

**SIMPLWINDOWS  
NAME:**

None

**CATEGORY:**

Conferencing

**VERSION:**

1.0

**SUMMARY:**

Controls EF1210 functions for gain control for 1 of the following 4 options:

- Codec output gain[GAINC]
- Playback input gain(on aux in input)[GAINP]
- Reference input gain[GAINR]
- Zone output gain[GAINZ]

**GENERAL NOTES:**

Each different ASPI device on the ASPI bus will have a unique unit ID. The module requires unit ID values as parameters. The unit ID has to be the HEX representation of the Unit ID. For example, for a unit ID of 00, the correct parameters on the module would be 30 for UNIT\_ID\_HIGH and 30 for UNIT\_ID\_LOW. For a unit ID of 01, the correct parameters on the module would be 30 for UNIT\_ID\_HIGH and 31 for UNIT\_ID\_LOW.

In addition, this module requires the entry of a channel number. This parameter is entered as the HEX representation of the channel you wish to control. Specifically:

**GAINC**

Channel parameter is not used and should be set to 30.  
GAINP Channel parameter is not used and should be set to 30.

**GAINR**

Channel parameter can be used for reference A or B. For A, set the channel number parameter to 41. For B, set the channel number parameter to 42.

**GAINZ**

Channel parameter can be used for output A or B. For A, set the channel number parameter to 41. For B, set the channel number parameter to 42.

The module uses real feedback from the ASPI unit for all outputs.

The POLL\_BEGIN and POLL\_END can be used to do an initial poll of the ASPI units for their current status. The modules which have these inputs should daisy chain together with the POLL\_END output of the first module triggering the POLL\_BEGIN input of the next module. POLL\_END of the last module does not get attached to another module. See the example program for proper implementation of this function.

The ASPI Serial String Que must be used to ensure that ASPI bus traffic is handled properly. Failure to implement this module may result in improper feedback from the ASPI units. See the example program for proper implementation of this function.

**CRESTRON  
HARDWARE  
REQUIRED:**

CNXCOM-2,  
ST-COM,  
CNXCOM,  
CEN-COM

**SETUP OF CRESTRON  
HARDWARE:**

Tested and verified at the following settings:

Baud Rate - 9600

Parity - None  
Data Bits - 8  
Stop Bits - 1

No Handshaking

**VENDOR FIRMWARE:** 1.01  
**VENDOR SETUP:** None  
**CABLE NUMBER:** CNSP-121

## CONTROL:

<b>UP</b>	D	Raise input volume level
<b>DOWN</b>	D	Lower input volume level
<b>SELECT_GAINC</b>	D	Selects GAINC option as described above
<b>SELECT_GAINP</b>	D	Selects GAINP option as described above
<b>SELECT_GAINR</b>	D	Selects GAINR option as described above
<b>SELECT_GAINZ</b>	D	Selects GAINZ option as described above
<b>POLL_BEGIN</b>	D	Digital trigger used to request an update poll for real feedback status. This only needs to be implemented at program startup a status update is desired
<b>ASPI -RX\$</b>	S	Digital trigger used to request an update poll for real feedback status. This only needs to be implemented at program startup a status update is desired

## FEEDBACK:

<b>LEVEL_FB\$</b>	S	Real feedback text string showing current volume level
<b>POLL_END</b>	D	Digital signal to be looped to the next ASPI module to continue status update request chain
<b>ASPI_TX\$</b>	S	Serial data string to be routed to the TX\$ of a com port

## PARAMETER DESCRIPTIONS:

<b>UNIT-ID-HIGH</b>	P	Hex version of EF1210's upper nibble of the unit ID. For ID 00, use 30. For ID 10, use 31.
<b>UNIT-ID-LOW</b>	P	Hex version of EF1210's lower nibble of the unit ID. For ID 00, use 30. For ID 01, use 31.
<b>CHANNEL_NUMBER</b>	P	See information above for proper usage.

**OPS USED FOR TESTING:** 5.10.11  
**COMPILER USED FOR TESTING:** SimplWindows Version 1.40.07  
**SAMPLE PROGRAM:** EF1210 TEST REV1.SMW  
**REVISION HISTORY:** ASPI EF1210 GAIN4 - Original