NetLinx Module Interface Specification
for the
Powersoft M Series Amplifier

IP Control Interface

Programmed for Powersoft by
www.romani-controls.com

TABLE OF CONTENTS

LIST OF TABLES

Revision History

Date	Initials	Revision	Comments
10-01-13	EV	v1.0.0	Initial release of Documentation for V1.0

Introduction

This document describes the software module for NetLinx in use with the Powersoft M Series
series Amplifiers. This module was written using NetLinx Studio version 3.3.1 and **Master Firmware version 4.1.373**.

It will not work on Netlinx Master with firmware below 4.xx

Overview

The Powersoft Amplifier may be controlled using this NetLinx module. This module requires a single
UDP port from a NetLinx Controller to the Amplifier being controlled. For installations containing more
than one amplifier, multiple instantiations of this module may be used. The module implements the
actual Powersoft protocol for communicating to the unit but exposes a more simplified, NetLinx-friendly
protocol to the programmer.

Implementation

To interface to the Powersoft module, the programmer must do each of the following:

1. Define the NetLinx UDP connection that will be used to communicate to the amplifier.
2. Define the virtual device id the module will use to communicate with the main program. NetLinx virtual devices start with device number 33001.
3. This NetLinx module must be included in the program with a DEFINE_MODULE command. This command starts execution of the module and passes in the following key information: NetLinx UDP port id to which the unit is connected and the virtual device id to be used for communicating with the main program. There should only be one instantiation of the module for a given amplifier device.

An example of how to do this is shown below.

```
DEFINE_DEVICE

dvPowerM      =    0: 5: 0            // Powersoft 8 Series Amplifier         Tcp-Ip


// Touch Panels
dvModero1a    = 11001: 1: 0
dvModero2a    = 11002: 1: 0

// Virtual Devices
vdvPowerM     = 33001: 1: 0          // Powersoft  8 Series Amplifier         Virtual



DEFINE_VARIABLE

// General Debug


constant TpCount = 2
volatile dev dvModeroA[TpCount] = { dvModero1a ,dvModero2a}
```

```
// Dynamic TEXT messages

volatile char DspTxtForMsgBox[28][50] =                // TXT MsgBoxs
{

   'Please Wait.....',                                 // Progress Bar Recall Presets
   'Load Preset.',
   '',
   '',
   'Attention !',                                      // MsgBox Module #xx Not Ready
   'Module is NOT Ready.',
   '',
   '',
   'Please Wait until',                                // MsgBox Connection
   'reloading DSP information...',
   '',
   '',
   'Please Wait until',                                // MsgBox Connection
   'the DPS is rebooting...',
   '',
   '',
   'Please Wait!',                                     // MsgBox Wait load preset
   'Preset is loading.',
   'Try later....',
   '',
   'Please Wait!',                                     // MsgBox Wait general
   'DSP is not ready.',
   'Try later....',
   '',
   'Attention!',                                       // MsgBox Device Status
   'DSP do not answer.',
   'Check Power or TCP/IP connection.',
   ''
}




volatile integer DsptxtMsgBox[] =                      // MsgBox TP address channels
{
   3996,
   3997,
   3998,
   3999
}
```

```
volatile char DspMsgBoxPopPages[4][50]=            // MsgBox PopUp Page Name
{
   'PPON-PopUpMSG box',
   'PPON-PopUpAnswerbox',
   'PPON-PopUpProgressBar',
   'PPOF-PopUpProgressBar'
}

volatile integer DspLvlProgressBar = 28            // Remember to change level value on the TP Graphic

volatile char DSP1IpAddress[32] = '192.168.2.30'   // DSP IP address
volatile long DSP1IpPort       =  8002             // DSP Connection Port
volatile long DSP1DiscoveryPort = 30718            // DSP Discovery Port

volatile integer Dsp1ID = 55                       // DSP ID

volatile integer DSP1DisableButton = 4001          // No Function Button
volatile integer Dsp1ConnectBtn = 1000             // 1 = connect , 0 = disconnect
volatile integer Dsp1ConnectToAmp = 1              // 1 = enabled , 0 = disable

volatile integer Dsp1moduleNumber = 2              // Number of DSP Module 1 or 2

volatile integer Dsp1EnterPageBtn = 12
volatile integer Dsp1PollingBtn = 1120

volatile integer Dsp2PollEnable = 1                // 1 or 0

// LEGEND  =  [C] : Channel value  - [A] : Address value – [L] : Level value

volatile integer Dsp1nchControl[]=                 // Main DSP Controls  [C]
{
   1001, // Power On        - Will Power ON if 12V PSU Is Present
   1002, // Power Off        - Will Power OFF if 12V PSU Is Present else the command recall reboot function.
   1003, // Power Toggle  - Will Power TOGGLE if 12V PSU Is Present

   // First Module
   1004, // Led Blinking
   1005, // Mute On
   1006, // Mute Off
   1007, // Mute Toggle
   1008, // SwMute Off Ch1
   1009, // SwMute On Ch1
   1010, // SwMute Off Ch2
   1011, // SwMute On Ch2
   1012, // SwMute Toggle Ch1
   1013, // SwMute Toggle Ch2
```

```
    4001, // Spare
    4001, // Spare
    4001, // Spare
    4001, // Spare

    // Second Module - If Present
    1018, // Led Blinking
    1019, // Mute On
    1020, // Mute Off
    1021, // Mute Toggle
    1022, // Mute Off CH3
    1023, // Mute On  CH3
    1024, // Mute Off CH4
    1025, // Mute On  CH4
    1026, // SwMute Toggle Ch3
    1027, // SwMute Toggle Ch4
    4001, // Spare
    4001, // Spare
    4001, // Spare
    4001, // Spare

// General
    1060, // Device Type
    1061  // Firmware Info
}

// Group MUTE

volatile integer Dsp1Mute[]=                // [C]
{
    1121, // Mute ch1
    1122, // Mute ch2
    1123, // Mute ch3 - If Present
    1124, // Mute ch4 - If Present

    1129, // UnMute ch1
    1130, // UnMute ch2
    1131, // UnMute ch3 - If Present
    1132, // UnMute ch4 - If Present

    1137, // HW Mute ch1&2
    1138, // HW Mute ch3&4 - If Present

    1141, // HW UnMute ch1&2
    1142, // HW UnMute ch3&4 - If Present
}
```

```
// Group Volumes

volatile integer nchDsp1VolUp[]=                    // [C]
{
   1201, // Module 1 Ch.1
   1202, // Module 1 Ch.2
   1203, // Module 2 Ch.1 - If Present
   1204  // Module 2 Ch.2 - If Present
}

volatile integer nchDsp1VolDown[]=                  // [C]
{
   1211, // Module 1 Ch.1
   1212, // Module 1 Ch.2
   1213, // Module 2 Ch.1 - If Present
   1214  // Module 2 Ch.2 - If Present
}

// Gesture Ipad controls
volatile integer nchDsp1ReloadVolume[]=             // [C]
{
   1301, // Module 1 Ch.1/2
   1302  // Module 2 Ch.3/4 - If Present
}

volatile integer lvlDsp1Volumes[] =                 // [L]
{
   11,12,     // TP volume levels
   13,14      // TP volume levels - If Present
}

volatile integer nchDsp1AddressValue[] =            // [C]
{
   1201, // Tp Channel Address for Module #1
   1202, // Tp Channel Address for Module #1
   1203, // Tp Channel Address for Module #2 - If Present
   1204  // Tp Channel Address for Module #2 - If Present
}


volatile float Dsp1VoldBStep = 1.0                       // dB - Start from 0.1 dB


volatile integer txtDsp1Info[] =                         // [A]Address and/or [C]channel Feedback
{
   1001, // Temperature Module #1                        [A]
```

```
    1002, // RMS Current Module #1 CH1              [A]
    1003, // RMS Current Module #1 CH2              [A]
    1004, // Impedence Module #1 CH1                [A]
    1005, // Impedence Module #1 CH2                [A]
    1006, // BitmapProtection Module #1             [A][C]
    1007, // Aux Positive Voltage                   [A]
    1008, // Aux Negative Voltage                   [A]
    4001, // Hardware Mute                          [A][C]
    1010, // Module #1 Ready                        [A][C]
    1011, // External Power                         [A][C]
    1012, // Preset Module #1                       [A]
    1013, // Temperature Module #2                  [A]
    1014, // RMS Current Module #2 CH3              [A]
    1015, // RMS Current Module #2 CH4              [A]
    1016, // Impedence Module #2 CH3                [A]
    1017, // Impedence Module #2 CH4                [A]
    1018, // BitmapProtection Module #2             [A][C]
    1019, // Aux Pos Voltage  Module #2             [A]
    1020, // Aux Neg Voltage  Module #2             [A]
    4001, // Hardware Mute                          [A][C]
    1022, // Module #2 Ready                        [A][C]
    4001, // Spare
    1024, // Preset Module #2                       [A][C]
    4001, // Spare
    4001, // Spare
    1061, // Alarm Mod #1                           [C]
    1062, // Alarm Mod #2                           [C]
    1065, // Bat Former Mod #1                      [C]
    1066, // Bat Former Mod #2                      [C]
    1069, // Device Type                            [A]
    1070, // Firmware Info                          [A]
    4001  // Device Name                            [A]

}
// Hold down channel to edit names
volatile integer nchDsp2Presets[] =           // [A][C]
{
    1314, // Preset #1 - 1
    1315, // Preset #2
    1316, // Preset #3
    1317, // Preset #4
    4001, // Preset Aux

    1328, // Preset #1 - 2 (if Present)
    1329, // Preset #2
    1330, // Preset #3
```

```
    1331, // Preset #4
    4001  // Preset Aux


}

volatile integer nchDsp2PresetControls[] =              // [C]
{
    1401, // Edit Name
    4001, // Spare
    1403  // Abort


}

persistent char Dsp2PresetsName[20][50] =
{
    'Preset #1','Preset #2','Preset #3','Preset #4','Preset Aux', // Module #1
    'Preset #1','Preset #2','Preset #3','Preset #4','Preset Aux'  // Module #2 if Present
}

volatile char Dsp2EditPopUpPageName[1][20] =
{
    'PresetSetup_PopUp'


}

volatile integer dsp2TpMask[2] =                // [A]
{
    1511,1512
}
```

// Only 1 DEFINE_MODULE defined per Amplifier

define_module 'Powersoft M Series rev1mod' AmpDSP1Mld1(dvPowerM,
                                                        dvModeroA,
                                                        vdvPowerM,

                                                        DSP1IpAddress,
                                                        DSP1IpPort,
                                                        DSP1DiscoveryPort,

                                                        Dsp1ID,
                                                        Dsp1nchControl,
                                                        Dsp1moduleNumber,
                                                        Dsp1EnterPageBtn,
                                                        Dsp1PollingBtn,
                                                        Dsp1PollEnable,
                                                        Dsp1Mute,
                                                        nchDsp1VolUp,
                                                        nchDsp1VolDown,
                                                        lvlDsp1Volumes,
                                                        nchDsp1ReloadVolume,
                                                        nchDsp1AddressValue,
                                                        Dsp1VoldBStep,
                                                        nchDsp1Presets,
                                                        nchDsp1PresetControls,
                                                        Dsp1PresetsName,
                                                        Dsp1EditPopUpPageName,

                                                        txtDsp1Info,

                                                        DsptxtMsgBox,
                                                        DspTxtForMsgBox,
                                                        DspMsgBoxPopPages,
                                                        DspLvlProgressBar,
                                                        Dsp1TpMask,
                                                        Dsp1ConnectBtn,
                                                        Dsp1ConnectToAmp,
                                                        Dsp1DisableButton)

See the Powersoft.axs source file for details on the module parameters.

Upon initialization the module will communicate with the Amplifier Device and information will be exchanged.
The Touch panel file supplied with the demo is designed for an iPad equipped with TPControl App.

Command Interface

The interface code will control the amplifier via command events (NetLinx command send_command). These commands will be sent to the module to affect control. Below are the commands supported. Each command processed provides, where possible, asynchronous feedback upon completion of the command.

SEND_COMMAND TO VIRTUAL DEVICE (CMD Format: <[COMMAND]>[VALUE].)

Command	Description
`<DEBUG>x.`	Command to Enable o Disable module debugger (X Parameters) 1 - for DEBUG ON 0 - for DEBUG OFF
`<POWER>x.`	Command to switch ON or OFF the amplifier. (X Parameters) 1 - for switch ON 2 - for switch OFF These commands take effect only if the 12v external power is present.
`<CONNECT>x.`	Command to Connect or Disconnect Device from Master (X Parameters) 1 - for CONNECT 0 - for DISCONNECT
`<PRESET>z-y,x.`	Command to recall presets from Device (X – Y – Z Parameters) **Note:** By default, this is set to off at startup. z = 1 - max TP count (Used for feedback) y = 1 - 4 (Module ID) x = 1 - 4 (Preset number)

Table 1 – Send Command Definitions

Device Notes

Please note that the Amplifier must be equipped with an external 12V DC
power supply in order to enable Power On/Off functions
The Amplifier accept only one connection at the time, so to be able to use
the Powersoft "Armonia" software you must disconnect the Amplifier from AMX with the
above specified button. (DSP1ConnectToDSP)