

White Paper

Backgrounder for the Creation of a Third-Party Device Driver for CS-Bus™ Controllers

Revision: 1.0.c

Date: July 16, 2010

This document is designed to provide some general background information for the creation of third-party device drivers using ASCII commands over RS-232-C or identical ASCII commands using UDP Protocol over Ethernet. An associated document entitled “Converging Systems’ Third Party CS-Bus Programmer’s Guide-ZGN Commands” provides the specific protocol information while this document describes user interface issues and addressing issues relevant to the creating of actual user interfaces.

The attached Appendix 1 outlines general addressing concepts that must be taken into consideration when creating specific menu systems. Also contained within this Appendix are actual code samples for reference.

Top Level Menu

Typically, pre-existing presets for Lighting Levels or Motor Presets are usually accessible to the user through a preliminary menu screens (i.e. Top Level Menu). An example of that menu screen can be found below for lighting controllers and motor controllers within **Figure 1** below.



Figure 1a—Lighting Control

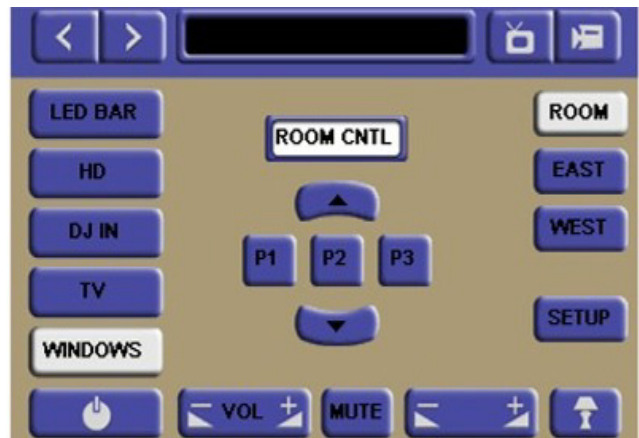


Figure 1b—Motor Control

Lighting Menu (Figure 1a). Each of the various square interface buttons (*P1-P8*) refer to pre-programmed lighting presets. Other user interface buttons in this configuration include (i) *Effect* and (ii) *OFF (red button)* and (iii) *ON (green button)*.

Motor Control Menu (Figure 1b). Each of the three square interface buttons (*P1-P3*) refer to pre-programmed intermediate stops (motor presets), while the up and down arrows refer to motor direction. The white “Room Cntl” button doubles as a motor stop button (a separate button could just have easily have been integrated for this function).

In situations where more than one lighting or window covering control needs to be accessed, additional Top Level Menus can be accessed. In this case, the rectangular buttons on the right side point to those secondary (Top Level) menus. In this case there are two top level menus, one entitled “ONYX” and another entitled “SHELF.”

Adjustment Menus

Even though a dealer may have preprogrammed specific lighting or motor presets upon initial installation, additional user controls may be useful to tweak the system. In this configuration, we have created adjustment menus (i.e. Adjustment Menus) where some type of adjustment commands can be accessed. These Adjustment Menus might be accessed by selecting the *Setup* button within the screen shown in **Figure 1**.

The screen shown below in **Figures 2** shows a typical Adjustment Menu screen for typical controls.

Lighting Menu (Figure 2a). In this case, this Adjustment Screen shows a lighting adjustment menu for a particular lighting control (“Onyx” bar top). A set of three up and down toggles for Brightness, Saturation and Hue (adjustments) has been positioned for quick access. Since these respective *Up* and *Down* buttons start an automatic fade (which continues until stopped or the end of the adjustment range is reached), it is necessary to have a “Stop” button to stop the automatic fade to suit a user’s preference. In this case, the white buttons marked *BRIGHT*, *SAT* and *HUE* serve this purpose. Once selected, any of these three buttons stop a previously invoked fade (i.e. a Brightness fade from full on to full off can be stopped anywhere along the continuum by pressing the *Bright* button).

Motor Control Menu (Figure 2b). A similar concept of a Motor UP or Motor DOWN button will evoke a motor to start moving in the selected direction and that motor will continue until it either reaches the end of its run or is stopped at an intermediate location by the pressing of a similar *Stop* button.



Figure 2a—Lighting Control



Figure 2b—Motor Control

Save Menus

Once a particular setting has been customized (i.e. a specific lighting level has been selected from a palette of 16.7 different levels or a specific motor intermediate stop has been selected), it is now time to save that particular level or preset into non-volatile memory.

The screens shown below in **Figures 3** show a typical Save Menu screen for a various controls.

Lighting Menu (Figure 3a). In this case, the *Save Menu* button in **Figure 2a** is selected and the screen below shown as **Figure 3a** displays. Here any specific preset button is selected and that level is stored into that location. There has also been an automatic “jump” programmed into each Preset button so after it is selected the first time, the screen in **Figure 2a** is immediately displayed thus preventing a user from easily storing the same specific preset location into all Preset locations by accident. This feature is obviously depending upon the particular software interface being programmed but in this case, it certainly is a great feature.

Motor Control Menu (Figure 3b). Similarly, in this case, the *Save Menu* button in **Figure 2b** is selected and the screen below shown as **Figure 3b** displays. Here any specific preset button is selected and that level is stored into that location. There has also been an automatic “jump” programmed into each Preset button so after it is selected the first time, the screen in **Figure 2b** is immediately displayed thus preventing a user from easily storing the same specific preset location into all Preset locations by accident. This feature is obviously depending upon the particular software interface being programmed but in this case, it certainly is a great feature.



Figure 3a—Lighting Control



Figure 3b—Motor Control

Appendix 1

A.1 Addressing Schemes

A.1.1 Addressing Scheme- Primary Address

For the most basic installation, each controller needs to have at minimum a primary address referred to as a “ZGN” (Zone/Group/Node) address. A device will respond if its primary ZGN address matches the address of the command on the bus. For example a device may have a ZGN address of 3.2.1. (i.e. Zone 3, a group number of 2, and a node of 1). It will respond if a message on the CS-Bus has a ZGN address of 3.2.1. It will also respond if a broadcast node ZGN address is used (i.e. 3.2.0), or a broadcast group ZGN address (i.e. 3.0.0), or a global broadcast ZGN address of 0.0.0 is used. Motor Devices are set to the factory default ZGN address of 1.1.0 while lighting control devices are set to the factory default ZGN address of 2.1.0.

A.1.2 Addressing Scheme - Secondary Addresses

Each device in addition to having a mandatory (single) primary ZGN address, may also have up to seven secondary ZGN addresses or Shadow addresses. A device will respond if one of its ZGN addresses (either the primary or one of the secondary addresses) matches the ZGN address of the command on the bus. For example a device may have a primary ZGN address of 1.1.1. (i.e. Zone 1, a group number of 1, and a node of 1) and a secondary ZGN address of 1.2.0. It will respond if a message on the CS-Bus has a ZGN address of 1.1.1 *or* 1.2.0. The shadow addresses also have to be further qualified by matching the type of device and command (see CS-Bus Messaging manual for more details). The device will also respond if a broadcast node ZGN address is used (i.e. 1.1.0), or a broadcast group ZGN address (i.e. 1.0.0), or a global broadcast ZGN address of 0.0.0 is used. Motor control devices are set to the factory default ZGN address of 1.1.0 while lighting control devices are set to the factory default of 1.1. Secondary addresses are utilized where a particular controller is desired to be activated in combination with other controllers but also be desired to be controlled individually.

A.1.3 Addressing Scheme - Typical Examples

Typically, where more than a few CS-Bus controllers all connected to the same bus, the general strategies below should be used.

Zone Address. The Zone address is the addressing element widest in scope. Usually, in large buildings, the Zone refers to the particular floor of a building. In a two floor building, relegate Zone 1 to the first floor and Zone 2 to the second floor. In the case for Zone 1, the ZGN reference for Zone “1” would be 1.X.X. In the case for Zone 2, the ZGN reference would be 2.X.X.

Group Address. The Group address is the addressing element one level narrower or more defined than the Zone Address. Specifically, in large buildings, the Group address refers to a particular room within a floor. In a three room single floor building (all within a single zone), the first room might be considered Group 1, while the second room would be considered Group 2 and the third room would be considered Group 3. All controllers within a particular room (i.e. Group) would be controlled as a unit by controlling that Group. In the case for a single floor building (single Zone or Zone=1) with 3 rooms,

the ZGN reference for the first room would be 1.1.X. The reference for the second room would be 1.2.X. And finally, the reference for the third room would be 1.3.X

Node Address. The Node address is the addressing element most narrowly defined. Specifically, in a particular room with just three controllers, the first controller might be considered Node 1, while the second controller might be considered Node 2 and the third controller might be considered Node 3. Each controller(s) within a particular room (i.e. Group) would be individually controllable through the use of this Node address. Specifically, in the case of a single floor building (single Zone or Zone=1) with a single room (single Group or Group=1), the ZGN reference for the first controller would be 1.1.1. The reference for the second controller would be 1.1.2. And finally, the reference for the third controller would be 1.1.3.

A.1.4 Implication for Device Driver (Lighting Controllers)

In typical installations, the installer/dealer would have created an addressing scheme which would have taken into consideration the above Zone/Group/Node (ZGN) theories. The graphical user interface would also be created with the general types of control available for CS-Bus controllers (as documented throughout the body of this White Paper). The final implementation of the device driver then would need to take into consideration the specific ZGN address for the device that is desired to be controlled through the device driver.

Top Level Menu (Figure 1a). Assuming that **Figure 1(a)** described within this white paper defined a setup where there were two lighting controllers within a single room of an installation, the general assumptions here would be as follows for a Top Level Menu page for one of those two lighting controllers:

1st LIGHTING CONTROLLER (ONYX) CONTROL (NODE=1)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 1			
	P1 (preset 1)	#2.1.1.LED=RECALL,1;	Lighting Preset #1 invoked
	P2 (preset 2)	#2.1.1.LED=RECALL,2;	Lighting Preset #2 invoked
	P3 (preset 3)	#2.1.1.LED=RECALL,3;	Lighting Preset #3 invoked
	P4 (preset 4)	#2.1.1.LED=RECALL,4;	Lighting Preset #4 invoked
	P5 (preset 5)	#2.1.1.LED=RECALL,5;	Lighting Preset #5 invoked
	P6 (preset 6)	#2.1.1.LED=RECALL,6;	Lighting Preset #6 invoked
	P7 (preset 7)	#2.1.1.LED=RECALL,7;	Lighting Preset #7 invoked
	P8 (preset 8)	#2.1.1.LED=RECALL,8;	Lighting Preset #8 invoked
	OFF button	#2.1.1.LED=OFF;	LED OFF
	ON button	#2.1.1.LED=ON;	LED ON
	Effect button	#2.1.1.LED=EFFECT;	Lighting Effect #1 invoked

The second lighting controller (bar shelf) would be similarly controlled by treating this second controller within the same room and zone as simply a different node (i.e. Node=2). The general assumptions here would be as follows:

2nd LIGHTING CONTROLLER (SHELF) CONTROL (NODE=2)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 2			
	P1 (preset 1)	#2.1.2.LED=RECALL,1;	Lighting Preset #1 invoked
	P2 (preset 2)	#2.1.2.LED=RECALL,2;	Lighting Preset #2 invoked
	P3 (preset 3)	#2.1.2.LED=RECALL,3;	Lighting Preset #3 invoked
	P4 (preset 4)	#2.1.2.LED=RECALL,4;	Lighting Preset #4 invoked
	P5 (preset 5)	#2.1.2.LED=RECALL,5;	Lighting Preset #5 invoked
	P6 (preset 6)	#2.1.2.LED=RECALL,6;	Lighting Preset #6 invoked
	P7 (preset 7)	#2.1.2.LED=RECALL,7;	Lighting Preset #7 invoked
	P8 (preset 8)	#2.1.2.LED=RECALL,8;	Lighting Preset #8 invoked
	OFF button	#2.1.2.LED=OFF	LED OFF
	ON button	#2.1.2.LED=ON	LED ON
	Effect button	#2.1.2.LED=EFFECT	Lighting Effect #1 invoked

Adjustment Menus (Figure 2(a)). Assuming that **Figure 2(a)** described within this white paper defined a setup where there were two lighting controllers within a single room of an installation, the general assumptions here would be as follows for an Adjustment Menu page for one of those two lighting controllers:

:

1st LIGHTING CONTROLLER (ONYX) ADJUSTMENT (Node 1)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 1			
	BRIGHT UP	#2.1.1.LED=FADE_UP;	Brightness Fade UP
	BRIGHT STOP	#2.1.1.LED=STOP;	Brightness Fade STOP
	BRIGHT DOWN	#2.1.1.LED=FADE_DOWN;	Brightness Fade DOWN
	SAT UP	#2.1.1.LED=SAT_UP;	Saturation Fade UP
	SAT STOP	#2.1.1.LED=STOP;	Saturation Fade STOP
	SAT DOWN	#2.1.1.LED=SAT_DOWN;	Saturation Fade DOWN
	HUE UP	#2.1.1.LED=HUE_UP;	Hue Fade UP
	HUE STOP	#2.1.1.LED=STOP;	Hue Fade STOP
	HUE DOWN	#2.1.1.LED=HUE_DOWN;	Hue Fade DOWN
	SAVE MENU		“jump to Figure 3 menu”

A similar set of commands would also be utilized the second controller (with Node 2 reference used instead of Node 1 as above).

Save Menus (Figure 3(a)). Assuming that **Figure 3(a)** described within this white paper defined a setup where there were two lighting controllers within a single room of an installation, the general assumptions here would be as follows for this specific menu which is used to save data for the first lighting controller (Onyx):

1st LIGHTING CONTROLLER (ONYX) STORE MENU (Node 1)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 1			
	P1 (preset 1)	#2.1.1.LED=STORE,1;	Setting stored as Preset #1
	P2 (preset 2)	#2.1.1.LED=STORE,2;	Setting stored as Preset #2
	P3 (preset 3)	#2.1.1.LED=STORE,3;	Setting stored as Preset #3
	P4 (preset 4)	#2.1.1.LED=STORE,4;	Setting stored as Preset #4
	P5 (preset 5)	#2.1.1.LED=STORE,5;	Setting stored as Preset #5
	P6 (preset 6)	#2.1.1.LED=STORE,6;	Setting stored as Preset #6
	P7 (preset 7)	#2.1.1.LED=STORE,7;	Setting stored as Preset #7
	P8 (preset 8)	#2.1.1.LED=STORE,8;	Setting stored as Preset #8

A similar set of commands would also be utilized the second controller (with Node 2 reference used instead of Node 1 as above).

A.1.5 Implication for Device Driver (Motor Controllers)

In typical installations, the installer/dealer would have created an addressing scheme which would have taken into consideration the above Zone/Group/Node (ZGN) theories. The graphical user interface would also be created with the general types of control available for CS-Bus controllers (as documented throughout the body of this White Paper). The final implementation of the device driver then would need to take into consideration the specific ZGN address for the device that is desired to be controlled through the device driver.

Top Level Menu (Figure 1(b)). Assuming that **Figure 1(b)** described within this white paper defined a setup where there were two motor controllers within a single room of an installation (east and west facing) were controlled in unison, the general assumptions here would be as follows for this master control menu (for both east and west facing windows):

SYSTEM WINDOW CONTROL (both East and West Facing in Unison)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 0 ¹			
	UP	#1.1.0.MOTOR=UP;	Motor moves UP
	P1 (preset 1) ²	#1.1.0.MOTOR=RECALL,1;	Lighting Preset #1 invoked
	P2 (preset 2) ²	#1.1.0.MOTOR=RECALL,2;	Lighting Preset #2 invoked
	P3 (preset 3) ²	#1.1.0.MOTOR=RECALL,3;	Lighting Preset #3 invoked
	DOWN	#1.1.0.MOTOR=DOWN;	Motor moves DOWN
	ROOM CONTROL (hidden motor "stop" button)	#1.1.0.MOTOR=STOP;	Motor stops

Notes:

¹Node of 0 is a wildcard which triggers all notes assigned between 1 and 255 within specified Zone & Group.

²Some IMC-100 controllers permit intermediate stops (IMC-100T), while other IMC-100 controllers with a built-in STI sensor (Screen Trigger Interface) lack this feature. It is suggested for drapery applications, that the IMC-100T product is sourced.

Two additional menu screens are possible here. One would be customized to control the first motor alone (the east-facing window), while a second and nearly identical menu screen would be created to control the second motor (west-facing window). The first window would be configured as Node=1 while the second window would be configured as Node=2. Examples for the two specific motor screens would be as follows:

1st MOTOR CONTROLLER (EAST-FACING) CONTROL (Node 1)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 1			
	UP	#1.1.1.MOTOR=UP;	Motor moves UP
	P1 (preset 1)	#1.1.1.MOTOR=RECALL,1;	Intermediate Stop #1 invoked
	P2 (preset 2)	#1.1.1.MOTOR=RECALL,2;	Intermediate Stop #2 invoked
	P3 (preset 3)	#1.1.1.MOTOR=RECALL,3;	Intermediate Stop #3 invoked
	DOWN	#1.1.1.MOTOR=DOWN;	Motor moves DOWN
	ROOM CONTROL (hidden motor "stop" button)	#1.1.1.MOTOR=STOP;	Motor stops

2nd MOTOR CONTROLLER (WEST-FACING) CONTROL (Node 2)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 2			
	UP	#1.1.2.MOTOR=UP;	Motor moves UP
	P1 (preset 1)	#1.1.2.MOTOR=RECALL,1;	Intermediate Stop #1 invoked
	P2 (preset 2)	#1.1.2.MOTOR=RECALL,2;	Intermediate Stop #2 invoked
	P3 (preset 3)	#1.1.2.MOTOR=RECALL,3;	Intermediate Stop #3 invoked
	DOWN	#1.1.2.MOTOR=DOWN;	Motor moves DOWN
	ROOM CONTROL (hidden motor "stop" button)	#1.1.2.MOTOR=STOP;	Motor stops

Adjustment Menus (Figure 2(b)). Assuming that **Figure 2(b)** described within this white paper defines a setup for the adjustment of a specific motor controllers (East-Facing or Node=1) within a single room of an installation, the general assumptions here would be as follows:

1st MOTOR CONTROLLER (EAST-FACING) ADJUSTMENT (Node 1)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 1			
	UP	#1.1.1.MOTOR=UP;	Motor moves UP
	STOP	#1.1.1.MOTOR=STOP;	Motor moves STOPS
	DOWN	#1.1.1.MOTOR=DOWN;	Motor moves DOWN
	P1 (preset 1)	#1.1.1.MOTOR=RECALL,1;	Intermediate Stop #1 invoked

	P2 (preset 2)	#1.1.1.MOTOR=RECALL,2;	Intermediate Stop #2 invoked
	P3 (preset 3)	#1.1.1.MOTOR=RECALL,3;	Intermediate Stop #3 invoked
	SAVE MENU		“jump to Figure 3 menu”

A similar set of commands would also be utilized the second controller (with Node 2 reference used instead of Node 1 as above).

Save Menus (Figure 3(b)). Assuming that **Figure 3(b)** described within this white paper defined a setup where there were two motor controllers within a single room of an installation, the general assumptions here would be as follows:

1st and 2nd MOTOR CONTROLLERS (EAST & WEST FACING) STORE MENU (Node 1 & 2)

Location	Button	ASCII command	Resulting Action
Zone 1 Group 1 Node 1			
	E(ast) Store	N/A	Information Only
	P1 (preset 1)	#1.1.1.MOTOR=STORE,1;	Intermediate Stop #1 stored
	P2 (preset 2)	#1.1.1.MOTOR=STORE,2;	Intermediate Stop #2 stored
	P3 (preset 3)	#1.1.1.MOTOR=STORE,3;	Intermediate Stop #3 stored
	W(est) Store	N/A	Information Only
	P1 (preset 1)	#1.1.2.MOTOR=STORE,1;	Intermediate Stop #1 stored
	P2 (preset 2)	#1.1.2.MOTOR=STORE,2;	Intermediate Stop #2 stored
	P3 (preset 3)	#1.1.2.MOTOR=STORE,3;	Intermediate Stop #3 stored

You will notice that the difference between the first triad of presets and the second triad of presets is simply the Node address (2 versus the 1).

Note: You cannot store a preset into a wildcard node (i.e. presets cannot be stored into node 0).