

HDANYWHERE™

mHub control module

For use with.....



Introduction

This module has been designed to control 4x4 and 8x8 HDAnywhere mHub devices, via IP or serial..

mHub Configuration

It is recommended that the mHub is given a static IP address in order to ensure stable communications with this module.

AMX Configuration

Adding the modules to your project

Copy the following files from the zip package to the folder where your project is stored:

HDAnywhere_mHub.tko

Add the above files to your project. Next, add parameter definitions and a `define_module` line to your master source file, similar to the following:

define_device

```
dvNetworking = 0:70:0;
vdvHDAnywhere = 33002:1:0;
```

define_variable

```
char g_deviceAddress[15] = 'x.x.x.x';
char IPorSerial = x;
```

```
define_module 'HDAnywhere_mHub' HDAnywhere_mHub( vdvHDAnywhere,
dvNetworking,
g_deviceAddress,
IPorSerial );
```

Configure Module Parameters

Name	Explanation
VdvHDAnywhere	The virtual HDAnywhere device.
dvNetworking	Networking device used to send data to mHub device.
g_deviceAddress	The IP address of the mHub to control (leave blank - "" - for serial).
IPorSerial	Whether to run over IP or serial communication ("0" for serial, "1" for IP).

Interfacing with the Module

For a complete example of the usage of this module, see the accompanying demonstration project included in the zip package. Usage however can be summarised as mainly involving display state strings. These are strings that state a range of outputs and which input they are set to, terminated by a semi-colon.

For example:

- `1=2; //Output 1 is/is desired to be displaying input 2`

- 1-3=4; //Outputs 1, 2 & 3 are/are desired to be displaying input 4

To switch:

- send_command vdvHDAnywhere, "1-4=2;"; //Sets outputs 1 to 4 to input 2
- send_command vdvHDAnywhere, "SET:5=7;"; //Sets output 5 to input 7

Power:

- [vdvHDAnywhere,1] = 1; //Power on
- [vdvHDAnywhere,1] = 0; //Power off

Beeping:

- [vdvHDAnywhere,2] = 1; //Beeps on
- [vdvHDAnywhere,2] = 0; // Beeps off

Feedback

The module supplies the following feedback, sent as strings to the virtual device.

Command Header	Explanation
<output>=<input>;	<Switch String> is given whenever there is an input/output change.