NetLinx Module Interface Specification

for the

Bosch
PLENA PLM-2xPx
DSP Power Amplifiers

V1.00

Bosch Communications Systems

BOSCH
Invented for life

TABLE OF CONTENTS

Introduction

This manual describes the use of the PLENA PLM-4Px2x NetLinx Modules designed to provide an interface between an AMX NetLinx system and a single or multiple Bosch PLENA PLM-4P125 and/or PLM-4P220 DSP Power Amplifiers. The interface communicates to the amplifiers via the Ethernet network, utilising UDP datagrams on ports 12128 and 12129.

Scope

The PLENA PLM-4Px2x DSP Power Amplifiers provide 4 channels of amplification, with each channel capable of having a unique mix of the four available line inputs.

The modules allow control and monitoring of the following parameters from the NetLinx system.

Global (all channels)

- Standby / Power
- Global Mute

For individual channels :-

- Channel Output Level
- Channel Output Mute
- Input 1 Mix Level
- Input 2 Mix Level
- Input 3 Mix Level
- Input 4 Mix Level
- Input 1 Mix Mute
- Input 2 Mix Mute
- Input 3 Mix Mute
- Input 4 Mix Mute

In addition, the following status values can be monitored.

- Channel 1 Over-Temperature (Thermal) .
- Channel 1 Fault.
- Channel 2 Over-Temperature (Thermal) .
- Channel 2 Fault.
- Channel 3 Over-Temperature (Thermal) .
- Channel 3 Fault.
- Channel 4 Over-Temperature (Thermal) .
- Channel 4 Fault.

The device discovery functionality of the protocol is not implemented.

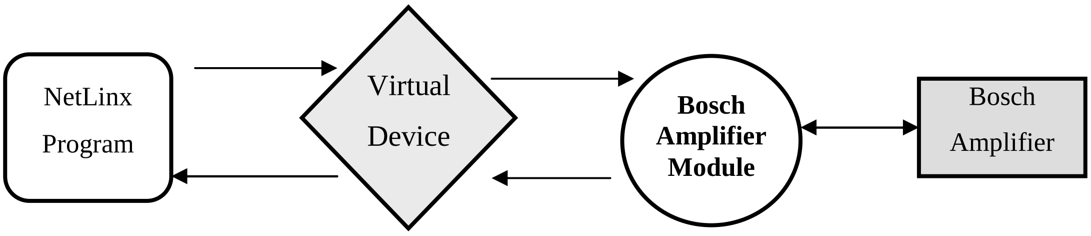
Overview

These are two separate modules, a communication module that handles the communications between the NetLinx system and all amplifiers and an amplifier module that provides access to the various amplifier controls and parameters. A single instance of the communication module is required as well as separate amplifier modules for each amplifier to be controlled. The amplifier communicate to the communications module and hence to the actual hardware via a common virtual device. Each amplifier is assigned a unique ID number (e.g. 1 for the first amp, 2 for the second etc) that is used by communication module to differentiate between the amplifiers.

The amplifiers functions are manipulated by sending commands to the respective amplifier modules. The status responses are return via strings sent from the module..

The setting of an amplifier’s IP address, presets, equaliser and other configuration settings are performed using the “PLM-4Px2x” application from Bosch, available for free download from its website.

The following diagram gives a graphical view of this interface.



A Few Notes of Caution

The Bosch application “PLM-4Px2x” which is used to configure the amplifier utilises the signal monitoring features of the protocol to display a real time level of each of the channel outputs. These monitoring packets are transmitted via UDP broadcasts and, as such, will be received and decoded by the communication module (although they are not passed on to the respective amplifier module). If multiple instances of the application are running on the LAN and connected to different amplifiers, the amount of broadcasted traffic may swamp the module and cause the NetLinx system to become slow and unresponsive.

In addition, ensure that your NetLinx has the latest firmware installed. Some earlier versions of firmware limited the size of received UDP packets to a maximum of 256 bytes which is less than the size of some of the packets sent from the amplifiers.

Implementation

To interface to the PLENA amplifiers, the programmer must do each of the following:

1. Define the virtual device for the communication module. NetLinx virtual devices start with device number 33001.
2. Define a virtual device for each amplifier module.
3. Define a character string variable for each amplifier module that specifies the actual hardware's IP address.
4. Define an integer variable for each amplifier that defines its unique ID.
5. Define an integer variable that contains the base port number for the UDP ports, The communication modules requires one port (0:<base port>:0) and each amplifier module also requires one port (0:<base port + id>:0.
6. Copy the "PLENA PLM Comms.tko" file into the same folder as the main program .axs file or into the shared TKOs folder.
7. Copy the "PLENA PLM-4Px2x.tko" file into the same folder as the main program .axs file or into the shared TKOs folder.
8. Define the modules within the main program with a DEFINE_MODULE command. .

An example of how to do this is shown below.

```
DEFINE_DEVICE

dvTP                        = 10001:1:0      // TOUCH PANEL
vdCOMMS                     = 33001:1:0      // AMPLIFIER COMMS
vdAMP1                      = 33002:1:0      // AMP #1
vdAMP2                      = 33003:1:0      // AMP #2

DEFINE_VARIABLE

CHAR         AMP1_ADDR[] = '192.168.0.31'  // IP ADDRESS OF AMP #1
CHAR         AMP2_ADDR[] = '192.168.0.32'  // IP ADDRESS OF AMP #2
INTEGER      AMP1_ID     = 1               // ID OF AMP #1
INTEGER      AMP2_ID     = 2               // ID OF AMP #2
INTEGER      BASE_PORT   = 10              // USES 0:10:0 UPWARDS

DEFINE_START

// Put this between the DEFINE_START and DEFINE_EVENT sections

DEFINE_MODULE 'PLENA Comms' COMMS1 (vdCOMMS, BASE_PORT)
DEFINE_MODULE 'PLENA PLM-4Px2x' AMP1 (vdAMP1, vdCOMMS, AMP1_ADDR, AMP1_ID)
DEFINE_MODULE 'PLENA PLM-4Px2x' AMP2 (vdAMP2, vdCOMMS, AMP2_ADDR, AMP2_ID)

DEFINE_EVENT

DATA_EVENT [vdAMP1]
{
   STRING :
   {
      // DECODE RESPONSES FROM AMPLIFIER
   }
}

…
```

Messages Overview

Amplifier Control Messages

Changes to an amplifier, are made from the main program by using the SEND_COMMAND function to send ASCII commands to the respective amplifier module. Status messages from the amplifier are sent back to the main program via SEND_STRING messages from the module that can be trapped using the STRING event of the DATA_EVENT handler. The following table defines the set of valid control messages.

Message	Format	Description
Power On	POWER:1	Force amplifier into operational mode.
Power Off	POWER:0	Force amplifier into standby mode.
Power Toggle	POWER:T	Switch between operational and standby modes.
Global Mute On	MUTE:1	Mute all channels
Global Mute Off	MUTE:0	Revert channels back to their local mute values.
Global Mute Toggle	MUTE:T	Switch between Global Mute On and Off.
Ping Time	PING_TIME:<value>	Time in 10ths secs between ping requests to try and detect an amplifier that is offline.
Poll Time	POLL_TIME:<value>	Time is 10ths secs between requesting level and status updates from the amplifier,
Trigger Preset	PRESET:<preset>	Trigger a stored preset (1-5)
Set Output Level	OUTPUT:<channel>:<level>	Set the channel output level (0-249)
Set Output Mute	OUTPUT_MUTE:<output><mute>	Mute/unmute the channel output.
Set an Input Mix Level	MIX:<output>:<input>:<level>	Set the mix level for a particular input mix
Set an Input Mix Mute	MIX_MUTE:<channel>:<input>:<level>	Set the mix mute for a particular input
Refresh	REFRESH	Refreshes the names, faults, levels and mutes.

Table 1 Request/Control Messages

Response/Feedback Messages

When changes on an amplifier are detected, the amplifier module will pass this information on to the main program via simple ASCII string messages. The following table defines the set of valid messages.

Message	Format	Description
Power Status	POWER:<status>	Describes the amplifier's current power mode. 0 – Standby 1 - Operational
Amplifier Name	NAME:<name of amplifier>	Describes the amplifier's name.
Online Status	ONLINE:<status>	Describes the online status of the amplifier. 0 – No responses from poll or ping requests. 1 – Amplifier responding to poll requests.
Fault Status	FAULTS:<status>	Describes the fault status of the amplifier. The faults status is a decimal number 0-255 that is bitmapped as follows:- Bit 0 – Channel 1 Thermal Fault. Bit 1 – Channel 1 Shutdown Fault. Bit 2 – Channel 2 Thermal Fault. Bit 3 – Channel 2 Shutdown Fault. Bit 4 – Channel 3 Thermal Fault. Bit 5 – Channel 3 Shutdown Fault. Bit 6 – Channel 4 Thermal Fault. Bit 7 – Channel 4 Shutdown Fault.
Override Status	OVERRIDE:<status>	Describes the current override status of the amplifier. 0 – Amplifier is operating normally. 1 – The override input has been triggered.
Input Names	INPUT_NAME:<input>:<name of input>	Describes the input names of the amplifier..
Output Names	OUTPUT_NAME:<output>:<name of output>	Describes the channel output names of the amplifier.
Global Mute Status	MUTE:<value>	Describes the status of the global mute. 0 – Global Mute not in effect. 1 – Global Mute if effect.
Output Level Status	OUTPUT:<output>:<level>:<mute>	Describes the level and mute status for an output.
Mix Level Status	MIX:<output>:<input>:<level>:<mute>	Describes the level and mute status for a mix crosspoint.
Preset Name	PRESET:<preset>:<valid>:<name of preset>	Describes the name and validity of a preset. 0 – Preset is not valid 1 – Preset is valid

Table 2 Response/Feedback Messages