KIMaip

INTERFACE DESCRIPTION

Author: thiere_m

Version 1.5 Last Modification: 24 October 2012

0 Version History

Version	Date	Comments
1.1	April 2011	First official issue
1.2	June 2011	1. New: Isolation of KIMaip, see Chapter 4.1.1. 2. Telegram direction new mentioned, see Chapter 7.1.2. 3. New examples for Value send and Value response send, see Chapter 7.3.1.1 and 7.3.1.3. 4. Note for Reset to Default, see Chapter 7.8.1.3. 5. System parameter ID 0011h: description extended, see Chapter 7.8.3. 6. I²C address description, see Chapter 7.8.3.1. 7. New: ETS Group Objects and Parameters, see Chap. 8. 8. New: Mechanical specifications, see Chapter 10.
1.3	22 June 2011	Text correction Chapter 3.
1.4	January 2012	1. Added: Chapter 4.2.2 “High/Low Voltages I/O-Pins”. Chapter 4.2.3 “Series Resistors I/O-Pins”. 2. Mechanical Specifications completed.
1.5	October 2012	1. Data direction Bit R/W revised. 2. New: list of Properties, see Chapter 9.

1 Contents

2 Commands

NOTE
Please note that all the commands not mentioned in this document are not used yet and are still reserved for later development of our software.

2.1 Overview

2.1.1 Messages to Module

CMD/Hex	Message	Page
10	ValueSend	20
11	ValueReadSend	20
12	ValueResponseSend	20
30	ObjectDescriptionWrite	22
31	ObjectDescriptionRead	22
34	GroupAddressAddSending	25
35	GroupAddressAdd	25
36	GroupAddressDelete	26
37	GroupAddressDeleteAll	26
38	GroupAddressRead	27
40	ParameterWrite	28
41	ParameterRead	28
50	InterfaceObjectWrite	29
51	InterfaceObjectRead	29
60	SystemParameterWrite	30
61	SystemParameterRead	30
70	TransparentGroupWrite	33
71	TransparentGroupRead	33
72	TransparentGroupResponseWrite	33
74	TransparentGroupWriteRcvd	34
75	TransparentGroupReadRcvd	34
76	TransparentGroupResponseRcvd	34
78	TransparentGroupWriteConf	34
79	TransparentGroupReadConf	34
7A	TransparentGroupResponseConf	35
7C	TransparentGroupWriteNegConf	35
7D	TransparentGroupReadNegConf	35
7E	TransparentGroupResponseNegConf	35
D0	ResetDeviceToDefault	30

2.1.2 Messages from Module

CMD/Hex	Message	Page
20	ValueReceive	21
21	ValueReadReceive	21
32	ObjectDescriptionResponse	22
39	GroupAddressResponse	27
42	ParameterResponse	28
52	InterfaceObjectResponse	29
62	SystemParameterResponse	31
80	SystemEvent	36
E0	ErrorHandling	38

3 GENERAL

The KIMaip is an easy to use interface to the KNX. The access to the KNX is realized as a simple I²C-Protocoll.
The KIMaip consists of a micro controller and all necessary components to connect to the KNX and contains the complete certificated KNX communication system. This module is designed to connect controller or other devices to the KNX. Due to its design it is also applicable for small and mid range quantities.

The KIMaip can be used in two different modes:

- Raw data mode
  In this mode the KIMaip transfers the data transparent from the I²C-Interface to KNX and vice versa. Here the transmission of the data is triggered completely by the I²C-Interface. The configuration of this mode is normally done via ETS or I²C-Interface – no external tool is required.

- Transparent mode
  Transparent mode enables receiving from and sending to all group addresses without any filtering. There are no data type limitations for sending.
  This mode is suitable for tracing the group oriented message traffic.

3.1 General features

Application Interface

- Inter-Integrated Circuit I²C
- I²C-Slave with additional wire for indicating new data
- Access to KNX group communication objects (runtime communication)
- Access to KNX interface objects (configuration)
- Configurable indication when group communication value was received
- Supply of the Application hardware till up to 30mA @ 12V

KNX features

- Device model 0705
- Mechanism for configuration via KNX integrated
- Number of group objects: 255

KNX group communication objects

- Transparent transmission of the group communication object data
- Telegram generation triggered via I²C
- Configuration via ETS or I²C

Transparent mode

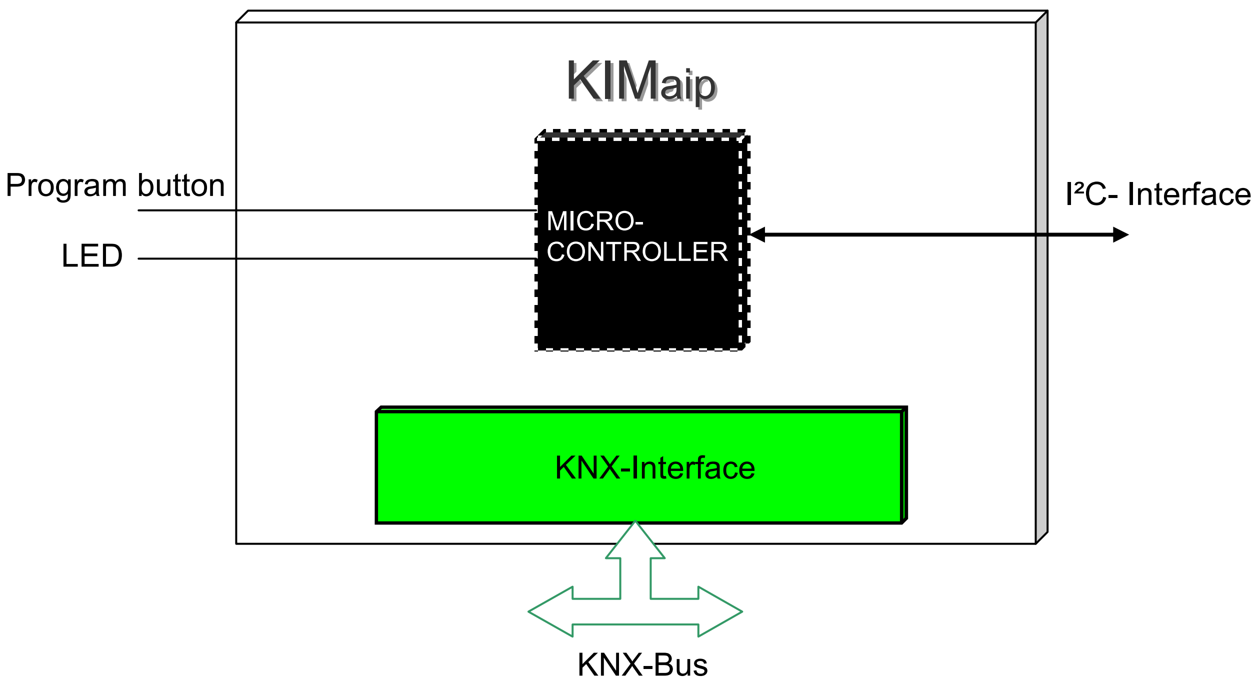
- receiving of all group oriented messages
- no filtering of messages
- sending on all group addresses
- no restriction of a specific data type to a specific group address for sending
- not needed protocol oriented information is removed
- also suitable for tracing and logging of messages
- maximum data length of 96 bytes

3.1.1 Resources

Number of group addresses: 254
Number of associations: 255
Number of Communication objects: 255 (0-254 with max. size of 14 bytes)
Number of application parameter: 2048 byte

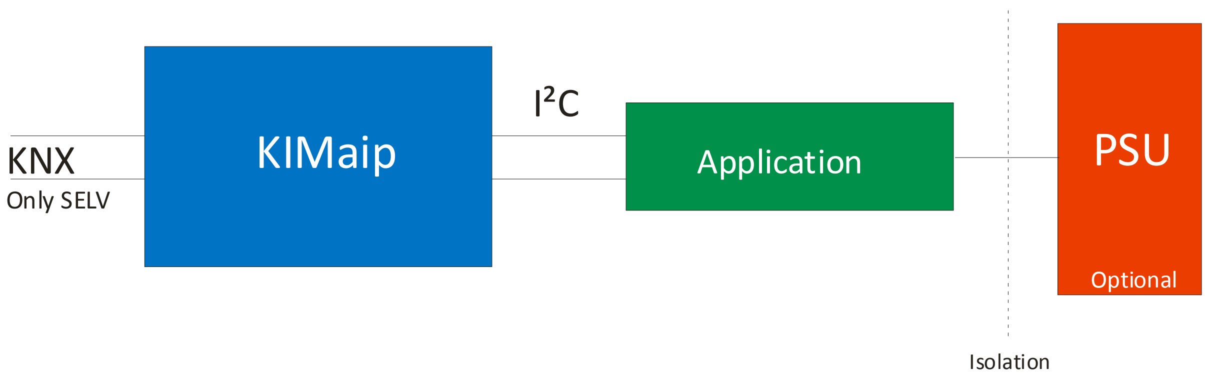
4 Hardware

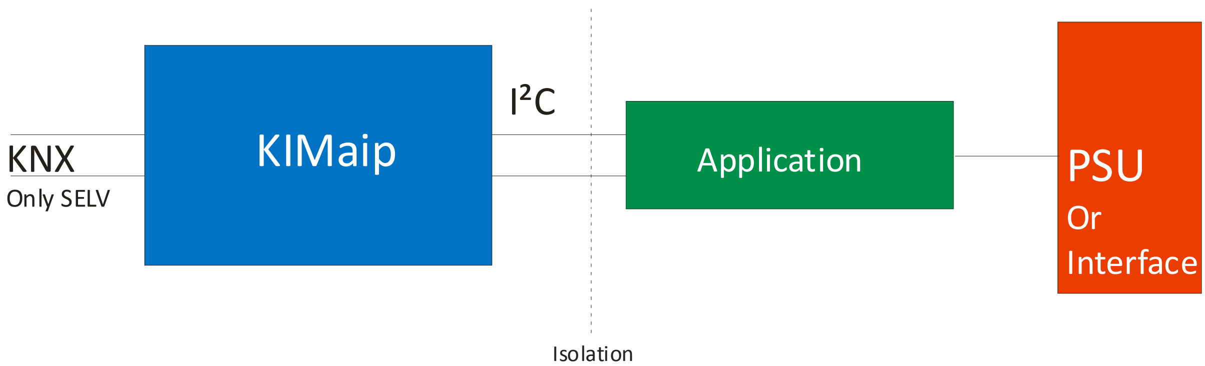
4.1 Block Diagram



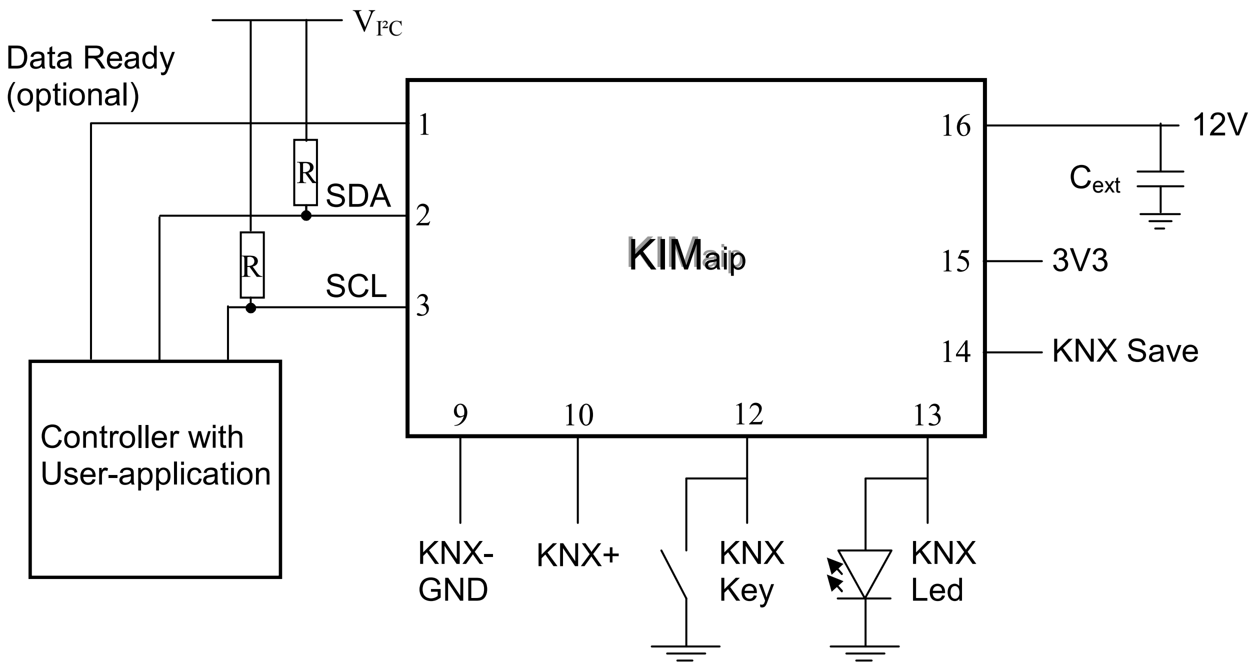
4.1.1 Isolation

There is no need to isolate KIMaip if no unisolated Power circuit is connected to the application. But if this is the case, a galvanic isolation should be projected between KIMaip and the application.





4.2 Pin assignment



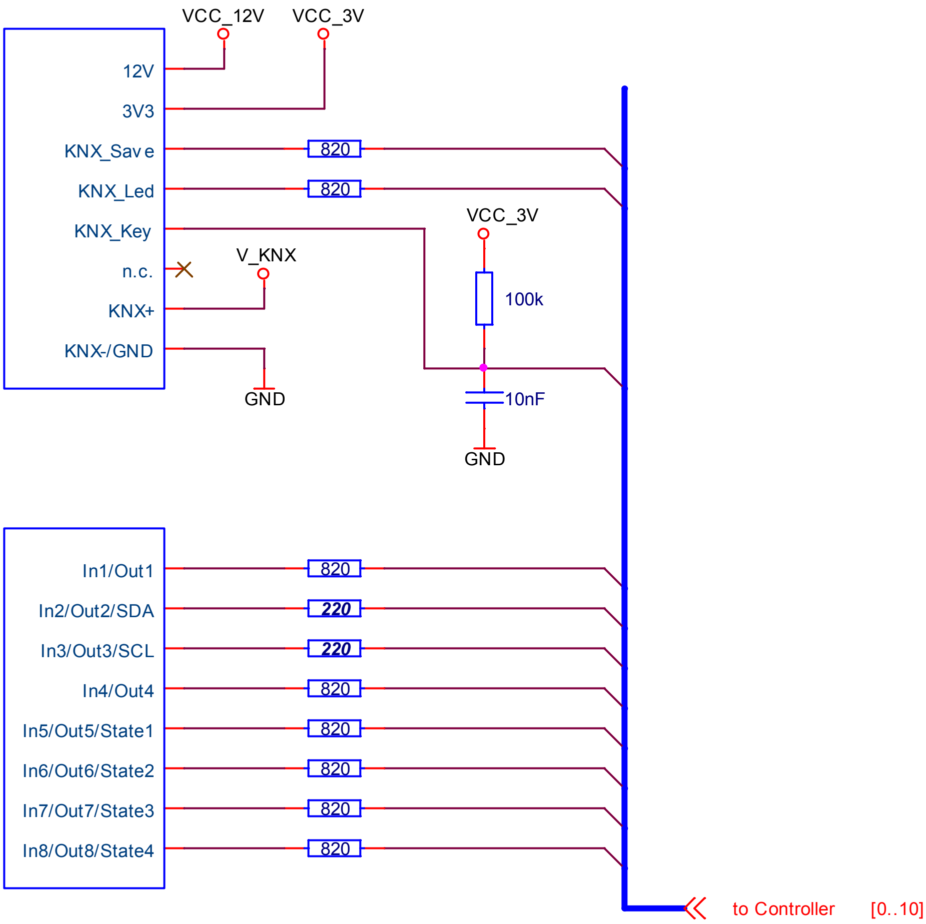
4.2.1 PIN Description

Pin	Description	Pin	Description
1/OUT 1	Data Ready (high active Pin)	12/KNX Key	Pin for KNX progr. Button
2/SDA	Serial Data	13/KNX Led	Pin for KNX progr. LED
3/SCL	Serial Clock	14/KNX Save	'1'= Bus Voltage OK '0'= Bus Voltage ≤ 18V
9/KNX-/GND	Negative Bus Pin	15/3V3	3,3 V for application
10/KNX+	Positive Bus Pin	16/12V	12 V for application

4.2.2 High/Low Voltages I/O-Pins

Inputs	Outputs
for Vcc = 3,3V: V_{low} = 0,45xVcc = 1,48V V_{high} = 0,75xVcc = 2,48V	Outputs without load: V_{low} = 0,25V max V_{high} = 3,05V min (for Vcc = 3,3V)

4.2.3 Series Resistors I/O-Pins



4.3 General Device specification

4.3.1 Absolute Maximum Ratings

All voltages are referring to GND. Currents are declared positive in case of flowing into pin.

Symbol	Parameter	Min	Typ.	Max	Unit
V_{ISO}	Isolation Voltage	No isolation between KNX and interface			
V_{Bus}	Bus Voltage (Bus+ to Bus-)	-45		45	V
I_{Bus}	Maximum current draw from KNX			400	mA
V_{data}	voltage on any interface pin	-0,5		3,5	V
I_{data}	current on any interface pin[1]	-4,0		+4,0	mA
	Storage temperature	-40		85	°C
	maximum power dissipation			1	W

[1] Internally limited, except SDA and SDC

4.3.2 Recommended Operating Conditions

4.3.2.1 General

Symbol	Parameter	Min	Typ.	Max	Unit
V_{Bus}	Bus Voltage (Bus+ to Bus-)	20	28	33	V
	Operating temperature	-25		85	°C
I_{Bus}	Bus power consumption		5		mA
I_{LED}				4,1	mA
	$V_{FLED} = 1{,}5V$		2,1		mA
I_{KEY}			33	50	µA
V_{Vcc3V}		3,2	3,3	3,4	V
I_{Vcc3V}	At $I_{Vcc12V} = 0mA$			30	mA
V_{Vcc12V}	At $I_{Vcc12V} < 30mA$	11,0	12,0	15	V
I_{Vcc12V}	At $I_{Vcc3V} = 0mA$			30	mA
$I_{Vcc12V} + I_{Vcc3V}$	(see note below)			30	mA

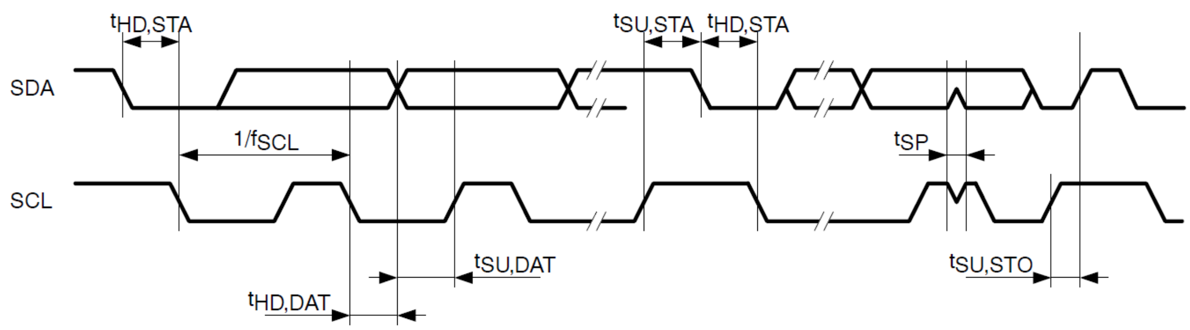
NOTE
If current is drawn from the KIMaip, there is an additional capacitor needed at Pin16.

$I_{Vcc12V} + I_{Vcc3V} > 10mA \rightarrow C_{ext} \geq 68\mu F$
$I_{Vcc12V} + I_{Vcc3V} > 15mA \rightarrow C_{ext} \geq 680\mu F$

4.3.2.2 I²C-Interface

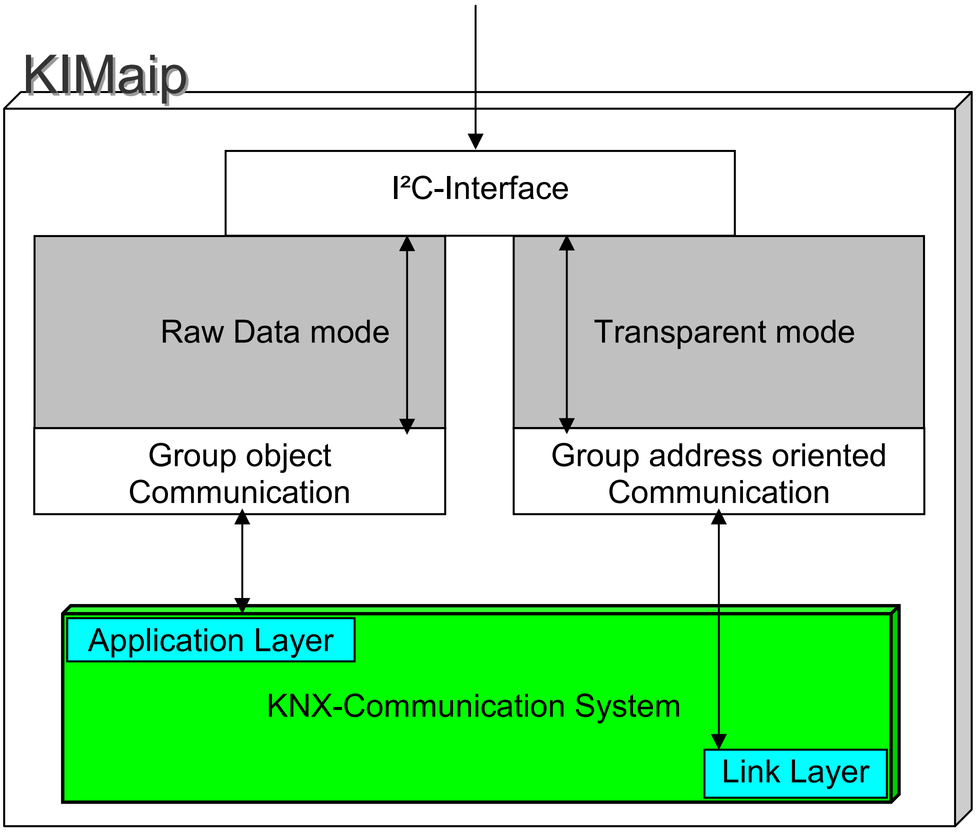
Symbol	Parameter	Min	Typ.	Max	Unit
V_{I^2C}	Supply Voltage of I²C		3,3		V
R_P	at SDA and SDC	2,2	10		kOhm
f_{SCL}	SCL clock frequency	0		400	kHz
$t_{HD,STA}$	Hold time START	4,0			µs
$t_{HD,DAT}$	Data hold time	0			ns
$t_{SU,DAT}$	Data setup time	250			ns
$t_{SU,STO}$	Setup time for STOP	4,0			µs
t_{SP}	Pulse width of spikes suppressed by input filter	50	100	600	ns

4.3.2.3 I²C Mode Timing



5 Software

5.1 Block Diagram



5.2 Introduction

There are two general ways of operation for the KIMaip:

- In the first way of operation the KIMaip behaves like a normal device. It has its own physical address, an address table, communication objects and a complete device management; this is called **Raw Data Mode**.
- In the second way of operation the KIMaip has an additional bypass activated in the lower layers for group oriented communication. Thus the complete group oriented traffic is routed to the I²C-Interface without any communication objects or any filtering; this is called **Transparent Mode**.

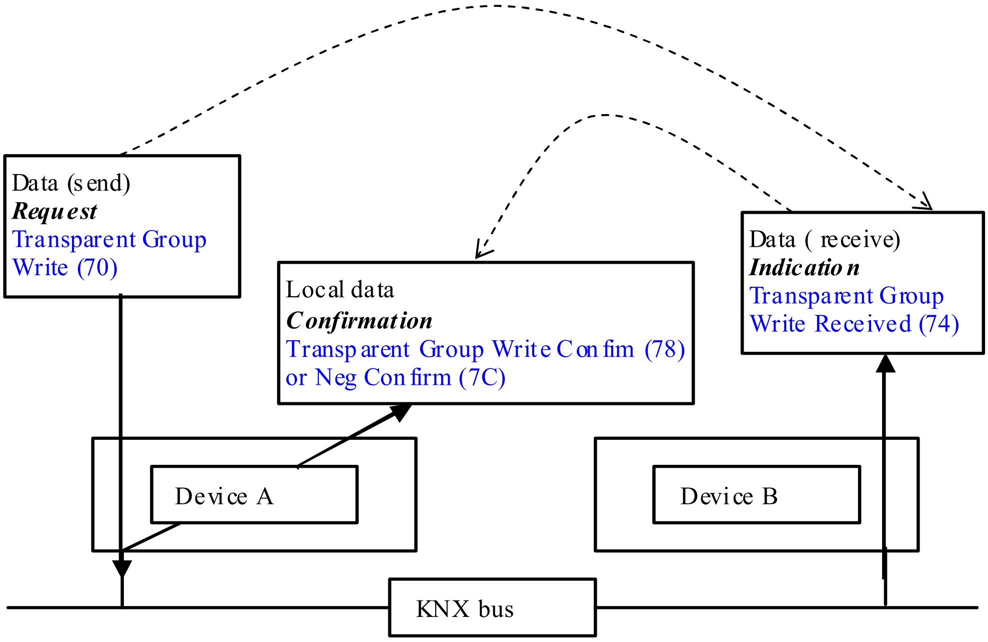
5.2.1 Raw data mode: Operation with Communication Objects

In this mode the KIMaip transfers the data transparent from the I²C-Interface to KNX and vice versa. Here the transmission of the data is triggered completely by the I²C-Interface. The configuration of this mode is normally done via ETS or I²C-Interface – no external tool is required.

5.2.2 Transparent mode: Operation without Communication Objects

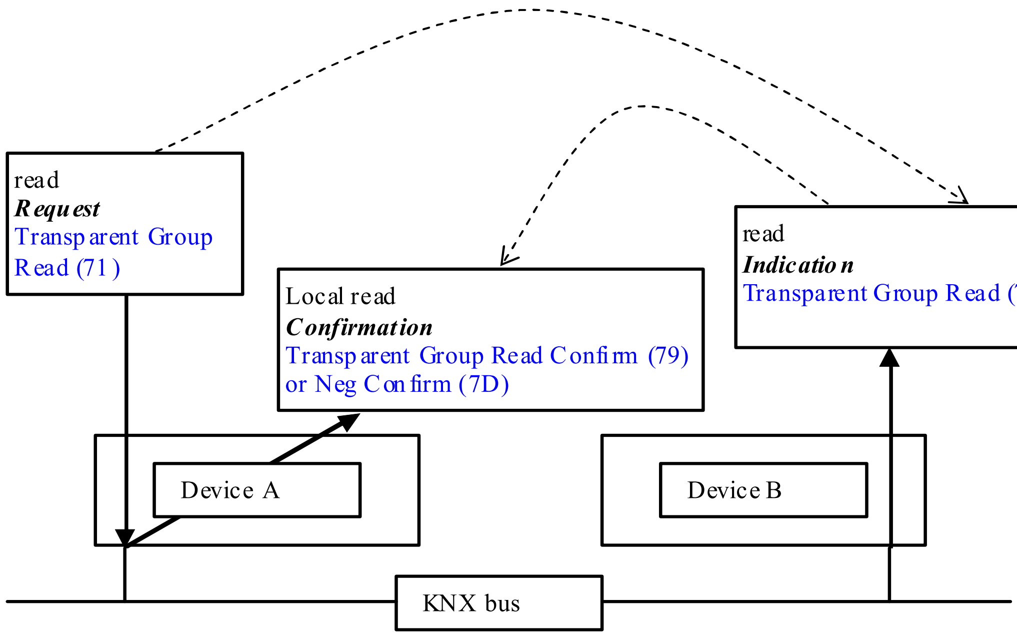
In Transparent mode the KIMaip activates a bypass that channels all group oriented messages from the lower layers directly to the I²C-Interface without filtering. Also all requests coming from the I²C-Interface are sent directly to the lower layers.
Transparent mode may be switched on or off with the Parameter ID 0100h.
For using the transparent mode, it is essential to understand the communication mechanism for group oriented communication. Here they will be described step by step in three communication situations.

Case 1) The most simple communication situation is when one device wishes to distribute a value.

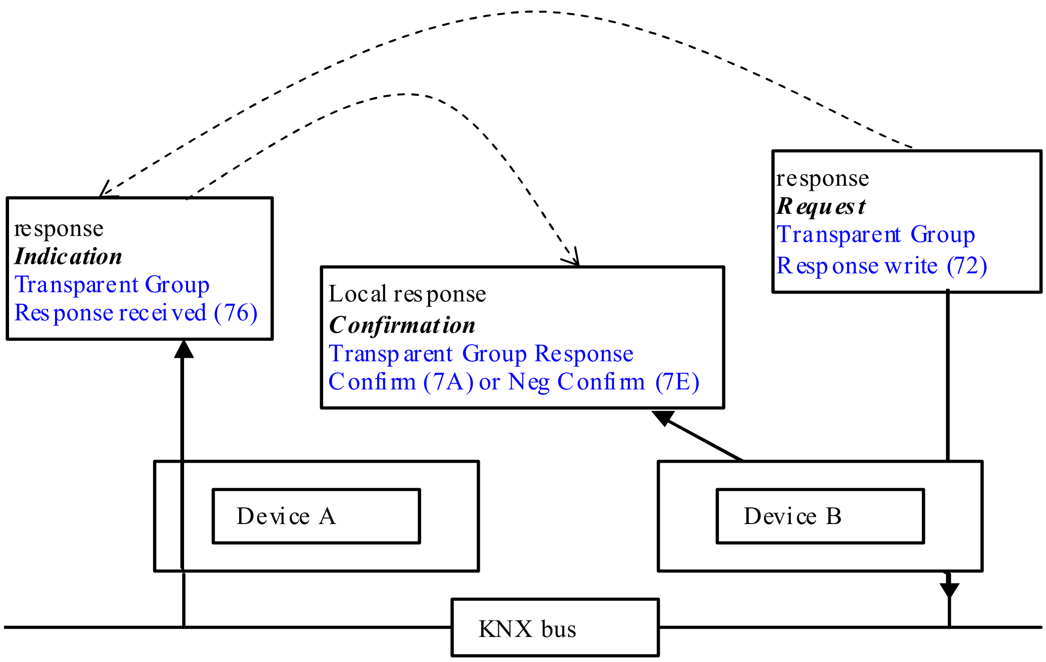


For that a 'send data request' is sent to the KIMaip. This request is forwarded by the device A to the bus. If local confirmations are enabled, a message is returned to the I²C-Interface with the information of success or fail in transmitting. This is based on the result of immediate acknowledge reception after transmission of the message on KNX bus. The remote device B is receiving the message from device A and indicates this fact to its own I²C-Interface by sending a data indication.

Case 2) The second possibility is when you ask a device to send you its actual value. This request for data is called a read request. A local confirmation is sent via I²C-Interface once the transmission is completed. The remote device receives the request and sends a read indication to its own I²C-Interface. This step does not yet involve getting the value requested!



Case 3) The last possibility is sending the answer as a result of the request of the case 2. The remote device sends a response message from its I²C-Interface to the KNX bus. It is locally confirmed in the remote device and the same message is distributed via the KNX bus to the other devices connected. Our device A receives this message and sends a response indication via I²C-Interface.



6 I²C Protocol

6.1 Settings

Default Address	08h
START condition	yes
STOP condition	yes
Acknowledge	yes
Clock stretching	yes
7-bit slave address	yes
General Call address	Not supported
Software Reset	Not supported
START byte	
Device ID	8

7 Serial Protocol

Messages to Module		Messages from Module	
Cmd/h	**Message**	**Cmd/h**	**Message**
10	ValueSend	20	ValueReceive
11	ValueReadSend	21	ValueReadReceive
12	ValueResponseSend	32	ObjectDescriptionResponse
30	ObjectDescriptionWrite	39	GroupAddressResponse
31	ObjectDescriptionRead	42	ParameterResponse
34	GroupAddressAddSending	52	InterfaceObjectResponse
35	GroupAddressAdd	62	SystemParameterResponse
36	GroupAddressDelete	80	SystemEvent
37	GroupAddressDeleteAll	E0	ErrorHandling
38	GroupAddressRead		
40	ParameterWrite		
41	ParameterRead		
50	InterfaceObjectWrite		
51	InterfaceObjectRead		
60	SystemParameterWrite		
61	SystemParameterRead		
70	TransparentGroupWrite		
71	TransparentGroupRead		
72	TransparentGroupResponseWrite		
74	TransparentGroupWriteRcvd		
75	TransparentGroupReadRcvd		
76	TransparentGroupResponseRcvd		
78	TransparentGroupWriteConf		
79	TransparentGroupReadConf		
7A	TransparentGroupResponseConf		
7C	TransparentGroupWriteNegConf		
7D	TransparentGroupReadNegConf		
7E	TransparentGroupResponseNegConf		
D0	ResetDeviceToDefault		

7.1 Command Description

7.1.1 Abbreviations example

Byte	Symbol	Description
0	I R/W	I²C Address Read 1 / Write 0
1	L	Length
2	C	Command
3/4	G	Group Object index
5…	V	Value

NOTE
The abbreviations I, W, L and C are available for all the following telegrams.
I2C-Address and Read/Write **are not mentioned** in the following commands description.
The first mentioned Byte is already the length of the telegram.

Example: Value Send

[0]	[1]	[2]	[3]	[4]	[5] …
IIIIIIIW 0	LLLLLLLL	CCCCCCCC 00010000	GGGGGGGG	GGGGGGGG	VVVVVVVV

Value Send: < 04 **10** 00 01 00

Value

Length Command Object

7.1.2 Telegram direction

In this documentation the direction of the telegrams will be differentiated as follow:
to module: <, from module: >

Example:	Value Send Object 5 ON:	< 04 **10** 00 05 01
	System Event "Object Conf…":	> 05 **80** 04 02 00 05

7.2 Message from Module

7.2.1 Empty data

[0]	[1]	[2]
IIIIIIIR 1	LLLLLLLL 00000001	CCCCCCCC 00000000

No data available: 01 **00**

NOTE
If there is no possibility to check the presence of messages directly from Pin 1 of the module "Data ready", it is possible to read this information with the command 00.

7.3 Accessing Object Values

G Group Object index
V Value

7.3.1 Messages to Module

7.3.1.1 ValueSend

[0]	[1]	[2]	[3]	[4]	[5] …
IIIIIIIW 0	LLLLLLLL	CCCCCCCC 00010000	GGGGGGGG	GGGGGGGG	VVVVVVVV

Example:
Value Send Object 5 ON: < 04 **10** 00 05 01
System Event "Object Confirmation OK": > 05 **80** 04 02 00 05

Value Send Object 8 OFF: < 04 **10** 00 08 00
System Event "Object not sent…": > 05 **80** 04 10 00 08

Example: Object 135 and 137 are 6 bytes long and have the same group address. Only "Value send" on Object 135
Value Send Object 135 < 00 09 **10** 00 87 11 22 33 44 55 66
Value Receive Object 137 > 00 09 **20** 00 89 11 22 33 44 55 66
System Event "Object 135 Confirm OK": > 00 05 **80** 04 02 00 87

7.3.1.2 ValueReadSend

[0]	[1]	[2]	[3]	[4]
90 IIIIIIIW 0	LLLLLLLL 00000011	CCCCCCCC 00010001	GGGGGGGG	GGGGGGGG

Example:
Value Read Object 6: < 03 **11** 00 06
System Event "Object Confirmation OK": > 05 **80** 04 02 00 06

7.3.1.3 ValueResponseSend

[0]	[1]	[2]	[3]	[4]	[5] …
IIIIIIIW 0	LLLLLLLL	CCCCCCCC 00010010	GGGGGGGG	GGGGGGGG	VVVVVVVV

A "Value Response" has to be sent, if the module gets a "Value Read command".

Example:
Value Response Object 4/00: < 04 **12** 00 04 00
Value Response Object 4/01: < 04 **12** 00 04 01
System Event "Object Confirmation OK": > 05 **80** 04 02 00 04

Example: Object 95 and 97 are 4 bytes long and have the same group address. Only "Value Response send" on Object 95
Value Response send Object 95: < 07 **12** 00 5F 44 33 22 11
Value Response Object 4/01: > 07 **20** 00 61 44 33 22 11
System Event "Object Confirmation OK": > 05 **80** 04 02 00 5F

7.3.2 Messages from Module

7.3.2.1 ValueReceive

[0]	[1]	[2]	[3]	[4]	[5] …
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 00100000	GGGGGGGG	GGGGGGGG	VVVVVVVV

Example: Values 0 and 1 for Object 8
Value Object 8 = OFF: > 04 **20** 00 08 00
Value Object 8 = ON: > 04 **20** 00 08 01

7.3.2.2 ValueReadReceive

[0]	[1]	[2]	[3]	[4]
IIIIIIIR 1	LLLLLLLL 00000011	CCCCCCCC 00100001	GGGGGGGG	GGGGGGGG

If a "Value Read" is received, the module will send a "Value Read Receive".

Example: Value read for Objects 6 and 10
Value Read Receive Object 6: > 03 **21** 00 06
Value Read Receive Object 10: > 03 **21** 00 0A

7.4 Accessing object description

G Group Object index
T Object type - see 7.4.2.2
F Configuration flags - see 7.4.2.3

7.4.1 Messages to Module

7.4.1.1 ObjectDescriptionWrite

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIW 0	LLLLLLLL 00000101	CCCCCCCC 00110000	GGGGGGGG	GGGGGGGG	TTTTTTTT	FFFFFFFF

Example: for Object 1, 1-bit type and communication flags are to be configured DFh.

Object Description Write: < 05 **30** 00 01 00 DF
Object Description Response: > 05 **32** 00 01 00 DF

7.4.1.2 ObjectDescriptionRead

[0]	[1]	[2]	[3]	[4]
IIIIIIIW 0	LLLLLLLL 00000011	CCCCCCCC 00110001	GGGGGGGG	GGGGGGGG

Example: Object 2 is 1-bit type and communication flags are configured D7h.

Object Description Read: < 03 **31** 00 02
Object Description Response: > 05 **32** 00 02 00 D7

7.4.2 Messages from Module

7.4.2.1 ObjectDescriptionResponse

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIR 1	LLLLLLLL 00000101	CCCCCCCC 00110010	GGGGGGGG	GGGGGGGG	TTTTTTTT	FFFFFFFF

See examples in Object Description Write and Read

7.4.2.2 GROUP OBJECT TYPES (OBJECTTYPE)

The group object types are according to the resource definition in the Konnex KNX Handbook.

value (code)	size	used length
0	1 bit	1 byte
1	2 bit	1 byte
2	3 bit	1 byte
3	4 bit	1 byte
4	5 bit	1 byte
5	6 bit	1 byte
6	7 bit	1 byte
7	1 byte	1 byte
8	2 byte	2 byte
9	3 byte	3 byte
10	4 byte	4 byte
11	6 byte	6 byte
12	8 byte	8 byte
13	10 byte	10 byte
14	14 byte	14 byte

7.4.2.3 STRUCTURE OF THE CONFIGURATION FLAGS (COMFLAGS)

The configuration-flags contain the information about the communication. They are identical to the flags in ETS in the „Edit Object“ dialog.

Bit	7	6	5	4	3	2	1	0
	read response enable	transmit enable	0	write enable	read enable	comm. enable	priority	priority
	U	T	0	W	R	C	P	P

PP priority
possible values are:
3 = low
1 = high
2 = urgent
0 = system (do not use!)

C Communication Enable
Enables the communication of the object

R Read Enable
Has to be set, if the Object-value should be readable from the KNX

W Write Enable
Enables the receiving from the KNX

T Transmit Enable
Enables the sending to the KNX

U Update Enable
Has to be set, if the Object-value should be updated with read-responses

0 not used

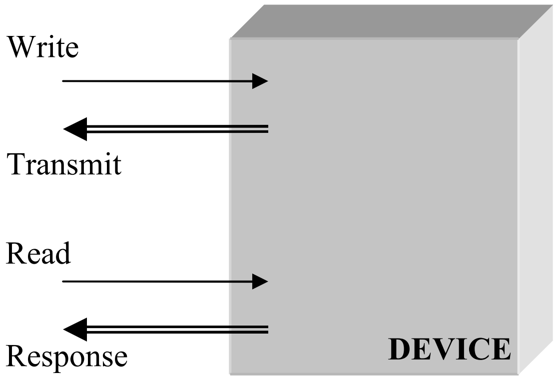
NOTE

Default value of the communication flags is D7h

7	6	5	4	3	2	1	0
read response enable	transmit enable	0	write enable	read enable	comm. enable	priority	priority
U	T	0	W	R	C	P	P
1	1	0	1	0	1	1	1

7.4.2.4 DEFINITION OF THE DIRECTION

Direction of telegrams:



7.5 Accessing GroupAdresses

G Group Object index
A Group address
X Index
M Maximal Number of addresses 1h..30h → 1dez..48dez
N Number of addresses 0..48 / effective value
O overflow (more addresses available) → No overflow = 0, overflow = 1

NOTE

Please note that the response telegramm “E0 00 00” means “RESPONSE OK”!

7.5.1 Messages to Module

7.5.1.1 GroupAddressAddSending

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIW 0	LLLLLLLL 00000101	CCCCCCCC 00110100	GGGGGGGG	GGGGGGGG	AAAAAAAA	AAAAAAAA

Example: Object 1, add Address 1 as “Sending Address”

Add Sending Group Address 1: < 05 **34** 00 01 00 01
Response: > 06 $\underbrace{\text{E0 00 00}}$ 34 00 01

Response for command “34”/Object 0001 OK

NOTE

Please note that the added/deleted “Address” is not mentioned in the Response telegram.

If no address is specified as a "sending address", the first entered address will be taken as the "sending address".
If several addresses are specified as "sending address", the last entered address will be taken as the "sending address".

7.5.1.2 GroupAddressAdd

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIW 0	LLLLLLLL 00000101	CCCCCCCC 00110101	GGGGGGGG	GGGGGGGG	AAAAAAAA	AAAAAAAA

Example: Object 1, add address 7
Add Group address 7:< 05 **35** 00 01 00 07
Response: > 06 E0 00 00 35 00 01

7.5.1.3 GroupAddressDelete

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIW 0	LLLLLLLL 00000101	CCCCCCCC 00110110	GGGGGGGG	GGGGGGGG	AAAAAAAA	AAAAAAAA

Example: Object 1, delete address 7
Delete Group address 7: < 05 **36** 00 01 00 07
Response: > 06 E0 00 00 36 00 01

7.5.1.4 GroupAddressDeleteAll

[0]	[1]	[2]	[3]	[4]
IIIIIIIW 0	LLLLLLLL 00000011	CCCCCCCC 00110111	GGGGGGGG	GGGGGGGG

Example: Object 1, Delete all addresses
Delete all Group address: < 05 **37** 00 01
Response: > 06 E0 00 00 37 00 01

7.5.1.5 GroupAddressRead

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIW 0	LLLLLLLL 00000101	CCCCCCCC 00111000	GGGGGGGG	GGGGGGGG	XXXXXXXX	MMMMMMMM

Example 1: Object 1, read all addresses (48 max, 30hex) from Index 00

Read Gr addrs for Object 1: < 05 **38** 00 01 00 30
Response: > 0D **39** 00 01 00 04 0001 0002 0003 0004

No Overflow
4 addresses are available (1 to 4)

Example 2: Object 1, read 3 addresses from Index 00

Read Gr addrs for Object 1: < 05 **38** 00 01 00 03
Response: > 0B **39** 00 01 00 83 0001 0002 0003

Only 3 addresses read (1 to 3)
Overflow because more than 3 addresses are available

7.5.2 Messages from Module

7.5.2.1 GroupAddressResponse

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7] …
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 00111001	GGGGGGGG	GGGGGGGG	XXXXXXXX	ONNNNNNN	AAAAAAAA

See examples in Group Address Read

7.6 Accessing Parameter

A Address (it's not a Group address!)
N number of data
V Value

NOTE
Maximal number of data length is 96 bytes.

7.6.1 Messages to Module

7.6.1.1 ParameterWrite

[0]	[1]	[2]	[3]	[4]	[5]	[6] …
IIIIIIIW 0	LLLLLLLL	CCCCCCCC 01000000	AAAAAAAA	AAAAAAAA	NNNNNNNN	VVVVVVVV

Example: from address 1, write the 5 bytes: 99 88 77 66 55
Parameter Write: < 09 **40** 00 01 05 99 88 77 66 55
Parameter Response: > 09 **42** 00 01 05 99 88 77 66 55

7.6.1.2 ParameterRead

[0]	[1]	[2]	[3]	[4]	[5]
IIIIIIIW 0	LLLLLLLL 00000100	CCCCCCCC 01000001	AAAAAAAA	AAAAAAAA	NNNNNNNN

Example: from address 3, read 2 bytes
Parameter Read: < 04 **41** 00 03 02
Parameter Response: > 09 **42** 00 03 02 77 66

7.6.2 Messages from Module

7.6.2.1 ParameterResponse

[0]	[1]	[2]	[3]	[4]	[5]	[6] …
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01000010	AAAAAAAA	AAAAAAAA	NNNNNNNN	VVVVVVVV

See examples in Parameter Write and Read.

7.7 Accessing Interface Objects

I	Object Index (Index 0 to 4: KNX-defined, other Indexes device-specific)
P	Property Id (please consult the KNX Manual "Resources" for this list)
S	Start Index (0 for Number of elements type otherwise 1 to N)
N	Number of Elements
V	Value

7.7.1 Messages to Module

7.7.1.1 InterfaceObjectWrite

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
IIIIIIIW 0	LLLLLLLL	CCCCCCCC 01010000	IIIIIIII	PPPPPPPP	NNNNSSSS	SSSSSSSS	VVVVVVVV

Example: Index 0, PID "11", new Serial Number 01 02 03 04 05 06
Interface Object Write: < 0B **50** 00 0B 10 01 01 02 03 04 05 06
Interface Object Response: > 0B **52** 00 0B 10 01 01 02 03 04 05 06

7.7.1.2 InterfaceObjectRead

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIW 0	LLLLLLLL 00000101	CCCCCCCC 01010000	IIIIIIII	PPPPPPPP	NNNNSSSS	SSSSSSSS

Example: Index 0, PID "11" Serial Number
Interface Object Read: < 05 **51** 00 0B 10 01
Interface Object Response: > 0B **52** 00 0B 10 01 FF FF FF FF FF FF

Example: Index 3, PID "13" Program (Application-) Version
Interface Object Read: < 05 **51** 03 0D 10 01
Interface Object Response: > 0A **52** 03 0D 10 01 00 72 F3 00 01

7.7.2 Messages From Module

7.7.2.1 InterfaceObjectResponse

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01010010	IIIIIIII	PPPPPPPP	NNNNSSSS	SSSSSSSS	VVVVVVVV

See examples in Interface Object Write and Read.

7.8 Accessing System Parameter

P Parameter ID
V Value

7.8.1 Messages to Module

7.8.1.1 SystemParameterWrite

[0]	[1]	[2]	[3]	[4]	[5] …
IIIIIIIW 0	LLLLLLLL	CCCCCCCC 01100000	PPPPPPPP	PPPPPPPP	VVVVVVVV

Example: System event disable
Syst Param Write: < 07 **60** 00 11 00 00 00 00
Syst Param Response: > 07 **62** 00 11 00 00 00 00

Example: Write Physical address
Syst Param Write: < 05 **60** 00 02 15 08
Syst Param Response: > 05 **62** 00 02 15 08

7.8.1.2 SystemParameterRead

[0]	[1]	[2]	[3]	[4]
IIIIIIIW 0	LLLLLLLL 00000011	CCCCCCCC 01100001	PPPPPPPP	PPPPPPPP

Example: Read Physical address
Syst Param Read: < 03 **61** 00 02
Syst Param Response: > 05 **62** 00 02 15 08

Example: Read Version
Syst Param Read: < 03 **61** 00 01
Syst Param Response: > 09 **62** 00 01 01 00 FF 01 00 FF

7.8.1.3 ResetDeviceToDefault

[0]	[1]	[2]	[3]	[4]
IIIIIIIW 0	LLLLLLLL 00000011	CCCCCCCC 11010000	11111111	11111111

Example: Reset device to default
Reset : < 03 **D0** FF FF
→ All system parameters are set back to their default value.

NOTE
Please note that a “Reset to default” should only be executed if **necessary**, because this command does overwrite the flash.

7.8.2 Messages from Module

7.8.2.1 SystemParameterResponse

[0]	[1]	[2]	[3]	[4]	[5] …
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01100010	PPPPPPPP	PPPPPPPP	VVVVVVVV

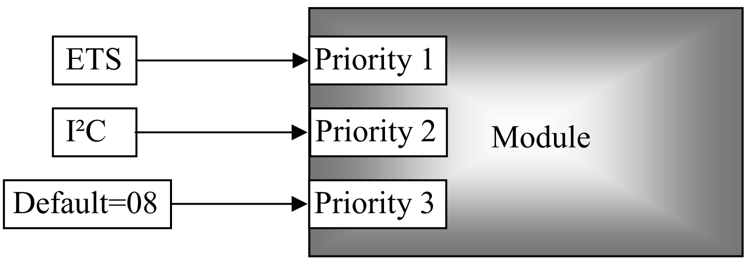
See examples in System Parameter Write and Read.

7.8.3 System Parameter-IDs

Parameter-ID	Data length [bytes]	Default-value	Read / Write	Description
0001h	6		R / -	Version
0002h	2	FFFFh	R/ W	Physical address
0003h	1	0	R/ W	Programming mode
0005h	1	0	R/ W	I^{2}C-Address, see Chap. 7.8.3.1
0006h	1	08h	R/ -	Actual used I^{2}C-Address, see Chap. 7.8.3.1
0011h	4	FFFFFFFFh	R/ W	Bit field to enable the System-Events Only the LSB is used: Bit 1: System Event Bit 2: Transport Layer Bit 3: Load/Run State Bit 4: Object confirmation remaining bytes for future development
0012h	1	1	R/ W	Enable error-messages
0020h	2	50	R/ W	Timeout over an I^{2}C-message
0021h	2	200	R/ W	Timeout for indications
0022h	2	10	R/ W	Timeout for indications if no communication to the master
0023h	1	3	R/ W	Retry sending indications, till no communication is detected
0100h	1	0	R/ W	Settings for transparent-mode Bit 0: Enable Bit 1: Enable confirmation (to I^{2}C) Bit 7: Send Acknowledge
0110h	1	0	R/ -	GroupLink disabled
0200h	1	83h	R/ -	State of Device, see Chap. 7.8.3.2
F0F0h	1		W	Reset the module

7.8.3.1 I^{2}C Address

There are three possibilities to determine or to set the I^{2}C Address.



If no address or address 00 is set from the ETS or from the I²C-Master, the used I²C-Slave address is the default address (08). The ETS has the highest priority to set the address.

7.8.3.2 Parameter "State of device" (0200)h

This parameter returns the various states of the device.

Bit	7	6	5	4	3	2	1	0
	Bus is OK	Programming mode	0	0	Transparent mode enabled	0	Application loaded	Application running
	1	0	0	0	0	0	1	1
	Normal mode 83h							

This parameter has a value of 83h in normal mode.
Syst Param Read: < 03 **61** 02 00
Syst Param Response: > 04 **62** 02 00 83

7.9 TRANSPARENT MODE

A Group address
D Data length (0 is equal to 1..6bit), otherwise number of bytes
V Value

To enable the Transparent Mode see Chapter 7.8, "Accessing System Parameter" Parameter ID 0100h

7.9.1 MESSAGES TO MODULE

7.9.1.1 TRANSPARENTGROUPWRITE

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIW 0	LLLLLLLL	CCCCCCCC 01110000	AAAAAAAA	AAAAAAAA	DDDDDDDD	VVVVVVVV

Example: Value 35h to address 0001 and 6 Bits-Data (Data length=0)
Transp. Group Write addr 0001: < 05 **70** 00 01 00 35
Transp. Group Write Conf.: > 05 **78** 00 01 00 35

Example: Value AFh to address 0002 and 2 Bytes-Data (Data length=2)
Transp. Group Write addr 0002: < 06 **70** 00 02 02 AF FF
Transp. Group Write Conf.: > 06 **78** 00 02 02 AF FF

7.9.1.2 TRANSPARENTGROUPREAD

[0]	[1]	[2]	[3]	[4]
IIIIIIIW 0	LLLLLLLL 00000011	CCCCCCCC 01110001	AAAAAAAA	AAAAAAAA

Example: Read address 0001
Transp. Group Read addr 0001: < 03 **71** 00 01
Transp. Group Read Conf.: > 05 **79** 00 01

7.9.1.3 TRANSPARENTGROUPRESPONSEWRITE

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIW 0	LLLLLLLL	CCCCCCCC 01110010	AAAAAAAA	AAAAAAAA	DDDDDDDD	VVVVVVVV

Example: in address 0003 Value 22h, 6 Bits (Data length=0)
Transp. Group Resp. Write addr 0003: < 05 **72** 00 03 00 22
Transp. Group Resp. Conf.: > 05 **7A** 00 03 00 22

7.9.2 Messages From Module

7.9.2.1 TransparentGroupWriteRcvd

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01110100	AAAAAAAA	AAAAAAAA	DDDDDDDD	VVVVVVVV

Example: address 1, 6 Bits length (Data length=0), Value 1
Transp. Group Write Received: > 05 **74** 00 01 00 01

Example: address 1, 1 Byte length (Data length=1), Value 12
Transp. Group Write Received: > 05 **74** 00 01 01 12

7.9.2.2 TransparentGroupReadRcvd

[0]	[1]	[2]	[3]	[4]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01110101	AAAAAAAA	AAAAAAAA

Example: from address 1
Transp. Group Read Received: > 03 **75** 00 01

7.9.2.3 TransparentGroupResponseRcvd

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01110110	AAAAAAAA	AAAAAAAA	DDDDDDDD	VVVVVVVV

Example: address 1, 1 Bit length (Data length=0), Value 1
Transp. Group Response Received: > 05 **76** 00 01 00 01

Example: address 1, 1 Byte length (Data length=1), Value 33
Transp. Group Response Received: > 05 **76** 00 01 01 33

7.9.2.4 TransparentGroupWriteConf

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01111000	AAAAAAAA	AAAAAAAA	DDDDDDDD	VVVVVVVV

See examples in Transparent Group Write.

7.9.2.5 TransparentGroupReadConf

[0]	[1]	[2]	[3]	[4]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01111001	AAAAAAAA	AAAAAAAA

See example in Transparent Group Read.

7.9.2.6 TransparentGroupResponseConf

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01111010	AAAAAAAA	AAAAAAAA	DDDDDDDD	VVVVVVVV

See example in Transparent Group Response Write.

7.9.2.7 TransparentGroupWriteNegConf

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01111100	AAAAAAAA	AAAAAAAA	DDDDDDDD	VVVVVVVV

Example: Value 37h to address 0001 but no Confirmation from Bus
Transp. Group Write addr 0001: < 05 **70** 00 01 00 37
Transp. Group Write Neg Conf.: > 05 **7C** 00 01 00 37

7.9.2.8 TransparentGroupReadNegConf

[0]	[1]	[2]	[3]	[4]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01111101	AAAAAAAA	AAAAAAAA

Example: Read address 0002 but no Confirmation from Bus
Read addr 0002: < 03 **71** 00 02
Read Neg Conf.: > 03 **7D** 00 02

7.9.2.9 TransparentGroupResponseNegConf

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIR 1	LLLLLLLL	CCCCCCCC 01111110	AAAAAAAA	AAAAAAAA	DDDDDDDD	VVVVVVVV

Example: Value 22h to address 0003 but no Confirmation from Bus
Transp. Group Resp. Write addr 0003: < 05 **72** 00 03 00 22
Transp. Group Resp. Write Conf.: > 05 **7E** 00 03 00 22

7.10 SYSTEM INDICATIONS

E Event-ID
D Data depending on the Event-ID

7.10.1 MESSAGES FROM MODULE

7.10.1.1 SYSTEMEVENT

[0]	[1]	[2]	[3]	[4]	[5]	[6]
IIIIIIIR 1	LLLLLLLL 00000101	CCCCCCCC 10000000	EEEEEEEE	EEEEEEEE	DDDDDDDD	DDDDDDDD or DDDDDDDD

7.10.2 EVENT-IDS

Event-ID	Definition
0100h	System events and…. 01 00h = Program button off 01 01h = Program button on 01 02h = Physical Address 02 00h = Bus error 02 01h = Bus OK 03 00h = Application started
0200h	Transport-layer connection and…. 00 00h = Disconnect AA AA = Address (Connection with…)
0300h	Load state machine and…. 01 **LS** h = Address table 02 **LS** h = Association table 03 **LS** h = Application table "**LS**" has three possible **L**oad **S**tatuses: 00h = is Unloaded 01h = is Loaded 02h = is Loading Please see for further information the KNX Manual, "System Specifications".
0402h	Object confirmation – OK and… Object Nr
0403h	Object confirmation – Error and… Object Nr
0410h	Object not sent, no group-address, or no communication and… Object Nr

Example: System Event, KNX Program button On/Off
Switched On: > 05 **80** 01 00 01 01
Switched Off: > 05 **80** 01 00 01 00

Example: System Event, KNX-Bus disconnected
Disconnected: > 05 **80** 01 00 02 00
Reconnected: > 05 **80** 01 00 02 01

Example: Transport Layer Connection with….
…Address A0 07: > 05 **80** 02 00 A0 07

Example:
1) Load State Machine during a Download from the ETS:

Address table/ Unloaded: > 05 **80** 03 00 01 00
Association table/ Unloaded: > 05 **80** 03 00 02 00
Application table/ Unloaded: > 05 **80** 03 00 03 00

2) Load State Machine after a Download from the ETS:

Address table/ Loading: > 05 **80** 03 00 01 02
Address table/ Loaded: > 05 **80** 03 00 01 01

Association table/ Loading: > 05 **80** 03 00 02 02
Association table/ Loaded: > 05 **80** 03 00 02 01

Application table/ Loading: > 05 **80** 03 00 03 02
Application table/ Loaded: > 05 **80** 03 00 03 01

Example: Object 1 confirmation
Value Send Object 1/ ON: < 04 **10** 00 01 01
Object Confirmation is OK: > 05 **80** 04 02 00 01

Example: Object EE not transmitted
Value Send Object EE/ ON: < 04 **10** 00 EE 01
Object not sent, no GA: > 05 **80** 04 10 00 EE

7.10.3 List of possible Errors

Error	ID, h	Comment
KIE_SEND_RESPONSE_OK	0000	!!!Not an error message!!!
KIE_LEN_DOESNT_MATCH	0010	received length and length-byte doesn't match
KIE_COMMAND_LENGTH	0020	length doesn't match to the command
KIE_COMMAND_NOT_FOUND	0100	wrong command, not found
KIE_APPL_STOP	0150	requested cmd denied because application stopped
KIE_NO_OBJ	0151	wrong object number
KIE_OBJ_LENGTH	1000	wrong object length
KIE_OBJ_TRANSP	1020	Access denied, if Transparent enabled
KIE_GA_ADD_DIS	3000	Access denied
KIE_GA_COUNT	3001	the maximal number (48) of GA is exceeded
KIE_GA_ADD_ERROR	3010	Error during add the GA
KIE_GA_DEL_ERROR	3011	Error during deleting a GA
KIE_PARAM_LENGTH	4000	Max. parameter length (96 bytes) is exceeded
KIE_PARAM_RANGE	4001	Max. parameter range (2 KB) is exceeded
KIE_NO_PROP	5000	Property not found
KIE_PROP_WRITE	5001	Error while writing the property
KIE_SYSID_NOT_FOUND	6000	System parameter not existing
KIE_SYS_LENGTH	6001	System parameter has the wrong length
KIE_SYS_ACCESS	6002	Error while writing system parameter
KIE_SYS_READ_ONLY	6003	Write not allowed
KIE_TRANSP_DISABLED	7000	wrong command, Transp. mode disabled
KIE_TRANSP_LEN_MISMATCH	7010	wrong data length
KIE_TRANSP_LEN_OV	7020	more than 96 bytes -> not allowed!?!?
KIE_TRANSP_GA	7030	GA not available
KIE_DEVICE_RESET	D000	Device not successfully reseted to default

7.10.3.1 Error Handling

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
IIIIIIIR 1	LLLLLLLL 00000110	CCCCCCCC 11100000	EEEEEEEE	EEEEEEEE	DDDDDDDD	DDDDDDDD	DDDDDDDD

Example: Error "wrong object nbr" after an "Object description read"
Object description read: < 03 **31** 01 0D
Error message: > 06 **E0** **01 51** 31 01 0D
→ Because……. there is no object number 269 → 010Dh!

Example: Error "parameter length"; from address 00 05, read 200 bytes
Parameter Read: < 04 **41** 00 05 C8
Error message: > 06 **E0** **40 00** 41 00 05
→ Because……. the maximal parameter length is 96 Bytes!

Note
In the Error message the three data bytes after the Error-ID are those just after the data length in the telegram sent.

8 ETS

8.1 Group Objects

Number of group addresses	254
Number of associations	255
Number of communication objects	255
Address of association table	0x4201
Address of communication object table	0x4400

8.2 Parameter

Address	Internal name	Description
4800	PAR_UI_8_0_deviceParam_iic_slaveAddress	
4801	PAR_UI_8_0_deviceParam_iic_slaveAddrSource	
4802	PAR_UI_8_0_deviceParam_groupIndPacked	To be put on “0”
4804	PAR_UI_32_0_deviceParam_enableSystemEvents	
4808	PAR_UI_8_0_deviceParam_enableErrorMsg	
4809	PAR_UI_8_0_deviceParam_disableGroupLink	Has to be different from “0”
480A	PAR_UI_8_0_deviceParam_transpModeEnable	
480B	PAR_UI_8_0_deviceParam_dummy	To be put on “0”
4850	PAR_UI_8_0_userParam	

9 Properties (KNX)

9.1 List

Property ID	**Type Read/Write**	**Element Index**	**Usage**
50	PDT_GENERIC_06 RO	APPL_iicVersion	KIMaip-Version. Only to be read.
60	PDT_GENERIC_02 RW	&IMS_msgTimeout	Timeout over an I²C-message (default 50ms).
61	PDT_GENERIC_02 RW	&IMS_indTimeout	Timeout for indications (default 200ms).
62	PDT_GENERIC_02 RW	&IMS_indTimeoutErr	Timeout for indications if no communication to the master (default 10ms).
63	PDT_GENERIC_01 RW	&IMS_indMaxRetry	Retry sending indications, till no communication is detected (default 3 times).

10 Mechanical Specifications

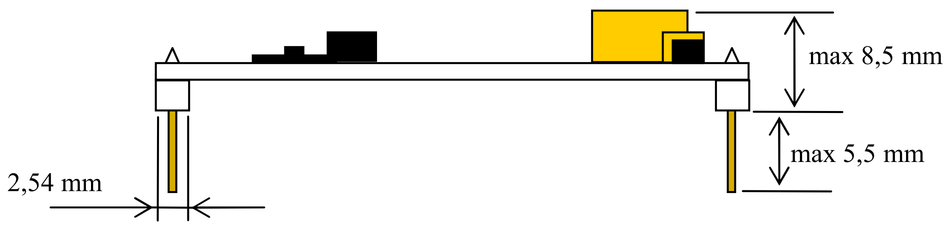


Figure 1: Side view

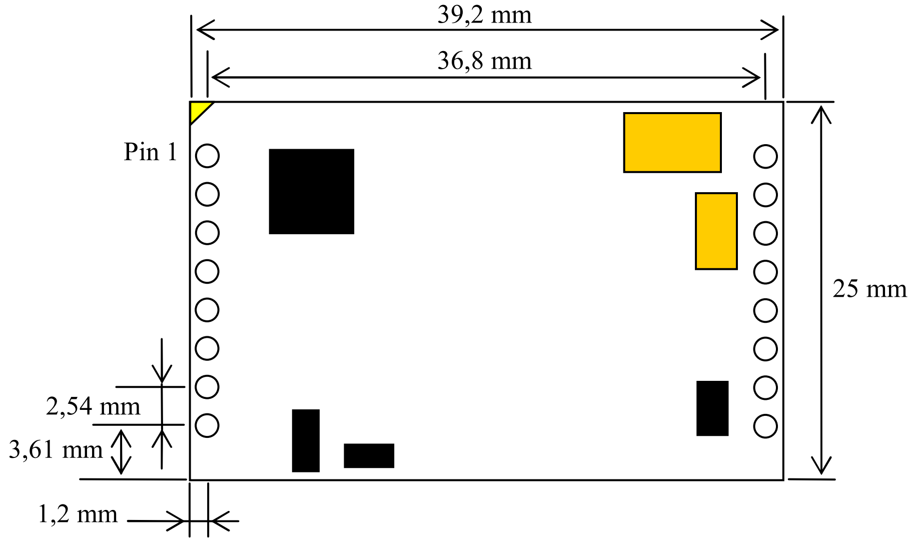


Figure 2: Top view

Figure 3: Bottom view

11 Glossary

Communication Objects
See Group Communication Object

Group address (GA)
Group addresses are used to link group communication objects.
See Group Communication Object

Group Object
See Group Communication Object

Group Communication Objects
Group communication objects contains the data points which are transmitted via runtime communication. One or more group addresses can be assigned to group communication objects. Always on group address is the sending address. Via this address the values of the group communication object are sent on the bus. The other group addresses are only used to receive values.
The sending group address is set via the command 34h or ETS. If the sending group address is deleted, the next group address becomes the sending one.
Other words for group communication object are

- group object
- communication objects

Physical Address
This address is the unique device address inside a KNX-System. This address is independent of the group addresses and is used for configuration of the device.