Fohhn

Fohhn-Net Device Module

Version	1.0.0
SIMPL Module filename	FohhnNet_Device_1.0.0.umc
SIMPL+ Module filename	FohhnNet_Device_1.0.0.usp
Simpl# Library filename	FohhnNet_Device_CSharp.clz
Tested on processor	CP3, CP4
Tested on processor firmware	1.8001.0146, 2.7000.00083
Tested on device model	Fohhn DI-4.2000
Tested on device firmware	3.2.2
Developed by	Niklas Westman – JaDeVa AB

Summary:

This module integrates with any Fohhn device that uses the Fohhn-Net protocol over UDP or RS485.

Important:

It is not recommended to use Fohhn Audio Soft at the same time as the processor is controlling the device. At the time of writing this module it has been noted that we don't get some responses while Fohhn Audio Soft is connected to the device. This has been noted when using Fohhn Audio Soft over network.

RS-485 settings:

Baud Rate	*Data Bits*	*Stop Bits*	*Parity*	*Handshaking*
19200	8	1	None	None

Release notes:

- 1.0.0
    - Initial release

PARAMETERS	
Connection Type	Defines which communication type should be used when communicating with the device. Available types: - *UDP – IP* - *Serial – RS485*
Device Id	The Device Id of the device to control
Host	The IP / FQDN / Hostname of the device when using UDP. If you want to set this during runtime, you can use serial input **Host** instead.
Port	The port to use when using UDP. If you want to set this during runtime, you can use analog input **Port** instead. *Default: 2101d*
Number of Inputs	The number of inputs on the device that you will be connecting to.
Number of Outputs	The number of outputs on the device that you will be connecting to.
Poll Rate	The time between polls sent by the module. It's recommended to not set this lower than 5s. *Default: 5s*

INPUTS	
Connect	Starts communication with the device. This has to be used for all connection types
Disconnect	Stops communication with the device
Poll_Routing	This polls the device for routing information. This is done automatically when the device starts responding, but after that you will have to do it yourself if you need it. That's because polling for routing requires a large amount of commands to be sent. Polling this affects the following signals: - Output_[X]_Input_[Y]_Current_Gain - Output_[X]_Input_[Y]_Mute_Is_On - Output_[X]_Input_[Y]_Mute_Is_Off
Recall_Preset	Recalls a preset in the device. This takes about 1.5 seconds to complete. *Range: 1 to 100*
Power_On	Takes the device out of standby
Power_Off	Puts the device in standby

Output_[X]_Volume	Controls the volume of output X in decibels. The value should be dB * 10. So -45.5 dB is sent as -455 *Range: -800 to 120*
Output_[X]_Mute_On	Mutes the audio on output X
Output_[X]_Mute_Off	Unmutes the audio on output X
Output_[X]_Mute_Toggle	Toggles the mute state on output X
Output_[X]_Input_[Y]_Gain	Controls the gain of input Y sent to output X. The value should be dB * 10. So -45.5 dB is sent as -455 *Range: -800 to 120*
Output_[X]_Input_[Y]_Mute_On	Mutes input Y on output X
Output_[X]_Input_[Y]_Mute_Off	Unmutes input Y on output X
Output_[X]_Input_[Y]_Mute_Toggle	Toggles the mute state of input Y on output X
Host	The IP / FQDN / Hostname of the device when using UDP. If you want to set this during compile-time, you can use parameter **Host** instead.
Port	The port to use when using UDP. If you want to set this during runtime, you can use analog input **Port** instead.
Serial_Rx$	Commands received from the serial port when using RS485. Connect this to the **Rx$** serial output on a Com-port.
Custom_Command	Send your own custom commands here, and get the response on serial output **Custom_Command_Response**
Log_Level	Used for debugging. This controls how much information is printed to the console. If this module is running on a server-based processor, messages will be written to ErrorLog instead. Available values: 0. Debug 1. Info 2. Errors So if you want all possible logging written to the console, set this value to 0. *Default: 2*

OUTPUTS	
Responding	The device is responding to requests.
Power_Is_On	The device is out of standby
Power_Is_Off	The device is in standby
Device_Alias	The alias set on the device. This will be empty if no custom alias is set.
Firmware_Version	The firmware version of the device. *Example: 3.2.2*
Temperature	The temperature reported by the device in Celsius. This output will report the temperature * 10, so a temperature of 33.5 is reported as 335
Status_Bit_[X]	This contains certain feedback for certain devices. Check ***Appendix A*** at the end of this manual for information on what is reported for your specific device.
Output_[X]_Current_Volume	Reports the volume of output X in decibels. The reported value is dB * 10. So -45.5 dB is reported as -455 *Range: -800 to 120*
Output_[X]_Mute_Is_On	Audio of output X is muted
Output_[X]_Mute_Is_Off	Audio of output X is not muted
Output_[X]_Input_[Y]_Gain	Reports the gain of input Y sent to output X. The reported value is dB * 10. So -45.5 dB is reported as -455 *Range: -800 to 120*
Output_[X]_Input_[Y]_Mute_Is_On	Audio of input Y on output X is muted
Output_[X]_Input_[Y]_Mute_Is_Off	Audio of input Y on output X is not muted
Serial_Tx$	Commands sent to the serial port when using RS485. Connect this to **Tx$** serial input on a Com-port.
Custom_Command_Response	Send your own custom commands on **Custom_Command**, and get the response here

Appendix A

Fohhn devices	Status_Bit_0	Status_Bit_1	Status_Bit_2	Status_Bit_3
DLI-130 DLI-230 DLI-330 DLI-430	Fault	Audio (AES)	Pilotton	
FV-100 FV-200	Fault	Audio (AES)		
LFI-120 LFI-220 LFI-350 LFI-450	Fault	Pilotton		
FMI-100 FMI-110 FMI-400	Fault	Pilotton		
DI-2.2000 DI-2.4000	Protect 1	Protect 2		
DI-4.1000 DI-4.2000	Protect 1	Protect 2	Protect 3	Protect 4
DFM-100 DFM-110 DFM-400	Fault	Audio (AES)	Pilotton	
MA-4.100 MA-4.600	Protect 1	Protect 2	Protect 3	Protect 4