



Module Interface Specification

for a

Generic Relay Screen

TABLE OF CONTENTS

- Control Life Pty Ltd License Agreement.....3
- Introduction5
- Overview5
- Implementation5
- Port Mapping.....6
- Channels6
- Command Control.....7
- Command Feedback.....8
- Programming Notes8

Date	Initials	Version	Comments
12-11-2007	KN	V1.0.0	Initial release
20-05-2008	KN	V1.0.1	Removed the duplicated "Real" device, only need two relay ports

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 Screen controlled by a single contact closure.

This module was written using NetLinx Studio version v2.7 build 2.7.0.210, based on Standard NetLinx API (SNAPI) R.1.9.0

Overview

The module translates between the standard interface described below and the disc a screen with contact closure.

Some functionality in the device interface may not be implemented in the API interface.

Implementation

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

1. Define the device ID for the screen that will be controlled.
2. Define the virtual device ID that the Screen module will use to communicate with the main program and User Interface. Virtual devices use device numbers 31000 - 32000.
3. The Control Life Screen 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 screen to be controlled.

An example of how to do this is shown below.

```
DEFINE_DEVICE
dvScreen = 5001:4:0
vdvScreen = 31001:1:0
```

```
DEFINE_CONSTANT
```

```
DEVCHAN ScrRly[2] = {{ dvScreen,1},{ dvScreen,2}} //1=Up, 2=Down
```

```
DEFINE_START
```

```
define_module 'CL_Screen_Relay_dr1_0_1' mScr1(vdvScreen, ScrRly)
```

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.

Channel	Description
2	PULSE: Screen Stop
4	PULSE: Screen Up
5	PULSE: Screen Down
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

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. Note: 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
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 " Adding Functions to Modules " section below. Note: Do not send any terminating characters, the module will append them. <channel> : channel number to send to unit ``PASSTHRU-1
?VERSION	Query for the current version number of the module. ?VERSION

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.

Command	Description
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
VERSION-<version>	Reports the version number of the module. <version> : x.y.z = module version number VERSION-1.0.1

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.
- It is assumed that the first Device channel in the DEVCHAN array move the screen up when a contact closure is triggered, the second channel for screen down, and both channels triggered at the same time will stop the screen's movement.