KNXnet/IP

AMX NetLinx Series Controllers

KNXnet/IP Module

Active Surroundings

March 17, 2013

Index

1 Overview 3
2 Versions 3
3 Licencing 3
4 Supported IP Gateways 3
5 Connection management 4
6 Status LEDS 4
6.1 LED States 4
7 Module definition 5
7.1 Parameters 5
8 KNX Telegrams 6
9 OPTIONAL KNXIP_GUI Module 6
9.1 Parameters 7

1 Overview

The KNXnet/IP module (KNXIP_BUS) for AMX NetLinx controllers facilitates an AMX NetLinx controller to communicate with the KNX bus via most KNX IP gateway devices without the need to add any additional hardware. The module implements the KNXnet/IP tunnelling protocol via a UDP connection with the IP device. The connection is fully managed and requires no code over and above the provision of the initial connection parameters.

The module may be used alone or in conjunction with the KNXIP_GUI wrapper module, this wrapper module provides an easy to use mechanism to add group addresses and implement switching and dimming operations in various ways. Sample code and a sample GUI for both implementation options may be found in the download section at Activesurroundings.com.

2 Versions

Product versions are output to debug as start up. When using the KNXIP_BUS in conjunction with the KNXIP_AMX, the versions should be matched.

3 Licencing

The module shall be licenced for use only on the AMX NetLinx controller specified at the time of purchase. Licence keys may not be cancelled, or re issued for any purpose without prior written agreement from Active Surroundings.

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.

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.

5 Connection management

The module initiates and manages the UDP connection to the IP gateway, the Link parameter indicates when the connection is in session.

The sample code available at Activesurroundings.com should be downloaded in order to see the correct implementation of the modules.

6 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.

6.1 LED States

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.

7 Module definition

The module is created as follows:

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);

7.1 Parameters

chLicenceKey (char[]) – licence key obtained from ActiveSurroundings

dvIPPort1 (dev) – the IP device the module will use for IP communication

s_chKNX_IP_Address (char[]) – IPV4 address of the IP gateway

s_uKNX_IP_Port (integer) – port number to which the gateway is set (KNX standard is 3671)

vdvKNX_INPUT (dev) – The input device to which commands and strings are sent to the KNX module.

An example command would be send_command *vdvKNX_INPUT, "'DEBUG ON'";*

An example string would be send_string vdvKNX_INPUT, "'W00/1/001=1:DPT_1BIT'";

vdvKNX_OUTPUT (dev) – The output device to which commands and strings are received from the KNX module.

uLINK (char) – KNX Link LED

s_uTX (char) – TX LED state

s_uRX (char) – RX LED state

s_uERR (char) – Error LED state

8 KNX Telegrams

The module allows the passing of ASCII string formatted telegrams which are then converted into the KNXnet/IP protocol for transmission to the KNX bus.

The telegram format is “W” or “R” (write or read) followed by a 3 level KNX group address such as “10/1/1” followed by the data type of the telegram. Currently supported data type options are:

DPT_1BIT

DPT_4BIT

DPT_6BIT

DPT_1BYTE

DPT_2BYTE

A typical write telegram would be W10/1/1=1:DPT_1BIT.

A typical read telegram might be R10/1/1.

9 OPTIONAL KNXIP_GUI Module

The optional KNXIP_GUI module is a wrapper module for the core KNXIP_BUS IP communication module. The purpose of this module is facilitate the easy definition of lighting functionality. It is recommended that the sample source code is reviewed for implementation details. The sample code is available in the downloads area as activesurroundings.com

In brief, KNXIP_GUI is defined in the following manner:

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

9.1 Parameters

vdvGUI (dev) – Device to which strings are sent adding the various button types.

vdvKNX_INPUT (dev) - The input device to which commands and strings are sent to the KNX module.

vdvKNX_OUTPUT dev () - The output device to which commands and strings are received from the KNX module.

s_dvLightButtons (dev[]) – Array of touch panel references.

uLINK (char) – KNX Link LED

s_uTX (char) – TX LED state

s_uRX (char) – RX LED state

s_uERR (char) – Error LED state

Add dimmer command format

DIMMER:CHANNEL:ACTIVATION GROUP:STATUS GROUP: DIM GROUP: BRIGHTNESS SET: BRIGHTNESS STATUS

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

Add switch command format

SWITCH:CHANNEL:ACTIVATION GROUP:STATUS GROUP

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

Add on only command format

ONONLY:CHANNEL':ACTIVATION GROUP:STATUS GROUP

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

Add off only command format

OFFONLY:CHANNEL':ACTIVATION GROUP:STATUS GROUP

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

SETLEVEL:CHANNEL':BRIGHTNESS SET:STATUS GROUP

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