



Module Interface Specification

for a

**Generic IR Disc Device**

# TABLE OF CONTENTS

- Control Life Pty Ltd License Agreement.....3
- Introduction .....5
- Overview .....5
- Implementation .....5
- Port Mapping.....6
- Channels .....7
- Command Control.....9
- Command Feedback.....12
- Programming Notes .....12
- Adding Functions to Modules.....13
  - Commands to the device .....13

Date	Initials	Version	Comments
12-11-2007	KN	V1.0.0	Initial release
23-01-2008	KN	V1.0.1	added IR stack control management

## **Control Life Pty Ltd License Agreement**

By using the software included with this End-User License Agreement ("EULA"), you ("Licensee") agree to be bound by, and Control Life Pty Ltd shall be entitled to enforce, this Control Life Pty Ltd License Agreement ("Agreement"). IF YOU DO NOT AGREE, DO NOT USE THE SOFTWARE, YOU MAY RETURN IT TO Control Life Pty Ltd FOR A FULL REFUND.

1. **LICENSE GRANT.** Control Life Pty Ltd grants to Licensee the non-exclusive right to use the Control Life Pty Ltd Software in the manner described in this License. This license does not grant Licensee the right to create derivative works of the Control Life Pty Ltd Software. The Control Life Pty Ltd Software consists of programming and development software, product documentation, sample applications, tools and utilities, and miscellaneous technical information. The Control Life Pty Ltd Software is subject to restriction on distribution described in this License Agreement. **YOU MAY NOT SUBLICENSE, RENT OR LEASE THE Control Life Pty Ltd SOFTWARE.** You may only use the Control Life Pty Ltd Software for its intended purpose. You may not reverse engineer, decompile, or disassemble the Control Life Pty Ltd Software.
2. **INTELLECTUAL PROPERTY.** The Control Life Pty Ltd Software is owned, by Control Life Pty Ltd and is protected by Australian copyright laws, patent laws, international treaty provisions, and/or state of NSW trade secret laws. Licensee may make copies of the Control Life Pty Ltd Software solely for backup or archival purposes. Licensee may not copy any written materials accompanying the Control Life Pty Ltd Software. The Software is licensed, not sold.
3. **TERMINATION.** CONTROL LIFE PTY LTD RESERVES THE RIGHT, IN ITS SOLE DISCRETION, TO TERMINATE THIS LICENSE FOR ANY REASON AND UPON WRITTEN NOTICE TO LICENSEE. In the event that Control Life Pty Ltd terminates this License, then Licensee shall return all copies of the Control Life Pty Ltd Software to Control Life Pty Ltd and certify in writing that all copies have been destroyed.
4. **PRE\_RELEASE CODE.** Portions of the Control Life Pty Ltd Software may, from time to time, as identified in the Control Life Pty Ltd Software, include PRE-RELEASE CODE and such code may not be at the level of performance, compatibility, and functionality of final code. THE PRE-RELEASE CODE may not operate correctly and may be substantially modified prior to final release or certain features may not be generally released. Control Life Pty Ltd is not obligated to make or support any PRE\_RELEASE CODE. ALL PRE-RELEASE CODE IS PROVIDED "AS IS" WITH NO WARRANTIES.
5. **LIMITED WARRANTY.** Control Life Pty Ltd warrants that the Control Life Pty Ltd Software will perform substantially in accordance with any accompanying written materials for a period of 90 days from the date of receipt. CONTROL LIFE PTY LTD DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH REGARD TO THE Control Life Pty Ltd SOFTWARE. THE LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS.

6. LICENSEE REMEDIES. Control Life Pty Ltd's entire liability and your exclusive remedy shall be repair or replacement of the Control Life Pty Ltd Software that does not meet Control Life Pty Ltd's Limited Warranty and which is returned to Control Life Pty Ltd. The Limited Warranty is void if failure of the Control Life Pty Ltd Software has resulted from accident, abuse, or misapplication. Any replacement Control Life Pty Ltd Software will be warranted for the remainder of the original warranty period or 30 days, whichever is longer. Outside of Australia, these remedies may not be available.

7. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. IN NO EVENT SHALL CONTROL LIFE PTY LTD BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR THE INABILITY TO USE THIS Control Life Pty Ltd PRODUCT, EVEN IF CONTROL LIFE PTY LTD HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/COUNTRIES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

This Agreement is governed by the laws of the State of NSW and Australia, and all disputes will be resolved in the courts of Australia.

## **Introduction**

This is a reference manual to describe the interface provided between an AMX NetLinx system and a Disc Device controlled by IR.

This module was written using NetLinx Studio version v2.6 build 2.6.0.191, based on Standard NetLinx API (SNAPI) R.1.9.0

## **Overview**

The module translates between the standard interface described below and the disc device IR protocol.

Some functionality in the device interface may not be implemented in the API interface. In cases where device functions are desired but not API-supported, the PASSTHRU command may be used to send any and all device-protocol commands to the device. See the PASSTHRU command and the [Adding Functions to Modules](#) section for more information.

## **Implementation**

To interface to the Control Life Disc Device module, the programmer must perform the following steps:

1. Define the device ID for the disc device that will be controlled.
2. Define the virtual device ID that the Disc Device IR module will use to communicate with the main program and User Interface. Virtual devices use device numbers 31000 - 32000.
3. The Control Life Disc Device 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: the virtual device ID for communicating to the main program, and the device ID of the disc device to be controlled.
4. Combine the touch panel device with the virtual device. When the channel is pushed, it will then trigger to corresponding function directly.

An example of how to do this is shown below.

```
dvDisc = 5001:4:0
vdvDisc = 31001:1:0
dvTP = 10001:23:0
```

```
DEFINE_COMBINE
(vdvDisc, dvTP)
```

```
DEFINE_START
```

```
define_module CL_DiscDevice_IR_dr1_1_0' mDiscDev1(vdvDisc, dvDisc)
```

## **Port Mapping**

This module uses multiple virtual devices in order distinguish events for one zone from another.

Virtual Device	Channels	Levels	Control	Feedback
31001:1:0	All Channels	All Levels	All Control Cmds	All Feedback Cmds

**Table 1 - Port Mapping**

## **Channels**

The channels supported by the module are listed below. These channels are associated with the virtual device(s).

Note: An ‘\*’ indicates an extension to the standard API.

<b>Channel</b>	<b>Description</b>
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
9	PULSE: Cycle Power
10	PULSE: 0 Digit Button
11	PULSE: 1 Digit Button
12	PULSE: 2 Digit Button
13	PULSE: 3 Digit Button
14	PULSE: 4 Digit Button
15	PULSE: 5 Digit Button
16	PULSE: 6 Digit Button
17	PULSE: 7 Digit Button
18	PULSE: 8 Digit Button
19	PULSE: 9 Digit Button
20	Pulse: +10 Digit Button
21	PULSE: Enter Button
27	PULSE: Set Power On Note: If the device supports discrete power function, IR channel 27 will be pulsed. Otherwise, IR channel 9 will be pulsed. IR will only be pulse when the power status is off
28	PULSE: Set Power Off Note: If the device supports discrete power function, IR channel 28 will be pulsed. Otherwise, IR channel 9 will be pulsed. IR will only be pulse when the power status is on
43	PULSE: Cancel Button
44	PULSE: Menu Button
45	PULSE: Move Menu Cursor Up
46	PULSE: Move Menu Cursor Down
47	PULSE: Move Menu Cursor Left
48	PULSE: Move Menu Cursor Right
50	PULSE: Exit button
51	PULSE: Move Menu Cursor Up Left
52	PULSE: Move Menu Cursor Up Right
53	PULSE: Move Menu Cursor Down Left
54	PULSE: Move Menu Cursor Down Right
55	PULSE: Set Disc Next
56	PULSE: Set Disc Previous
57	PULSE: VIDEO button
63	PULSE: Sleep button
65	PULSE: Function button
66	PULSE: Setup Button
80	PULSE: Clear Button
99	PULSE: Display Button
100	PULSE: Subtitle Button

101	PULSE: Info Button
102	PULSE: Favorite Button
103	PULSE: Continue Button
104	PULSE: Return Button
105	PULSE: Guide Button
112	PULSE: AB Repeat Button
114	PULSE: Title Button
117	PULSE: Angle Button
118	PULSE: Audio Button
120	PULSE: Cycle the disc tray open and closed
124	PULSE: Cycle the random state
125	PULSE: Cycle the repeat state
178	PULSE: Random is on
179	PULSE: Random-all is on
180	PULSE: Random-off is on
181	PULSE: Repeat-disc is on
182	PULSE: Repeat-track is on
183	PULSE: Repeat-all is on
184	PULSE: Repeat-off is on
185	PULSE: Frame Forward
186	PULSE: Frame Reverse
188	PULSE: Slow Forward
189	PULSE: Slow Reverse
192	PULSE: Cycle the scan speed
*238	Tx: Transmitting Data - 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: Scan Forward is active - provides feedback only
247	ON: Scan Reverse is active - provides feedback only
249	ON: Slow Forward is active - provides feedback only
250	ON: Slow 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: Set power on - used for feedback also OFF: Set power off

**Table 2 - Virtual Device Channel Events**

## Command Control

The UI module controls the disc device via command events (NetLinx command *send\_command*) sent to the COMM module. The commands supported by the COMM module are listed below.

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

Command	Description
?DEBUG	Request the state of the debug feature. ?DEBUG
DEBUG-<value>	Set the state of debugging messages in the UI module and the Comm. module. <b>Note:</b> See Programming Notes section.  <value> : 1 = set only error messages on 2 = set error and warning messages on 3 = set error, warning & info messages on 4 = set all messages on  DEBUG-1
?DISCCAPACITY	Get the number of disc slots supported by the device. ?DISCCAPACITY
PASSTHRU-<channel>	Allows user the capability of sending commands directly to whatever unit is attached with minimal processing by the module. User must be aware of the channel numbers of the AMX IR code to use this command. This gives the user access to features that may not be directly supported by the module. For more information, see the " <a href="#">Adding Functions to Modules</a> " section below. <b>Note:</b> Do not send any terminating characters, the module will append them.  <channel> : channel number to send to unit  ``PASSTHRU-1

<p>?PROPERTY-&lt;key&gt;</p>	<p>Get the module properties. If a property is not set, the module will return an empty string.</p> <p>&lt;key&gt;:</p> <p>MODULE-NAME (Name of this module)</p> <p>MODULE-VERSION (Version number of this module. The version is in the format: major.minor.micro)</p> <p>DEVICE-MAKE (The manufacturer name)</p> <p>DEVICE-MODEL (The specific model number of the device being configured.)</p> <p>DEVICE-CATEGORY (The control method used by the device)</p> <p>DEVICE-REVISION (The firmware version installed within the device being used.)</p> <p>DEVICE-CHANNELS (The number of available device channels)</p> <p>DEVICE-LEVELS (The number of available device levels)</p> <p>UI-TEMPLATE (Specifies the type of UI required for a given module.)</p> <p>DEVICE-GUID (This is an abbreviation for Device Global Unique Identification.)</p> <p>MODULE-CONTACTADDRESS( The contact address of Control Life Pty Ltd)</p> <p>MODULE-DESCRIPTION (This is a short description of this module.)</p> <p>MODULE-DOCURL (This is a URL used to document this module.)</p> <p>MODULE-UPDATELOCATION (If the module is ever updated at some later date, this is the location that should be used (if present) to retrieve the updated tko files.)</p> <p>MODULE-COPYRIGHT (The copyright information for this module.)</p> <p>MODULE-VENDOR (description of the vendor: Control Life Pty Ltd)</p> <p>DEVICE-TYPE (Type of Device: LightSystem)</p> <p>SUPPORT-DISCRETE-POWER (does it support discrete power)</p> <p>RECORD-CAPABLE (can it record?)</p> <p>IRONTIME (Pulse Time)</p> <p>IROFFTIME (Interval between two pulses)</p> <p>Example:</p> <p>?PROPERTY-</p>
------------------------------	---

PROPERTY-<key>,<value>	<p>Sets the module property. You must REINITialize the module for the new settings to take affect.</p> <p>&lt;key&gt;:</p> <p>DEVICE-MAKE (The manufacturer name)</p> <p>DEVICE-MODEL (The specific model number of the device being configured.)</p> <p>DEVICE-REVISION (Update the firmware version of the device)</p> <p>&lt;value&gt;: xxx.xxx.xxx</p> <p>SUPPORT-DISCRETE-POWER (does it support discrete power)</p> <p>&lt;value&gt; 1 = yes, 0 = no</p> <p>RECORD-CAPABLE (can it record?)</p> <p>&lt;value&gt; 1 = yes, 0 = no</p> <p>IRONTIME (Pulse Time)</p> <p>&lt;value&gt; 1-100 (in 1/10 sec)</p> <p>IROFFTIME (Interval between two pulses)</p> <p>&lt;value&gt; 1-100 (in 1/10 sec)</p> <p>Example:</p> <p>PROPERTY- DEVICE-REVISION,4.05.00</p>
IOLINK-<dps>,<channel>	<p>where &lt;dps&gt; is the DPS in string form, i.e. 17:1:0, and &lt;channel&gt; is the channel on the IO device to which the power sensor is connected.</p>
?VERSION	<p>Query for the current version number of the module.</p> <p>?VERSION</p>

**Table 3 – Send Command Definitions**

## **Command Feedback**

The COMM module provides feedback to the User Interface module for disc device changes via command events. The commands supported are listed below.

**PLEASE NOTE:** Feedback is only provided when there is a state change. If no state change resulted from the command sent in, then no feedback will be returned.

<b>Command</b>	<b>Description</b>
DEBUG-<value>	Returns the state of debugging messages in the UI module and the Comm. module.  <value> : 1 = set only error messages on 2 = set error and warning messages on 3 = set error, warning and info messages on 4 = set all messages on  DEBUG-1
DISCCAPACITY-<discs>	Reports the number of disc slots that the device supports.  <discs> : 1  DISCCAPACITY-1
PROPERTY-<key> ,<value>	Returns the module properties. If a property is not set, the module will return an empty string. All property keys are case-sensitive.  Example:  PROPERTY-MODULE-VENDOR,Control Life Pty Ltd
VERSION-<version>	Reports the version number of the module.  <version> : x.y.z = module version number  VERSION-1.0.0

**Table 4 - Command Feedback Definitions**

## **Programming Notes**

- At startup and when the 'REINIT' command is used, all values are set to default values. If these values are not initialized during the startup or re-initialization sequence, then they remain set to their default values and may be returned if a query/get command is sent.

## **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 IR code to be passed through the module. The device-specific protocol must be known in order to use this feature.

```
send_command vdvDevice, "PASSTHRU-', $01"
```