



IP/ KNX Convertor

User manual-Ver.1

BTIC-01/00.1

Contents

| | |
|--|-----------|
| 1. Summary | 3 |
| 2. Technical Properties & Dimension and Circuit Diagram | 4 |
| 2.1 Technical data | 4 |
| 2.2 Dimension diagram | 5 |
| 2.3 Circuit diagram | 5 |
| 3. Project design and programming | 6 |
| 3.1 Overview of the functions | 6 |
| 3.2 Parameter setting description in the ETS | 7 |
| 3.2.1. Parameter window "IP settings" | 7 |
| 3.2.2. Parameter window "IP address" | 8 |
| 4. Communication Protocol | 10 |
| 4.1 Default Communication Protocol | 10 |
| 4.2 Command ID | 11 |
| 4.3 READ GROUP ADDRESS | 12 |
| 4.4 WRITE GROUP ADDRESS | 14 |
| 4.5 CHANGE THE MONITOR MODE | 16 |

1. Summary

The IP/KNX convertor is designed for an intelligent building control system, which is used to connect the KNX bus and the LAN. It serves as an interface between KNX installations and IP networks. TCP telegrams from the client devices on the LAN can be converted to KNX telegrams via the convertor, and be sent to the KNX bus to control KNX devices.

For convenient to control the KNX/EIB devices by a LAN devices, we define a communication protocol which is transmitted on the data field of the IP packet transferred by the IP/KNX convertor in the chapter 4 of this manual. The users can configure or program the IP network devices according to the protocol to achieve the control for KNX devices.

The IP/KNX convertor has two working modes: the monitor mode and the filter mode. In the monitor mode, all active telegrams can be forwarded. In the filter mode, only the telegrams those with a group address in the filter table in the IP/KNX convertor will be forwarded to the other side.

The filter table can be set via the Filter Table Setting Command from the IP network.

The IP address of the IP/KNX convertor is fixed and can be configured before running. The power supply is not only from the KNX bus, but also need an additional 30V DC supply.

The functions of the IP/KNX convertor are summarized as follows:

- Support TCP/IP protocol, always works in TCP server mode
- Transfer the communication control telegrams transparently
- Support up to 10 TCP connections at the same time
- Transfer the TCP telegram to the KNX/EIB system to control the KNX device
- Monitor the telegrams on KNX bus and transfer them to all the clients connected
- Read respond telegram will only be sent to the initiator
- To read and write any data type which the max length is 14 bytes defined in the KNX standard

This manual provides the technical information about the IP/KNX Convertor as well as assembly and programming in detail for users, and explains how to use the convertor by the application examples.

The IP/KNX convertor is a modular installation device. It can be installed in the distribution board on 35mm mounting rails according to EN 60 715. The device adopts an RJ45 interface to connect with LAN. The network interface can be operated with a transmission speed of 10/100Mbit/s adaptively. The bus connection and auxiliary power supply connection are carried out via EIB bus connection terminals.

The IP/KNX convertor is able to use the Engineering Tool Software ETS (ETS3 or later) which can import a VD4 files to allocate the physical address and set the parameters.

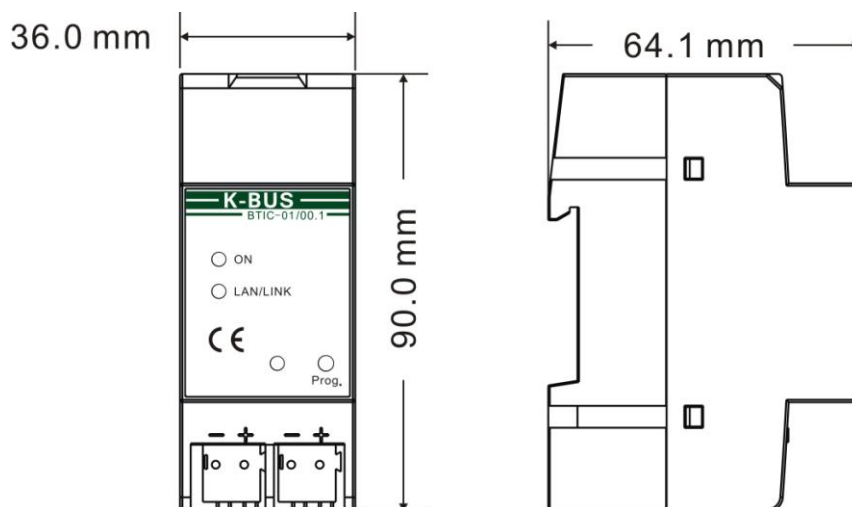
2. Technical Properties, Dimension and Circuit Diagram

2.1 Technical data

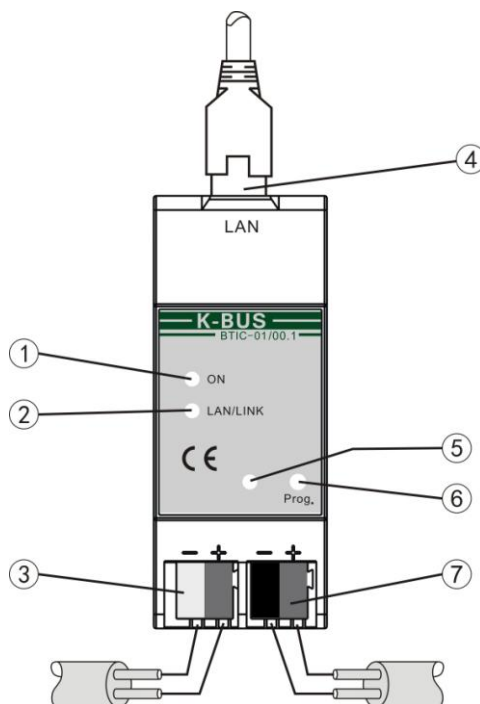
| | | |
|------------------------------|--|---|
| Power supply | Operation voltage | 21-30V DC, via the EIB bus |
| | Current consumption, KNX | <12mA |
| | Power consumption, KNX | <360mW |
| | Auxiliary power supply | 20-30V DC |
| | Auxiliary power consumption | <2.5W |
| Connections | EIB / KNX Interface | Via bus connection terminals (black/red) |
| | Auxiliary power supply | Via bus connection terminals(grey/yellow) |
| | LAN | RJ45 socket for10/100Base-T, IEEE 802.3 network, Adaptively |
| Operating and display | Programming LED and button | For assignment of the physical address |
| | LED Green flashing | The device running indicator |
| | LED ON | Network connection indicator |
| | LAN/LINK LED flashing | Telegram traffic indicator |
| Temperature | Operation | -5 °C ... + 45 °C |
| | Storage | -25 °C ... + 55 °C |
| | Transport | - 25 °C ... + 70 °C |
| Ambient | Humidity | <93%, except condensation |
| Design | Modular installation device, on 35mm mounting rail | |
| Dimensions | 90×36×64mm (H×W×D) | |
| Weight | 0.1kg | |
| Housing, colour | Plastic housing, grey | |

| Application program | Max. number of communication objects | Max. number of group address | Max. number of associations |
|---------------------|--------------------------------------|------------------------------|-----------------------------|
| IP/KNX convertor | 0 | 0 | 0 |

2.2 Dimension diagram



2.3 Circuit diagram



- ① LED ON, indicate that the network connect status.
- ② LAN/LINK LED flashing, indicate that telegram traffic is ongoing.
- ③ Auxiliary power supply connection
- ④ LAN connection RJ45
- ⑤ Programming LED, red for assignment of physical address, green flashes for the device running well.
- ⑥ Programming button
- ⑦ EIB /KNX bus connection

3. Project design and application

3.1 Overview of the functions

◆ Open communication protocols

The communication protocols for IP Interface is open, anyone can follow the text of the protocols to achieve communication between the IP network devices and KNX devices via IP/KNX Converter. So you can easy to connect the KNX/EIB bus system with the IP network into a whole, and ensure the reliability, commonality and scalability of the system internal real-time communication, and really realize the openness of the system.

The IP/KNX convertor always works as a TCP server, support up to 10 TCP connections at the same time.

◆ For receiving and sending communication control telegram

The communication control telegrams are received and sent in accordance to the format which described in the chapter 4, “communication protocol”, i.e. command format. Each data in these command present different meaning. The detail could be seen from the table in the chapter 4.

◆ For converting the telegram

The telegram received from the IP client by the IP/KNX convertor should be generated in strict accordance with the communication protocol format. While receives the telegram from the IP client, the IP/KNX convertor extract the EIB information from the telegram, such as the group address, length, data type and value, etc, and then converted them into the KNX/EIB telegram, and sent to the KNX/EIB system to control the KNX devices.

◆ ACK report and executed result report

When receives a telegram from the IP client, the IP/KNX convertor will check the CRC in the telegram first. If the telegram is correct, the IP/KNX convertor will respond to the IP client via sending an ACK report. If the telegram is not correct, it will not respond. When the telegram is successfully transmitted to the EIB bus, the IP/KNX convertor will send a successful telegram to the IP client. If failure transmitted, it will send a fail telegram to the IP client (see “RESULTPKT” telegram).

◆ Bus monitor and filter functions

The IP client can monitor the data on the EIB bus via the IP/KNX convertor. The IP/KNX convertor also provides filter function to group address. When the filter function is set, the IP/KNX convertor only forwards the telegram which group address is in the filter table to the EIB bus, and reverse. The filter table can be modified or set by the IP client device. The Maximum number of

group address in the filter table is 254. General, the bus monitor mode is suitable for most of the users.

◆ Reading and writing data type

The data telegram that the length is less than or equal to 14 bytes can be sent to or read from the EIB bus via the IP/KNX convertor.

The data which is read from the EIB bus is only forwarded to the client of the read request initiated.

3.2 Parameters setting description for the ETS

3.2.1. Parameter window “IP settings”

Parameter window “IP setting” is shown in fig. 3.2. Here can set the name and port number of the IP/KNX Convertor.

Because IP convertor and RS485 convertor use the same product database, so before setting IP convertor parameters, you must select devices first, as follow (fig.3.1):

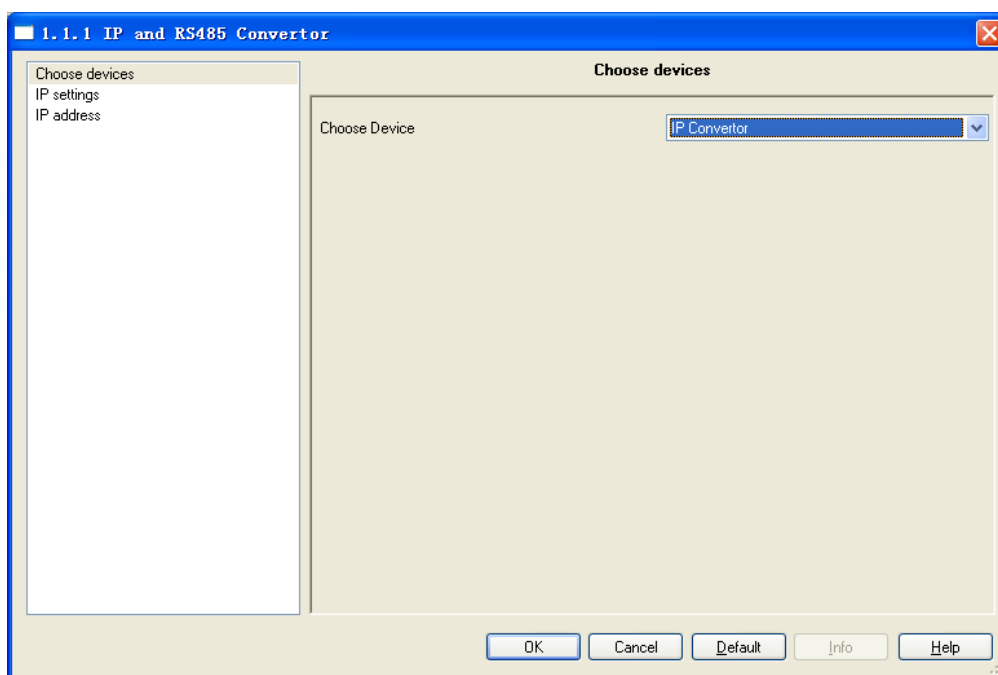


Fig 3.1 Choose device

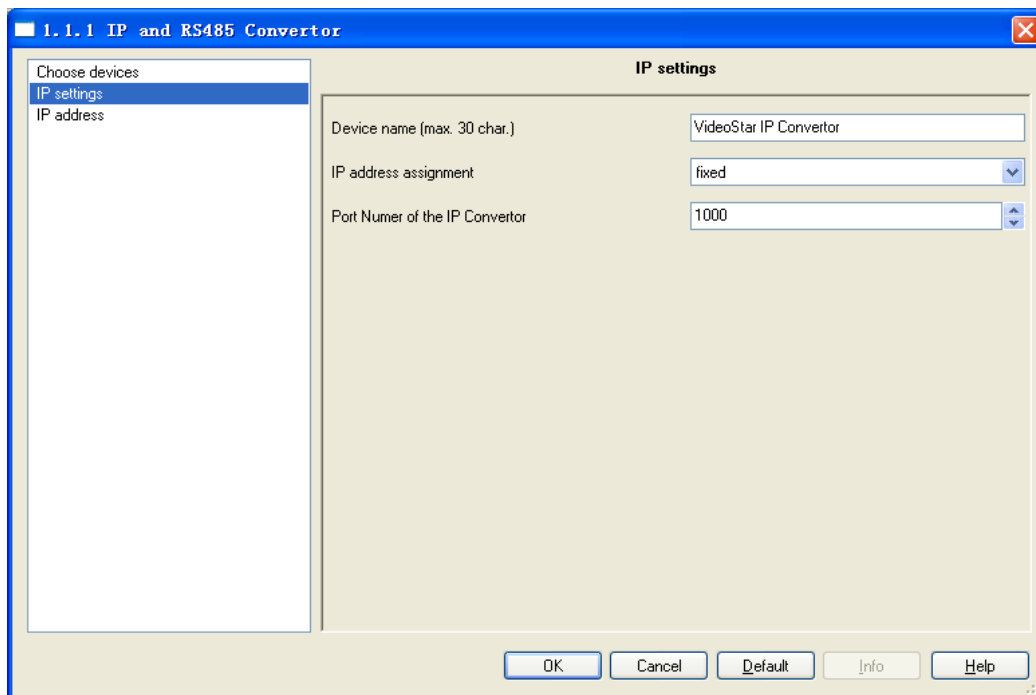


Fig.3.2 parameter window “IP settings”

Parameter “Device name (Max. 30 Char.)”

This parameter defines the name of the IP convertor to identify the device on the LAN. Here can entry a maximum of 30 characters in length. It is a spare parameter in current version.

Parameter “IP address assignment”

This parameter indicates that the IP address assignment of the convertor is fixed. The IP address can be assigned in the follow parameter window “IP address”.

Parameter “Port Number of the IP Convertor”

This parameter defines the port number of the convertor, the port number of client devices must be the same with the convertor, or their communications will fail.

3.2.2. Parameter window “IP address”

Parameter window “IP address” is shown in fig. 3.3. Here set the network connection information, such as IP address, Subnet mask and Default gateway.

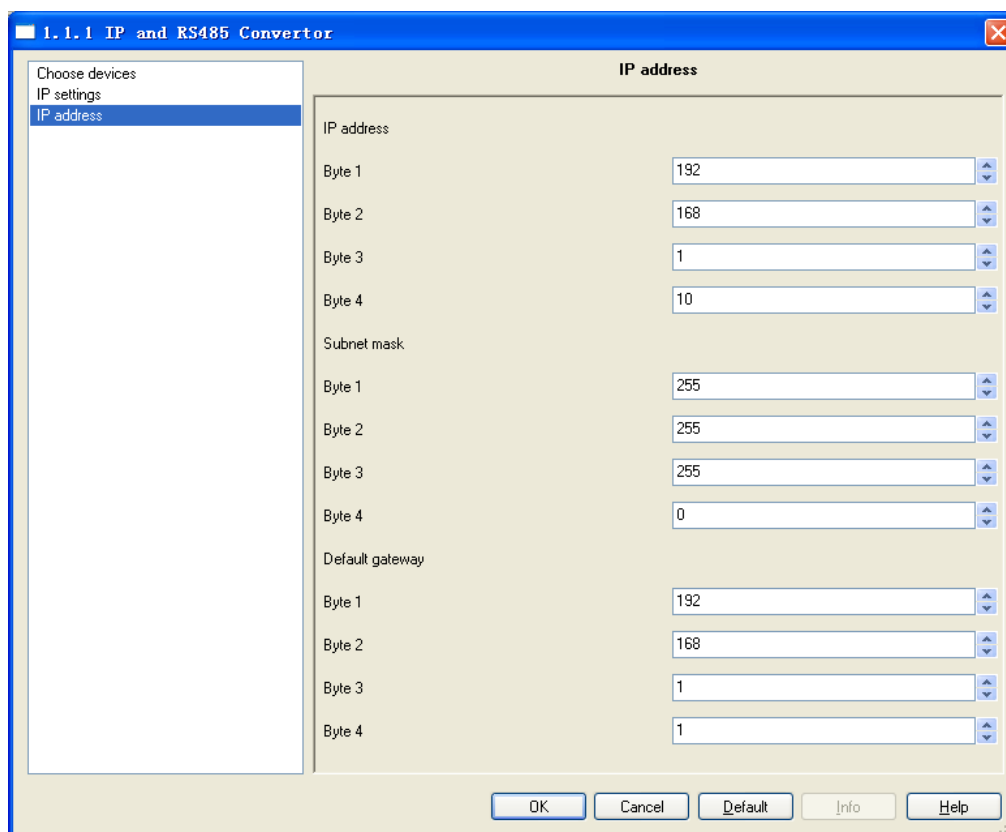


Fig 3.3 parameter window "IP address"

Parameter "IP address"

This parameter defines the IP address of the IP convertor, the IP address is unique on the LAN, to avoid collide. Option:

Byte x: 0...255

The IP address should be entered in a byte-by-byte manner, e.g. as follows for address 192.168.1.10:

Byte1: 192

Byte2: 168

Byte3: 1

Byte4: 10

Parameter "Subnet mask"

This parameter defines the subnet mask of the IP convertor. The subnet mask must be set to reflect the number and structure of the subnet. Option:

Byte x: 0...255

In a small network the subnet mask 255.255.255.0 should be set as follows:

Byte1: 255

Byte2: 255

Byte3: 255

Byte4: 0

Parameter “default gateway”

This parameter defines the default gateway. The default gateway must be the same network segment with the IP address. Option:

Byte x: 0...255

The default gateway 192.168.1.1 should be set as follows:

Byte1: 192

Byte2: 168

Byte3: 1

Byte4: 1

4. Communication Protocol**4.1 Default Parameters**

IP address: 192.168.1.10, port number: 1000, TCP Server mode, these mean that the device does not need to configure via ETS and can be used directly after leave factory if the client information match with it.

Command packet

```
typedef struct {
    BYTE    Head1;        // Always is 0x55=STX1.
    BYTE    Head2;        // Always is 0xAA=STX2.
    WORD    MachineID;    // Not use, default to 0.
    WORD    Reserved;    // Reserved
    WORD    Command;      // Command ID
    DWORD   Length;      // Input Parameter 1
    WORD    InParam;      // Input Parameter 2
    WORD    ChkSum;      // Checksum
} CMDPKT;                // Size of command packet is 16 Bytes.
```

Command acknowledge packet

```
typedef struct {
    BYTE    Head1;        // Always is 0x5A=STX1.
    BYTE    Head2;        // Always is 0xA5=STX2.
    WORD    MachineID;    // Not use, default to 0.
    WORD    Response;     // 0x00-NAK, 0x01-ACK.
    WORD    ChkSum;      // Checksum
} ACKPKT;                // Size of command acknowledge packet is 8 Bytes.
```

Command execution result packet

```
typedef struct {
    BYTE    Head1;        // Always is 0xAA=STX1.
    BYTE    Head2;        // Always is 0x55=STX2.
    WORD    MachineID;    // Not use, default to 0.
    WORD    Reserved;     // Reserved
    WORD    Ret;          // 1: OK 0: Error
    DWORD   OutParam;     // Output Parameter
    WORD    ChkSum;       // Checksum
} RESULTPKT;             // Size of command execution result packet is 14 Bytes.
```

Instruction:

- A. The IP client send command using format of CMDPKT to the convertor. When the command is arrived, the IP/KNX convertor will reply an ACK using the format of ACKPKT to the IP client, if the IP client does not receive the ACK after a time, it will be considered that the transmission is failure and decide whether to re-send.
- B. After the command was executed, the convertor will report the execute result using the format of structure RESULTPKT.
- C. When write a Group Address Value which has more than 2 bytes Object Value, the Object Value should be sent in another data packets, the Setting Data Command, to the Convertor.
- D. When read a Group Address, need to ensure that the Communication Object has the read/write/transmit enable. All the responds will be sent to the client in data packets.
- E. The protocol for monitor mode and filtering mode is not suitable for all users now. We will provide them in another document when need.
- F. You should know the IP address and the port number of the IP Converter and have to link to it with TCP protocol before using it. The IP Converter could be configured by ETS Tools!
- G. The IP/KNX convertor always works as TCP server. In the monitor mode, the telegrams received from the KNX bus will be sent to all the clients that have connected to IP/KNX convertor. But the read respond telegram will only be sent to the initialize.
- H. Only 10 TCP connections allowed at a time. Do not support UDP mode temporary.

4.2 Command ID

```
enum // The command ID for the connection.
{
    CMD_SET_MONITOR_MODE = 0x101,
    CMD_SET_FILTER_MODE,
    CMD_SET_RANDOM_MODE,    //This command is not available now
    CMD_GROUP_ADDR_READ,
    CMD_GROUP_ADDR_WRITE
};
```

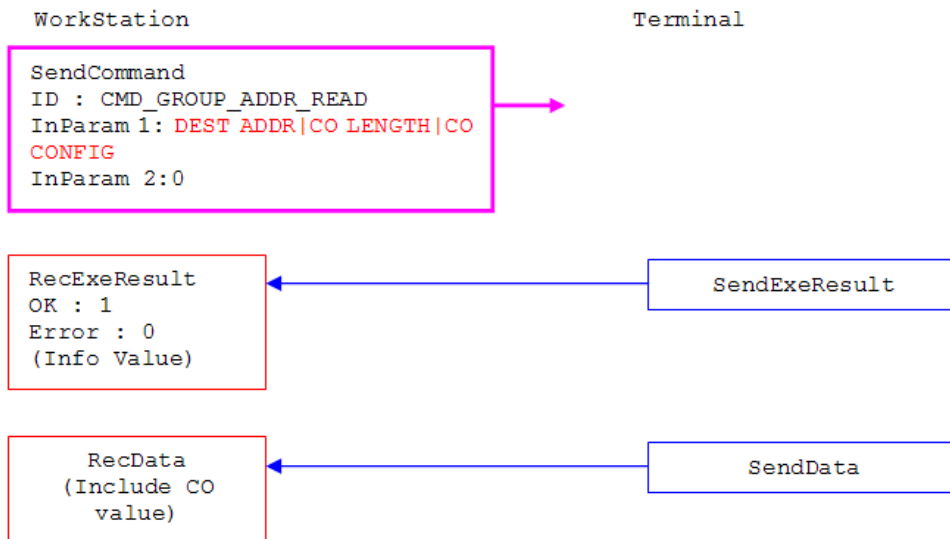
4.3 BOOL ReadGroupAddr(CMDPKT* pCMD)

Function: Read the Group Address

Parameters:

Return: 1—OK;

0—Error;



PROTOCOL:

| COMMAND(Client----IP/KNX Convertor) | | |
|-------------------------------------|---|------------------------------|
| Byte | Value | Meaning |
| 1 | 0x55 | Always is 0x55 |
| 2 | 0xAA | Always is 0xAA |
| 3 | 1--255, 0 = default | Not use, default to 0. |
| 4 | | |
| 5 | 0 | Reserved |
| 6 | | |
| 7 | CMD_GROUP_ADDR_READ | Command ID |
| 8 | | |
| 9 | nConfig ^(NOTE 1) (We used the lowest 2 bits as the priority, usually it equals 0x03) | Input Parameter 1 |
| 10 | nLength ^(NOTE 2) | |
| 11 | DEST_ADDR ^(NOTE 3) | |
| 12 | | |
| 13 | 0 | Input Parameter 2 |
| 14 | | |
| 15 | ChkSum | Checksum ^(NOTE 4) |
| 16 | | |

| ACK(IP/KNX Convertor ---- Client) | | |
|---|----------------------------------|------------------------------------|
| Byte | Value | Meaning |
| 1 | 0x5A | Always is 0x5A |
| 2 | 0xA5 | Always is 0xA5 |
| 3 | 1--255, 0 = default | Not use, default to 0. |
| 4 | | |
| 5 | 0x00-NAK | Response |
| 6 | 0x01-ACK. | |
| 7 | ChkSum | Checksum ^(NOTE 4) |
| 8 | | |
| EXERESULT (IP/KNX Convertor ---- Client) | | |
| Byte | Value | Meaning |
| 1 | 0xAA | Always is 0xAA |
| 2 | 0x55 | Always is 0x55 |
| 3 | 1--255, 0 = default | Not use, default to 0. |
| 4 | | |
| 5 | 0 | Reserved |
| 6 | | |
| 7 | 1: OK 0: Error | Result |
| 8 | | |
| 9 | 0 | Output Parameter |
| 10 | | |
| 11 | | |
| 12 | | |
| 13 | ChkSum | Checksum ^(NOTE 4) |
| 14 | | |
| Data respond(IP/KNX Convertor ---- Client) | | |
| Byte | Value | Meaning |
| 1 | 0xA5 | Always is 0xA5 |
| 2 | 0x5A | Always is 0x5A |
| 3 | 1--255, 0 = default | Not use, default to 0. |
| 4 | | |
| 5 | SRC ADDR ^(NOTE 5) | The Physical Address of the source |
| 6 | | |
| 7 | DEST ADDR ^(NOTE 3) | Destination Group Address |
| 8 | | |
| 9 | nLength | The Length of the Value |
| 10 | // Group communication Multicast | APCI of EIB telegram |
| 11 | APCI_VALUE_READ 0x0000 | |
| | APCI_VALUE_WRITE 0x0080 | |
| | | 1~14 Bytes of Value |

| | | |
|----|--------|------------------------------|
| 19 | ChkSum | Checksum ^(NOTE 4) |
| 20 | | |

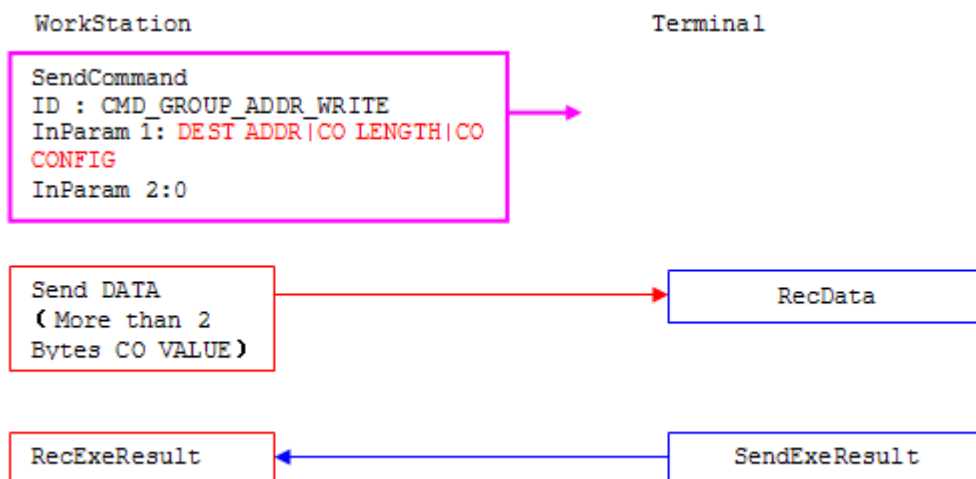
4.4 BOOL WriteGroupAddr(CMDPKT* pCMD)

Function: Write the Group Address

Parameters:

Return: 1—OK;

0—Error;



PROTOCOL:

| COMMAND(Client----IP/KNX Convertor) | | |
|-------------------------------------|---|------------------------------|
| Byte | Value | Meaning |
| 1 | 0x55 | Always is 0x55 |
| 2 | 0xAA | Always is 0xAA |
| 3 | 1--255, 0 = default | Not use, default to 0. |
| 4 | | |
| 5 | 0 | Reserved |
| 6 | | |
| 7 | CMD_GROUP_ADDR_WRITE | Command ID |
| 8 | | |
| 9 | nConfig ^(NOTE 1) (We used the lowest 2 bits as the priority, usually it equals 0x03) | Input Parameter 1 |
| 10 | nLength ^(NOTE 2) | |
| 11 | DEST ADDR ^(NOTE 3) | |
| 12 | | |
| 13 | (nLength <= 2) : Group Value | Input Parameter 2 |
| 14 | Else : 0 | |
| 15 | ChkSum | Checksum ^(NOTE 4) |

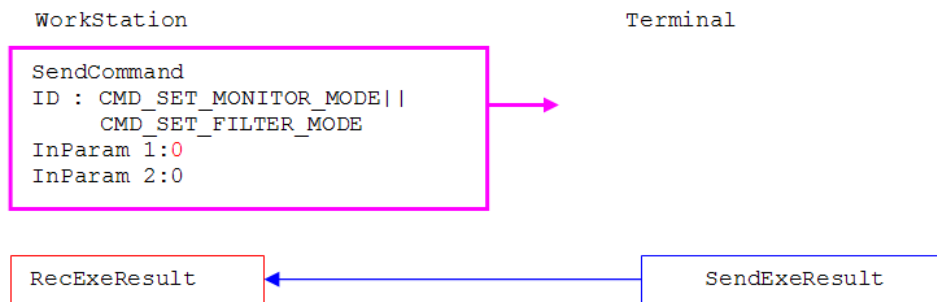
| 16 | | |
|---|---------------------|---|
| ACK(IP/KNX Convertor ---- Client) | | |
| Byte | Value | Meaning |
| 1 | 0x5A | Always is 0x5A |
| 2 | 0xA5 | Always is 0xA5 |
| 3 | 1--255, 0 = default | Not use, default to 0. |
| 4 | | |
| 5 | 0x00-NAK | Response |
| 6 | 0x01-ACK. | |
| 7 | ChkSum | Checksum ^(NOTE 4) |
| 8 | | |
| EXERESULT (IP/KNX Convertor ---- Client) | | |
| Byte | Value | Meaning |
| 1 | 0xAA | Always is 0xAA |
| 2 | 0x55 | Always is 0x55 |
| 3 | 1--255, 0 = default | Not use, default to 0. |
| 4 | | |
| 5 | 0 | Reserved |
| 6 | | |
| 7 | 1: OK 0: Error | Result |
| 8 | | |
| 9 | 0 | Output Parameter |
| 10 | | |
| 11 | | |
| 12 | | |
| 13 | ChkSum | Checksum ^(NOTE 4) |
| 14 | | |
| Setting Data(Client----IP/KNX Convertor) | | |
| Byte | Value | Meaning |
| 1 | 0x5A | Always is 0x5A |
| 2 | 0xA5 | Always is 0xA5 |
| 3 | 1--255, 0 = default | Not use, default to 0. |
| 4 | | |
| 5 | | 3~14 Bytes of Value(nLength >= 3) : Group Value |
| 6~ | | |
| 17 | | |
| | | |
| | | |
| | | |
| 18 | ChkSum | Checksum ^(NOTE 4) |
| 19 | | |
| 20 | | |

4.5 BOOL ComMonitorModeChange(CMDPKT* pCMD)

Function: Change the monitor mode;

Parameter:

Return: 1 -- OK;
0 -- Error;



If Setting the CMD_SET_FILTER_MODE Mode, then the filter table should be send in 3 seconds after the command.



PROTOCOL:

| COMMAND(Client----IP/KNX Convertor) | | |
|-------------------------------------|---|------------------------------|
| Byte | Value | Meaning |
| 1 | 0x55 | Always is 0x55 |
| 2 | 0xAA | Always is 0xAA |
| 3 | 1--255, 0 = default | MachineID |
| 4 | | |
| 5 | 0 | Reserved |
| 6 | | |
| 7 | CMD_SET_MONITOR_MODE | Command ID |
| 8 | CMD_SET_FILTER_MODE | |
| 9 | CMD_SET_MONITOR_MODE: 0 CMD_SET_FILTER_MODE: (1~254),the length of filter table. | Input Parameter 1 |
| 10 | | |
| 11 | | |
| 12 | 0 | Input Parameter 2 |
| 13 | | |
| 14 | ChkSum | Checksum ^(NOTE 4) |
| 15 | | |
| 16 | | |
| ACK(IP/KNX Convertor ---- Client) | | |
| Byte | Value | Meaning |
| 1 | 0x5A | Always is 0x5A |

| 2 | 0xA5 | Always is 0xA5 |
|--|--------------------------------|--|
| 3 | 1--255, 0 = default | MachineID |
| 4 | | |
| 5 | 0x00-NAK | Response |
| 6 | 0x01-ACK. | |
| 7 | ChkSum | Checksum ^(NOTE 4) |
| 8 | | |
| EXERESULT(IP/KNX Convertor ---- Client) | | |
| Byte | Value | Meaning |
| 1 | 0xAA | Always is 0xAA |
| 2 | 0x55 | Always is 0x55 |
| 3 | 1--255, 0 = default | MachineID |
| 4 | | |
| 5 | 0 | Reserved |
| 6 | | |
| 7 | 1: OK 0: Error | Result |
| 8 | | |
| 9 | 0 | Output Parameter |
| 10 | | |
| 11 | | |
| 12 | | |
| 13 | ChkSum | Checksum ^(NOTE 4) |
| 14 | | |
| Setting Filter Table(Client----IP/KNX Convertor) | | |
| Byte | Value | Meaning |
| 1 | 0x5A | Always is 0x5A |
| 2 | 0xA5 | Always is 0xA5 |
| 3 | 1--255, 0 = default | MachineID |
| 4 | | |
| | High Byte of the Group Address | Max Data:254*2 Bytes(The Lower Byte of the Group Address follows the higher Byte, and they should be marked in ascending order, for example, Group Address 00-05 should be placed before 00-06, also 00-05 should be placed before 01-05). |
| | Low Byte of the Group Address | |
| | High Byte of the Group Address | |
| | Low Byte of the Group Address | |
| | ...High Byte | |
| | ...Low Byte | |
| | ...High Byte | |
| ...Low Byte | | |
| 19 | ChkSum | Checksum ^(NOTE 4) |
| 20 | | |

NOTE 1: According to KNX Standard, it has 4 types, corresponding to 4 Transmission Priority:

11 = low operational priority

10 = high operational priority

01 = alarm priority

00 = system priority

Usually we use LOW priority, so the nConfig will be 0x03, others are not recommended.

NOTE 2: The length of the Communication Object which was linked to the DEST ADDR according to the KNX standard, e.g. when the Communication Object less than 1 byte it equals to 1, otherwise it equals to length + 1. That is say, when we have a 1 bit CO, then the nLength should equals to 0x01, while a 1 byte CO will have the nLength equals to 0x02.

NOTE 3: Destination Group Address which you want to read or write.

NOTE 4: The sum of all the data bytes before.

NOTE 5: The Physical Address of the device from whom sends out the telegram.