NetLinx Module Interface Specification

for a

JVC
SR-HD1500US DVR

Professional Control Solutions LLC • 3804 Parchment Circle • Henrico, VA • 23233-7034
804-677-6794 • Info@ProfessionalControlSolutions.com
www.ProfessionalControlSolutions.com

TABLE OF CONTENTS

Overview 3
Channels 4
Command Interface 6
Table 1 – Send Command Definitions String Feedback 6
String Feedback 7
Adding Functions to Modules 8
Commands to the device 8
Additional Feedback from the device 8

LIST OF TABLES

Table 1 – Send Command Definitions 6
Table 2 - String Feedback Definitions 7

Revision History

Date	Initials	Comments
01/30/2011	DC	Initial Release v1.0
03/06/2011	DC	Added additional functions
08/13/2014	DC	Added subtitle language functions and lapsed time

Overview

The AMX module communicates to the JVC SR-HD1500USat 9600, O, 8, 1; without hardware handshaking. The communication cable is a db-9 female with the following connections:

JVC	NetLinx
2 Rx	2 Rx
3 Tx	3 Tx
5 Gnd	5 Gnd (or 1 on NXI Phoenix)

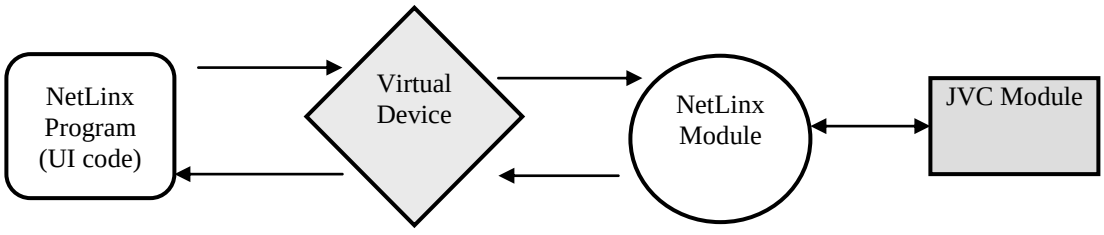
The communication module is called by adding the following line of code:

```
DEFINE_MODULE 'JVC_SRHD1500US_Comm' JVC_code(virtual_name, real_name)
```

This document will define the common NetLinx module interface for a JVC SR-HD1500USplayer. Obviously there will always be features one system supports that another does not (or cannot). The model is not designed to be static. It is designed to be ever-growing while always supporting backwards compatibility. It is up to the programmer of each module to adhere to the model and to find the best way to fit the protocol of a piece of equipment to the model.

For features that are not part of the model the programmer may support additional commands that extend beyond the model to support those features. This is desirable because manufacturers want to expose the features of their system that make them unique and differentiate them from their competitors. Exposing control for those features should be done even if they are not part of the model. In this module, the 'PASSTHRU=' command provides this functionality.

The following diagram gives a graphical view of the interface between the interface code and the NetLinx module.



Channels

The UI module controls the disc device via channel events (NetLinx commands *pulse, on, and off*) sent to the COMM module. The channels supported by the COMM module are listed below. These channels are associated with the virtual device(s) and are independent of the channels associated with the touch panel device.

Note: An '*' indicates an extension to the standard API.

Channel	Description
1	PULSE: Play
2	PULSE: Stop
3	PULSE: Pause
4	PULSE: Next
5	PULSE: Previous
6	PULSE: Scan Forward
7	PULSE: Scan Reverse
8	PULSE: RECORD
20	PULSE: Exit/Cancel
21	PULSE: Enter/Select Button
22	PULSE: Open/Close/Eject
23	ON: Select SD CARD
24	ON: Select HDD
25	ON: Select BLU-RAY/DVD
27	ON: Set Power On
28	ON: Set Power Off
29	PULSE: OK
30	PULSE: Delete
31	PULSE: RED
32	PULSE: GREEN
33	PULSE: BLUE
34	PULSE: YELLOW
41	PULSE: Dub Menu
43	PULSE: Disc Menu
44	PULSE: Menu Button for unit navigation
45	PULSE: Move Menu Cursor Up
46	PULSE: Move Menu Cursor Down
47	PULSE: Move Menu Cursor Left
48	PULSE: Move Menu Cursor Right
49	PULSE: EDIT MODE
55	PULSE: Finalize
61	PULSE: SELECT VIDEO IN
62	PULSE: SELECT SVIDEO IN
63	PULSE: SELECT DV IN
71	PULSE: SELECT REC MODE XP
72	PULSE: SELECT REC MODE SP
80	PULSE: SELECT NO SUBTITLES
81	PULSE: SELECT ENGLISH SUBTITLES
82	PULSE: SELECT FRENCH SUBTITLES
83	PULSE: SELECT SPANISH SUBTITLES
84	PULSE: REQUEST CURRENT TRACK NUMBER
85	PULSE: Request Lapsed Time
240	ON: Record is active – provides feedback only
241	ON: Play is active – provides feedback only

242	ON: Stop is active – provides feedback only
243	ON: Pause is active – provides feedback only
246	ON: Forward is active – provides feedback only
247	ON: Reverse is active – provides feedback only
251	ON: Device is Online – used for feedback only OFF: Device is not Online
252	ON: Data is Initialized – use for feedback only OFF: Data is not Initialized
255	ON: Power on – used for feedback only OFF: Power off

Table 1 - Virtual Device Channel Events

Command Interface

The interface code will control the JVC SR-HD1500USplayer via command events (NetLinx command *send_command)*. These commands will be sent to the module to affect control. Below are the commands supported.

Command	Description
DEBUG=<state>	Set the state of the debugging flag. Note: Setting debug to on will prevent the polling for current status. This will eliminate the feedback. <state>: 0 = off 1 = on 'DEBUG=1'
PASSTHRU=<string>	Allows user the capability of sending commands directly to whatever unit is attached without processing by the NetLinx module. User must be aware of the protocol implemented by the unit to use this command. This gives the user access to features which may not be directly supported by the module. The communication does not add automatically any characters to the passthru string. For more detail please read the Adding Functions to Modules section at the end of this document. <string> : string to send to unit PASSTHRU=THIS IS A COMMAND PASSTHRU=RESET
TITLE=<value>	REQUESTS the Title/Track from the DVD/HDD. Title will be selected and should start to play. TTILE=099
VERSION?	Query for the current version number of the NetLinx module. VERSION?

Table 1 – Send Command Definitions

String Feedback

The NetLinx module will provide feedback to the interface code for JVC SR-HD1500USplayer changes via string events. Below are the strings supported.

String	Description
DEBUG=<state>	State of the debug flag. Non-solicited feedback. <state>: 0 = off 1 = on 'DEBUG=1'
TIME=<value>	Returns the elapsed time for the selected device. (Return from Channel 85 Pulse) TIME=01:20:33
TITLE=<value>	Returns the Title/Track from the DVD. (Return from channel 84 Pluse) TTILE=099 ***NOTE: The unit returns a different title number than the one requested. But the correct title does appear to be playing.***
VERSION=<value>	Reports the current version number of the NetLinx module. <value> : current version number in xx.yy format VERSION=1.06

Table 2 - String Feedback Definitions

Adding Functions to Modules

Commands to the device

This module supplies a mechanism to allow additional device features to be added to software using the module. This is the PASSTHRU command, which allows protocol strings to be passed through the module. The device-specific protocol must be known in order to use this feature.

As an example, suppose that a module for a projector has not implemented the 'white balance adjustment' feature. The command that the projector protocol requires is 03H, 10H, 05H, 14H, followed by a checksum. The documentation for the PASSTHRU command specifies that the module will automatically generate the checksum. In this case, the following string should be sent from the UI code to implement 'white balance adjustment'.

```
send_string vdvDevice,'"PASSTHRU=',$03,$10,$05,$14"
```

The reason to use PASSTHRU instead of sending a protocol string directly to the device port is that the device may require command queuing, calculation of checksums, or other internal processing, which would not be done if the string was sent directly. Because of this, it is best to filter all communication TO the device through the module. (The module documentation will indicate any processing that will be automatically done to the PASSTHRU string like checksum calculation.)

Additional Feedback from the device

The module documentation indicates what feedback is provided. If additional feedback is required, a CREATE_BUFFER for the device must be implemented in the user code to process the strings from the device manually. Note that the module will still be processing the response strings independently and sending the interpreted feedback up to the user code.