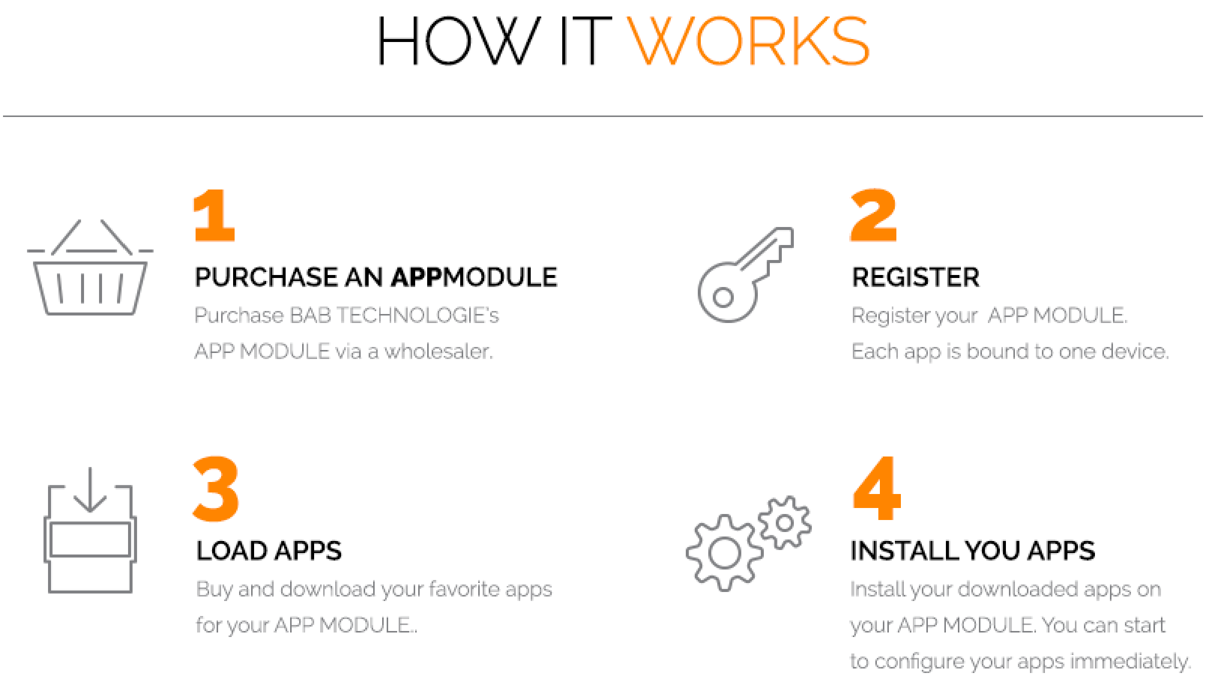
- Tripping address (1 Bit / 1 Byte) and tripping value (static or equal)
- Command text incl. maximum one variable taken from an optional value address
- Option of sending the command regularly at any interval (e.g.: value polling)

Per response:

- Text of the expected response incl. one variable (integer or float) which is then selected and sent to a value address
- Alternative text for the response without variable and self-defined value which is sent to the value address in this case

3 THE INNOVATIVE, MODULAR APP-CONEPT FOR THE BUILDING AUTOMATION

The innovative, modular app concept for building automation. The **APP**MODULE brings the innovative, modular app concept into building automation. You can mix and match any of the diverse applications that are available to ingrate third-party solutions. With these apps from the dedicated **BAB** APPMARKET, the **APP**MODULE becomes a tailor-made integration unit for your building automation.



HOW IT WORKS

1 PURCHASE AN APPMODULE
Purchase BAB TECHNOLOGIE's APP MODULE via a wholesaler.

2 REGISTER
Register your APP MODULE. Each app is bound to one device.

3 LOAD APPS
Buy and download your favorite apps for your APP MODULE..

4 INSTALL YOU APPS
Install your downloaded apps on your APP MODULE. You can start to configure your apps immediately.

Manufacturer of the **APP**MODULE BAB TECHNOLOGIE GmbH

Distribution of all apps for the **APP**MODULE BAB APP MARKET GmbH

App developer REAL SMART HOME GmbH

3.1 INFORMATION ABOUT THE APPMODULE

Please refer to the separate product documentation of the **APP**MODULE for a detailed product description and setup instructions.

http://www.bab-tec.de/index.php/download_de.html

Product variants:
The **APP**MODULE is available in three variants:

- **APP**MODULE KNX/TP – for stand-alone use on KNX/TP Bus
- **APP**MODULE EnOcean – for stand-alone use in the EnOcean wireless network
- **APP**MODULE Extension – for use in an IP-based KNX installation (KNXnet/IP) or as extension for an EIB**PORT**

4 APP INSTALLATION / UPDATE

Please proceed as follows to install an App.

1. Open the APPMODULE web page: Enter <IP Address of APPMODULE> into your browser's address bar and press Enter. The APPMODULE web interface will appear.

2. Log in with your user credentials. Please refer to the APPMODULE documentation for login details.

3. Click on the menu entry "App Manager"

4. You are now on the page where already installed Apps are listed. The list will be empty if no apps have been installed. Click "Install App" in order to install a new app.

5. Now click on "Select App"; a file selector window will appear. Choose the app »ASCII Tool« and click "OK".
   The Smart Home App "ASCII Tool" must first be downloaded from the BAB APP MARKET (www.bab-appmarket.de).

6. After the message "Installation successful" appears, click "OK". You are ready to configure the App.

7. To update an already installed app, click on the App icon in the "App Manager".

8. The detail view of the App appears. Click on "Update App" to select the app package and start the update. The update version must be downloaded from the BAB APP MARKET.

After the message "Installation successful" appears, click "OK". The app has been updated. Your instance configurations will remain unchanged.

Information
To configurate the App please use Google Chrome.

5 APP SETTINGS

This App allows for bi-directional TCP/IP or UDP ASCII. Create a protocol from some 32 freely definable commands and responses including unique EOL marker.

5.1 INSTANCE

As soon as the App is installed, you can create so called “Instance”. An Instance is one of several objects of the same class.

In order to create an instance, click on the following symbol "Create Instance".

Instance Name:

Choose a name for this new instance.

Comment:

Insert a description what this instance does.

5.1.1 CONNECTION PARAMETERS

Destination IP:

Insert the IP address of your destination device (it must have a static IP for the app to work reliably).

Communication Port

The port number on which the device listens (default is 23 for Telnet). This port is used for UDP and TCP/IP protocol.

Protocol Type:

Select the protocol for communication with the device. The TCP and UDP protocols are available for selection.

Receive Port for UDP Communication:

Local port in the **APP**MODULE for responses during bi-directional UDP communication. Please watch that this port can then only be used by this app instance and must differ from receive ports in other instances. Port 1735 and 1900 are being already used for other function in the **APP**MODULE and will abort this instance when set as receive port.

Socket Timeout:

Defines the timeout for closing the connection:

- -1: close connection immediately (any potential reply by the device will not be received)
- 0: never close connection
- 1 - 3600: time in seconds after which the socket will be closed. This time will be reset for each command that has been sent to the server.

Connect at Instance Start:

If the device you are connecting to is sending information on its own, i.e. without a prior request command, then you should connect to it immediately as the instance starts (and set the socket timeout to 0). In all other cases, it is enough to establish a connection as soon as you send the first command to the device and responses will be evaluated depending on the socket timeout configuration.

5.1.2 PROTOCOL

Command End-of-Line Marker:

Define how outgoing commands should be terminated. This string will be attached to any command you define below.

5.1.2.1 COMMANDS

Up to 32 commands (256 in the pro version) can be triggered by telegrams or automatically sent in intervals.

Comment:

Insert a comment to describe the command.

Trigger Address (EIS 1 / EIS 14):

A telegram on this address will trigger the command if it matches the trigger value.

Trigger Value (-1 .. 255):

Defines which value on the Trigger Address will trigger this command:

- -1: Wildcard. Any telegram will trigger this command
- 0-255: This value will trigger this command

Auto Send Interval (in seconds):

Instead of manual triggering, or in addition to it, you can have the command sent automatically. This is useful for state polling. If you set this up, the state of the trigger address will be ignored of course, the command will be sent in the desired interval in any case.
0 means no automatic sending.

Text to Send:

This text will be sent to the destination host. Add %d or %f for dynamic values (see help text for 'Value Address').

Value Address:

If a dynamic value (e.g. music volume) needs to be sent within the ASCII command, you can send it on this address and it will be inserted at the desired position. To do this, just add %d for decimal values or %f for floating point values at any position into the text. Please note that expert settings for the wildcard are possible. Check sprintf documentation. For example, %02d will output one-digit integer values as 0x instead of x.

Value Data Type:

The data type of the dynamic value:

- 0%...100%
- 2 Byte 0...65535
- 1 Byte 0...255
- EIS 5: 2 Byte Floating Point
- EIS 6: 0-100%
- EIS 10: 2 Byte 0-65535
- EIS 14: 1 Byte 0-255

Response End-of-Line Marker:

Define how your protocol responses are terminated.

5.1.2.2 RESPONSES

Up to 32 responses (256 in the pro version) can trigger telegrams.

Comment:

Insert a comment to describe the response.

Expected Response:

This is the answer you expect from the destination host.
If the expected response contains a dynamic value that you want to send to your KNX devices (e.g. music volume), you can add patterns to be matched into this field. To do this, just add %d for decimal values or %f for floating point values into the text at any position. E.g. 'VOLUME NEW VALUE %d' for an imaginary protocol that sends responses likes 'VOLUME NEW VALUE 42'. The value 42 would then be sent on your response value group address, ignoring any response value you might have set in the field Response Value.

Response Value:

This is the value you want to send via KNX when you get your expected answer. Leave empty if you use %d or %f wildcards in 'Expected Response', because then the response value will be what matches the wildcard.

Response Value Group Address:

This is the group address to which your response value will be sent.

Response Value Data Type:

The data type for response values:
- 0%...100%
- 2 Byte 0...65535
- 1 Byte 0...255
- EIS 1: 1 Bit
- EIS 5: 2 Byte Floating Point
- EIS 6: 0-100%
- EIS 10: 2 Byte 0-65535
- EIS 14: 1 Byte 0-255

6 ATTACHMENT

function	EIS type	DPT	typical function	typical values	data	identifier
PriorityPosition	EIS1	DPT1	Wind alarm	1=high and inhibit	1 Bit	1-bit
Switch	EIS1	DPT1	Light switching	0=Off; 1=On	1 Bit	1-bit
DimControl	EIS2	DPT3	Dimming	0=Off; 1=On xxxx=relative dimming 0-255=absolute dimming	1Bit 4Bit 8Bit	3-bit controlled
Time	EIS3	DPT10	Time	Hhh:mm:ss	3 Byte	Time
Date	EIS4	DPT11	Date	dd:mm:yyyy	3 Byte	Date
Value	EIS5	DPT9	Value	0-255	1Byte	2-byte float value
DimValue	EIS6	DPT5	Percent	0-100%	1Byte	8-bit unsigned value
DriveBlade Value	EIS6	DPT5	Position value	0-100%; 0-255	1Byte	8-bit unsigned value
DriveShutter Value	EIS6	DPT5	Position value	0-100%; 0-255	1Byte	8-bit unsigned value
Position	EIS6	DPT5	Control value Heating	0-100%; 0-255	1Byte	8-bit unsigned value
DriveMove	EIS7	DPT1	Move shutter	0=up 1=down	1Bit	1-bit
DriveStep	EIS7	DPT1	Adjusting the slat blind	0=up; 1= down; 0 or 1 during movement=stop	1Bit	1-bit
PriorityControl	EIS8	DPT2	Priority	0,1 switch; 3=forced off; 4=forced on	2Bit	1-bit controlled
FloatValue	EIS9	DPT14	IEEE	Floating-point value	4 Byte	4-byte float value
Counter 16bit	EIS10	DPT7	Counter 16 bit	0 - 65.535	2Byte	2-byte unsigned value
Counter 16bit	EIS10	DPT8	Counter 16 bit with sign	-32.768 - 32.767	2Byte	2-byte signed value
Counter 32bit	EIS11	DPT12	Counter 32 bit	0 - 4.294.967.295	4Byte	4-byte unsigned value
Counter 32bit	EIS11	DPT13	Counter 32 bit with sign	0 - 4.294.967.295	4Byte	4-byte signed value
Access Control	EIS12	DPT15	Access control	Card number	4Byte	Entrance access
Char	EIS13	DPT4	ASCII characters	Character	1Byte	Character
Counter 8bit	EIS14	DPT5	Value	0 - 255	1Byte	8-bit unsigned value
Counter 8bit	EIS14	DPT6	Value with sign	-128 - 127	1Byte	8-bit signed value
String	EIS15	DPT16	String	max. 14 characters	14 Byte	Character string

EIB/KNX devices exchange fixed prescribed data formats with each other. These are defined in types.
The old designations of the types are EIS (EIB Interworking Standard)
The new designations are DPT (Data Point Type)