

BaosModuleConfig

User Manual for Version 2.2.1

WEINZIERL ENGINEERING GmbH

Achatz 3-4

84508 Burgkirchen / Alz

E-Mail: info@weinzierl.de

Web: www.weinzierl.de

Table of Content

BaosModuleConfig	3
1 Description	3
2 Supported Modules	3
3 Installation.....	3
4 Overview.....	3
4.1 Menu bar	4
4.2 Tool bar.....	4
4.3 Main area.....	4
5 Configure a BAOS module	4
5.1 Establish a connection.....	4
5.1.1 Select Interfaces to scan	5
5.1.2 Port Configuration.....	5
5.2 BAOS Module Address	6
5.3 Authorize Properties (other than 830.1).....	7
5.4 Authorize OEM Production (830.1).....	7
5.5 Properties	8
5.5.1 Read	8
5.5.2 BAOS Configuration.....	8
5.5.3 Write.....	9
5.5.4 Log folder.....	9
6 Automation	10
7 FDSK <-> QR Code Converter.....	10

Change History

Summary of Changes	Revision Date	Author
Creation	2013-12-02	Ms
Adapted formatting	2014-03-18	Ms
Extended for new Modules 83x	2016-10-18	Ef
Translate the page numbers to English, remove telephone and fax number	2016-11-29	Ef
Update version to 2.0.1	2018-09-11	Ef
Add section Installation	2018-09-17	Ef
Update version to 2.1.0	2019-04-01	Ef
Update version to 2.1.1	2019-12-03	Ef
Update version to 2.2.0, add support for Modules 830.1	2025-07-22	Ef
Update version to 2.2.1, fix support for Modules 830.1	2026-08-12	MEf

BaosModuleConfig

1 Description

BaosModuleConfig is a configuration tool to customize the vendor specific data of Weinzierl KNX BAOS modules. These settings are necessary for a correct device discovery by the ETS® Software.

2 Supported Modules

- KNX BAOS Module 830
- KNX BAOS Module 832
- KNX BAOS Module 830.1
- KNX BAOS 83x Chip

3 Installation

To run the BaosModuleConfig program, the Microsoft Visual C++ Redistributable Package for Visual Studio 2019 (x86) is required. If it's not already installed on the system an error message like Figure 1 will appear on start. In this case the package needs to be installed on the system. It can be downloaded from the Microsoft homepage (<https://www.microsoft.com>).

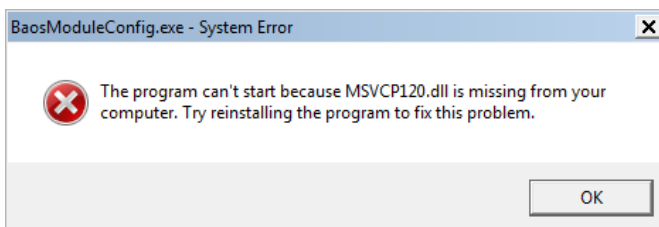


Figure 1: Missing Microsoft Visual C++ Redistributable Package error message

4 Overview

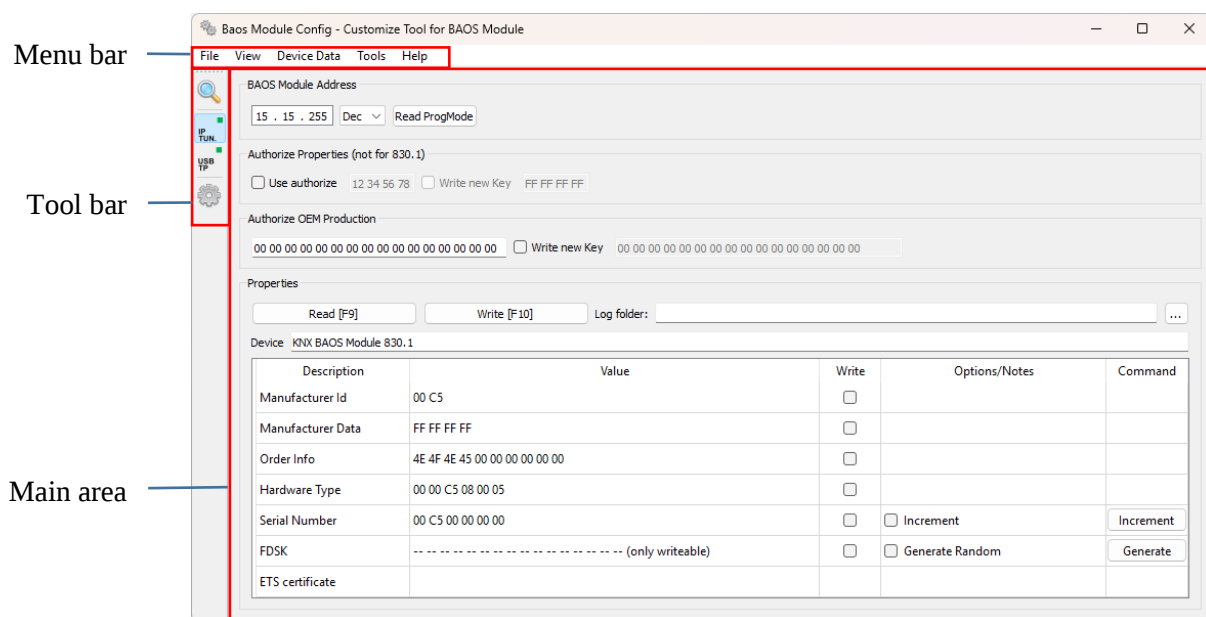


Figure 2: Overview with Module 830.1

4.1 Menu bar

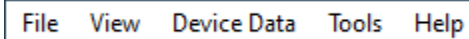


Figure 3: Menu bar

The menu bar offers additional options:

- | | |
|--|---|
| - File → Exit (Shortcut <i>Alt+F4</i>) | Close the program. |
| - View → Access Port Configuration | Show and configure the configuration of the Interfaces. |
| - Device Data → Read (Shortcut <i>F9</i>) | Read properties, see also section 5.5.1 Read. |
| - Device Data → Write (Shortcut <i>F10</i>) | Write properties, see also section 5.5.3 Write. |
| - Tools → FDSK <-> QR Code Converter | Opens a dialog to convert the FDSK to QR Code, see also section 7 FDSK <-> QR Code Converter. |
| - Help → Help (Shortcut <i>F1</i>) | Open the manual. |
| - Help → About | Show the 'About dialog' for the BaosModuleConfig program. |

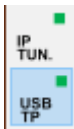
4.2 Tool bar



Scan:

Search for connected KNX interface devices of certain type (refer to 5.1.1 Select Interfaces to scan).

The scan is executed automatically on application start.



Discovered interfaces:

All discovered KNX interface devices:

Select or close an interface for the bus access.



Configuration:

Display or hide a configuration view of an interface.

4.3 Main area

In this section (*Figure 2*→*Main area*) all necessary options to customize your KNX BAOS module are shown and can be changed.

For more details, see 5.2 *BAOS Module Address* and following.

5 Configure a BAOS module

5.1 Establish a connection

To establish a connection to the KNX bus select a KNX interface device on the *Tool bar*. A tool tip with the name of the device will be shown when you hover with the cursor over the icon.

A new port scan can be started by pressing the *Scan* button, which clears all ports and adds the current ones.

5.1.1 Select Interfaces to scan

Clicking on the button *Configuration* shows the ‘*Access Port Configuration*’ with a list of supported interface types at the top. In this list the interface types to scan for, can be selected. The behaviour of this *Scan* button is like the button in 4.2 *Tool bar*→*Scan*.

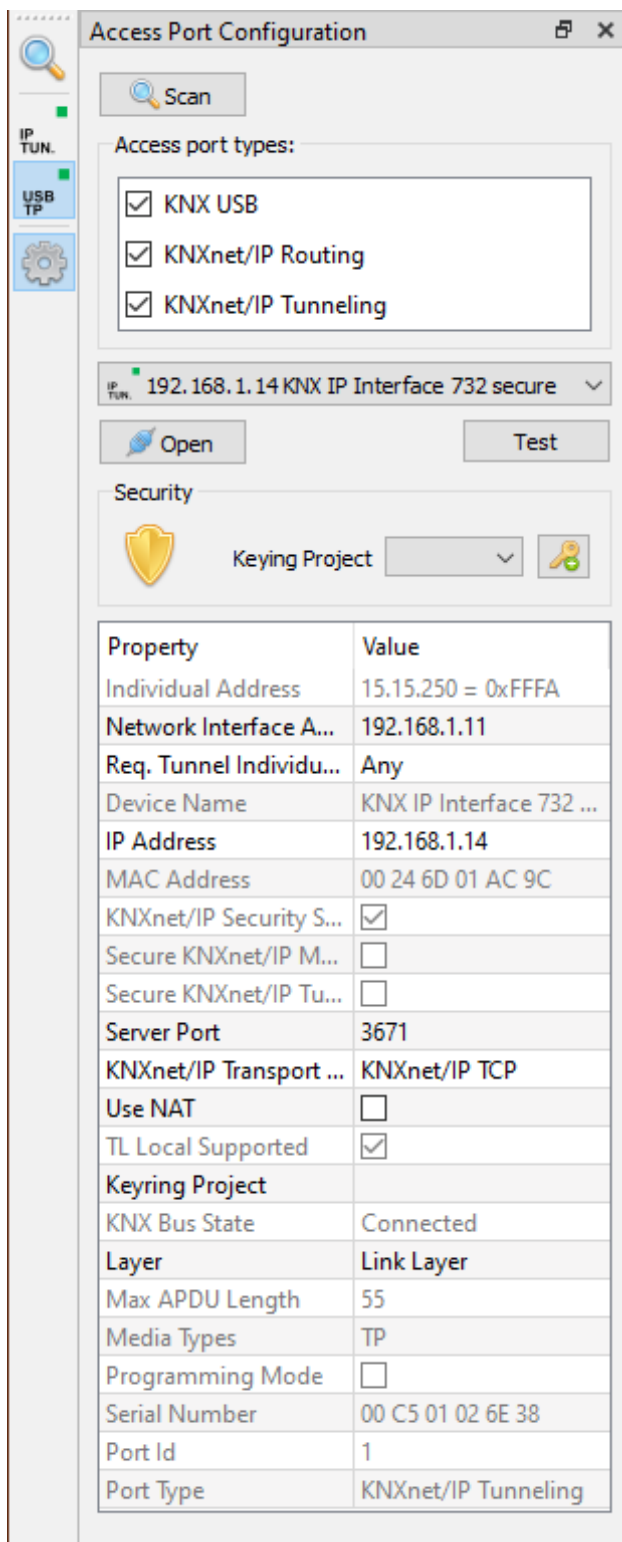


Figure 4: Access Port Configuration

5.1.2 Port Configuration

At the bottom of the ‘*Access Port Configuration*’ (Figure 4) a list of properties of the selected KNX interface will be shown. In normal operation it is not necessary to change anything here.

All discovered interfaces can be selected on the drop down menu. Open or close a connection to the selected KNX interface device with the respective button. With the *Test* button the established connection can be checked and the result is shown on the left side of the button.

5.2 BAOS Module Address

The field '*BAOS Module Address*' holds the KNX individual address of the KNX BAOS Module to be configured. If the individual address of the connected KNX BAOS module is known, it can be entered in the text field. The adjacent combo box allows to switch between decimal and hexadecimal representation of the address.

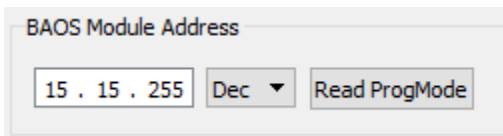


Figure 5: BAOS Module Address

Alternatively the individual address can be read from the device, when it is in programming mode. To switch on the programming mode, see the datasheet of the used module.

On a click on the '*Read ProgMode*' button the individual address of the device will be read out of the device and shown in a pop up window. On this window the programming mode can be switched off by pressing *Yes*.

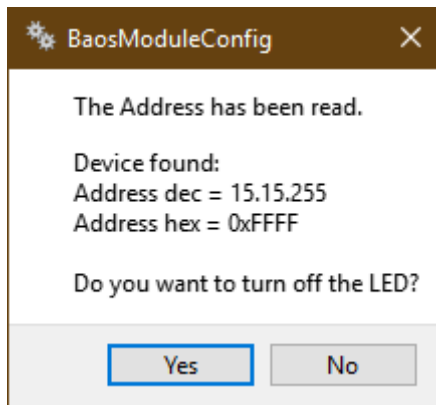


Figure 6: Read ProgMode

Now the discovered individual address of the BAOS module is stored into the address field. When no or more than one device are in programming mode a pop up dialog will report an error.

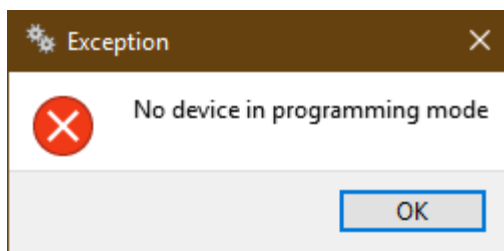


Figure 7: No device in prog. mode

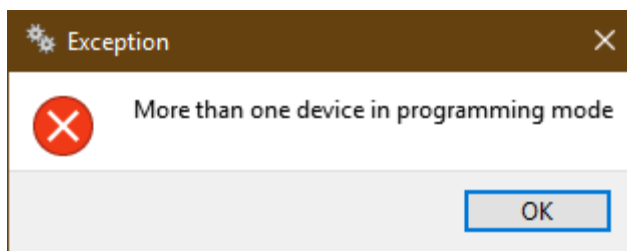


Figure 8: More than one device in prog. mode

5.3 Authorize Properties (not for 830.1)

Configuration data can be protected by authorization keys. Such keys are managed by the KNX management system. It supports 4 access levels:

Level 0: Reserved for system manufacturer, key is not public

Level 1: Reserved for device manufacturer, default key: 0x12345678

Level 2: Used by ETS, default key: 0xFFFFFFFF (not protected)

Level 3: No key for unprotected data, e.g. reading the serial number of a device

Level 1 is protected just with a default key. This key can be used to configure the IDs for ETS compatibility. The BaosModuleConfig Tool allows changing the key to any (non-public) value. The KNX BAOS Module 820 does not support access protection.

If the check box 'Use authorize' is set, the tool uses the key in the edit box for authorization.

A new, user defined authorization key can be set by enabling 'Write new Key'. The new key will be written when selecting the *Write* button on the *Properties* section (referring to 5.5.3 *Write*).



Figure 9: Authorize Properties (other than 830.1)

5.4 Authorize OEM Production (for 830.1)

Configuration data is protected by authorization OEM Production key.

To read the properties the current key must be used. When default key is set, any key is possible to read the properties. The key from the line edit is used automatically for this.

To write the properties the OEM Production key must be set to a key different to the default one (00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00).

A new, user defined authorize OEM Production key can be set by enabling 'Write new Key'. The new key will be written when selecting the *Write* button on the *Properties* section (referring to 5.5.3 *Write*).



The OEM Production key must be changed, otherwise the device can't be individualized.

Figure 10: Authorize OEM Production (830.1)

5.5 Properties

Description	Value	Write	Options/Notes	Command
Manufacturer Id	00 C5	☐		
Manufacturer Data	FF FF FF FF	☐		
Order Info	4E 4F 4E 45 00 00 00 00 00	☐		
Hardware Type	00 00 C5 08 00 05	☐		
Serial Number	00 C5 00 00 00 00	☐	☐ Increment	Increment
FDSK	----- (only writeable)	☐	☐ Generate Random	Generate
ETS certificate				

Figure 11: Properties

5.5.1 Read

To read the current configuration data from a device just press the *Read* button (shortcut key *F9*). After a while the property description and the actual values are shown, see *Figure 11*. All values are hexadecimal notation.

If it is a BAOS module 82x or 83x by Weinzierl Engineering GmbH, the type is shown in the *Device* field. If it is an unknown device, the data can't be read and written.

By pressing the *Read* button again the properties will be read again, so all unwritten changes will be lost.

The shown properties where from a BAOS module 830.1, other modules can also have less properties.

5.5.2 BAOS Configuration

Change the values of the properties to your needs corresponding to the usage of them.

The properties are used for:

- **Manufacturer ID:** This is a two byte identifier for the KNX manufacturer. The KNX manufacturer code is assigned to KNX member by the KNX Association on request. The manufacturer code is checked by ETS before starting a download.
- **Manufacturer Data:** This is manufacturer specific device information (e.g. manufacture date). ETS will not evaluate this value.
- **Order Info:** This contains a manufacturer specific Order Information (e.g. name or order number of the product). This value is read and shown by ETS in *Diagnostic/Device Info*.
- **Hardware Type:** This contains an identification number of the hardware. The highest octet always is 00h, then the manufacturer id (e.g. 00h C5h) and then three octets of user defined hardware number. This number should be unique for every hardware type. It can be used by ETS to check whether an application is compatible with a specific hardware.
- **Serial Number:** This is the KNX Serial Number of the device. The two highest octets shall be the manufacturer id and the next four octets are a device unique number. So every KNX serial number is world-wide unique.

When *Increment* is checked, it increments the Serial Number by one after every write. The *Increment* button also increments the value on every click.

- **FDSK:** This is the KNX Factory Default Setup Key of the device. This is only for devices which supports KNX security. This property is only writeable so on every read it will be reseted to unknown.

When *Generate Random* is checked, it generates a random FDSK after every write. The *Generate* button also generates it on every click.

- **ETS certificate:** This is the ETS Device Certificate, which is generated from the Serial Number and the FDSK. This value is only for your information and read-only.

With the *Export* button an image as SVG can be exported.



The FDSK must be locked prior sale to ensure the security of the module.

For more information about properties please refer to the KNX specifications.

Enter the new value as hexadecimal notation.

Every changed data will be highlighted blue until it is written to the device or module.

5.5.3 Write

Write the new values of the selected properties in the *Write* column into the KNX BAOS Module by clicking the *Write* button (Shortcut key *F10*). Now all properties should be shown in black font, unless it was incremented see 5.5.2 BAOS Configuration, to indicate that the write process was successful.

If the authorization key is missing and the device needs one to grant write access, an error dialog appears. Try to use the authorization key for the BAOS module, if necessary (refer to 5.3 *Authorize*).

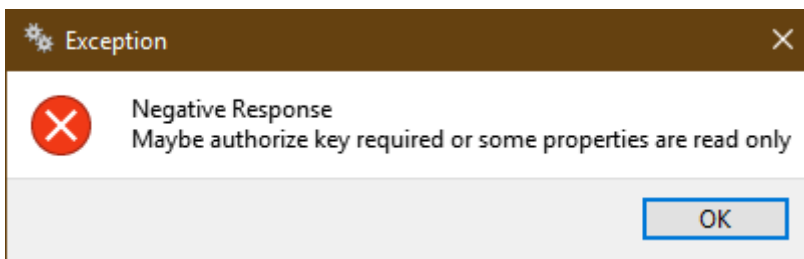


Figure 12: Negative Response, use key

If the authorization key is wrong, another error dialog appears.

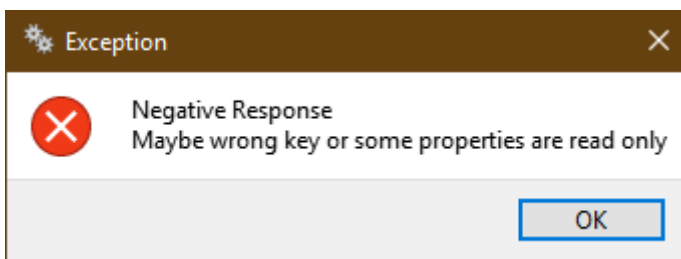


Figure 13: Negative Response, wrong key

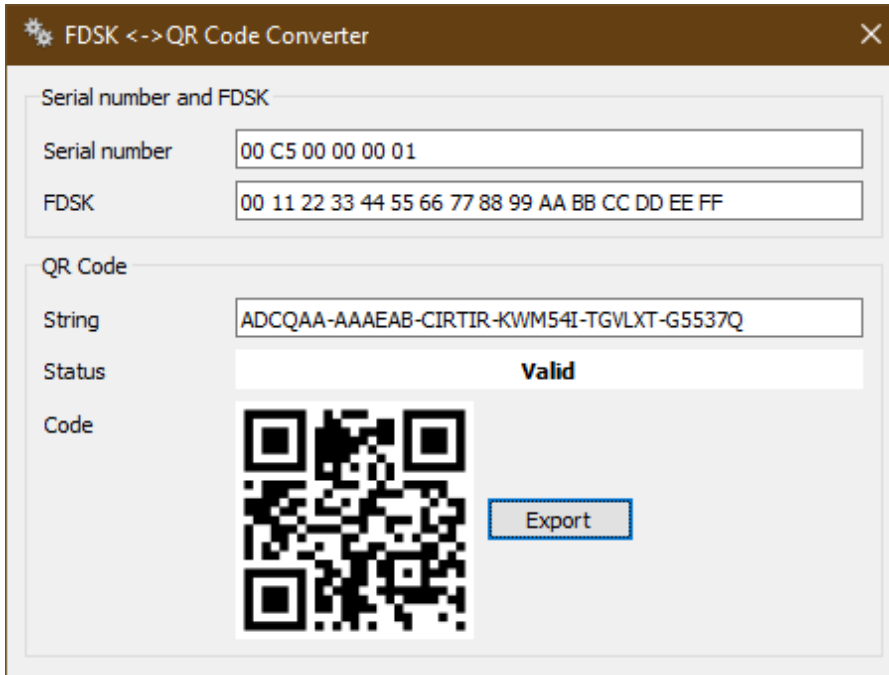
5.5.4 Log folder

A folder can be selected here in which a *logfile.txt* will be generated. These logfile contains some information about the written Serial Numbers, FDSK's and ETS certificates. Also a subfolder *images* is generated with the corresponding QR-Codes as SVG.

6 Automation

The configuration task can be automated and integrated in individual tools for production. The BaosModuleConfig SDK allows the implementation of individual tools for configuration and test.

7 FDSK <-> QR Code Converter



The screenshot shows a software window titled "FDSK <-> QR Code Converter". It has a standard Windows-style title bar with a close button. The window is divided into two main sections. The top section, "Serial number and FDSK", contains two text input fields. The "Serial number" field contains the hexadecimal string "00 C5 00 00 00 01". The "FDSK" field contains the hexadecimal string "00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF". The bottom section, "QR Code", contains a "String" input field with the value "ADCQAA-AAAEAB-CIRTIR-KWM54I-TGVLXT-G5537Q". Below this, the "Status" is displayed as "Valid". To the left of the QR code is the label "Code". To the right of the QR code is an "Export" button.

Figure 14: FDSK <-> QR Code Converter

In this dialog the Serial number and FDSK can be entered to check the corresponding ETS Certificate (QR Code→String). Or enter the ETS Certificate and check if it's valid and also the included Serial number and FDSK.

Furthermore the QR Code of the certificate is shown and can be exported as SVG via the *Export* button.