KNXnet/IP

AMX NetLinx Series Controllers

KNXnet/IP Module

Programmers Manual

Updated September 3, 2013

Index

1 Overview

The KNXnet/IP modules for AMX NetLinx controllers allow integration with the KNX bus using any standard KNXnet/IP or KNXnet/IP routing gateway. The communication module implements KNXnet/IP tunnelling protocol eliminating the need for any specialist gateway types, although such hardware types usually also support the KNX tunnelling protocol and would thus be also supported by the KNXnet/IP module. The module supports all the normal KNX data types required for switching, light dimming, temperature manipulation and date time transmission.

2 Modules

The solution comprises of two modules, KNXIP_BUS, the communication module, and KNXIP_GUI which provides a ready to use set of interfaces for buttons and sliders. The communication component is required and the GUI module is optional should the integrator prefer to write the GUI code. The below sections provide more detail on each module.

2.1 KNXIP_BUS

The KNXIP_BUS module creates and maintains the communication connection with the KNX bus via the KNXnet/IP gateway. The module can receive KNX telegrams translated into an ASCII text format for easy integration with data consumers.

This module may be defined in the following way;

DEFINE_MODULE 'KNXIP_BUS' KNXCOMM (s_chLicenceKey, dvIPPort1, s_chKNX_IP_Address, s_uKNX_IP_Port, vdvKNX_INPUT, vdvKNX_OUTPUT, s_uLINK, s_uTX, s_uRX, s_uERR);

The parameter definitions are;

Char[] Licence Key - Alpha numeric licence key. If an empty string is provided, the module will operate fully for 1 hour in DEMO mode.

Dev IP Port – A reference to the IP device the module will communicate with.

Char[] KNX IP Address – The IPV4 address of the IP gateway e.g. 192.168.1.20

Integer KNX IP Port – the port number specified IP gateway listens on. The default and KNX standard port is 3671.

Dev KNX Input – The virtual device through which KNX telegrams and configuration commands may be sent.

Dev KNX Output – The virtual device though which KNX telegrams are received from the KNX bus.

Char Link – Used for LED status, value is 1 when KNX connection exists.

Char TX – Used for LED status, value blinks 1 on sending of a telegram.

Char RX – Used for LED status, value blinks 1 when receiving a telegram.

Char Err – Used for LED status, value is 1 when an error condition exists.

2.1.1 Sending KNX Telegrams

KNX Telegrams as send as strings in the following format;

Send_string vdvKNX_INPUT, Command

The Command is defined as;

"W" (Write) or "R" (Read)

"0/1/1" (Specimen group address)

"=" (Equals)

"1" (One or On for a switch)

":" (Separator)

"DPT_1BIT" (Data type of group object)

So the above switching telegram will be send as;

send_string vdvKNX_INPUT, "'W0/1/1=1:DPT_1BIT'";

DPT_1BIT, DPT_4BIT, DPT_6BIT, DPT_1BYTE, DPT_2BYTE, DPT_3BYTE types are supported.

See sample file *SAMPLE_STANDALONE.zip* available at www.activesurroundings.com for examples on how the module might be used.

2.1.2 Setting Time and Date Stamps

It is possible to request the module to send periodic time and date stamps to the KNX bus, this is typically used for syncing various devices or displays on the bus.

The below command will cause the module to send a time stamp to the bus on startup and once a minute, possible options are STARTUP DAY HOUR MINUTE SECOND NONE,

```
 SEND_STRING vdvKNX_INPUT, "'TIMESTAMP:11/1/1:STARTUP MINUTE'";
```

The below command will cause the module to send a date stamp to the bus on startup and once an hour, possible options are STARTUP DAY HOUR MINUTE SECOND NONE,

```
SEND_STRING vdvKNX_INPUT, "'DATESTAMP:11/1/2:STARTUP HOUR'";
```

2.1.3 Debug

Debug information may be turned on and off using the below commands. The debug output is available on the AMX controller port 23.

```
SEND_COMMAND vdvKNX_INPUT, "DEBUG ON’";
SEND_COMMAND vdvKNX_INPUT, "DEBUG OFF’";
```

2.2 KNXIP_GUI

The optional KNXIP_GUI module wraps the KNXIP_BUS module to provide ready button functionality with on, off, dim buttons and brightness sliders.

This module may be defined in the following way;

DEFINE_MODULE 'KNXIP_GUI' KNXGUI (vdvGUI, vdvKNX_INPUT, vdvKNX_OUTPUT, s_dvLightButtons, s_uLINK, s_uTX, s_uRX, s_uERR);

The parameter definitions are;

Dev GUI – The virtual device through which the definition commands are sent.

Dev KNX Input – The virtual device through which KNX telegrams and configuration commands may be sent.

Dev KNX Output – The virtual device though which KNX telegrams are received from the KNX bus.

Dev[] LightButtons – Array of panels defined in the project.

Char Link – Used for LED status, value is 1 when KNX connection exists.

Char TX – Used for LED status, value blinks 1 on sending of a telegram.

Char RX – Used for LED status, value blinks 1 when receiving a telegram.

Char Err – Used for LED status, value is 1 when an error condition exists.

Each group and associated GUI structure are defined when the application initializes. Commands are as follows;

The below command defines a switch object to a button with channel code 15, with activation address 0/1/1 and status address of 7/1/1. This is send in the following way

SEND_STRING vdvGUI, "'SWITCH:15:0/1/1:7/1/1'";

The below command defines a dimmer object to a button with channel code 11, with activation address 0/1/1, status address of 7/1/1, dim step address of 1/1/1, brightness value set address and brightness value get address. This is send in the following way

SEND_STRING vdvGUI, "'DIMMER:11:0/1/1:7/1/1:1/1/1:2/1/1:2/1/1'";

The below commands define on only and off only buttons.

SEND_STRING vdvGUI, "'ONONLY:16:0/1/1:7/1/1'";

SEND_STRING vdvGUI, "'OFFONLY:17:0/1/1:7/1/1'";

The below command sets a specific brightness level and maintains status to the defined channel code.

SEND_STRING vdvGUI, "'SETLEVEL:18:2/1/1:7/1/1:128'";

The module supports 3 byte time and date types, which can be decoded into usable data. The below command registers a specified group as a given type so that the correct decode can occur.

The below command defines that 12/1/2, when received from the bus should be decoded as a date.

SEND_STRING vdvKNX_INPUT, "'T12/1/2:PDT_DATE'";

The above command will result in telegram data 0f, 0e, 0c being decoded as 15:30:12

The below command defines that 12/1/1, when received from the bus should be decoded as a time.

SEND_STRING vdvKNX_INPUT, "'T12/1/1:PDT_TIME'";

The above command will result in telegram data 08, 03, 0d being decoded as 2013-03-08.

If no definition for a 3 byte telegram is sent, any 3 byte telegram data will be returned as raw data.

See sample file *SAMPLE_FULL.zip* available at www.activesurroundings.com for examples on how the module might be used.

3 Licencing

Licence keys are supplied as a string value that is passed to the module are one its instantiation parameters. The confirmation of the key check is available from debug at start up. Licence keys are available at www.activesurroundings.com or via email from info@activesurroundings.com.

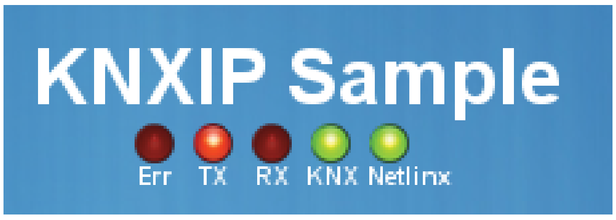
4 Supported IP Gateways

The module implements the KNXnet/IP tunnelling protocol and is designed to be compatible with any IP gateway with proven compatibility with KNX ETS programing software. The module has been tested against BAOS gateways and successfully completed the QA cycle without problems.

Sample Files

Several sample files may be found at www.activesurroundings.com

5 Status LEDS



The sample GUI contains several LED type images representing the states of the communication link, err status, transmission and receive states. These states are representative of the module parameters LINK, TX, RX and ERR. This functionality is entirely optional.

5.1 LED States

The GUI LEDS are provided to give an easy idea of the current status of the KNX driver.

ERR LED: Lit when the connection state is in an error condition. There are several reasons why an error condition exists, but usually it concerns network or connection problems.

TX LED: Lit when a KNX frame is transmitted to the KNX gateway.

RX LED: Lit when a KNX frame is received from the KNX gateway.

LINK LED: Lit true when a connection is being maintained between the KNXnet/IP driver and the KNX gateway.