



SERIAL INTERFACE MODULE KNX

SIM-KNX EVALUATION KIT

DOCUMENTATION

V1.1

Author: Klaus Adler

Last Modification: 11.01.2007

© 2005-2007 TAPKO Technologies GmbH
Yorckstr. 22, 93049 Regensburg, Germany

The data contained herein are subject to change without notice. Tapko does not warrant for correctness or completeness of the document.
The reproduction, transmission or use of this document or it's contents is not permitted without written authority.
All rights reserved.

1 CONTENTS

1	Contents.....	3
2	Licence.....	4
3	SIM-KNX Evaluation Boards	5
3.1	SIM-KNX Evaluation Board USB	5
3.1.1	Getting started with the Evaluation Board USB.....	5
3.1.2	Block-Diagram.....	5
3.1.3	Schematic	6
3.1.4	Placement	7
3.1.5	Pictures.....	8
3.2	SIM-KNX Evaluation Board RS232.....	9
3.2.1	Getting started with the Evaluation Board RS232.....	9
3.2.2	Block-Diagram.....	9
3.2.3	Schematic	10
3.2.4	Placement	11
3.2.5	Pictures.....	12
4	SIM-KNX Application examples	13
4.1	Application examples overview	13
4.2	SIM-KNX APPLICATION EXAMPLE with VC++	14
4.2.1	Description.....	14
4.2.2	Command window	15
4.2.3	Trace Window	16
4.2.4	System requirements	16
4.3	SIM-KNX APPLICATION EXAMPLE with Excel / VBA 1	17
4.3.1	Description.....	17
4.3.2	Command window	18
4.3.3	Trace Window	19
4.3.4	System requirements	20
4.4	SIM-KNX APPLICATION EXAMPLE with Excel / VBA 2.....	21
4.4.1	Description.....	21
4.5	SIM-KNX APPLICATION EXAMPLE with Configuration Manager.....	23
4.6	Using SIM-KNX with CONRAD C-CONTROL I STATION 2.0.....	24
4.7	Using SIM-KNX and with Doepke fingerprintsensor.....	26
4.8	Using SIM-KNX and Sensirion SHT1X as a climate sensor.....	28
4.9	Using SIM-KNX with Microsoft HyperTerminal.....	29
4.9.1	Preparation	29
4.9.2	Receiving	31
4.9.3	Sending.....	31
4.10	Using SIM-KNX with Sharp XG-PH50 Beamer	32

2 LICENCE

All here shown demo program and database entries are free software and can be downloaded and redistributed. There is no guarantee for the correctness of any information which is given on these pages. We are in no way responsible for any damage resulting from the use or abuse of this information.

NO WARRANTY

BECAUSE THE SOFTWARE IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE SOFTWARE, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE SOFTWARE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

3 SIM-KNX EVALUATION BOARDS

3.1 SIM-KNX EVALUATION BOARD USB

3.1.1 GETTING STARTED WITH THE EVALUATION BOARD USB

CONNECTING USB INTERFACE

The SIM-KNX Evaluation Board USB uses a RS232-USB converter from FTDI. Before you can use the evaluation board you need to install the virtual com port driver for the RS232-USB converter.

You will find this drivers and the manuals on the web site of FTDI:

<http://www.ftdichip.com/Drivers/VCP.htm>

After installing the driver and connection the SIM-KNX evaluation board, you can communicate with SIM-KNX via any terminal program or your own application.

CONNECTING KNX

Connect the Bus line to the KNX-connector.

The best way to test if KNX is connected correct and SIM-KNX is working is to press the program button. The LED should switch, each time you press the program button.

3.1.2 BLOCK-DIAGRAM

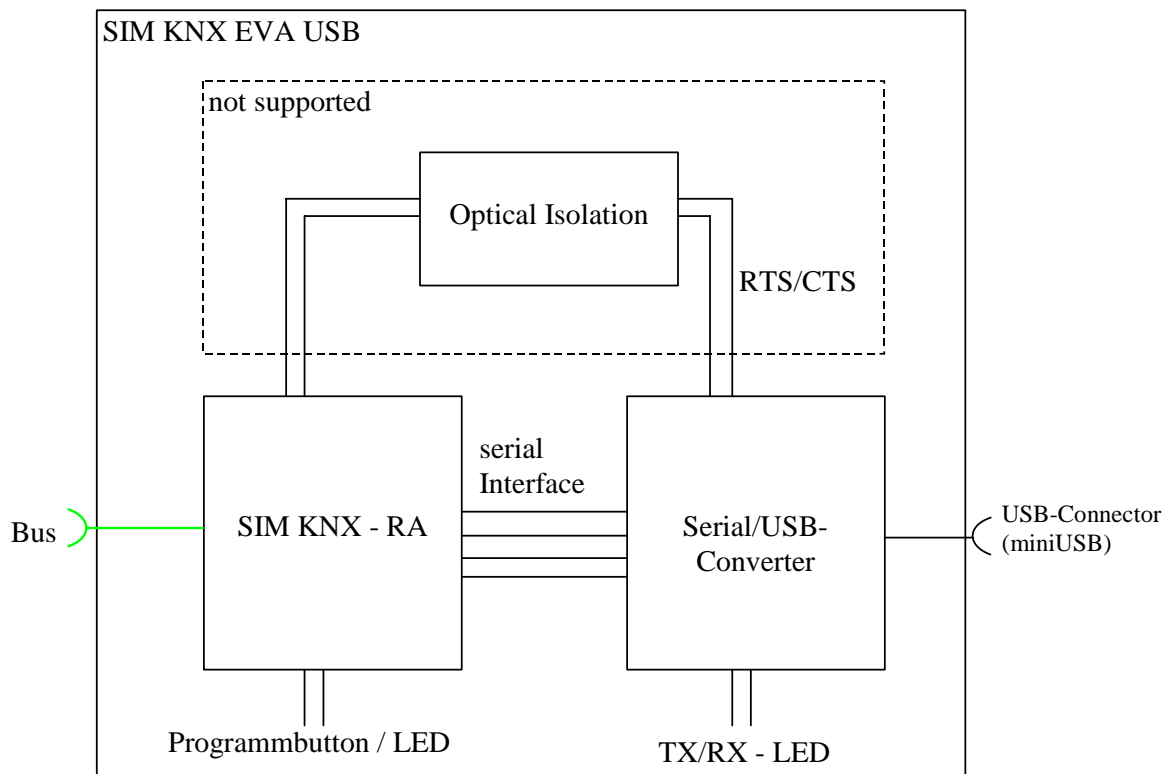


Figure 1: SIM-KNX Evaluation Board USB: Block Diagram

SIM-KNX Evaluation Kit

3.1.3 SCHEMATIC

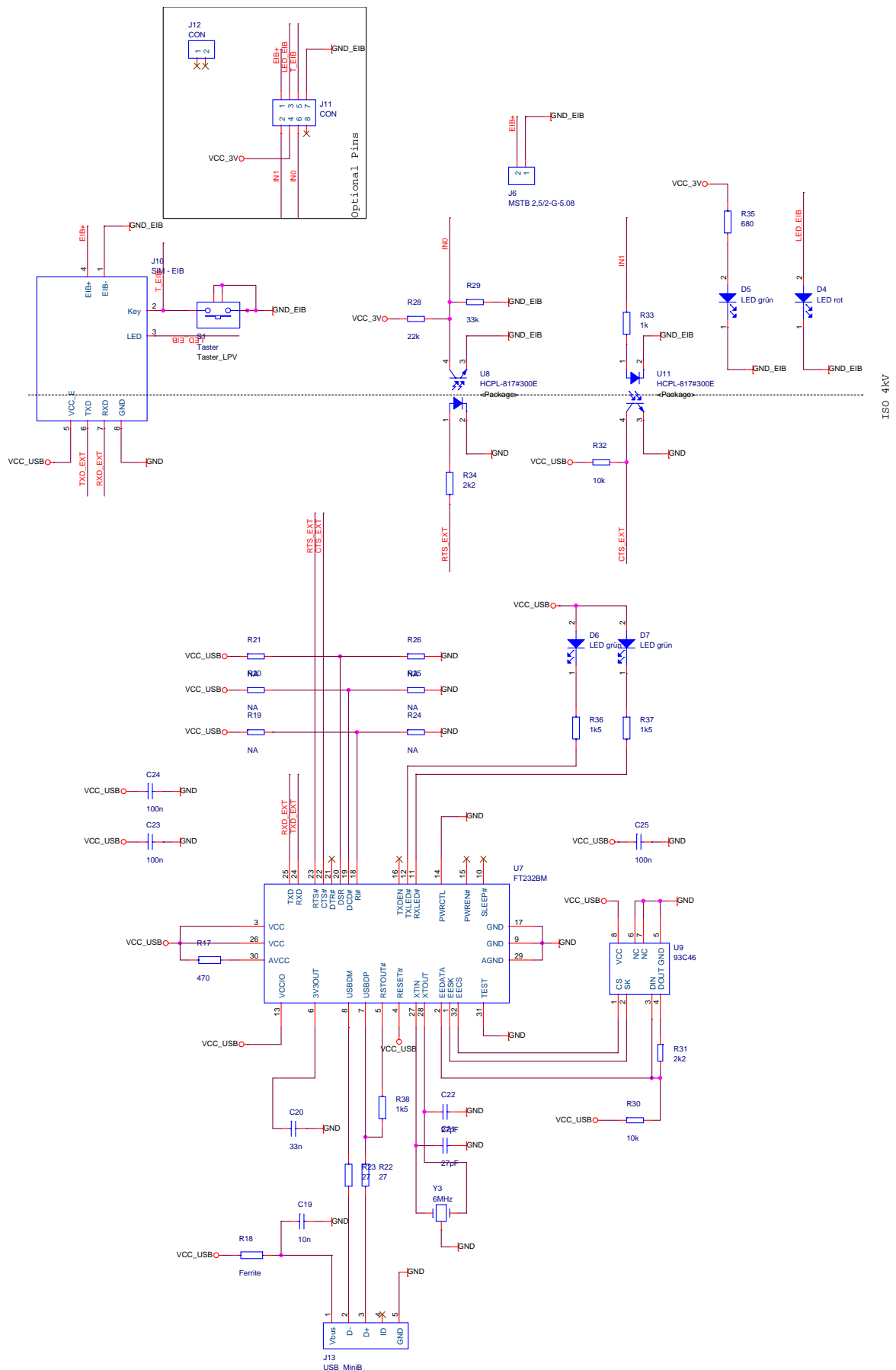


Figure 2: SIM-KNX Evaluation Board USB: Schematic

3.1.4 PLACEMENT

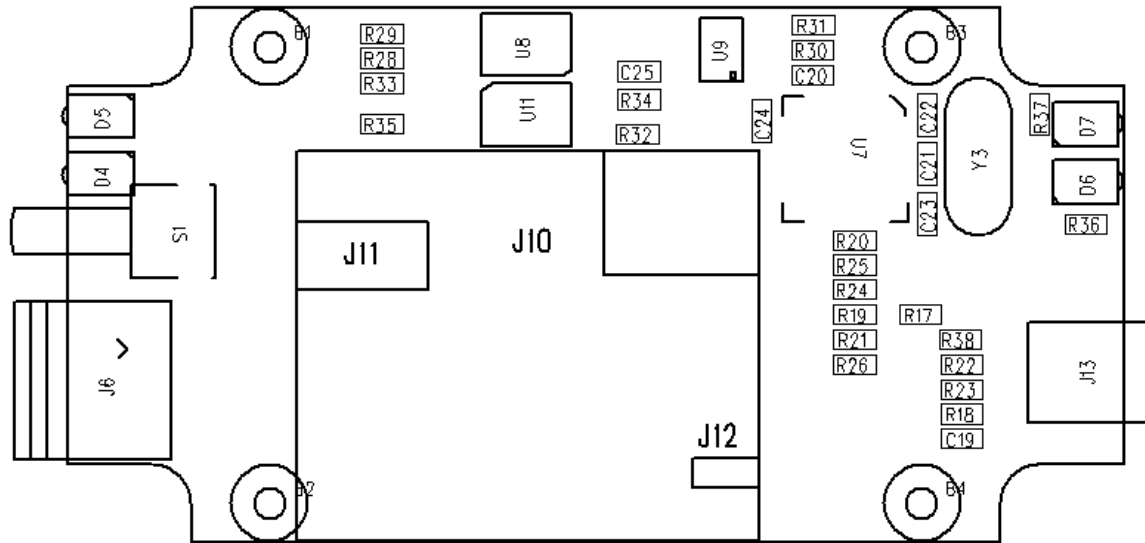


Figure 3: SIM-KNX Evaluation Board USB: Placement

3.1.5 PICTURES

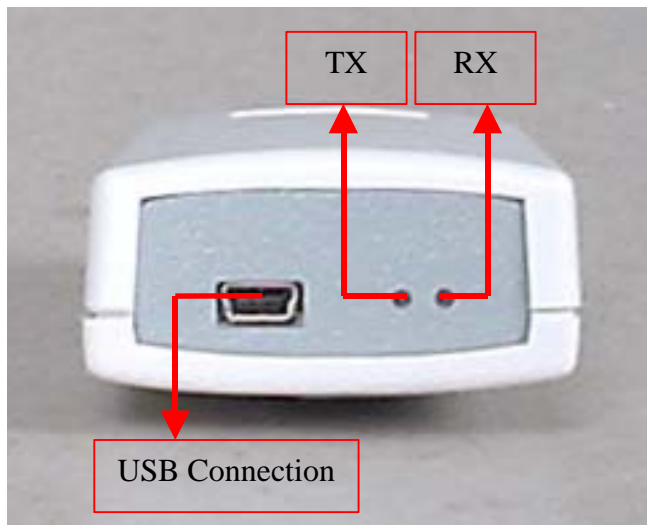


Figure 4: SIM-KNX Evaluation Board USB: Front view

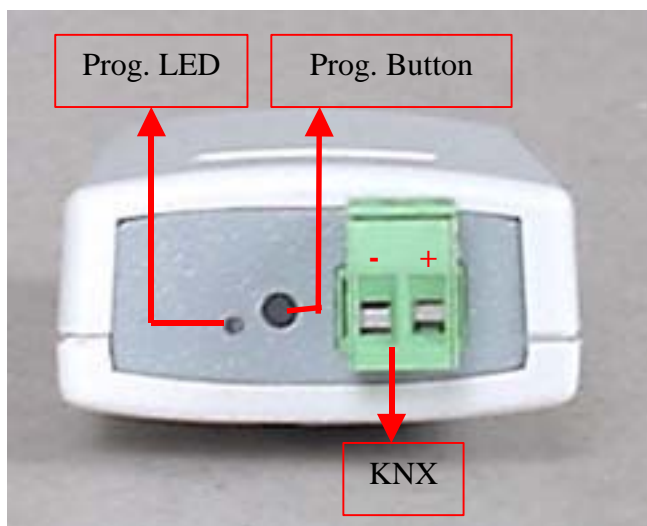


Figure 5: SIM-KNX Evaluation Board USB: Rear view

3.2 SIM-KNX EVALUATION BOARD RS232

3.2.1 GETTING STARTED WITH THE EVALUATION BOARD RS232

CONNECTING RS232 INTERFACE

You can connect the SIM.-KNX Evaluation Board RS232 to the serial interface of your computer with a normal 9-pin RS232 cable.

CONNECTING KNX

Connect the KNX line to the KNX-connector.

The best way to test if KNX is connected correct and SIM-KNX is working is to press the program button. The LED should switch, each time you press the program button.

3.2.2 BLOCK-DIAGRAM

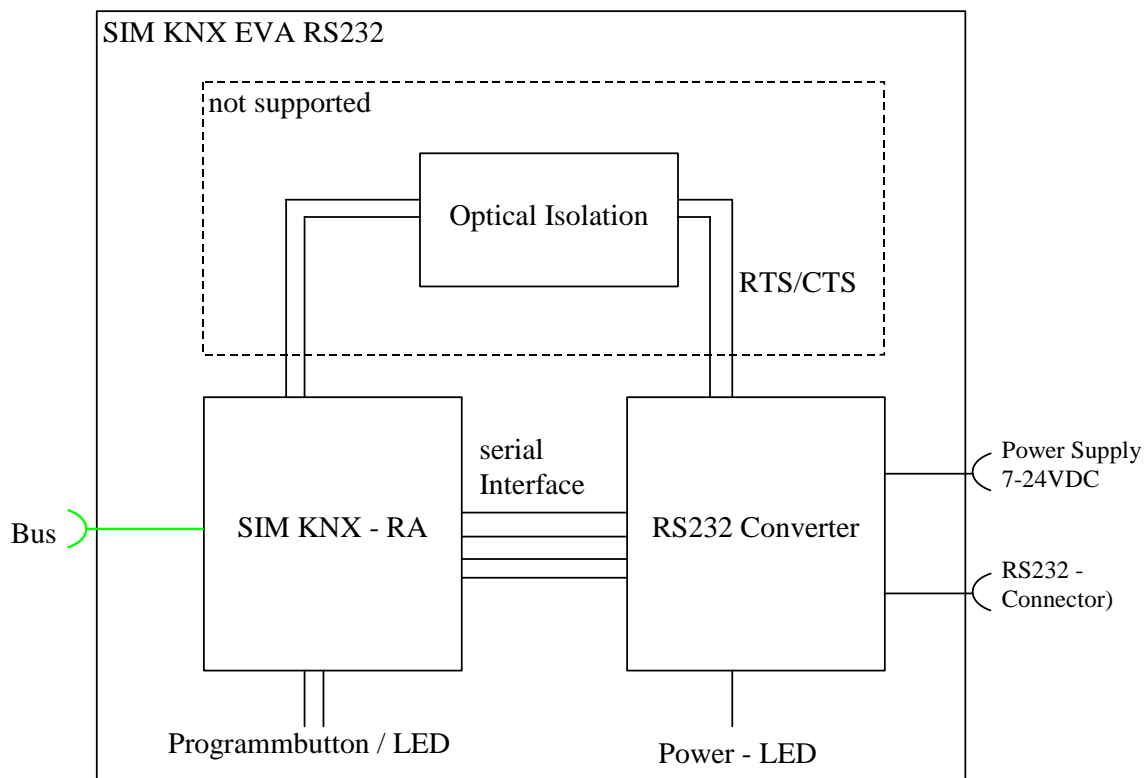


Figure 6: SIM-KNX Evaluation Board RS232: Block Diagram

SIM-KNX Evaluation Kit

3.2.3 SCHEMATIC

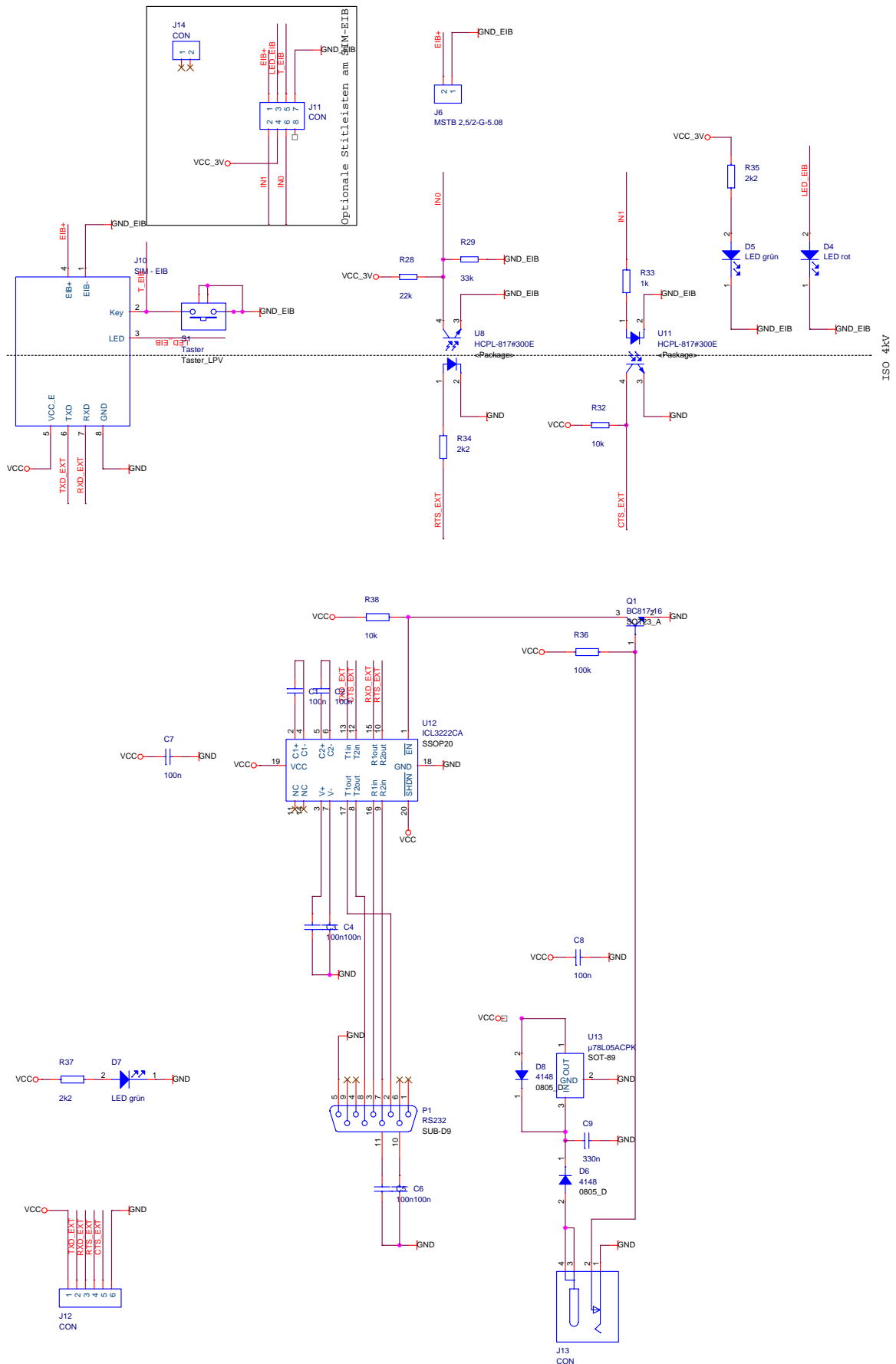


Figure 7: SIM-KNX Evaluation Board RS232: Schematic

3.2.4 PLACEMENT

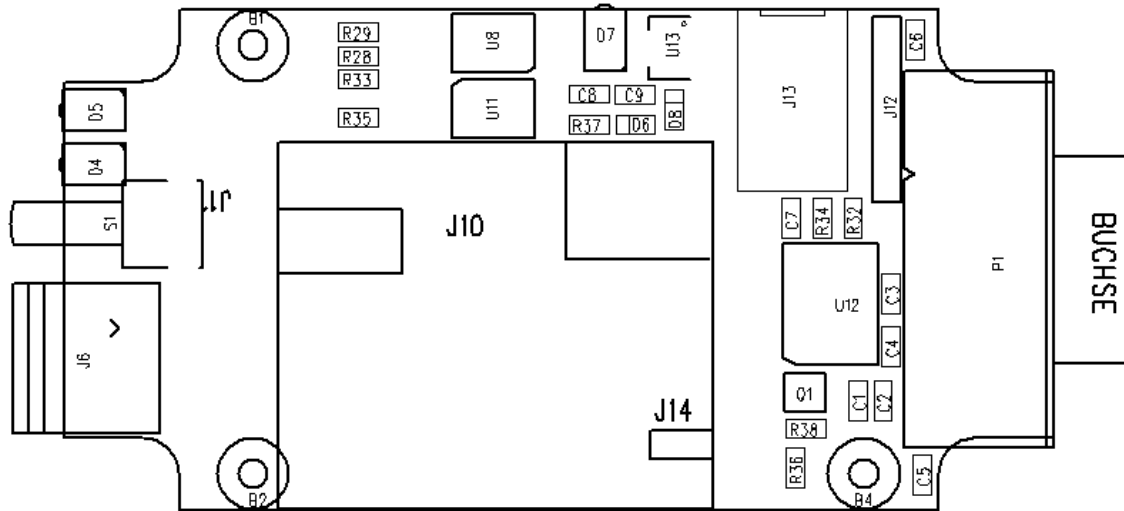


Figure 8: SIM-KNX Evaluation Board RS232: Placement

3.2.5 PICTURES



Figure 9: SIM-KNX Evaluation Board RS232: Front view

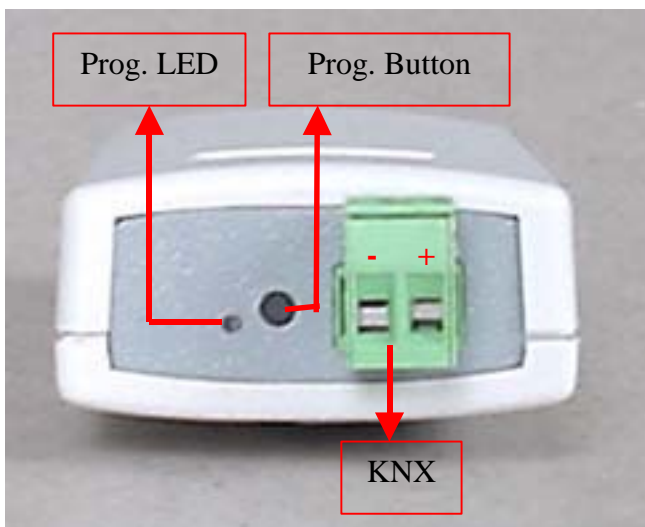


Figure 10: SIM-KNX Evaluation Board RS232: Rear view

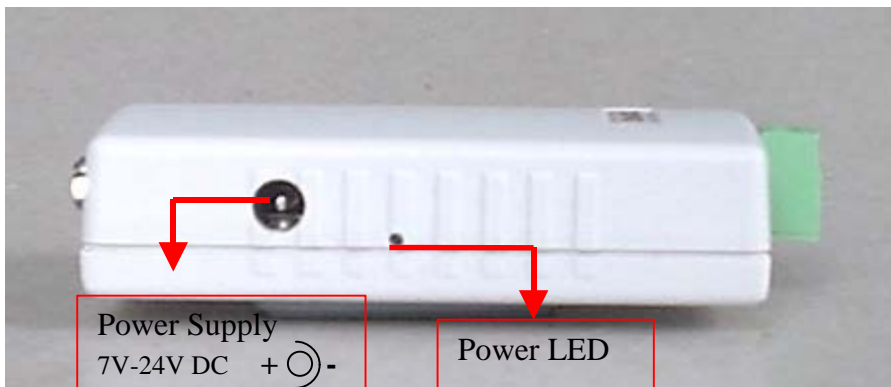


Figure 11: SIM-KNX Evaluation Board RS232: Side view

4 SIM-KNX APPLICATION EXAMPLES

4.1 APPLICATION EXAMPLES OVERVIEW

On the following pages you will find a set of application examples and hints for how to use the SIM-KNX module or the SIM-KNX Eval-board.

The intention is to provide a useful help. We tried to describe and explain in a shortest possible way. If you miss essential clues or you need further assistance please feel free to contact us by email under sim-knx@tapko.de

- 1 SIM-KNX APPLICATION EXAMPLE with VC++
- 2 SIM-KNX APPLICATION EXAMPLE with Excel / VBA 1
- 3 SIM-KNX APPLICATION EXAMPLE with Excel / VBA 2
- 4 SIM-KNX APPLICATION EXAMPLE with Configuration Manager
- 5 Using SIM-KNX with CONRAD C-CONTROL I STATION 2.0
- 6 Using SIM-KNX and with Doepke fingerprintsensor
- 7 Using SIM-KNX and Sensirion SHT1X as a climate sensor
- 8 Using SIM-KNX with Microsoft HyperTerminal
- 9 Using SIM-KNX with Sharp XG-PH50 Beamer

4.2 SIM-KNX APPLICATION EXAMPLE WITH VC++

4.2.1 DESCRIPTION

The intension of this example is to show an application running on a Windows PC. This application is written in Visual C++.

The application program handles 5 communication objects with different communication directions and different usage.

The usage of communication objects and the appropriate communication direction is briefly shown in the following table:

Used Objects:

OBJECT- No.	Comm.. direction	Type	Group address	Usage
0	Sending	DPT1.001	1/1/100	Sending of "on/off" Cyclic sending (10s) is possible
1	Sending	DPT5.001	1/1/101	Sending of 0..100%
2	Sending	DPT9.001	1/1/102	Sending of Temperatures
3	Receiving	DPT1.001	1/1/103	Receiving of "on/off" Receive-timeout (10s) is possible
4	Receiving	DPT5.001	1/1/104	Receiving of 0..100%
5	Receiving	DPT9.001	1/1/105	Receiving of Temperatures

Figure 12: SIM-KNX application example with VC++: Usage of objects

The communication direction mentioned above is oriented from communication objects to KNX bus. That means:

- sending communication object - sending data from the SIM-KNX to the KNX bus.
- Receiving communication object - receiving data from the KNX bus to SIM-KNX.

The application appears with two windows. One window is a console window for tracing the communication between the application and the SIM-KNX. This window we call trace window. The second window is a normal application window that we call command window. With the command window we have the possibility to send value changes from the application to the communication objects in the SIM-KNX. Depending of the configuration of the objects the values will be sent to the KNX bus immediately or on explicit request with a Send button that sends an appropriate command to the SIM-KNX.

4.2.2 COMMAND WINDOW

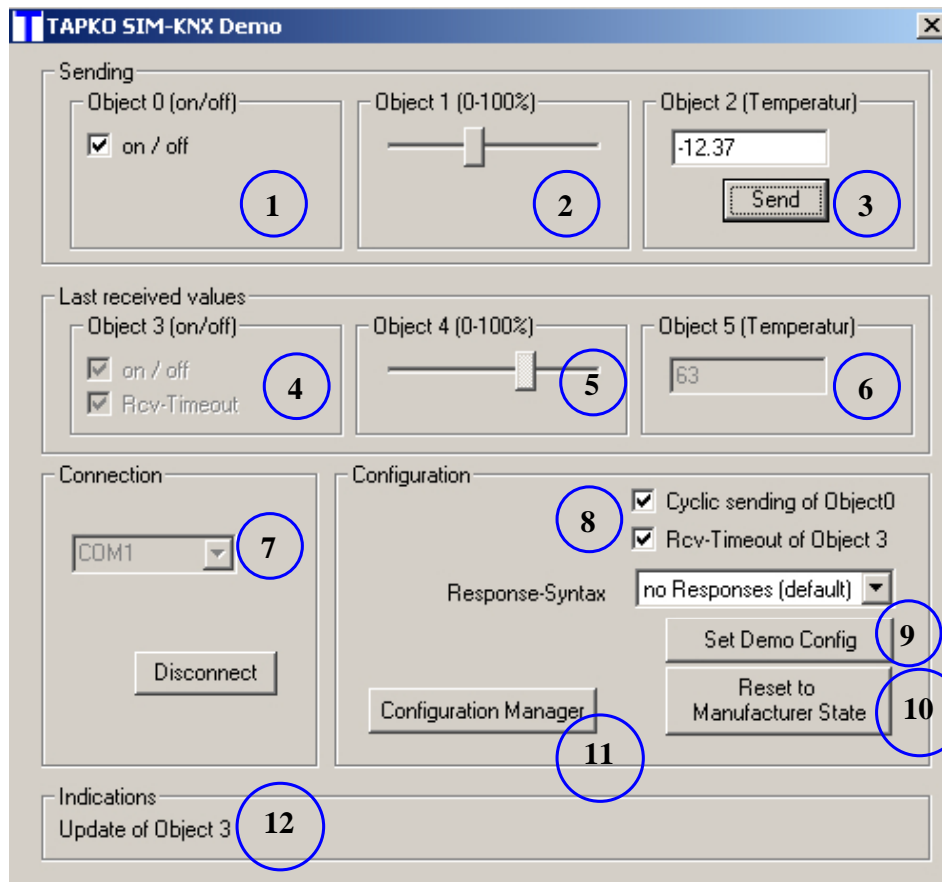


Figure 13: SIM-KNX application example with VC++: Command window

1. send on / off on object 0 (mark checked = on / mark uncheck = off)
2. send value on object 1 – regulator can be moved by mouse click or by keyboard
3. send value on object 2 – type the value (maximum 2 decimal places) and press send
4. indication of value on object 3 – value (DPT1), receive-timeout
5. indication of value on object 4 – value 0-100% (DPT5)
6. indication of value on object 5 – value of temperature (DPT9)
7. chosen COM-Port for connection
 - button “Connect” is visible when disconnected
 - button “Disconnect” is visible when connected
8. basic settings for demo config
 - cycling sending of object 0: on / off
 - Rcv-Timeout of Object 3
 - response syntax
 - no response
 - Echo Command
 - Echo “OK”
9. press button to program demo configuration with settings of (8)
10. press this button for reset to manufacturer state – all made settings get lost
11. press this button to open SIM-KNX APPLICATION EXAMPLE with Configuration Manager
12. shows indications

4.2.3 TRACE WINDOW

```

<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
Connection Closed
Connection succesful
>>
<< oui $03 $80 1
Connection Closed
Connection succesful
>>
<< oui $03 $80 1
>> ovs (2) -12.37
>> ovs(1) 85
>> ovs(1) 44
>> ovs(0) 0
>> ovs(0) 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1

```

Figure 14: SIM-KNX application example with VC++: Trace window

In the trace window every action and command sent and received by SIM-KNX is shown.

“<<” SIM-KNX sends to the Application via serial interface.

“>>” Application sends via serial interface commands to SIM-KNX

Example 1 (blue):

>> **ovs (2) -12.37**

SIM-KNX receives command “set object value” (ovs) on object 2 with value -12.37

Example 2 (green)

<< **oui \$03 \$80 1**

SIM-KNX sends global update flag (oui) on object (\$03) with RAM-flags of the communication object(\$80) and actual value of the communication object (1)

4.2.4 SYSTEM REQUIREMENTS

- Microsoft Windows OS (tested on MS Windows 2000 SP4)
- either a free USB or Serial Com Port
- there is no installation necessary – starting directly from CD is possible

4.3 SIM-KNX APPLICATION EXAMPLE WITH EXCEL / VBA 1

4.3.1 DESCRIPTION

The intension of this example is to show an application written in Visual Basic for Applications embedded in Excel that communicates with the KNX bus.

The application program handles 5 communication objects with different communication directions and different usage.

The usage of communication objects is the same like in the previous example.

Used Objects:

OBJECT- No.	Comm.. direction	Type	Group address	Usage
0	Sending	DPT1.001	1/1/100	Sending of "on/off" Cyclic sending (10s) is possible
1	Sending	DPT5.001	1/1/101	Sending of 0..100%
2	Sending	DPT9.001	1/1/102	Sending of Temperatures
3	Receiving	DPT1.001	1/1/103	Receiving of "on/off" Receive-timeout (10s) is possible
4	Receiving	DPT5.001	1/1/104	Receiving of 0..100%
5	Receiving	DPT9.001	1/1/105	Receiving of Temperatures

Figure 15: SIM-KNX application example with Excel /VBA: Usage of objects

The application appears in one spread sheet. On the left side we have an area that we call command area. On the right side you see a console window for tracing the communication between the application and the SIM-KNX. This area we call trace area or simply trace. In the command area we have the possibility to send value changes from the application to the communication objects in the SIM-KNX. Depending of the configuration of the objects the values will be sent to the KNX bus immediately or on explicit request with a Send button that sends an appropriate command to the SIM-KNX.

The sending objects are in the upper region(1, 2 and 3 on the next picture), the receiving objects are in the middle region (4, 5 and 6 on the next picture)

The connection port selection and activation of connection is done in the mid lower section.

In the lower region are some possibility to set certain object behavior, to configure the connected SIM-KNX according the object usage in this application and to clear all configuration data in the SIM-KNX. There is also a button to call the Configuration manager that is described as a separate application in the chapter SIM-KNX APPLICATION EXAMPLE with Configuration Manager. For a successful call the SIMKNX_demo.exe has to be in the same directory like the SIMKNX_demo.xls

4.3.2 COMMAND WINDOW

Sending Objects		
Object 0 (on/off)	Object 1 (0..100%)	Object 2 (Temperature)
<input checked="" type="checkbox"/> on/off	<input type="text" value=""/>	20,50 °C
1	2	Send 3

Last received values		
Object 3 (on/off)	Object 4 (0..100%)	Object 5 (Temperature)
<input checked="" type="checkbox"/> on / off <input type="checkbox"/> Rcv-Timeout	<input type="text" value=""/>	63,00 °C
4	5	6
Update of Object 5 12		

Connection	
COM-Port: <input type="text" value="COM1"/>	7
	Disconnect

Configuration	
<input checked="" type="checkbox"/> Rcv-Timeout of Object 3	<input type="checkbox"/> Cyclic Sending of Object 0
8	Response-Syntax: <input type="text" value="no Responses (default)"/>
	Set Demo Config 9
Reset to Manufacturer State 10	

Figure 16 SIM-KNX application example with Excel /VBA 1 : Command area

1. send on / off on object 0 (mark checked = on / mark uncheck = off)
2. send value on object 1 – regulator can be moved by mouse click or by keyboard
3. send value on object 2 – type the value (maximum 2 decimal places) and press send
4. indication of value on object 3 – value (DPT1), receive-timeout
5. indication of value on object 4 – value 0-100% (DPT5)
6. indication of value on object 5 – value of temperature (DPT9)
7. chosen COM-Port for connection
 - button “Connect” is visible when disconnected
 - button “Disconnect” is visible when connected
8. basic settings for demo config
 - cycling sending of object 0: on / off
 - Rcv-Timeout of Object 3
 - response syntax
 - no response

- Echo Command
 - Echo "OK"
9. press button to program demo configuration with settings of (8)
 10. press this button for reset to manufacturer state – all made settings get lost
 11. shows indications

4.3.3 TRACE WINDOW

```

Connection succesful
<< oui $03 $80 1
>>
>> ovs(1) 1
>> ovs(1) 48
<< oui $03 $80 1
>> ovs(2) 20.5
<< oui $03 $80 1
Disconnected
Connection succesful
>>
<< oui $03 $80 1
<< oui $03 $18 0
<< oui $03 $80 0
<< oui $05 $08 63
<< oui $03 $80 0
<< oui $04 $18 100
<< oui $04 $18 66
<< oui $03 $80 0
<< oui $03 $80 0
<< oui $03 $18 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1
<< oui $03 $80 1

```

Figure 17: SIM-KNX Exel VBA 1 Demo Software: Trace window

In the trace window every action and command send and received by SIM-KNX is shown.

“>>” SIM-KNX receives from KNX

“<<” SIM-KNX sends on KNX

Example 1 (blue):

>> ovs (2) 20.5

SIM-KNX receives command “set object value” on object 2 with value 20.5

Example 2 (green)

<< oui \$03 \$80 1

SIM-KNX sends global update flag on object (\$03) with RAM-flags of the communication object(\$80) and actual value of the communication object (1)

4.3.4 SYSTEM REQUIREMENTS

- Microsoft Windows OS (tested on MS Windows 2000 SP4)
- either a free USB or Serial Com Port
- there is no installation necessary – starting directly from CD is possible

4.4 SIM-KNX APPLICATION EXAMPLE WITH EXCEL / VBA 2

4.4.1 DESCRIPTION

The intension of this example is to show how easy a private visualization can be built with an application written in Visual Basic for Applications embedded in Exel that communicates with the KNX bus.

In this application the SIM-KNX is used in transparent mode. In this mode no communication objects are used. All group oriented messages are passed through. The application program handles 8 data points with different communication directions and different usage.

The usage of data points is.

Used datapoints identified by their group address:

Group address	Comm.. direction	Name	Type	Usage
1/2/3	Sending	Output 1	DPT1.001	Sending of "on/off"
1/2/3	Sending	Output 2	DPT1.001	Sending of "on/off"
1/1/1	Receiving	Temperature	DPT5.001	Receiving of Temperatures in °C
1/1/2	Sending	Rel. humidity	DPT9.001	Receiving of rel. humidity in 0..100%
1/2/1	Receiving	ON/OFF	DPT1.001	Receiving of "on/off"
1/2/2	Receiving	ON/OFF	DPT5.001	Receiving of "on/off"
1/3/1	Receiving		DPT5.001	Receiving of "on"
1/3/2	Receiving		DPT5.001	Receiving of "on"

Figure 18: SIM-KNX application example with Excel /VBA 2: Usage of objects

The application appears in one spread sheet. On the left side we have our visualization elements area.called command area. On the right side you see a console window for tracing the communication between the application and the SIM-KNX. This area we call trace area or simply trace.

SIM-KNX Evaluation Kit

In the command area we have the possibility to send values from the application to the KNX by clicking on the Output 1 or Output 2 buttons. The values will be sent to the KNX bus immediately.



Figure 19: SIM-KNX application example with Excel /VBA 2: Command and trace area

The connection port selection and activation of connection is done on page COM.

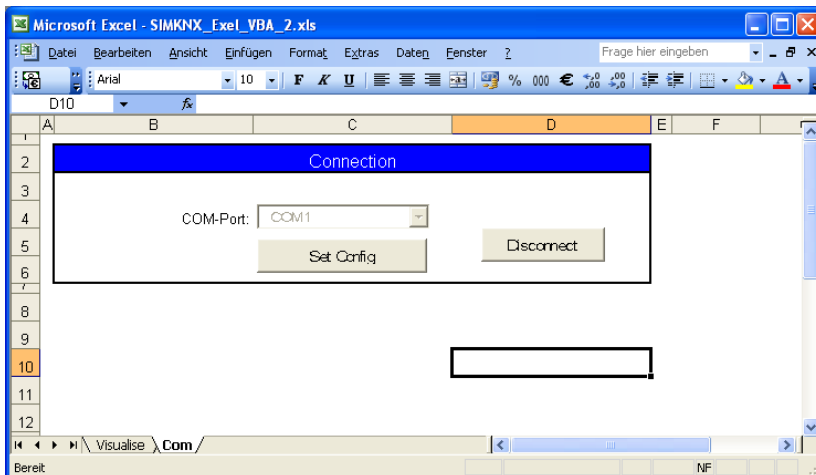


Figure 20: SIM-KNX application example with Excel /VBA 2: Connecting to serial and configuring device

There is also a button for configuring the SIM-KNX for this application. Actually it is switching into transparent mode. You may observe this command on the Visualise page in the trace – dts \$17. For a description of transparent mode and details of the dts command referre to the SIM-KNX documentation.

4.5 SIM-KNX APPLICATION EXAMPLE WITH CONFIGURATION MANAGER

You can start this configuration manager either with the buttons in the demosoftware or manually by starting the SIMKNX_demo.exe with parameter “ConfigManager”, for example “C:\SIMKNX_demo.exe /ConfigManager”

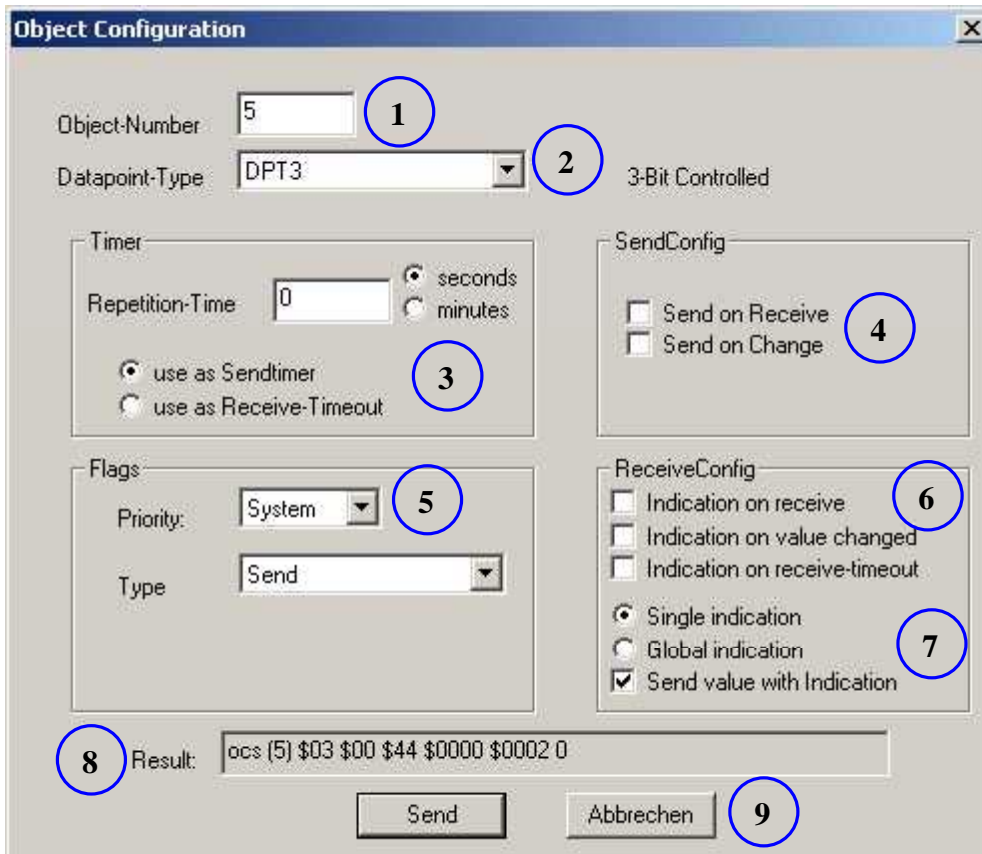


Figure 21: SIM-KNX Demo Software: Configuration manager

1. set object number to configure
2. set DPT type – explanation is shown on the right
3. timer
 - set repetition time and unit (seconds or minutes)
 - use timer as sendtimer or receive-timeout
4. configuration when the value is send (receive and/or change)
5. set priority flag (low, high, alarm, system)
6. set object flags
7. set receive configuration
 - set configuration when the indications will be sent
 - set usage of the indication
 - set if value is send with indication (marked check = yes)
8. see the telegram, send on the KNX
9. press “Send” to send the command or “Abbrechen” to abort

4.6 USING SIM-KNX WITH CONRAD C-CONTROL I STATION 2.0

This demonstration program shows how a SIM-KNX connected to the RS232 port of a controller, can be driven by basic software to obtain a KNX enabled controller for easy programmable automation tasks.

This SIM-KNX demo program for CONRAD C-CONTROL I STATION 2.0 is written in BASIC++2006 from Stefan Tappertzhofen.

This basic compiler can be obtained for free from <http://www.tappertzhofen.eu/bpp2>.

The SIM-KNX eval board is connected with a standard male to female 9 pin sub D cable to the CONRAD C-CONTROL I STATION.

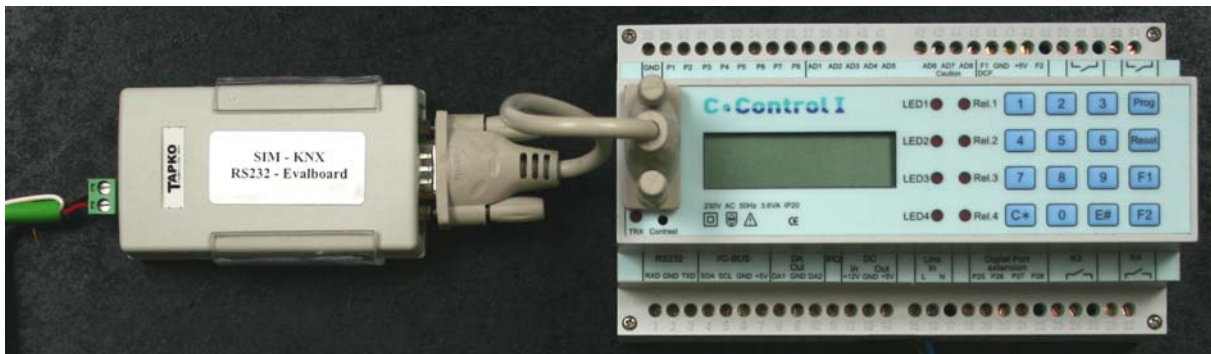


Figure 22: SIM-KNX Demo Software: Configuration manager

The software performs actions when a key is pressed or if an appropriate message is received. Four group addresses and 4 objects are used.

Following actions are implemented:

on press of key:

"F2" a reset configuration of SIM-KNX to manufacturer state is performed

"F1" the first 4 communication objects are configure for usage in the SIM-KNX.

For first 4 group addresses are added and every communication object is connect to one group address. The object 0 and 1 are configured for sending. The object 2 and 3 are configured for receiving and transmitting individual indications with values on the serial interface.

The group addresses added are in 3 level ETS notation: 1/1/1, 1/1/2, 1/1/3, 1/1/4

"1" The object 0 is set to the value **0** and this is immediately send to the bus.

The resulting message on the bus looks like this:

```
BC XXXX 0901 81 00 80 YY 1)
```

"2" The object 0 is set to the value **1** and this is immediately send to the bus.

The resulting message on the bus looks like this:

BC XXXX 0901 81 00 80 YY

"3" The object 1 is set to the value **0** and this is immediately send to the bus.
The resulting message on the bus looks like this:

BC XXXX 0901 81 00 80 YY

"4" The object 1 is set to the value **1** and this is immediately send to the bus.
The resulting message on the bus looks like this:

BC XXXX 0901 81 00 80 YY

¹⁾ Where XXXX is the own device address and YY the checksum of the frame.

4.7 USING SIM-KNX AND WITH DOEPKE FINGERPRINTSENSOR

This demop shows the integration of Doepke Finger Identification System with almost no effort with SIM-KNX by using the command translation ability.

The Doepke FIS-Controller is configured to send the 11 digit ID number of a recognised person on the serial interface (e.g.: 00099801997). This ID string is translated in the receive buffer of SIM-KNX to a object value set command. This way a send communication object is performed.

You see below how the two portions of the incoming string are identified and translated.

000998 is replaced by ovs (
997 is replaced by)1

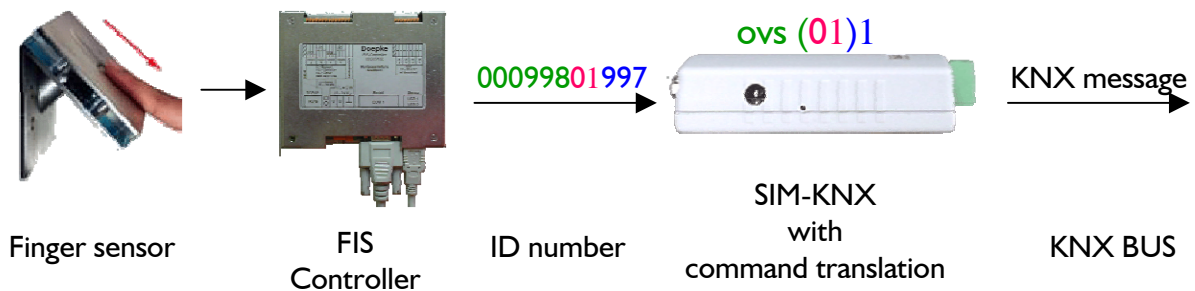


Figure 23: SIM-KNX and with Doepke fingerprintsensor: System setup

The command translation function is configured with the provided ETS project. A real installation with a SIM KNX and a KNX power supply looks like this:



Figure 24: SIM-KNX and with Doepke fingerprintsensor: Demo board

For information about the Doepke FIS-Controller and Doepke please contact:
gerold.rooofs@doepke.de

4.8 USING SIM-KNX AND SENSIRION SHT1X AS A CLIMATE SENSOR

In this demo we provide a complete schematic and the source code including the resulting binary file to build a complete climate sensor. A small MSP430 is used to convert the IIC bus needed to address the sensor signals to the asynchrony serial protocol for the SIM-KNX.

The original design is done on a breed board.

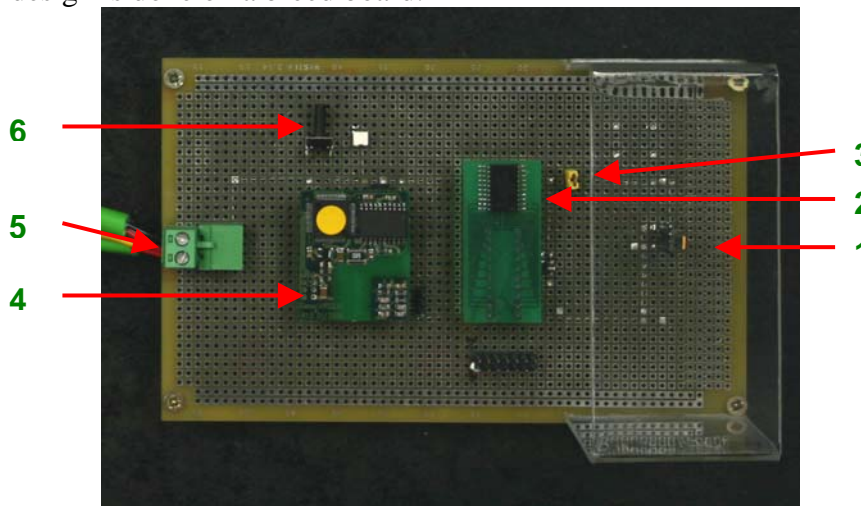


Figure 25: SIM-KNX and Sensirion SHT1X as a climate sensor: breed board setup

On the right side under the transparent shield is the sensor SHT1X (1). This sensor is capable of detecting temperature and relative humidity. This information is available through a serial protocol similar to the I2C protocol.

In the middle you see a protocol changer based on a small MSP430F1121 (2). It's task is to translate the I2C protocol on one side to the ASCII on the SIM-KNX side. Among others it controls also the send conditions and the refresh rate of the measuring cycle. This cycle consists of starting a measuring of relative humidity, reading this result, converting it to an ASCII string, sending it to the SIM-KNX, starting a temperature measurement, reading this result, converting it to an ASCII string and sending it to the SIM-KNX. On the end of this loop a delay of 1 second is added. The values transferred to the SIM-KNX are sent immediately after reception. There is no send control based on value provided in this example. In a product a small hysteresis would be better to suppress unnecessary sending. A jumper (3) is used to enable sending the one time necessary SIM-KNX configuration strings on startup.

On the picture above you see on the left side the SIM-KNX module(4), the programming key and LED(5) and the KNX bus connector(6).

Following documents are provided:

SIMKNX KLIMASENSOR.DSN
 clima sensor.pdf
 software for U3 MSP430F1121AIDW

Orcad 10 schematic design
 same as above but as pdf
 project for IAR Embedded Workbench

4.9 USING SIM-KNX WITH MICROSOFT HYPERTERMINAL

for using the Microsoft HyperTerminal with SIM-KNX you first have to make certain settings in HyperTerminal to communicate correctly with SIM-KNX. Please follow this step by step instruction.

We use a German version of HyperTerminal for illustration purposes here.

4.9.1 PREPARATION

- 1) Connect the SIM-KNX to the bus. Check if the module is powered and running by pressing on the program button. The LED should go on. By repeated press the LED should go off. This behavior is an important check if the SIM-KNX is running.
- 2) Connect the
 - RS232 connector with the provided RS232 serial connection cable 1:1 9pin D-sub plug / socket cable to a free com port. On PCs especially notebooks the RS232 Levels may be too low to power the galvanic isolated part of SIM-KNX. When you connect the RS232 connector to the PC the power LED on the side beside the additional power connector should go on For situations where the state lines do not provide enough or any power, an additional power supply is added to the Evaluation kit.
 - USB connector with the provided USB cable plug A / mini plug B to a free USB connector on your PC. If this is the first time to connect a SIM-KNX USB version, your PC will recognize a new device and ask for drivers. Please download from <http://www.ftdichip.com/Drivers/VCP.htm> the appropriate driver for your operating system and install these.
- 3) Start HyperTerminal
- 4) Enter the name SIMKNX as a name for this settings
- 5) Select the appropriate COM port where you have connected the SIM-KNX COM1 or COM2 for RS232 or COMx for USB. (x can be 3, 4, 5,...)
- 6) Set 9600 bits per second, 8 data bits, no parity, 1 stop bit, no flow control

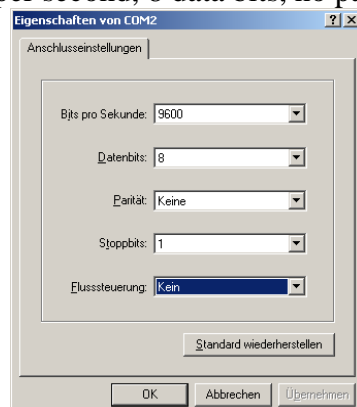


Figure 26: using SIM-KNX with Microsoft hyperterminal: communication setup

- 7) Open File – Properties (Datei – Eigenschaften) and select the tab settings (Einstellungen).



Figure 27: using SIM-KNX with Microsoft hyperterminal: where to find ASCII configuration

There please click on ASCII-Configuration... (ASCII-Konfiguration...)

- 8) Make following settings:

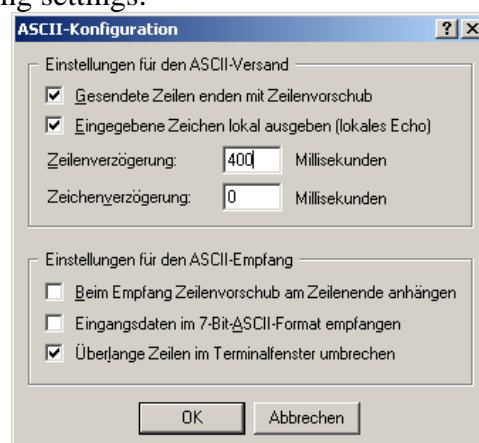


Figure 28: using SIM-KNX with Microsoft hyperterminal: ASCII configuration

Normally it is not necessary to introduce a delay between lines except when configuration data in flash is changed. There we have to give the SIM-KNX the opportunity to finish writing into flash memory. Since our examples change non volatile flash data this is a way to ensure nothing goes wrong.

- 9) Test the communication with the SIM-KNX.

Please type “dvg↵”.

The answer should look like this:

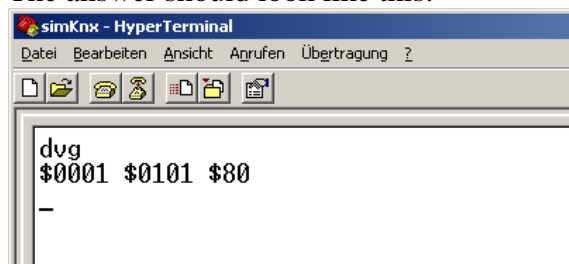


Figure 29: using SIM-KNX with Microsoft hyperterminal: testing communication with SIM-KNX

(see command `dvg` for details about the response)

If you see this or a similar response the communication between HyperTerminal and SIM-KNX is OK.

10) Save the connection settings for later use by file – save (Datei – Speichern)

4.9.2 RECEIVING

As an example, for receiving the command ‘switching light’ from the KNX bus, you have to introduce the group address of this light to the SIM-KNX.

- 1) find out what group address is used for this switching by having a look to the ETS project. Let us assume it is the address in hex \$0901 or 1/1/1 (this is the same address only written in 3 level ETS notation 1/1/1. Use online converter http://www.tapko.de/tapko_main/en/tools/eib_converter.shtml to convert formats)
- 2) add this group address to the address table in the SIM-KNX
- 3) configure an object for receiving
- 4) type on the HyperTerminal:

gci

oga(0) 1/1/1

ocs(0) 1 0 \$17 \$0000 \$0802 0

Explanation of the lines above:

Gci

erase all configuration data stored previously

oga(0) 1/1/1

add address 1/1/1 or written hex \$0901 to address table and connect this to object number 0

ocs(0) 1 0 \$17 \$0000 \$0802 0

configure object 0 for receiving

after this, when you send a command switch on or switch off on the bus you will see:

oui \$00 \$08 1 or

oui \$00 \$08 0

on the terminal output.

4.9.3 SENDING

An other way to send commands is to upload a text file where you have stored previously your configuration, your scene or similar sets of commands.

For demonstration, please upload the test.txt file under transfer-send text file.

This configures the SIM-KNX for 2 object sending and 2 objects receiving, sends 4 telegrams and asks the version of the firmware.

4.10 USING SIM-KNX WITH SHARP XG-PH50 BEAMER

The intension of this example is to show command translation.

In this example we control a beamer SHARP XG-PH50 without a remote control. The possibility of the beamer to receive and send ASCII commands via a serial interface is essential for this application.

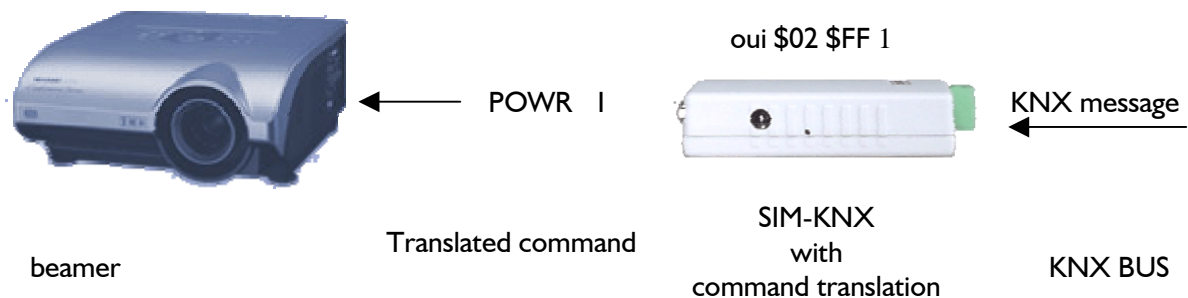
The target device, the beamer, has a fixed command set that can not be altered. Therefore we need the ability of the SIM-KNX to translate commands

Here a fixed string of 8 characters, a command (4 character) with a parameter (4 character), is sent to the beamer.

After this command is processed, a response OK or ERR is returned to the SIM-KNX.

Command samples:

POWR 0, POWR 1, MUTE 1, MUTE 0, ...



The command translation is configured with a database entry.

On reception of object 0 to 10 the predefined string like 'POWR 1' are sent.

The response string OK or ERR is associated to the status object, so the success of the last command could be retrieved or send automatically.

If you need more details about this application please contact us by email under sim-knx@tapko.de