



NetLinx Module Interface Specification

for the

Powersoft Duecanali & K series Amplifier

IP Control Interface

PRELIMINARY

Programmed for Powersoft by
www.romani-controls.com



TABLE OF CONTENTS

Introduction	2
Overview	2
Implementation	3
Command Interface	10
String Feedback.....	11

LIST OF TABLES

Table 1 – Send Command Definitions	6
Table 2 - String Feedback Definitions	11

Revision History

Date	Initials	Comments
07/30/2012	EV	Initial release of Documentation for V1.0

Introduction

This document describes the software module for NetLinx in use with the Powersoft Duecanali & K series Amplifiers.

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

dvPower1      =      0: 5: 0 // Powersoft K-Class Amplifier

// Touch Panels
dvModerola    = 10001: 1: 0
dvModero2a    = 10002: 1: 0

// Virtual Devices
vdvPower1     = 33002: 1: 0 // Powersoft K-Class Amplifier

DEFINE_VARIABLE

volatile integer txtMsgBox[] =
{
    3996,
    3997,
    3998,
    3999
}

volatile char MsgBoxPopPages[3][20] =
{
    'PopUpMSG box',
    'PopUpAnswerbox',
    'PopUpProgressBar'
}

volatile integer LvlProgressBar = 18

constant TpCount = 2

volatile dev dvModeroA[TpCount] = { dvModerola, dvModero2a}

```

```

volatile char DSP1IpAddress[32] = '192.168.2.22' // Ip Address of Powersoft
volatile long DSP1IpPort = 8002
volatile long DSP1DiscoveryPort = 30718

volatile integer DSP1ID = 00

volatile integer DSP1DisableButton = 4001 // No Function Button
volatile integer DSP1EnterPageBtn = 1 // Button number for entering DSP Page

volatile integer DSP1InchControl[]= // Main DSP Controls
{
    1001, // Power On - Will Power On if 12V PSU Is Present
    1002, // Power Off - Will Power Off if 12V PSU Is Present
    1005, // Power Toggle - Will Power if 12V PSU Is Present

    1003, // Mute On ch1
    1004, // Mute Off ch1
    1011, // Mute On ch2
    1012, // Mute Off ch2
    1015, // Toggle Mute ch1
    1016, // Toggle Mute ch2
    1017 // Toggle Main Mute
}

volatile integer DSP1ConnectToDSP = 1301 // Connect or Disconnect Amplifier

// General Info from Module - * indicate Modero's address and/or channel number -

volatile integer DSP1txtInfo[]=
{
    101, // Pos Aux Voltage
    102, // Neg Aux Voltage
    103, // Aux Analog Voltage
    104, // Main Power Voltage
    105, // Main Power Ampere
    106, // Hub Voltage
    107, // Alim + CH1
    108, // Alim - CH1
    109, // Alim + CH2
    110, // Alim - CH2
    111, // Clock *
    112, // Vaux *
    113, // IGBT *
    114, // BOOST *
    115, // LED *
    116, // Output Attenuation ch1
    117, // Output Attenuation ch2
    118, // HW Mute ch1 *
    119, // HW Mute ch2 *
    120, // Current Temp C°
    121, // Protection ch1 *
    122, // Protection ch2 *
    123, // HW Protection ch1 *
    124, // HW Protection ch2 *
    4001, // AlarmTriggered ch1 *
    4001, // AlarmTriggered ch2 *
    4001, // DSPAlarmTriggered ch1 *
    4001, // DSPAlarmTriggered ch2 *
}

```

```

129, // Presence *
130, // Last ON/OFF
131, // ModlReady *
132, // DeviceON *
133, // Idle CH 1 *
134, // Idle CH 2 *
135, // Presence Signal ch1 *
136, // Presence Signal ch2 *
137, // ProtectionCount
138, // Impedance Value ch1
139, // Impedance Value ch2
140, // Gains ch1
141, // Gains ch2
142, // OutVoltages ch1
143, // OutVoltages ch2
144, // Max Main Ampere
145, // Clip CH1 *
146, // Clip CH2 *
147, // Gate CH1 *
148, // Gate CH2 *
149, // ModCounter
4001, // Boards #1
4001, // Boards #2
4001, // Boards #3
4001, // Boards #4
4001, // Boards #5
155, // InputRouting
156, // IdleTime
157, // DSPModCounter
158, // DSPCRC1
159, // DSPCRC2
160, // DSPCRC0
161, // KAESOPModCounter
162, // KAESOPCRC
163, // Firmware Info
164, // Current Active Preset
165, // Tone IN Alarm Status CH1 *
166, // Tone IN Alarm Status CH2 *
167, // Tone OUT Alarm Status CH1 *
168, // Tone OUT Alarm Status CH2 *
169, // Load Monitor Status CH1 *
170 // Load Monitor Status CH2 *

}
persistent integer DSP1TypeTemp = 0 // 1- Celsius , 0- Fahrenheit
volatile integer DSP1TypeTempBtn = 1305
volatile integer DSP1moduleNumber = 1

volatile integer DSP1PollEnable = 1
volatile integer DSP1PollinStatusBtn = 1303

volatile integer DSP1MaxGains = 2 // Only Single Gains

volatile integer DSP1InchFadersUP[]=
{
    1021, // Attenuation ch1
    1022, // Attenuation ch2

```

```

    1023 // Attenuation ch1&2
}

volatile integer DSPInchFadersDN[]=
{
    1031, // Attenuation ch1
    1032, // Attenuation ch2

    1033 // Attenuation both ch1&2
}

//
volatile integer DSP1lvlFadersLevels[]=
{
    1, // Attenuation ch1
    2, // Attenuation ch2
    3 // Attenuation Main Fader
}

// Set 0 if you want disable automatic Faders Unmute when you move the volumes
volatile integer DSPInchFadersUnmute[]=
{
    1,
    2
}

// Sets input signal routing
volatile integer DSPInchInputRouting[]=
{
    1041, // from analogical source, to output.
    1042, // from analogical source, through DSP, to output
    1043, // from AES3 source CH2, to output
    1044, // from AES3 source CH2, through DSP, to output
    1045, // from AES3 source, to output - ETH
    1046, // from AES3 source, through DSP, to output - ETH
    1047, // from KAESOP source, to output (Not implemented yet)
    1048 // from KAESOP source, through DSP, to output
}

}

volatile integer DSP1MaxPresets = 50 // Max Presets's Number

persistent char DSP1PresetName[50][40] // Presets's Names [MaxPresets][Max Lenght
char = Fixed to 40]

volatile integer DSP1HoldValue = 10 // Hold Time Button for recall change name
persistent integer DSP1CurrentPreset = 0 // Actual Selected Preset

volatile integer DSP1MediaPresetsBtn = 1201 // This button select the Stored Media
Presets (EEPROM - Smart Card)
volatile integer DSP1CurrentMediaPresets = 0 // By default is set to EEPROM Media
Support
volatile integer DSP1GetPresetNameBtn = 1302 // Update Preset Database

```

```

volatile integer DSPlenableChanEdit[TpCount] = // Witch TP is Enabled to Editing
{
    1,1
}

volatile integer DSPlnchFunctionsPreset[]=
{
    1241, // Recall button
    1242, // Store button
    1243 // Delete button
}

volatile char DSP1txtEmptyPreset[] = '- Empty -' // Empty preset name

// There are two graphic formats for recall preset

// Standard page style, all presets button in one page

volatile integer DSPlnchPresets[]=
{
    1071,1072,1073,1074,1075,1076,1077,1078,1079,1080,
    1081,1082,1083,1084,1085,1086,1087,1088,1089,1090,
    1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,
    1101,1102,1103,1104,1105,1106,1107,1108,1109,1110,
    1111,1112,1113,1114,1115,1116,1117,1118,1119,1120
}

// Vertical Scrolling list Presets

volatile integer DSP1MaxPresetPerPage = 5

volatile integer DSP1UpBtn    = 1211
volatile integer DSP1DownBtn = 1212

volatile integer DSPlnchVChannel[] =
{
    1221,1222,1223,1224,1225
}

volatile integer DSP1TxtVChannel[] =
{
    1221,1222,1223,1224,1225
}

volatile integer DSP1TpMaxPages      = 10 // this value indicate the maximum
                                       number of pages
volatile integer DSP1TxtPages        = 1231

```

```

volatile char DSP1TxtForMsgBox[36][50] = // TXT MsgBoxes
{
    'Attention!', // Answer Box Delete Presets
    'Are you Sure to delete',
    'this stored PRESET ?',
    '',

    'Please Waiting.....', // Progress Bar Reload Preset Name
    'Reload Preset Name.',
    '',
    '',

    'Please Waiting.....', // Progress Bar Recall Presets
    'Load Preset.',
    '',
    '',

    'Attention !', // Msg Box for Power ON/OFF/Toggle
    'External 12v power supply',
    'is not present.',
    '',

    'Current Presets', // Msg Box for Store Presets
    'has been stored!',
    '',
    '',

    'Attention !', // Msg Box for Store Presets
    'The presets has not been stored.',
    'Please Check and try again!',
    '',

    'Current Preset', // Msg Box for Delete Presets
    'has been deleted.',
    '',
    '',

    'Attention !', // Msg Box for Delete Presets
    'The presets has not been deleted.',
    'Please Check and try again!',
    '',

    'Attention !', // Msg Box for Reload Presets Database
    'Reload Preset Database.',
    '',
    ''
}

```



```

// Only 1 DEFINE_MODULE defined per Amplifier

define_module 'Powersoft K Series rev1mod' AmpDSP1Mld1(dvPower1,
    dvModeroA,
    vdvPower1,

    DSP1IpAddress,
    DSP1IpPort,
    DSP1DiscoveryPort,

    DSP1ID,
    DSP1InchControl,
    DSP1txtInfo,
    DSP1TypeTemp,
    DSP1TypeTempBtn,
    DSP1moduleNumber,
    DSP1PollEnable,
    DSP1PollinStatusBtn,
    DSP1ConnectToDSP,

    DSP1MaxGains,
    DSP1InchFadersUP,
    DSP1InchFadersDN,
    DSP1lvlFadersLevels,
    DSP1InchFadersUnmute,
    DSP1DisableButton,
    DSP1InchInputRouting,
    DSP1MaxPresets,
    DSP1InchPresets,
    DSP1PresetName,
    DSP1HoldValue,
    DSP1CurrentPreset,
    DSP1MediaPresetsBtn,
    DSP1CurrentMediaPresets,
    DSP1TxtForMsgBox,

    DSP1MaxPresetPerPage,
    DSP1UpBtn,
    DSP1DownBtn,
    DSP1InchVChannel,
    DSP1TxtVChannel,
    DSP1enableChanEdit,
    DSP1EnterPageBtn,
    DSP1TpMaxPages,
    DSP1TxtPages,
    DSP1GetPresetNameBtn,
    DSP1InchFunctionsPreset,
    DSP1txtEmptyPreset,

    txtMsgBox,
    MsgBoxPopPages,
    LvlProgressBar)

```

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.

Command	Description
<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.
<PRESET>XX.	Command to recall the preset number from 1 to 50 (XX Parameters) 1 to 50 for desiderated preset number.
<ROUTING>X.	Command to set Input Signal Routing (X Parameters) 1 - Analogical Source to Output 2 - Analogical Source to Output, through DSP 3 - AES3 Source CH2 to Output 4 - AES3 Source CH2 to Output, through DSP 5 - AES3 Source to Output 6 - AES3 Source to Output, through DSP 7 - KAESOP Source to Output 8 - KAESOP Source to Output, through DSP
<DEBUG>X.	Command to Enable o Disable module debugger (X Parameters) 1 - for DEBUG ON 0 - for DEBUG OFF

Table 1 – Send Command Definitions

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)